Practical error correction codes and near-term algorithms for quantum computation

Xiaosi Xu

St Annes College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2020

Acknowledgements

I would like to express my gratitude to many people, both in work and in life. This work would not have been completed without the help, support and care from all of you.

First and foremost, I would like to thank my supervisor Simon Benjamin. He is the best supervisor I have ever known in this world, not only because of his wisdom, vast of knowledge and enthusiasm in research, but also his patience and understanding on nearly everything in work and life. I feel extremely lucky to join this group and have Simon as my supervisor. I would forever appreciate his stimulation and encouragement for me, motivation and determination for work, and manner and attitude to the surrounding world.

I would also like to specially thank Xiao Yuan, for the astonishment he brought to me on how smart and productive a young researcher could be, for all the stimulations and guidance on my project, for his humour and personality, and for all those old good times that we drank, ate crabs and had spicy hotpots together.

I also want to thank Ying Li, for his patience answering my stupid and naive academic questions in the early stage of my PhD , for the constructive suggestions and guidance he always gave, and for his kindness and optimism towards life.

I would also like to express my gratitude to Markus Müller, Alejandro Bermudez and Mauricio Gutiérrez for generous help and collaborations within the 'eQual' project; Jonathan Home, for hosting me for two months in Zurich, for all those stimulating discussions and for showing me how real experiments run. I also feel fortunate to have worked with all other collaborators, for the nice experience and for all the help they gave.

For all those people in this group, I couldn't thank you more. Suguru Endo, for his diligence and enthusiasm, for our friendship since the start of PhD and for his jokes – they always make me laugh. Zhenyu Cai, for his kindness answering my stupid questions, for all those discussions and encouragement. Sam McArdle, for his knowledge on everything in quantum industry, his interesting long stories and his 'quantum jokes' to make me smile as a start of the day. Tyson Jones, for his expertise on computer science and his patience helping me solving computing problems, for his easygoing personality, for always being energetic and dynamic. And all the other people in the group, either former or present, Joe O'Gorman, Leonard Wossnig,

Andrea Rocchetto, Bálint Koczor, Takahiro Tsunoda, Carlos Outeiral, Sam Jaques, Armands Strikis and Richard Meister, for all those nice pub trips, for all of their unique personalities that make this group so fun and pleasant to stay.

Beyond all those people helped me for my academic work, there are many friends that I feel lucky to have them with me. Jonathan Baugh, for all interesting discussions on research and others, for sharing his attitudes about life, for his encouragement and those days having lunch together. Xinya, Monica, Yang, Heng, Junjie, Kun and so many friends in this long list, for staying with me, for your support and for sharing happiness and sorrows.

Finally, I am grateful for all my family members, for my mom and dad, who have raised me up and support me on everything in life.

Abstract

The development of quantum computers requires not only experimental advances, but also theoretical efforts to tackle the bottlenecks and identify potential applications. This thesis addresses some essential elements in each stage of the development of quantum computers, from the currently available hardware to the long-term devices.

The critical factor that hinders the quantum hardware from performing perfect operations is the presence of noise, however, it can be overcome by error correction codes. We first consider a large-scale quantum processor with asymmetrical noise and preferably a modular structure. By concatenating the surface code with an error-detecting code, and utilising the information from detection with a customised decoder, we find very high thresholds for the designed devices. The second topic is small error correction codes which are more realistic to be implemented in the near future. A measure 'integrity' is introduced to benchmark the performance of quantum memories to demonstrate beneficial error correction, associated with four milestones. We show integrity is closely related with other commonly-used measures and can be easily assessed by experimentalists. This framework is then studied in detail in the context of an ion-trap quantum device implemented with the seven qubit code. We present a protocol based on detailed sequences of crystal-reconfiguration operations and stabiliser mappings to perform a full error correction cycle. Realistic error models and parameters from reported works are also given.

While experimental efforts continue to battle against the noise, some hybrid quantum-classical algorithms have been developed to be implemented on the near-term noisy devices. The variational algorithm is one of the hybrid algorithms that have been widely applied to simulations of condensed matter physics and quantum chemistry. In the next chapter, we introduce a variational quantum compiler to automatically search for the encoding circuits of small error correction codes. By tailoring the ansatz to optimise the circuit for particular hardware systems, we show some novel circuits discovered with the compiler. The final chapter applies the variational algorithm to solve linear algebra problems including linear systems of equations and matrix-vector multiplications. We show that the linear algebra problems can be solved with near-term devices, with circuit depth linear to the condition number and super-linear to the number of qubits.

List of publications

This thesis is based on the following publications:

- Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation. A. Bermudez, X. Xu, R. Nigmatullin, J. O'Gorman, V. Negnevitsky, P. Schindler, T. Monz, U. G. Poschinger, C. Hempel, J. Home, et al, Physical Review X, 2017, 7(4): 041061.

- An integrity measure to benchmark quantum error correcting memories. X. Xu, N. de Beaudrap, J. O'Gorman, S. C. Benjamin, New Journal of Physics, 2018, 20(2): 023009.

- Fault-tolerant protection of near-term trapped-ion topological qubits under realistic noise sources. A. Bermudez, X. Xu, M. Gutiérrez, S. C. Benjamin, M. Müller, Physical Review A, 2019, 100(6): 062307.

- A high threshold code for modular hardware with asymmetric noise. X. Xu, Q. Zhao, X. Yuan, S. C. Benjamin, Physical Review Applied, 2019, 12(6): 064006.

- Variational algorithms for linear algebra. X. Xu, J. Sun, S. Endo, Y. Li, S. C. Benjamin, arXiv preprint arXiv:1909.03898, 2019.

- Variational circuit compiler for quantum error correction. X. Xu, X. Yuan, S. C. Benjamin, arXiv preprint arXiv:1911.05759, 2019.

There are other related works which are not covered in detail by this thesis:

- Network architecture for a topological quantum computer in silicon. B. Buonacorsi, Z. Cai, E. B. Ramirez, K. S. Willick, S. M. Walker, J. Li, B. D. Shaw, X. Xu, S. C. Benjamin, J. Baugh, Quantum Science and Technology, 2018.

- Demonstration of Adiabatic Variational Quantum Computing with a Superconducting Quantum Coprocessor. M. Chen, M. Gong, X. Xu, X. Yuan, J. Wang, C. Wang, C. Ying, J. Lin, et al, arXiv preprint arXiv:1905.03150, 2019.

- Mitigating Coherent Noise Using Pauli Conjugation. Z. Cai, X. Xu, S. C. Benjamin, npj Quantum Information, 2020, 6(1): 1-9.

Statement of author contributions

In this thesis, Chapter 1 and Chapter 2 give an overview and introduction of essential elements that provide the basic background of works described in later chapters. The chapters from 3 to 7 contain the original research works conducted by the author, in collaboration with other authors named.

Chapter 3 describes the work which was reported in [Physical Review Applied, 2019, 12(6): 064006]. In collaboration with Simon Benjamin, I designed a concatenated code which is optimised for quantum systems with asymmetric noise. Xiao Yuan and Qi Zhao wrote part of the code. I am the first author of the paper and have carried out the numerical simulations and analysed the results. All the results presented in this chapter are my own work.

Chapter 4 describes a benchmarking method to quantify the performance of small error correction codes. This work is in collaboration with Simon Benjamin and has been reported in [New Journal of Physics, 2018, 20(2): 023009]. The analytic treatments were carried out in collaboration with Simon Benjamin, with inputs from Niel de Beaudrap. I am the first author of the paper and have performed the numerical simulations and result analysis. Joe O'Gorman has verified the result. The numerical results presented are all of my own work.

Chapter 5 describes a potential protocol to implement fault-tolerant error correction on an ion-trap quantum processor. The content of this chapter is based on [Physical Review X, 2017, 7(4): 041061] and [Physical Review A, 2019, 100(6): 062307]. Alejandro Bermudez as the first author of both papers, developed the protocols and carried out the analytical work, with inputs from Markus Müller and other authors. This thesis concentrates on numerical simulations and the results. I applied the technique introduced in Chapter 4 and performed all the numerical simulations. The numerical results presented are my own work, and where noted, some graphs were generated by other authors.

Chapter 6 describes a quantum compiler to search for optimal encoding circuits of error correction codes, based on [arXiv:1911.05759]. I am the first author of the paper, and with inputs from Xiao Yuan and Simon Benjamin, I conceived the idea, developed the protocol and carried out the numerical simulations. All the results presented are my own work.

Chapter 7 is based on [arXiv:1909.03898]. Together with Xiao Yuan, and with inputs from other authors, we conceived the idea and developed the technique to solve linear algebra problems assisted with the variational algorithm. The analytical results were mainly produced by Xiao Yuan, and I as the first author, carried out the numerical simulations. The results presented are my own calculations.

Chapter 8 is the summary of all the works included in this thesis.

Contents

1 Introduction and overview

The classical computer is undoubtedly one of the most important and influential inventions in the world that has brought revolutionary changes to our life in many ways. Over the decades it has been developed to be smaller, faster and more powerful. While computers' capacity of solving problems continuously increases, problems that are intractable by classical algorithms remain unsolvable, as predicted by the famous Church-Turing Thesis. For example, the 'electronic-structure problem' to find the low energy levels in chemical systems, quickly becomes impossible to solve even with the most powerful supercomputers, as the system size increases. Besides, as transistors are designed to be smaller and more sophisticated, their size is already within the quantum regime where quantum effects emerge. Therefore in order to avoid unpredictable phenomena in such a quantum system, as well as other quantum systems that people are interested in, one needs to simulate their behaviour equipped by classical computers, which however can not be easily realised since the memory required for such simulations grows exponentially. All these factors triggered the introduction of the concept of quantum computer by Feynman in 1982 [1], followed by a more rigorous formalisation by David Deutsch in 1985 [2]. After this it was not long before Deutsch and Jozsa justified that quantum computers are capable of solving a certain class of problems exponentially faster than the classical machines [3]. Two years later, Peter Shor proposed a quantum

algorithm which achieves prime-factorisation exponentially faster than any of the known classical algorithms [4]. This is a remarkable discovery as it suggests that a quantum computer could be used to break RSA, the most widely used public cryptosystem. Another example of famous quantum algorithm is the Grover's algorithm, which provides quadratic speed up on searching of some unstructured space over the classical algorithms [5]. All of these discoveries have brought great excitement into the field of quantum computation, but also lead to an important question, how feasible it is to make a practical quantum computer?

To answer the question one has to firstly engage with several levels of issues. The basic level is to find a physical platform where the quantum computer can be built upon. This requires the system to have a full set of building blocks analogous to the classical computers, which allow for sophisticated and precise control and manipulations. Potential systems include superconducting circuits, trapped ions in electro-magnetic fields, NV centres in a diamond, quantum-dots in silicon-based architectures and so on. Despite that each system has different advantages and disadvantages, one common problem shared by all is that the quantum state stored in the device can not last forever but has only a limited life time. The so-called *decoherence* originates from unwanted interactions with the environment and from imperfect controls and operations. To address this problem, the concept of *quantum error correction* has been introduced, followed with the development of *fault-tolerant* quantum computing and the concept of a *threshold*. The latter suggests that noise can be suppressed to an arbitrarily low level by having a sufficiently large resource overhead, if the error affecting each elementary step is lower than a certain value. On the other hand though, with the imperfect performance of quantum devices currently present in the lab, such a large physical system is out of reach.

While full-fledged quantum computers are years away, the *near-term noisy intermediate-scale quantum (NISQ)* hardware has attracted great interests recently, as machines with hundreds of physical qubits are expected to come out in the next few years. Early works have already demonstrated the potential of NISQ devices with applications in quantum chemistry and many-body physics. On the other

hand, it remains an open question to identify practical near-term algorithms which enable shallow circuits to be performed with a certain level of fault tolerance. The *hybrid classical-quantum algorithms* satisfy such a need by having only the core and classically-intractable problems computed by the quantum device, while the other parts are handled by classical machines. *Variational quantum algorithms* are the most commonly used ones, including the *variational quantum eigensolver (VQE)*, *variational quantum simulator (VQS)* and *quantum approximate optimisation algorithm (QAOA)*. While those algorithms are continuously improved, it is also interesting to explore possible applications in near-future quantum computers.

This thesis addresses some of the issues mentioned above. We start with a general introduction to quantum computation in Chapter 2, including basics in quantum information processing, foundations of quantum error correction and hybrid algorithms that will be used in the later chapters of the thesis. In Chapter 3 we present a possible route for scaling up the quantum system with error correction codes, while boosting the performance by utilising more error information. Chapter 4 focuses on the more experimentally-feasible small error correction codes and a means to quantify performance of quantum memories. The measure described in Chapter 4 is then modelled in detail in the context of an ion-trap quantum computer in Chapter 5, with realistic protocols and noise models analysed. In Chapter 6, we discuss a near-term algorithm and apply it to search for the encoding circuit of small error correction codes. This leads to Chapter 7 where another potential application of near-term algorithms is explored, on the mathematical challenge of linear algebra problems. Finally in Chapter 8 we summarise all the works presented in this thesis. Overall, a wide range of problems have been discussed, from the long-term large-scale quantum computing, in combination with the fault-tolerant hardware design, to the practical near-term applications, with the goal to at least offer potential routes from today's small devices to the far-reaching future of mature quantum technology.

2

Introduction to quantum computation

Contents

2.1 Basics in quantum computation

For classical computers, a *bit* is the elementary memory unit which can either be 0 or 1, and the counterpart of such unit in a quantum computer is called a *qubit*. To fully describe the state of an n-qubit system, we represent it using a vector in a Hilbert space with a dimension of 2^n. The standard *basis* (computational basis) vectors for a single qubit are $|0\rangle$ and $|1\rangle$:

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

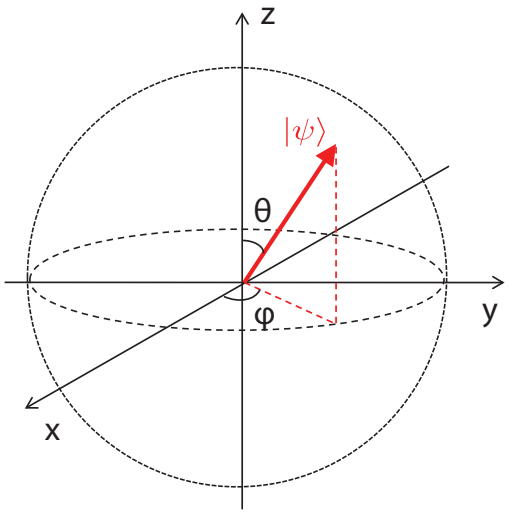


Figure 2.1: Schematic view of a qubit on the surface of a Bloch sphere.

An arbitrary *state* is written as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2.1}$$

where α and β are complex numbers which satisfy $|\alpha|^2 + |\beta|^2 = 1$. Note the $\{0, 1\}$ is not the unique choice for a basis, since any state and its orthogonal state can form a basis. Another useful and commonly used basis is $\{+, -\}$, called the Hadamard basis, which relates with the computational basis by: $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Another useful representation of a qubit is the *Bloch sphere*, with which we can rewrite Eq. (2.1) as:

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle\right), \tag{2.2}$$

where $e^{i\gamma}$ is a global phase that has no observable effect. The quantum state $|\psi\rangle$ on a Bloch sphere is illustrated in Fig. 2.1. Here $|\psi\rangle$ is a *pure state*, as it is a single ket vector. A quantum state that is a statistical ensemble of pure states is called the *mixed state*. Mixed state can be described with a *density matrix* ρ:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|, \tag{2.3}$$

where p_i is the fraction of the ensemble for each pure state $|\psi_i\rangle$.

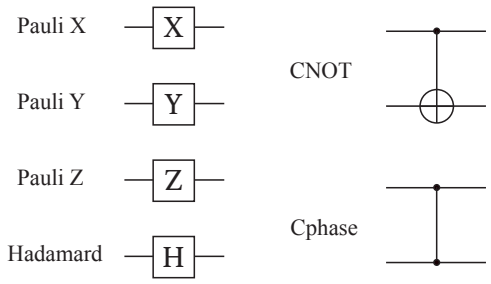


Figure 2.2: Symbols for common single- and two-qubit gates.

Quantum gates are unitary operators acting on the quantum state. Any single-qubit operation can be expressed as a linear combination of the *Pauli matrices*:

$$U = a_I I + a_x X + a_y Y + a_z Z, \tag{2.4}$$

where the coefficients a_I, a_x, a_y and a_z satisfy: $|a_I|^2 + |a_x|^2 + |a_y|^2 + |a_z|^2 = 1$, and I, X, Y, Z are:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Another commonly-used single-qubit gate is the *Hadamard* gate, which has the matrix form of:

$$H = \frac{\sqrt{2}}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Hadamard gate is important because it maps between $\{0, 1\}$ and $\{+, -\}$.

For a quantum system with more than one qubit, a useful and important class of operations is the controlled operations. The most common one is the *Controlled-Not* or *CNOT* gate, which is applied from the 'control' qubit to the 'target' qubit. Expressed as $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$, where $\otimes$ refers to the tensor product, it applies an X gate to the target qubit if the control qubit is in state $|1\rangle$. The matrix form is:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Similarly a *Controlled-Z* or *Cphase* gate is a controlled Z operation, which applies a Z gate to the target qubit if the control qubit is at state $|1\rangle$. The matrix form is:

$$Cphase = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

The circuit symbols for the gates described above are shown in Figure 2.2.

A quantum state can also be changed by measurement, which is a non-unitary operator projecting the quantum state into a certain basis. For example, if measuring a state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ in the $\{0, 1\}$ basis, the result would either be $|0\rangle$ with probability of $|\alpha|^2$ or $|1\rangle$ with probability of $|\beta|^2$. Clearly the measurement procedure destroys the phase information stored within the quantum state, and this procedure is irreversible.

2.2 Introduction to quantum error correction

One of the largest challenges to building a practical quantum computer is to fight against noise. Even if single-qubit gates can be performed with the fidelity up to 99.99%, errors can accumulate over thousands of operations and gradually decohere the quantum state. For classical computers, *bit flips* can happen which turn the 0 digit into 1 and vice versa. However, such errors can be easily detected with the common data-copying techniques and corrected by resetting the bit back to 0 or 1 in each step. Based on its intrinsic nature, quantum states are more vulnerable to noise, as they are susceptible to both bit and *phase flips* that occur continuously (A single qubit bit flip error is equivalent to an unwanted X operation, and a phase flip to an unwanted Z operation.). Besides, quantum states are continuous while the read-out of qubits is digital, meaning that measurement would destroy the entangled information. Therefore, error correction for quantum states is essential but also challenging. Unlike classical states, quantum states cannot be copied from one qubit to another as predicted by the *non-cloning* theorem [6]. Thus, error correction should be employed in a way that never attempts to determine any information of the qubit. Pioneering works by Shor, Calderbank and Steane [7–9] have shown that it is possible

to encode the quantum information into several *physical qubits*, as an ensemble of one (or more) *logical qubit(s)*, and apply error correction codes to indirectly detect the error information. On the basis of their work, many error correction codes have been identified, along with the introduction of *stabiliser* formalism and fault-tolerant error correction. More recently, scalable quantum computing is investigated and proved to be achievable, equipped with *topological codes*. The following sections give a brief introduction of essential elements in quantum error correction.

2.2.1 Simple error correction and error detection codes

The repetition code is the simplest code for classical error correction. The code protects the bit from binary errors simply by making three copies of the state:

$$0 \to 000, \qquad 1 \to 111,$$

where the bit string 000 is called the *logical 0* while 111 is called the *logical 1*. The way it works is that after some time of corruption, we measure the three bits and recover the logical bit back into either logical 0 or 1, depending on the majority vote. The advantage of such a method can be easily explained: suppose the probability for flip of one single bit is p, which is exactly the error probability for an unprotected bit. For the logical bit, since the majority vote fails if two or three bits are flipped, the error probability is thus $p_c = 3p^2(1-p) + p^3$. We can see that $p_c < p$ when $p < 1/2$. With the currently mature technology, the error rate for a single operation in a classical computer can be controlled to very low levels, thus even this simple repetition code could reduce the error to a significant level.

For qubits in quantum computers, we can apply the same trick by encoding the single qubit state into a logical state. Here we take the three-qubit correction code as an example, to introduce the idea behind the Shor code, the first quantum error correction code which was proposed by Peter Shor in 1995 [7]. A logical qubit is encoded into three physical qubits by:

$$|0\rangle_L = |000\rangle, \qquad |1\rangle_L = |111\rangle, \tag{2.5}$$

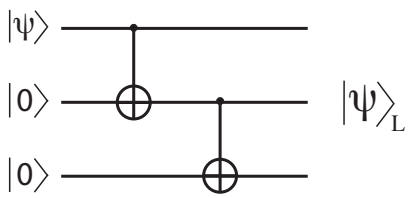


Figure 2.3: Encoding circuit for the three-qubit code.

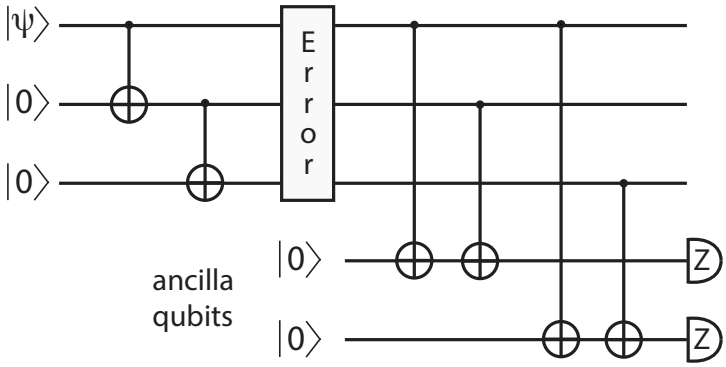


Figure 2.4: Circuit for the three-qubit bit flip correction code with two ancilla qubits to perform parity checks.

where the label 'L' means it is a logical state. Thus an arbitrary state $|\psi\rangle$ is expressed as:

$$|\psi\rangle_L = \alpha|0\rangle_L + \beta|1\rangle_L = \alpha|000\rangle + \beta|111\rangle. \tag{2.6}$$

This state can be easily realised by the circuit shown in Fig. 2.3.

By doing so the state is stored in the three qubits as a whole. Any bit flip would corrupt the logical state and project it out of the logical subspace. Such an error can be detected by measuring all the three qubits, however, as explained before, direct measurement of the qubits corrupts the phase information of the quantum state, thus we need to seek for approaches that require no direct measurement of the state of each qubit.

Suppose a bit flip occurs to the first qubit of the logical qubit with the state $\alpha|0\rangle_L + \beta|1\rangle_L$: $\alpha|000\rangle + \beta|111\rangle \rightarrow \alpha|100\rangle + \beta|011\rangle$. By comparing the first and the second qubits we would find them to be different, and by comparing the first and

error location	no error	qubit 1	qubit 2	qubit 3
measured state	00	11	10	01

Table 2.1: Error syndromes of the three-qubit bit error correction code with two ancillae. The states shown in the table are measurement results of the ancillae.

the third qubits we find them also different, thus by deduction we find the bit error happens on the first qubit. This procedure is called *parity check*. Therefore, by measuring and checking the parity of two of the qubits Z_1Z_2 and Z_1Z_3, we would know if an error occurs and if so, where it is. Importantly, we do not measure the data qubits directly, instead, we only measure the difference between two of them, through measurement of additional qubits called *ancillae*. The circuit for such parity checks for the three-qubit bit correction code are shown in Fig. 2.4. The parity checks can be conducted by two CNOT gates acting from the data qubits to one of the ancillae which is initialised at $|0\rangle$. After the CNOT gates, the ancillae are measured in Z basis, i.e. measured in the $\{0, 1\}$ basis. Any bit flip on one of the three qubits leads to a unique result, called an *error syndrome*, which is summarised in Table 2.1. Using this syndrome information one knows where the error occurred and thus can correct it by applying a Pauli X gate (i.e. a flip) to the certain qubit.

Two things need to be noted here. Firstly, this code can correct only one error. If two errors occur, while we still refer to the syndrome table as if there is only one error and apply the corresponding recovery operation, a logical error would be introduced and the code will have failed. In fact, the *distance* between two codeword states limits the number of errors a certain code can correct. In this case, $d = 3$ (i.e. 3 qubits must change to go from one legitimate state to another), thus the maximum number of errors can be corrected by this code is $t = [(d - 1)/2] = 1$. Secondly, it is a bit flip correction code which detects and corrects only bit flip errors. The detection and correction of a single phase error can be realised with the same encoding circuit but with a different parity check circuit, as shown in Fig. 2.5. The essence of parity checks is based on the fact that for a CNOT gate, a bit error propagates from the control qubit to the target qubit, while a phase error propagates

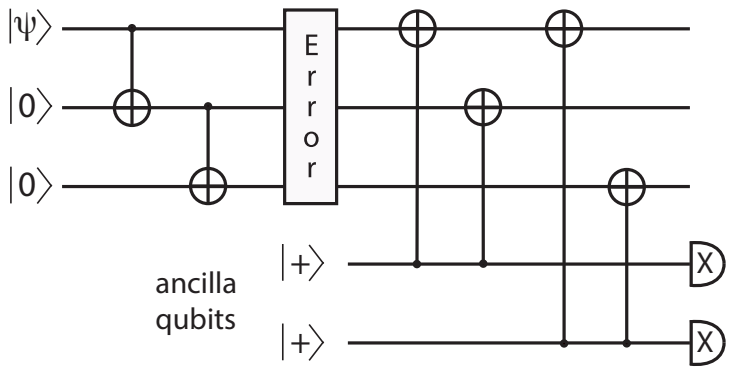


Figure 2.5: Circuit for the three-qubit phase flip correction code with two ancilla qubits to perform parity checks.

from the target qubit to the control qubit. An odd number of errors propagating to the ancilla qubit will flip its original state and can be detected by measurement.

The error correction codes as described above are powerful in that they allow us to not only detect but also correct errors. There is another category of codes that only detects errors but is not capable of knowing where they are. Such codes are called error detection codes, which were developed by Vaidman and Grassl [10, 11], right after the introduction of error correction codes. Compared with the *error correction codes*, error detection codes have fewer requirements for hardware. The benefit of error detection is further reinforced with the development of post-selected quantum computation, which was proposed by Knill in 2004 as a feasible approach for large-scale quantum computing with high error rates [12].

The four-qubit code is an $[[n, k, d]] = [[4, 2, 2]]$ quantum detection code, where n is the number of physical qubits in the memory, k is the number of logical qubits that are encoded, and d is the *code distance*. The code distance implies the maximum number of errors t one code could correct, $t = (d - 1)/2$. The code thus encodes two logical qubits into four physical qubits [13]:

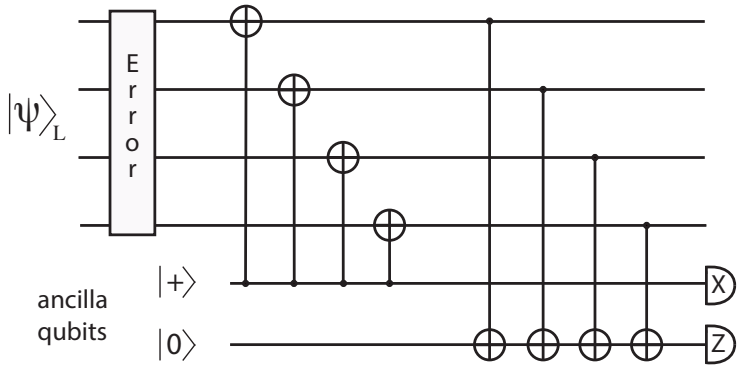


Figure 2.6: Circuit for the four-qubit error detection code.

$$\begin{aligned} |00\rangle_L &\rightarrow |0000\rangle + |1111\rangle \\ |01\rangle_L &\rightarrow |0011\rangle + |1100\rangle \\ |10\rangle_L &\rightarrow |1010\rangle + |0101\rangle \\ |11\rangle_L &\rightarrow |0110\rangle + |1001\rangle. \end{aligned} \tag{2.7}$$

The detection of a bit error and a phase error is realised separately with a single ancilla (two ancillae are drawn in Fig. 2.6 aiming for better understanding of the process). The first ancilla, initialised in $|+\rangle$, applies CNOT gates to each four data qubits before measured in the X basis, is to detect any bit error, while the second one is to detect any phase error. Since the parity checks for bit and phase flips are independent, only one ancilla is actually needed to perform checks sequentially. Therefore, this code is capable of detecting at most one bit error plus one phase error. In fact, owing to the codeword structure, any double errors of the same type would not project the logical qubit out of the logical subspace but would result in either a logical error or no error with 50/50 probability. This property, and the property that it can be easily encoded in a fault-tolerant way (which will be elaborated later), makes the four-qubit detection code a good choice as a first step towards demonstration of error correction on a logical qubit in a real lab system [14].

Up to now we have introduced simple error correction and error detection codes. The smallest code that can detect a single X (or a single Z) error requires two

physical qubits, while at least three physical qubits are required for correction of a single X (or a single Z) error. The [[4,2,2]] code can detect either one X or Z error, and the smallest code that is capable of correcting any X or Z error (full error correction) has five physical qubits. Larger codes generally encode more logical qubits and can correct more errors, however, they also require more complicated error correction circuits. We will introduce more error correction codes in the later chapters.

2.2.2 Error correction with stabiliser codes

The stabiliser formalism

So far we have described codes by their state representation, however, as codes differ their logical states are different, and it would be cumbersome to write down each state as the code distance increases. First discussed by Gottesman in 1997 [15], stabiliser codes are a useful class of codes that provides a general rule for code construction, gate operation and error correction. With the knowledge of the stabilisers, one can easily find out the logical operator, error correction circuit and understand how errors corrupt the logical state without needing to work with the codeword states of the logical qubit.

If we take a look at the $|\psi+\rangle$ Bell state of two qubits:

$$|\psi+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \tag{2.8}$$

we can find it satisfies: $X_1X_2|\psi\rangle = |\psi\rangle$ and $Z_1Z_2|\psi\rangle = |\psi\rangle$, so we say that the state $|\psi\rangle$ is stabilised by the operator X_1X_2 and Z_1Z_2. By definition, for an arbitrary state $|\psi\rangle$, if it is an eigenstate of some operator M: $M|\psi\rangle = |\psi\rangle$, we say that M is the stabiliser of the state $|\psi\rangle$.

The idea of the stabiliser formalism is by use of subgroup. For all unitary operators of a single qubit, the Pauli group P forms a subgroup which contains the following elements:

$$P = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}. \tag{2.9}$$

Any two elements within the Pauli group anticommute with each other: $\{S^i, S^j\} = 0$. For N qubits, the Pauli group is extended as: $P = P^{\otimes N}$. So the code space of an N-qubit quantum memory of state $|\psi\rangle$ is defined to be invariant under the action of the stabiliser group M:

$$L = \{M|\psi\rangle = |\psi\rangle, \forall M \subset P^{\otimes N}\}, \tag{2.10}$$

where M is an Abelian subgroup of the N-qubit Pauli group. Therefore, instead of being described with the state vector representation which is 2^N dimensional, the stabiliser state can be described easily with the generators from the Pauli group which stabilise those vectors. For error correction, stabiliser codes, which involve subgroups of stabiliser operators, reflect the logical subspace of a logical qubit and are used to detect and correct errors.

The four-qubit error detection code revisited

Let us take another look at the [[4,2,2]] error detection code introduced in the last section. As explained above, the code encodes the two logical qubits into four physical qubits, and the realisation of error detection of a bit flip and a phase flip is done independently with the use of an ancilla qubit. The stabiliser set for this code is specified by two operators:

$$\mathcal{S}^1 = X_1X_2X_3X_4, \qquad \mathcal{S}^2 = Z_1Z_2Z_3Z_4. \tag{2.11}$$

All valid codestates are stabilised by the two stabiliser operators. $\mathcal{S}^1$ is used for check of a phase error, corresponding to the first parity check in Fig. 2.6, while $\mathcal{S}^2$ is for check of a bit error. The four-qubit memory has a dimension of 2^4, while the stabiliser defines a subspace with a dimension of 2^2, leaving the freedom of $2^{4-2} = 2^2$, that corresponds to two logical qubits. The logical operators of this code can be chosen as $\bar{X_1} = X_1X_3$, $\bar{X_2} = X_1X_2$, $\bar{Z_1} = Z_1Z_2$ and $\bar{Z_2} = Z_1Z_3$ (the bar on top of an operator represents it to be a logical operator).

2.2.3 Fault-tolerant error correction

So far in previous sections when introducing the error correction codes, we seem to have made two major assumptions. Firstly, errors occur only to the data qubits and are stored statically within the encoded state; secondly, all the logic gates within the error correction cycle are perfect. Both of the two assumptions are very unrealistic. Quantum computation can only be useful when the quantum information stored in the memory undergoes operations dynamically and continuously. Besides, all the logic gates in a quantum channel are noisy, including state preparation, information processing, stabiliser checks, measurement, and decoding. All such errors gradually accumulate and corrupt the logical state. Nevertheless, quantum computation can tolerate faulty gates, provided that the severity of the noise is below some *threshold*. Known as fault-tolerant computing, the concept was firstly introduced for classical computers [16, 17], but now becomes an important milestone for quantum computing which paves the path towards realisation of physical quantum computers [18–20].

One major reason for accumulation of errors stems from error propagation, i.e. errors propagate from one physical qubit to another through logic gates. Single-qubit gates do not yield cascades of errors, however, any two- or more-qubit gate can copy errors from one qubit to another. For example, through a CNOT gate X errors propagate from the control qubit to the target qubit while phase errors propagate from the target qubit to the control qubit. Additionally, the gate operation itself may create errors even when no prior error happens to either of the qubits. This is the major problem that hinders the effectiveness of the error correction codes (as for example, distance-3 codes cannot correct more than one error). Besides, an error correction cycle itself is not noise-free, while two-qubit gates and measurement are essential for any codes. A faulty measurement has a certain probability to generate a wrong outcome, based on which, a wrong fix would be applied to the data qubit, creating one more error in the quantum memory. Therefore, to achieve a fault-tolerant quantum memory, logic circuits should be designed to diminish error propagation or to control errors to a certain low and correctable level. For distance-3 error correction codes, a fault-tolerant circuit is defined such that the

failure of a single element will cause at most one error in the output of the quantum channel. For codes with higher distance, the 'one error' criterion can be relaxed to the maximum number of errors that can be corrected [19]. Generally fault-tolerant quantum computing requires more hardware resources and has more complicated circuits than the non-fault-tolerant case.

2.2.4 Concatenation of codes and the threshold theorem

Concatenation is to encode (an) encoded logical qubit(s) by constructing a hierarchy of quantum circuits [21, 22]. We use $C_0, C_1, C_2...$ to describe the hierarchy levels, where C_0 is the ground level where the logical qubit is encoded with physical qubits. In the second level C_1, each logical qubit is encoded with the same circuit but with logical gates, and so forth. To protect the code, a cycle of error correction is applied at each logical level. Let us take the Steane code as an example. In the first level one logical qubit is encoded into seven physical qubits, and if in a fault-tolerant way, at most one bit flip and one phase flip errors are allowed; Then in the second level, seven logical qubits, containing a total of 49 physical qubits, are encoded with the same fault-tolerant circuit (but each gate is now a logical gate with transversal gates to each individual data qubit). Now the code distance becomes $3^2 = 9$, thus allowing the code to correct up to $(9-1)/2 = 4$ errors. Therefore, at the cost of using more physical qubits, the approach of concatenation can correct more errors. One would then find that a concatenated code with a higher level leads to a smaller logical failure rate, if the probability for occurrence of a single error is smaller than a certain value. The critical error rate is called the threshold of the particular quantum system. If the error rate of the physical qubits can be controlled below the threshold, we can achieve arbitrary accuracy by implementation of a suitably large quantum circuit, provided with abundant resources [15, 23].

The threshold is an important measure for the potential of quantum systems. It relies heavily on the specific platforms and architectures, therefore the reported thresholds in literature differ significantly with different physical systems and error correction codes [12, 21, 24–29]. The *surface code* [30], which has one of the highest

threshold above 1% at circuit level with the standard depolarising noise model, will be discussed in more detail in Chapter 3.

2.2.5 Topological error correction codes

So far we have limited our focus on the most simple and basic codes for an introduction of fundamental concepts and principles. We now introduce a more advanced and modern class of codes called *topological codes*, proposed by Kitaev and Yu in 1997 [31]. Having been receiving growing interests over the years for designs of quantum systems and constructions of large-scale architectures, these are so far the most promising and realisable codes towards realistic and fault-tolerant quantum computation.

Topological codes are a group of error correction codes that the stabilisers can correspond to qubits that are always physically close in a lattice of a certain dimension [28, 32]. By design the topological codes can scale up by inherently concatenating with itself or with other codes of the same type, therefore offering great potential of scalability. Another big advantage of the topological codes is that they have demonstrated a very high threshold, which is within reach of the current experimental technologies. The family includes the surface code, *colour code* and *gauge colour code*, and we will introduce the first two in the following chapters.

The group of topological codes are particularly attractive solutions because they typically have a local structure which is favourable for low-weight stabiliser check operations that are performed to identify errors. This leads to relatively simple protocols supporting fault-tolerant quantum computing, i.e, ensuring a single error occurring during an error-correction cycle will not itself corrupt the logical qubit(s) that are the subject of the cycle. This leads to the threshold rate for errors at the physical level as described in the last section; for many topological codes, when error rates are within this threshold, one can achieve an arbitrarily low logical error rate by having a suitably large ratio of physical qubits to logical qubits.

2.3 Introduction to hybrid quantum algorithms

As introduced in the previous section, quantum error correction plays an essential role in large-scale quantum computations to protect the quantum processors from corruption of noise. For fault-tolerant quantum computing, implementation of any classically intractable problem, like Shor's algorithm, requires a precise control of individual gates of millions of physical qubits [33, 34], which is however beyond the current technology [35].

While this goal is challenging to be achieved within a short period of time, efforts have been taken to seek for quantum algorithms that can be implemented in the near future. As substitutes for fully-fledged quantum computation, quantum-classical hybrid algorithms have been invented, where the quantum computer is used as a coprocessor that only executes the core and classically hard part of the calculation, while the classical computer mainly solves the higher level calculation. Especially, as a typical hybrid method, variational quantum simulation has been applied to find the ground state energy of a Hamiltonian of interest– this approach is referred as the variational quantum eigensolver (VQE) [36, 37]. Another related approach that can be applied to combinatorial problems is the quantum approximate optimisation algorithm (QAOA) [38]. Compared to the classical variational simulation, variational quantum simulation encodes the quantum state via a parameterised quantum circuit that is proven hard with any classical means [35, 39, 40].

Recently, the variational algorithm has been widely used in quantum simulations in the Noisy Intermediate Scaled Quantum (NISQ) regime, for finding energy spectra of a many-body Hamiltonian [36, 41–46], applications with machine learning [47–52], circuit learning [53–56], and others [57, 58]. As variational algorithms can work with shallow circuits, errors may be less damaging and can be suppressed with error mitigation algorithms [43, 59–64], thus enabling the NISQ devices to perform non-trivial tasks.

Variational algorithms have also been invented to simulate real [60] and imaginary [65] time evolutions of quantum systems. We introduce the key elements of the variational algorithm in the following sections.

2.3.1 The variational quantum eigensolver

The variational quantum eigensolver is a variational algorithm to search for the eigenstates of a given Hamiltonian H. To find the ground state

$$E = \frac{\langle\Psi|H|\Psi\rangle}{\langle\Psi|\Psi\rangle},$$

the main idea is to consider a trial state with a set of parameters, $|\phi(\vec{\theta})\rangle = U(\vec{\theta})\,|0\rangle$, with $U(\vec{\theta}) = U_L(\theta_L)\ldots U_2(\theta_2)U_1(\theta_1)$, $\vec{\theta} = (\theta_1, \theta_2, ..., \theta_L)$. The number of the parameters is polynomial to the number of qubits. The trial state is called the state 'ansatz', and the circuit to prepare it is called the circuit ansatz. A good ansatz should well represent the target state. The first VQE experiment [36] implemented the hardware efficient ansatz, which has repeated structures with gates that are easy to implement in the currently available quantum hardware. Owing to its practicability, the hardware efficient ansatz has been applied in many quantum chemistry simulations [45, 66]. However, it is unfavourable for large systems, as classical optimisations that are used to find the solution become extremely difficult as the system size increases [67]. Another widely adopted ansatz is the unitary coupled cluster (UCC) method which is inspired by classical chemistry problems [68, 69]. The UCC ansatz is a natural choice for computations of quantum chemistry [70, 71], however, it might be difficult to implement as it requires a complicated circuit structure.

Suppose the ground state of H can be represented by the ansatz with certain parameters, the ground state finding problem is then converted to an optimisation problem where the set of parameters corresponding to the minimum energy is to be determined,

$$\vec{\theta}_{\min} = \arg\min_{\vec{\theta}} \langle\phi(\vec{\theta})|H|\phi(\vec{\theta})\rangle. \tag{2.12}$$

The optimisation can be accomplished with various methods, including the gradient based classical optimisation [36], global search algorithms [72], imaginary time evolution [65, 73], etc. We introduce the gradient decent and imaginary time evolution methods below.

2.3.2 Optimisation methods

Gradient descent

Gradient descent is a first-order classical optimisation algorithm to search for the minimum of a given function. Applying it to the variational eigensolver, we start with a guess of the solution, $\vec{\theta}_0$ and update the parameters along the negative gradient of the energy $E(\vec{\theta}) = \langle\phi(\vec{\theta})|H|\phi(\vec{\theta})\rangle$,

$$\vec{\theta}_{i+1} = \vec{\theta}_i - a\nabla E(\vec{\theta}_i),\ \forall i = 0, 1, \ldots, T \tag{2.13}$$

where a is the time step, and T is the total number of steps. As the energy E monotonically decreases with sufficiently small a, the solution via VQE always at least corresponds to a local minimum. Starting from several random initial positions, one may find the true solution, i.e., the global minimum.

Imaginary time evolution

The imaginary time is a mathematical concept that has been applied in various physical domains. The idea is to replace the real time it with the imaginary time τ under the Wick rotation. For a system with Hamiltonian H, the propagator e^{-itH} thus becomes $e^{-\tau H}$. The imaginary time evolution could be used to solve dynamical problems [65, 73]. We can start from a variational principle to find the equations that govern the parameter evolution such that a parameterised circuit follows an imaginary time evolution. We introduce the McLachlan's variational principle [74] here. We begin by considering simple real time evolution. The Schrödinger equation can be written as:

$$\frac{d|\Psi(t)\rangle}{dt} = -iH|\Psi(t)\rangle. \tag{2.14}$$

Suppose at time t the state $|\Psi(t)\rangle$ can be represented by the trial state $|\phi(t)\rangle$, then at time $t+\delta t$ we wish to find the trial state $|\phi(t+\delta)\rangle$ also a good representative of the state $|\Psi(t+\delta t)\rangle$. Therefore, we project the state into the trial state manifold and wish to solve

$$\frac{d|\phi(\vec{\theta}(t))\rangle}{dt} = -iH|\phi(\vec{\theta}(t))\rangle. \tag{2.15}$$

McLachlan's variational principle minimises the distance of the left and right terms by

$$\delta \left\| (d/dt + iH)|\phi(\vec{\theta}(t))\rangle \right\| = 0, \tag{2.16}$$

where $\||\phi\rangle\| = \sqrt{\langle\phi|\phi\rangle}$ is the norm of $|\phi\rangle$.

If all the parameters are real, the equation becomes

$$\sum_j M_{i,j}\dot{\theta}_j = V_i, \tag{2.17}$$

where

$$M_{i,j} = \mathrm{Re}\left(\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}\frac{\partial|\phi(\vec{\theta})\rangle}{\partial\theta_j}\right), \quad V_i = \mathrm{Im}\left(\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}H|\phi(\vec{\theta})\rangle\right), \tag{2.18}$$

with the detailed derivations given in Appendix A. The elements of M and V can be found using the quantum computer, while solving Eq. (2.17) for $\dot{\theta}_j$ is the task for the classical computers. By updating $\vec{\theta}$ via:

$$\vec{\theta}_{i+1} = \vec{\theta}_i + \Delta T\dot{\vec{\theta}_i} \tag{2.19}$$

with a sufficiently small time step ΔT, we can find the evolution of parameters and thus the evolution of the states over time.

For VQE where a static problem is considered, if the initial state has a non-zero overlap with the ground state, the state at $\tau \to \infty$ corresponds to the ground state [65]. This is because the amplitudes of all excited eigenstates decay faster than the ground state, making $|\Psi(\infty)\rangle$ the ground state of the Hamiltonian H. In such a case, the quantum state under the Wick rotation is

$$|\Psi(\tau)\rangle = \frac{e^{-H\tau}|\Psi(0)\rangle}{\sqrt{\langle\Psi(0)|e^{-2H\tau}|\Psi(0)\rangle}}, \tag{2.20}$$

and the Schrödinger equation becomes

$$\frac{\partial|\Psi(\tau)\rangle}{\partial\tau} = -(H - E_\tau)|\Psi(\tau)\rangle, \tag{2.21}$$

where $E_\tau = \langle\Psi(\tau)|H|\Psi(\tau)\rangle$.

Assuming the state $|\Psi(\tau)\rangle$ can be well approximated by a parameterised state $|\phi(\tau)\rangle = |\phi(\theta_1, \theta_2, ...)\rangle$ with real parameters θ_i, applied with the McLachlan's

variational principle, according to Ref. [73] the imaginary time evolution of the quantum state $|\phi(\tau)\rangle$ can be mapped to the evolution of the parameters with

$$\sum_j A_{i,j}\dot{\theta}_i = -B_i, \tag{2.22}$$

where

$$\begin{aligned} A_{i,j} &= \mathrm{Re}\left(\frac{\partial\langle\phi(\vec{\theta}(\tau))|}{\partial\theta_i}\frac{\partial|\phi(\vec{\theta}(\tau))\rangle}{\partial\theta_j}\right) \\ B_i &= \mathrm{Re}\left(\frac{\partial\langle\phi(\vec{\theta}(\tau))|}{\partial\theta_i}H|\phi(\vec{\theta}(\tau))\rangle\right). \end{aligned} \tag{2.23}$$

Suppose for the initial state we have $|\phi(\tau)\rangle = |\phi(\theta_1(0), \theta_2(0), ...)\rangle$, thus we can update the parameters $\vec{\theta} = (\theta_1, \theta_2, ...)$ via $\vec{\theta}(\tau + \Delta\tau) = \vec{\theta}(\tau) + \Delta\tau * \dot{\vec{\theta}}(\tau)$ to emulate the imaginary time evolution, where $\Delta\tau$ is a sufficiently small step size.

The imaginary time evolution can also be used to simulate quantum systems with noise. In such a case, a mixed state is applied to model the unitary evolution of the Schrödinger equation, which is given by

$$\frac{d\rho(t)}{dt} = -i[H, \rho(t)], \tag{2.24}$$

Similar to the case with the pure state, a parameterised trial density matrix $\rho = \rho(\vec{\theta}(t))$ is applied. The Schrödinger equation with the trial density matrix is

$$\sum_i \frac{\partial\rho(\vec{\theta}(t))}{\partial\theta_i}\dot{\theta}_i = -i[H, \rho(t)]. \tag{2.25}$$

With the McLachlan's variational principle, we have

$$\delta\|d\rho/dt + i[H, \rho(t)]\| = 0. \tag{2.26}$$

Switching to the imaginary time domain, where the Wick-rotated density matrix becomes

$$\rho(\tau) = \frac{e^{-H\tau}\rho(0)e^{-H\tau}}{\mathrm{Tr}\left[e^{-2H\tau}\rho(0)\right]}, \tag{2.27}$$

the evolution is transformed into

$$\frac{d\rho(\tau)}{d\tau} = -(\{H, \rho(\tau)\} - 2\langle H\rangle\rho(\tau)), \tag{2.28}$$

where $\{H, \rho(\tau)\} = H\rho(\tau) + \rho(\tau)H$. When considering a parameterised density matrix $\rho(\vec{\theta}(\tau))$, the imaginary time evolution of $\rho(\vec{\theta}(\tau))$ is mapped to the evolution of the parameters as

$$\sum_i \frac{\partial \rho(\vec{\theta})}{\partial \theta_i}\dot{\theta}_i = -(\{H, \rho(\vec{\theta}(\tau))\} - 2\langle H\rangle\rho(\vec{\theta}(\tau))). \tag{2.29}$$

Applying the McLachlan's variational principle, we have

$$\delta\|d\rho/d\tau + (\{H, \rho\} - 2\langle H\rangle\rho)\| = 0, \tag{2.30}$$

which can be converted into

$$\sum_i C_{j,i}\dot{\theta}_i = -D_j, \tag{2.31}$$

where

$$\begin{aligned} C_{j,i} &= \mathrm{Tr}\left[\frac{\partial \rho(\vec{\theta}(\tau))}{\partial \theta_j}\frac{\partial \rho(\vec{\theta}(\tau))}{\partial \theta_i}\right], \\ D_j &= \mathrm{Tr}\left[\frac{\partial \rho(\vec{\theta}(\tau))}{\partial \theta_j}\{H, \rho(\vec{\theta}(\tau))\}\right], \end{aligned} \tag{2.32}$$

as shown in Ref. [73]. The detailed derivations are given in Appendix A.

2.3.3 Circuit implementation

Both the optimisation methods of gradient descent and imaginary evolution can be not only realised by classical methods, but also implemented with quantum circuits.

Gradient descent

With the gradient descent method, we need to obtain $\nabla E(\vec{\theta})$. Considering $E(\vec{\theta}) = \langle\phi(\vec{\theta}(t))|H|\phi(\vec{\theta}(t))\rangle$, we have

$$\begin{aligned} \frac{\partial E(\vec{\theta})}{\partial \theta_i} =& \frac{\partial}{\partial \theta_i}(\langle\phi(\vec{\theta}(t))|H|\phi(\vec{\theta}(t))\rangle) \\ =& \frac{\partial\langle\phi(\vec{\theta}(t))|}{\partial \theta_i}H|\phi(\vec{\theta}(t))\rangle + \langle\phi(\vec{\theta}(t))|H\frac{\partial|\phi(\vec{\theta}(t))\rangle}{\partial \theta_i} \\ =& 2\,\mathrm{Re}\left(\frac{\partial\langle\phi(\vec{\theta}(t))|}{\partial \theta_i}H|\phi(\vec{\theta}(t))\rangle\right). \end{aligned} \tag{2.33}$$

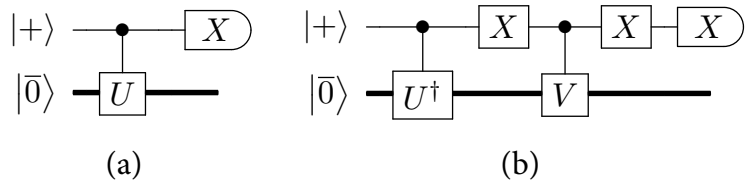


Figure 2.7: The circuits to evaluate (a) $\mathrm{Re}\left(\langle\bar{0}|\, U\, |\bar{0}\rangle\right)$ and (b) $\mathrm{Re}\left(\langle\bar{0}|\, UV\, |\bar{0}\rangle\right)$. The ancilla qubit is initialised in the $|+\rangle$ state and measured in the X basis.

Suppose the Hamiltonian H can be decomposed into $H = \sum_i \alpha_i h_i$, where h_i are unitary operators. As $|\phi(\vec{\theta})\rangle = U(\vec{\theta})|\bar{0}\rangle$ where $|\bar{0}\rangle$ represents a (multi-qubit) quantum register at state zero, we can write the derivative of $|\phi(\vec{\theta}(t))\rangle$ as

$$\frac{\partial|\phi(\vec{\theta})\rangle}{\partial\theta_i} = \frac{\partial}{\partial\theta_i}U(\vec{\theta})|\bar{0}\rangle = Q_i(\vec{\theta})|\bar{0}\rangle, \tag{2.34}$$

and if $U(\vec{\theta}) = G_n(\theta_n)G_{n-1}(\theta_{n-1})...G_1(\theta_1)$, i.e. a gate sequence, then $Q_i(\vec{\theta}) = G_n(\theta_n)G_{n-1}(\theta_{n-1})...\frac{\partial}{\partial\theta_i}G_i(\theta_i)...G_1(\theta_1)$. Thus we have

$$2\,\mathrm{Re}\left(\frac{\partial\langle\phi(\vec{\theta}(t))|}{\partial\theta_i}H|\phi(\vec{\theta}(t))\rangle\right) = 2\,\mathrm{Re}\left(\sum_j \alpha_j\langle\bar{0}|Q_i^*(\vec{\theta})h_k U(\vec{\theta})|\bar{0}\rangle\right). \tag{2.35}$$

It has been pointed out in Ref. [75] that any term with a form of $\mathrm{Re}\left(\langle\bar{0}|U|\bar{0}\rangle\right)$ can be efficiently measured with the circuit shown in Fig. 2.7(a) by measuring $\langle X\rangle$, and it is further proved that $\mathrm{Re}\left(\langle\bar{0}|UV|\bar{0}\rangle\right)$ could be measured with the circuit in Fig. 2.7(b). Therefore, each term in Eq. (2.35) can be evaluated with a quantum circuit.

Imaginary time evolution

For imaginary time evolution with pure states, we need to evaluate each term of M and V. As

$$\begin{aligned} M_{i,j} &= \mathrm{Re}\left(\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}\frac{\partial|\phi(\vec{\theta})\rangle}{\partial\theta_j}\right) \\ &= \mathrm{Re}\left(\langle\varphi_i(\vec{\theta}(t))|\varphi_j(\vec{\theta}(t))\rangle\right) = \mathrm{Re}\left(\langle\bar{0}|Q_i^*(\vec{\theta})Q_j(\vec{\theta})|\bar{0}\rangle\right), \end{aligned} \tag{2.36}$$

and

$$V_i = \mathrm{Im}\left(\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}H|\phi(\vec{\theta})\rangle\right) = \mathrm{Re}\left(-i*\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}H|\phi(\vec{\theta})\rangle\right), \tag{2.37}$$

all the elements of M and V can be determined with the same circuit in Fig. 2.7.

3

A concatenated surface code

Contents

As mentioned in the previous chapters, topological error correction codes offer a possible solution to build a large-scale quantum computer in the presence of noise. As a 2D topological code, the surface code is regarded as a highly promising code due to its modest requirement of a 2D nearest-neighbour connectivity, a high threshold (resulting from low degree stabiliser checks), and simple grid-like lattice structure [31, 76, 77]. The surface code deals with bit flip and phase flip errors independently, so we can perform checks for X-type and Z-type errors alternatively. However, in practice many systems experience an asymmetric error source that makes the standard surface code no longer the optimal choice; relevant cases are reported in

Refs. [78–80], which lead to a rising interest in quantum systems associated with biased noise [81–85]. Commonly in real systems the noise processes are complex, involving both environmental elements and aspects due to the active gates, but generally phase processes take place with a different severity to flip processes [86–88].

In this chapter, we present a new topological code which is designed for quantum systems with asymmetrical noise model. We consider the scenario where phase errors are more prevalent than bit flip errors (it immediately applies to the converse case where bit flip, rather than phase, is prevalent). Our approach is to introduce a variant of the canonical surface code by concatenating it with a two-qubit phase detection code. In particular we propose a new surface code decoder (the algorithm that attempts to infer optimal error-correcting operations) by feeding in the information on likely error locations obtained by the lower-level error detection.

We give a brief introduction on the standard surface code in Sec. 3.1, including the code structure, decoders and threshold calculation. In Sec. 3.2, we introduce the concatenated code by the code structure and the new surface code decoder. Numerical implementation of the concatenated surface code is also presented. The chapter is concluded in Sec. 3.3 with a discussion of potential future works.

The concept and protocol for this code was created collectively by all authors of Ref. [89]. All numerical implementation and modelling were performed by the present author, except for a contribution to the implementation of Dijkstra's algorithm which was provided by Qi Zhao.

3.1 Introduction to the surface code

There are mainly two types of surface code depending on the code topology, the *toric code* and the *planar code*, which differ only on the boundary conditions. The toric code has a periodic boundary condition thus every data qubit is geometrically equivalent, while the planar code does not have the boundary condition but with qubits located in a flat 2D surface. For large surface codes (like those to be used in real quantum computers), the performance of the two types is very similar–it even is for medium sizes we can simulate [90]. Throughout this work we consider a toric code.

3.1.1 Code structure

The surface code has a 2D square lattice structure where the data qubits and ancillae sit one next to one another. As shown in Fig. 3.1, in one representation we can locate four data qubits at the edge of each plaquette (the white dots), while the black and purple ones are ancilla qubits located at the center. In this picture, each plaquette defines an operator of either $\bar{X} = X_1X_2X_3X_4$ (with the purple ancilla) or $\bar{Z} = Z_1Z_2Z_3Z_4$ (with the black ancilla), where X and Z are Pauli matrices, referring to the stabiliser generators. When performing the X parity check, a suitable protocol is to perform CNOT gates from the ancilla qubit to each of the four data qubits, then to measure the ancilla qubit in order to learn the parity of the four data qubits. Consequently, one error on a given data qubit will be identified by two ancilla qubits adjacent to that data qubit. Specifically, a bit flip error will be identified by the Z parity check, while a phase flip error by the X parity check. As shown in the figure, a Z error happening to a data qubit (depicted in red) can be identified by the two neighbouring ancilla qubits (in yellow). If the error location can be exactly identified, such an error can be corrected by applying another gate of the same type.

The logical operations of the surface code are depicted in the figure as the green and the red lines. In fact, for a toric code, any complete curve with Pauli-Z operators spanning from one boundary to the other is a logical Z gate, and any complete curve with Pauli-X operators spanning from one boundary to the other is a logical X gate. The two types of logical operators commute with all stabiliser operators and anticommute with one another.

3.1.2 The decoder

The error correcting process on the surface code involves repeated stabiliser checks, i.e. parity checks. We refer to stabiliser checks of all plaquettes as one cycle. Normally the X and Z stabiliser checks are performed in an alternating order.

An error occurring to a data qubit leads to a change of the error syndrome, which helps us to identify where the error is and thus apply the recovery operation.

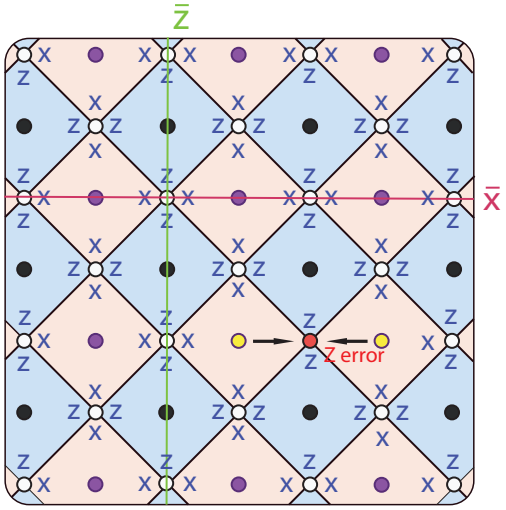


Figure 3.1: Code structure of the surface code. The white dots are the data qubits, which sit at the edge of each plaquette. The black and purple ones are ancilla qubits. In an error correction cycle, CNOT gates are applied between the ancilla qubit and the neighbouring four data qubits, and the ancilla qubit is measured to determine the parity of the four data qubits so as to identify whether an odd number of errors have occurred in the particular plaquette. The red and green lines denote a logical X and a Z operator respectively. The data qubit in red represents the occurrence of a Z error, which can be identified by the adjacent ancillae (in yellow) through the X parity check.

However, as error syndromes do not map to a unique error pattern, stabiliser checks cannot determine precisely the error locations. Errors in measurements can also lead to wrong syndrome outcomes. Therefore, a classical algorithm is used to infer the error locations given the stabiliser check information, to determine what operations should be performed in order to recover the correct logical state of the quantum system. This inference algorithm is referred to as a *decoder*. For a system with given physical error rate, we hope to achieve a logical error rate as low as possible with a particular decoder. Among many decoders such as minimum weight perfect matching (MWPM) [91–93], maximum likelihood based on a tensor network [29, 30, 32, 90, 94–97] and renormalisation group methods [98–100], the MWPM algorithm is most commonly used as the algorithm at the heart of decoders for topological codes. Studies based on this approach have reported high thresholds for the surface code ranging from 0.75% to 1.4%, according to the specific variant and error model [29, 30, 90, 96, 97]. In this work we use it in our decoder for the concatenated code, and we

also employ it when we compute results for the standard surface code as a reference.

3.1.3 Threshold calculation

Threshold is an important measure for the surface code, not only because a code with a higher threshold can permit fault-tolerant quantum information processing on a more noisy system, but also because the higher-threshold code can be expected to achieve a given target logical error rate with a smaller resource overhead. For the threshold estimates, we use a standard approach of simulating a certain number of full stabiliser cycles, recording classical information at each stage, and finally applying the error inference process, i.e. the decoder, as a single-shot analysis which may either 'succeed' or 'fail'.

To estimate a surface code threshold for a given quantum machine, a typical numerical model involves the following stages. Note one does not model the qubits as full quantum entities (which would obviously be exponentially costly in time and memory) but rather one tracks the errors as discrete Pauli events. This implies that we must use an error model where the error events can be described as random (multi-qubit) Pauli operations. Fortunately many realistic error models can be described this way.

1. A total of n imperfect stabiliser check cycles are modelled, where n is proportional to the size of the code. In each cycle, the full set of surface code stabilisers, Z and X types, are measured. (This can be done simultaneously, if the modelled quantum hardware would permit, or in a set of sub-tasks).

2. The result of each stabiliser measurement is compared to the previous recorded outcome for that stabiliser – if it differs then we record that point in time and space as a stabiliser 'syndrome event'.

3. After all cycles are complete, we apply our classical decoder software to analyse the recorded information.

The decoding process is based on the observation that any data qubit error is identified by two adjacent ancillae, while a chain of errors are identified by ancillae adjacent to the end points of the chain. Therefore, a successful correction operation can be applied if we identify the correct ancillae to pair up, and apply the recovery gates all the way between the two ancillae. Note that the recovery path does not need to follow the same path as the error chain, as any closed loops within the surface are equivalent to a stabiliser operator. The failure only happens when a curve is formed that connects one boundary to another, in which case a logical error is generated and cannot be identified.

It is necessary to conduct more than one stabiliser cycle only because the operation of stabiliser checks can be faulty, in which case a wrong error syndrome may be given. Suppose, for example, that there is no error on the data qubits in a region, but that a stabiliser incorrectly reports a change due to a measurement error. Assuming prior and subsequent stabiliser measurements were correct, then we will see that stabiliser outcome 'toggle' its state, and then immediately revert back – and we can pair those two switching events thus identifying the error.

Therefore, the challenge of decoding is to successfully sort all such events into matched pairs. To do so, we first assign a weight to each potential pairing of events according to their separation (in time and space), with a higher weight indicating that it is less likely that the specific pair is associated with one another. For a toric code with the standard depolarising error model, the weight is proportional to the separation between two syndrome events; specifically, the separation in space is given by the 'Manhattan distance'.

In order to find the most likely set of pairings, the MWPM algorithm is used. As the name suggests, the algorithm pairs all events in such a way that the total weight is minimised. This task can be solved by mapping the events into a graph matching problem and applying algorithms from graph theory. Here we use the blossom algorithm introduced by Edmonds in 1965 [101].

Once the optimal pairing is given, this proposed matching is used to derive a corrective action that should map the entire array back to a correct surface code

state; by comparison to the actual record of quantum errors that were introduced in the simulation (which of course would be unknown in a real quantum machine) we are able to record the decoding effort as either a 'success' or 'failure'. Repeating the entire experiment many times, we determine the percentage success rate. Restarting the complete exercise with the same error rate but a different code size, we discover whether increasing the code size lowers the probability of failure; if so, we are within threshold for quantum error correction. By continuously increasing the error rate from below the threshold, we can finally determine the threshold error rate.

3.2 Concatenation of the surface code with the error detection code

3.2.1 The two-qubit error detection code

The two-qubit error detection code is the smallest code that can detect one type of single error. We choose the phase detection code because biased phase noise is more commonly seen in experiments, however, for any system with biased X errors, a bit flip detection code can be adopted under basically the same concept. We employ the encoding $|+\rangle_L = |++\rangle$, and $|-\rangle_L = |--\rangle$. The logical gates are $\bar{X} = XI = IX$, $\bar{Z} = ZZ$, and the stabiliser is XX. Any single phase flip error will be detected when the stabiliser is measured. However, it will not be possible to determine which of the two qubits received the phase error; therefore the code is detecting but not correcting. It remains possible to map the defective state back into the code space, by applying a Z operation to either qubit, but this will lead to a logical error $\bar{Z}$ with a significant probability (a 50% probability, if no other information influences our choice).

3.2.2 The concatenated code

As more information concerning error locations is obviously beneficial for the decoder to determine the correct recovery operations more accurately, we concatenate the standard surface code with the two-qubit error detection code. Consequently our basic building block, corresponding to the white dots in Fig. 3.2(a), is actually

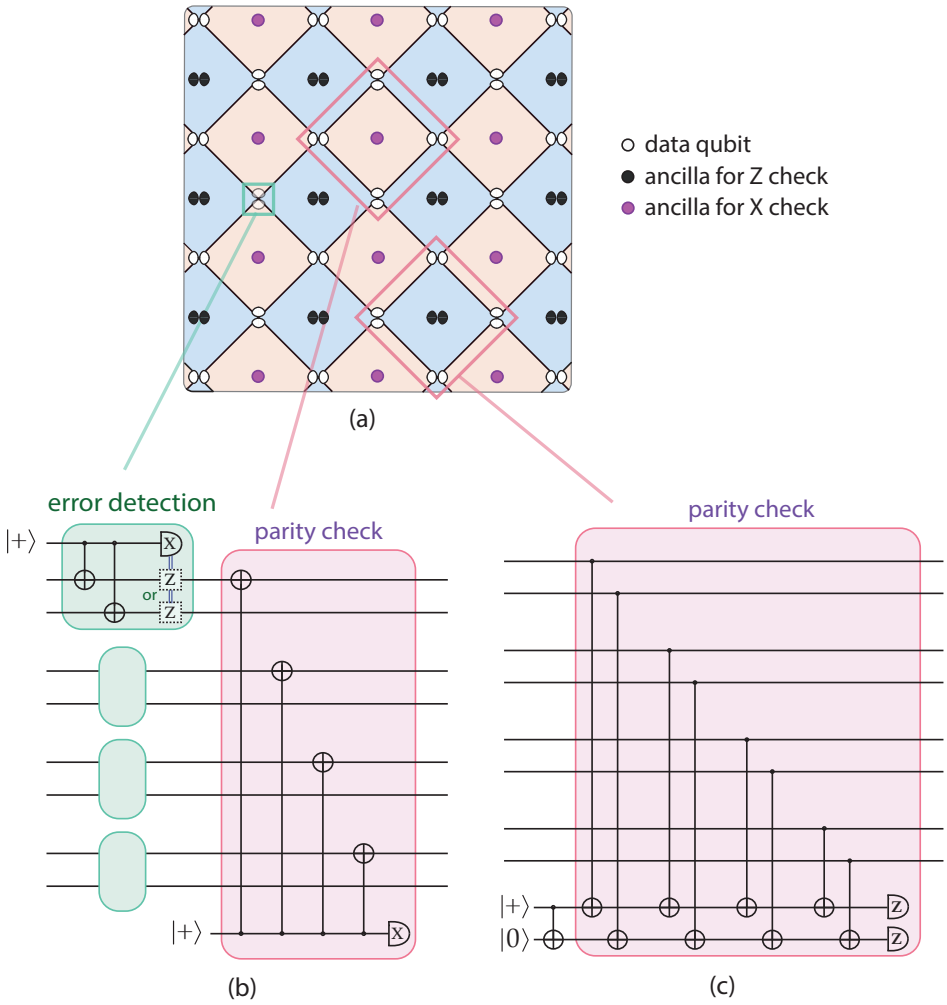


Figure 3.2: Schematic view of the concatenated code and the circuits used for error detection and stabiliser checks. (a) Our surface code variant. When concatenated with a two-qubit phase detection code, each data qubit becomes a logical qubit that consists two physical qubits (plus one ancilla for error detection, though not shown here). Two physical qubits are required to do a Z stabiliser check, and a single qubit is needed for an X stabiliser check. (b) The circuit for an X check, starting with error detection on each qubit. Once an error is detected, a phase gate is applied to either of the two qubits. (c) The circuit for a Z check, where transversal CNOT gates are applied. In the end, both the two ancillae are measured and the parity of the four qubits is represented by the parity of the measurement outcomes.

three physical qubits: the two encoded qubits and one additional ancilla qubit (not shown in the graph). In Fig. 3.2, the black dots represent the ancilla qubit for a Z stabiliser check of the surface code; it involves two physical qubits. However the ancilla for X checks is different: it is simply a single qubit.

With the concatenation, the number of physical qubits is doubled. Undesirable as it is to increase resource costs, we can expect that the new code may offer advantages because information obtained at the (lower) error detection level can be a powerful resource for acting correctly at the (higher) surface code level. We show the circuit of error detection as the green-shaded area in Fig. 3.2(b). When detecting a phase error, we choose to flip one physical qubit and thus restore the data qubit to the proper code space, albeit with the significant probability of having thus implemented an unwanted phase flip on that data qubit. Therefore we record the location of all data qubits where we have observed a phase error; these are now at 'high risk' of an error whereas data qubits that have passed the phase-error-detect stage without issue are at 'low risk'. This partitioning is very valuable for surface code level inference as we presently discuss.

To do a Z stabiliser check, two ancilla qubits are initialised and prepared to be at $|0\rangle_L$, followed with transversal CNOT gates applied from the data qubit to the ancilla qubits, with the circuit depicted in Fig. 3.2(c). Note that by doing so, we also ensure that this circuit is fault-tolerant. Measuring an X stabiliser needs only one ancilla qubit since $\bar{X} = XI = IX$. For simplicity we apply the CNOT gate always on the first qubit, as shown in Fig. 3.2(b). The merit of this simplification relies on the data qubit being within its correct two-qubit code space; to maximise this probability we apply the X stabiliser check on all data qubits immediately after the local error detection. Conversely since the Z stabiliser check detects only X errors in the data qubits, we opt not to perform an error detection cycle ahead of it, since this would provide no beneficial information but could add more errors to the system. Thus the overall cycle is: local error-detect, then the X stabiliser checks and finally the Z stabiliser checks, before repeating.

3.2.3 Decoder for the higher-level code

Wizard decoder

The novelty of the present approach is that there is additional information to feed into the classical decoder, in addition to the record of syndrome events. We introduce this by the following thought experiment: Suppose that we augment the standard surface code with a 'wizard' who has the power to detect errors perfectly, and can correct any error with a 50% chance. Whether he corrects it or not, he always records the information into a list; thus half of the list (on average) refers to data qubits with errors, while half refers to those without errors. Note all data qubits that have suffered an error are certainly on the list. In the remainder of the chapter when we refer to 'the list' we mean this record of the qubits that are at a high risk of error, here provided by the 'wizard' but in practice coming from the error detect circuits.

During the decoding phase, we have access to this list in addition to our usual syndrome information. We will then only permit pairing of syndrome events that can be connected by a path along which all data qubits lie in the list (we could give infinite 'weight' to pairs that cannot be so linked, to prevent our MWPM algorithm from matching them). If the errors are sparsely located, the decoder would then be very powerful – its pairings are correct with a high probability. In fact, we have confirmed that the threshold data qubit error rate in this circumstance is 59%, which is in fact the lattice percolation threshold [102, 103]: the decoder fails only when the errors are so dense that we can always find a path, connected by listed potential errors, from one boundary to another – this is a logical error and can not be corrected. Finally note that if, in the above story, the wizard were only able to detect phase errors, then a very high threshold would still be achieved but would apply specifically to errors of that type.

Realistic decoder

For our real concatenated code we do indeed have such a list, which is simply the record of the space/time coordinates where phase errors were detected. However the idealisation described above cannot be achieved for two main reasons (1) The error

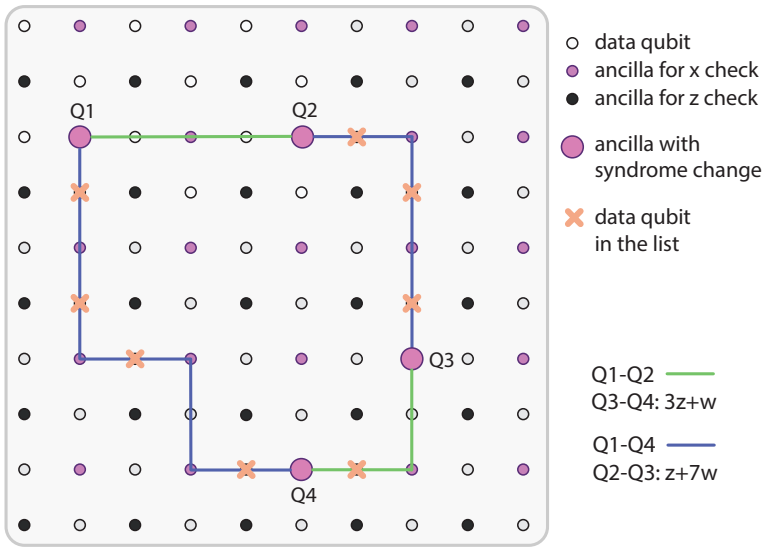


Figure 3.3: Pairing of ancilla qubits based on spatial locations and potentially faulty qubits. All of the qubits shown are in the higher-level surface code. The ancilla qubits with a change of error syndrome are denoted as the big purple dots. The orange crosses represent data qubits identified as likely to have suffered an error according to the lower-level error detection code, i.e. they are on the 'list' as described in the main text. Two possible pairings are shown in this graph: Q1&Q2 and Q3&Q4, or Q1&Q4 and Q2&Q3, connected by the green and purple curves, respectively. The weights for the connections between two adjacent ancilla qubits with and without the listed data qubits are w and z. Therefore, while the algorithm based purely on spatial distance will always opt to the first pairing option, the second option gives a lower total weighted distance if $z > 3w$.

detection process is imperfect due to noise (it can create errors, or incorrectly report the error-status of the pair), (2) It is possible that both physical qubits constituting a data qubit have received errors leading to a logical error which is undetectable. Because of these imperfections we need to permit our decoder to pair syndrome events even when we cannot connect the two through a path along which all data qubits lie the list. However, we can assign such cases a higher weight, and thus input the knowledge represented by our list in a 'softer' form. This is illustrated in Fig. 3.3, which depicts a scenario where we would chose a different pairing than a regular surface code decoder would select because of the additional information from the list.

We now describe the approach we have taken to implement a decoder that exploits the valuable, albeit imperfect, information represented by our list. We apply the Dijkstra's algorithm [104] which is widely used to efficiently find the

shortest paths between nodes in a graph. We employ it to determine a suitable weight for each possible pairing; this replaces the simple 'Manhattan distance' calculation that is usually employed in surface code analysis. We can regard each ancilla qubit for X checks as a node, and similarly (but as a separate problem) for Z checks. For each two adjacent nodes, if the data qubit in the middle is on the list, the distance of these two nodes are set to be w, otherwise z. After setting all the distances for each pair of adjacent nodes, Dijkstra's algorithm can determine the minimal distance for each pair of syndrome ancilla qubits and the corresponding path connecting them. It is this 'shortest path' that then provides the proper weighting for each pair of syndrome events. Given this weighting, we can use the standard MWPM algorithm to match them. Note that the weight parameters w and z influence the performance of decoder significantly and can be optimised in different practical cases. The ratio $w : z$ is of course dependent on the hardware error rates, with an infinite weight for z being the idealised limit corresponding to the 'wizard' in the earlier illustration. In addition to these spatial weights, we also consider the syndrome events occurring in different stabiliser cycles, therefore apart from w and z, we introduce the time weight t. Paths over time are permitted only when the ancilla qubit is connected to itself. The distance of two syndrome events is thus the sum of distance in space and time.

There is a further non-trivial feature related to the use of Dijkstra's algorithm. Since it specifies the entire optimal path connecting each pair, once we have opted to pair two syndrome events then we should correct errors along the lowest weighted path, instead of correcting errors following the shortest spatial path.

Dijkstra's algorithm is computationally expensive. Fortunately, in order to reduce the complexity of Dijkstra's algorithm in a large scale graph, we do not always need to calculate the accurate minimal distance between two syndrome events but instead we can give an approximate distance. Here we predetermine a rather large cut down threshold which is related to the hardware error rates and weight parameters z, w, t and can be optimised in different cases. If the distance between a certain pair of nodes exceed this threshold, the program will stop and

set this value as their distance. This modified Dijkstra's algorithm can improve the efficiency at the expense of the performance of the decoder.

Another important factor in the new decoder is the frequency to apply X and Z checks. As described above, in the case with equal probability of X and Z errors, one should perform equal number of X and Z checks. However, we find that a biased environment in which phase errors dominate necessitates a higher rate of X checks in order to obtain the lowest logical error rate. With the standard surface code, one finds that the ratio of the total number of X and Z checks for the optimal behaviour of the code is roughly the same as the ratio of the probabilities for occurrence of Z and X errors, defined as α. Such a discovery is not surprising, as the surface code handles phase and bit errors independently with equal power. Strictly speaking, this is true when the number of basic gate operations involved in an X check is the same as that for a Z check, so that the rate of introduced errors is the same. This will be approximately true in real devices. However, our concatenated code is rather different, not only because it is capable of correcting more phase errors, but also it requires more gates to perform a Z check. In a sense, these two facts have conflicting implications, since when we increase the number of Z checks to achieve balanced performance versus the X checks, more gates are conducted and thus we introduce more errors into the quantum hardware. In our simulation, we find that the code handles Z errors much better than the X errors, therefore more Z checks leads to a higher threshold. The optimal ratio of X/Z checks applied in the simulations is found to be smaller than α. We will discuss this further in the next section.

3.2.4 Performance of the concatenated code

Error models

In our simulation, we consider a mixture of dephasing and depolarising noise, motivated by their popularity and practical soundness. The noise is stochastic such that each operation can be modelled by a superoperator $\mathcal{NU}$ with

$$\mathcal{N} = (1-p)\mathcal{I} + p\mathcal{E}.$$

Here $\mathcal{U}$ is the ideal operation, and $\mathcal{N}$ is the noisy superoperator with identity channel $\mathcal{I}$ and error channel $\mathcal{E}$ occurring with probabilities $1-p$ and p, respectively. For simplicity, we consider the same error rate p for both single- and two-qubit operations. For a single qubit, the dephasing error is a Z error and the depolarising error is a uniform mixture of X, Y, and Z errors. For two qubit gates, the dephasing error is a uniform mixture of errors: ZI, IZ and ZZ; and the depolarising error is a uniform mixture of the 15 errors: $IX, IY, IZ, XX, XY, \ldots, ZZ$. We add noise to every gate, ancilla initialisation and ancilla measurement.

When mixing the two error models together, we choose the depolarising model and the dephasing model with probabilities p_{depo} and p_{deph}, respectively. That is, the noise superoperator is $\mathcal{N} = (1-p)\mathcal{I} + p(p_{depo}\mathcal{E}_{depo} + p_{deph}\mathcal{E}_{deph})$ with depolarising noise operator $\mathcal{E}_{depo}$ and dephasing noise operator $\mathcal{E}_{deph}$, and $p_{depo} + p_{deph} = 1$. In the simulation, a gate is applied perfectly with probability $1-p$, otherwise an error is applied with either the depolarising model or the dephasing model based on the biased ratio. Note this ratio is not the actual ratio of the probabilities of X and Z errors. When the ratio is unity, i.e, when the dephasing model and depolarising models are applied with an equal chance, the probability for a Z error is roughly 2.8 times of that for an X error.

To further generalise our model, we also consider a quantum hardware with a modular structure, where local gates involved in error detection (green boxes in Fig. 3.2) may have a lower error rate than that of the long-range gates involved in the surface code parity checks (red boxes). Therefore we have two overall error rates, p_d and p_g, which we refer as the local error rate, and the global error rate, respectively. In summary, the noise superoperator of local and global errors are

$$\begin{aligned} \mathcal{N}_d &= (1-p_d)\mathcal{I} + p_d(p_{depo}\mathcal{E}_{depo} + p_{deph}\mathcal{E}_{deph}), \\ \mathcal{N}_g &= (1-p_g)\mathcal{I} + p_g(p_{depo}\mathcal{E}_{depo} + p_{deph}\mathcal{E}_{deph}), \end{aligned} \tag{3.1}$$

respectively. In the threshold calculation, the threshold is based on p_g, as it is no smaller than p_d and may thus be more dominant. In the following, we will consider scenarios with different ratios of global to local error rates p_g/p_d, and different ratios of depolarising to dephasing error rates p_{depo}/p_{deph}.

Simulation results

We numerically test the capacity of the concatenated code. Each simulation cycle follows the procedure as described before, including stabiliser checks, pairing of ancillae, correction of errors and finally determination of whether there is a logical error. To reduce correlated errors, the CNOT gates between the data and the ancilla qubits follow a "Z" shape, as pointed out in [105]. A given experiment is successful if it suffers neither a logical X nor a logical Z error, i.e. errors can be perfectly corrected. A Monte-Carlo simulation is applied, with each data point being the average result of at least forty thousand runs. Our main focus here is the threshold of the code under different circumstances.

Case 1: $p_{deph} = 1$, $p_{depo} = 0$, and $p_d \leq p_g$. To begin with, we consider the case with pure dephasing errors, i.e., $p_{deph} = 1$ and $p_{depo} = 0$. Since X errors do not occur in this scenario, we only apply X stabiliser checks for both the concatenated and the standard surface codes for a fair comparison. We first do not distinguish between the fidelity of local and global gates, i.e., with $p_d = p_g$. It is found that the threshold of the concatenated code is 1.42%, as shown as the first data point in Figure 3.4, higher than that of the standard surface code which is 1.20%. The result suggests that with only phase noise, the benefits obtained from the extra information exceeds the extra noise introduced from the error detection circuits.

We now move to the case that qubits exist in a modular structure such that certain short range two-qubit gates have higher fidelity than other longer range gates, i.e., $p_d \leq p_g$. As shown in Fig. 3.4, when we gradually increase the ratio p_g/p_d, the threshold error rate is observed to have a continuous gain. We see that the increase of the threshold slows down with a larger ratio p_g/p_d. However, if the error detection is perfect, the code should never fail, as no error is introduced to the second qubit (in the circuit in Fig. 3.2 (b)), while the error detection always identifies the error – it can only possibly be on the first qubit and can be simply corrected once it is found.

To further assess the performance of the concatenated code versus the standard surface code, we also compare each code's probability of successfully protecting its

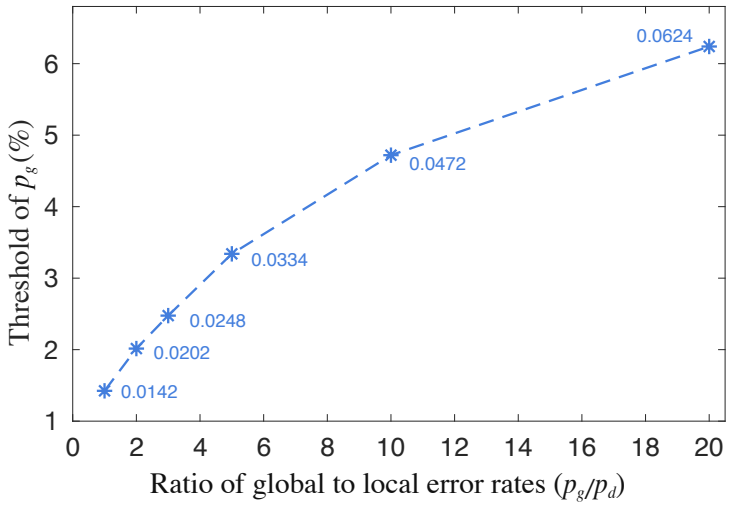


Figure 3.4: The increase in noise threshold as we alter the relative error rates for local error detection and global parity check, in a scenario with pure phase errors. The threshold error rate represents the error rate applied for global parity check. A larger ratio leads to an increase of threshold.

logical qubit given that each embodies the same number of physical qubits, e.g, with the same resource requirements. Here we consider the standard surface code with a size of $20 * 20 (= 400)$, and the concatenated code of size $14 * 14$ which actually requires 392 qubits when we allow for the additional resources needed for our phase error detect layer. Thus the two have nearly the same number of physical qubits. A total of $3 * n$ stabiliser cycles are performed, where n is the code size. The result is shown in Figure 3.5, where the y-axis 'success rate' is 1 - logical error rate, and the x-axis is the error rate in the parity check cycle. The orange curve represents the concatenated code where error rate for local error detection p_d is the same as the error rate in parity check p_g, while the yellow one is the same code but with $p_d = 0.5p_g$. We see that reducing the error rate for error detection yields a large gain in the success rate considering the same error rate of parity check cycle. Over the whole range, both the curves for the concatenated codes are superior to the blue curve, which corresponds to the standard surface code. The grey dashed curves from left to right indicate the thresholds for the blue, orange and yellow curves as references.

For the concatenated code in the simulations above, as previously discussed, the

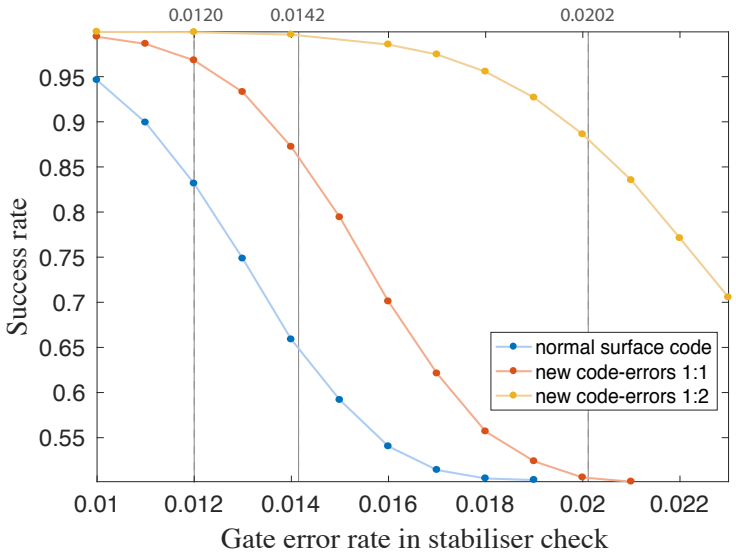


Figure 3.5: Code success rate as a change of the gate error rate applied in the parity check cycle under the circumstance of pure phase error. The blue curve represents a standard surface code with a size of 20*20. The orange and yellow curves refer to the concatenated surface code of size 14*14, with $p_d = p_g$ and $p_d = 0.5p_g$, respectively. The code sizes are chosen such to make the total number of physical qubits (almost) the same for the two different codes. The grey dashed lines (from left to right) denote where the threshold is, for the blue, orange and yellow curves, respectively.

weights for space and time are adjusted for the highest success rate. We do this empirically though a certain trend has been found. For convenience we make the weight w=1, and refer to t and z as time weight and space weight, respectively. Table 3.1 shows the weights used in the simulations. The precise reason why the optimal weights vary in this fashion is elusive, however, we found that the space weight is not independent as we introduce the cut down threshold as another variable into the decoder. In the calculation of the distance of the two syndrome events, if the distance found is higher than this threshold, the program will stop and make the distance infinity, as to not pair the two syndrome events. By doing so we shorten the time cost for each simulation run, and we found it also constrains the space weight — which should be smaller than the threshold so as to permit the connection of two data qubits that are not in the list.

p_d/p_g	1	1/2	1/3	1/5	1/10	1/20
Space weight z	12	12	4.5	4	3	3
Time weight t	3.5	3.5	0.85	0.5	0.4	0.35
Cut down threshold	$3n+1$	$3n+1$	$2n+4$	$n+10$	$n+6$	$n+5$

Table 3.1: Different weights and cut down thresholds used in the simulations. n represents the size of the code.

Relative strength		0.2	0.3	0.5	0.7	0.9	1.0
F	standard code	9	6	4	4	3	3
	$p_d = 1/3p_g$	5	3	2	2	2	1
	$p_d = p_g$	4	3	2	1	1	1
Z/X error rate		9.918	6.738	4.352	3.597	2.958	2.786

Table 3.2: The optimal frequencies of X/Z checks for the concatenated and the standard codes as the relative strength of depolarising and dephasing noise changes. F refers to frequency, e.g. how many rounds of X stabiliser checks are applied before one round of Z stabiliser checks. The Z/X error rate is the relative probability for Z and X errors happening in the standard surface code, when the frequencies of X and Z checks are as above.

Case 2: $p_{depo}/p_{deph} \in [0, 1]$**.** So far our comparisons made in the previous simulations are for gates with pure phase noise, now we move on to a more realistic scenario where both phase and bit flip noise is present but the former still has greater severity. In this case Z stabiliser checks are required to detect X errors, and as mentioned above, in a biased environment, we may employ multiple X stabiliser cycles for each Z stabiliser cycle. The reason to do so stems from the fact that the overall success rate is the product of the success rates for logical Z and X errors, thus the highest success rate is achieved when the success rate for the two types of logical errors is the same. For both the standard surface code and the concatenated code, we determined the optimal pattern by trialing different frequencies and selecting the one that leads to the highest success rate. As shown in Table 3.2, the frequency F refers to the rounds of X stabiliser checks before one round of Z stabiliser check is applied. Unsurprisingly as the relative strength of depolarising error reduces, the frequency increases. With the standard surface code, the optimal frequency is roughly the same as the rounded relative probability to

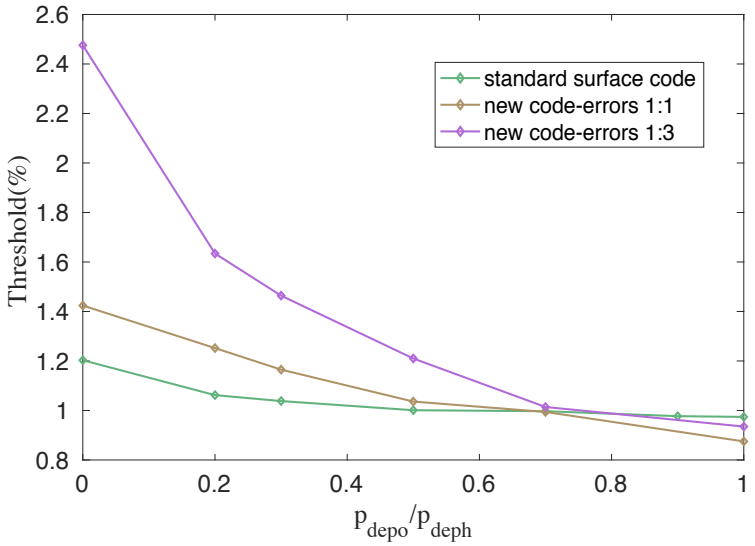


Figure 3.6: The dependence of threshold on the relative strength of dephasing and depolarising errors. The 'threshold' quantity along the y-axis is the total probability of a physical error; each such error is assigned to one of the two possible models with relative probabilities given by the x-axis. The purple (brown) curve corresponds to the concatenated code where the gates in error detection have three times the error rate (the same error rate) as the gates in parity check. For reference the standard surface code is plotted as the green curve.

find a phase and bit-flip error, as indicated in the last row of the table. Note the frequency may not be strictly ideal, since we do not consider the case when it is a non-integer (e.g. 3/2 means three rounds of X checks followed with two rounds of Z checks), as it may change the error pattern significantly. As for the concatenated code, smaller frequencies are observed as the p_d/p_g increases. This can be explained by the fact that as p_d becomes larger, more errors are introduced during the error detection cycle which increases the logical error rate. Given that the total number of stabiliser checks is fixed, a smaller number of X checks is therefore more beneficial.

Note that for accurate simulations, we need to apply a deep enough simulation (i.e. sufficient number of stabiliser cycles) to ensure that a large number of X errors occur in each run – otherwise, the results would be skewed by the 'edge effect' that the simulation starts from a clean, error-free state.

The data plotted in Fig. 3.6 shows the threshold change as we gradually increase the relative strength of the depolarising versus the dephasing error model. It is

not surprising to see that all the three curves representing the threshold, which is the error rate in the stabiliser check cycle, decline as the ratio rises. The green curve referring to the standard surface code starts from the lowest value but goes down slowly. The purple curve stands for the concatenated code whose gate error rate in the error detection is one third of that in the stabiliser check, i.e., $p_d = p_g/3$. We see that it decreases fast and crosses the green curve when the ratio is roughly 0.8, indicating that the concatenated code is no longer beneficial when the strength of depolarising error is higher than this value. The concatenated code which has an equal probability for errors happening in error detection and parity check, $p_d = p_g$, is plotted as the brown curve, which as we would expect lies between the other two curves.

3.3 Conclusion and outlook

In this chapter we describe a variant of the standard surface code by concatenating it with a simple two-qubit phase detection code. We developed a new decoder that efficiently takes into account the likely location of errors, obtained from the error detection stage, with a modified minimum weight perfect matching algorithm. The concatenated code substantially outperforms the canonical surface code (by allowing a significantly higher threshold) for the common scenario where phase error dominates. The concatenated code becomes yet more advantageous when one considers the likely scenario that local gates (among the groups of three qubits associated with error detection) may have a lower noise rate compared to the gates that link between such groups. Such a scenario would correspond to modular hardware with exquisite local quantum control and more noisy global links. Modular quantum computing has been extensively studied and shown to be feasible even with near-term hardware [106, 107].

It remains an interesting question to consider other codes for local modules. For instance, one can consider combining the decoding algorithm described in this chapter with the four-qubit error detection code, as has been studied by Criger and Terhal [84], or a three-qubit phase-error correction code. For the former case, as the

four-qubit error detection code detects both phase and bit-flip errors, the modular architecture could work with any error model. For the latter case which is ideal with a phase-dominated error model, errors can be more accurately located, even with a high chance to be exactly corrected in the lower-level code. However, as a tradeoff, more noise is introduced into the system due to a larger number of gates applied. It is an open question whether general concatenated codes incorporated with the customised decoder will outperform the standard surface.

4

Error correction with small codes

Contents

In Chapter 3 we described a concatenated topological code for quantum systems with modular structures and experiencing biased noise. While a high threshold is found, limited by the currently inaccurate fabrication process and imperfect gate controls, a large-scale device with gate error rate below the threshold is still beyond reach. Small error correction codes thus stand out as the most promising choice to protect near-future quantum computers from corruption due to noise. As noted in Chapter 2, in recent years, rapid progress has been reported in the implementation of small quantum codes, across platforms as diverse as ion traps [14, 108, 109], superconducting qubits [110–113], and crystal defect systems [114].

Several small error correction codes are particularly popular with outstanding

properties. The five-qubit 'perfect' code, proposed by Laflamme et al. in 1996 [115], is the smallest code that is capable of correcting an arbitrary physical error. Around roughly the same time the seven-qubit Steane code was proposed by Steane [116]. The Steane code is also known as the smallest 2D colour code, which has low-weight and symmetrical X and Z stabilisers. Another popular example is the nine-qubit 'rotated' surface code. It is the smallest surface code that is capable of correcting any arbitrary physical error.

A comprehensively successful quantum code will have been achieved when one can demonstrate a full set of quantum operations on encoded qubits with a fidelity that exceeds that of the best possible unencoded physical qubits [13]. However this criterion is very challenging to achieve; it means 'beating' the superb fidelities exceeding 99.9% that can now be achieved with single physical qubits [117–119]. Even the task of achieving a superior coherence time with a memory based on an encoded qubit, versus a single physical qubit, is not trivial. Individually controlled physical qubits can persist for the order of a minute when not actively manipulated [119], or 10 minutes using dynamical decoupling [120].

It is therefore interesting to find a measure for the efficacy of memories based on small quantum codes, using which we can identify reasonable milestones for near-future experimental realisations. Equally importantly we wish to be able to fairly compare memories based on platforms that might have very different inherent timescales. In this chapter, we introduce a measure called 'integrity' to benchmark the performance of small error correction codes on general hardware systems. We show that this measure can be related to other commonly used measures including the *trace distance* [121, 122], the *logical fidelity* [113, 123] and the *pseudo-threshold* [124, 125], but with some extra advantages.

This chapter is organised as follows. We give a introduction to several popular small error correction codes in Sec. 4.1, including the five-qubit code (4.1.1), the seven-qubit code (4.1.2) and the nine-qubit code (4.1.3). In Sec. 4.2 we introduce a measure called the *integrity* of the logical qubit. We give the definition of integrity in 4.2.1, based on which we offer a set of four milestones to benchmark

the performance of an encoded quantum memory. Intensive numerical results are given in 4.2.3. We finally conclude in Sec. 4.3.

The concept of integrity, and the investigation of its basic properties, was achieved jointly in discussions with all the authors of Ref. [126]. All numerical analysis and investigation were performed by the present author. Additionally, the present author led the effort to find qualitative explanations of the features observed from the numerics.

4.1 Small error correction codes

4.1.1 The five-qubit code

The [[5,1,3]] five-qubit code encodes one logical qubit into five physical qubits. The stabiliser set for this code contains four operators:

$$\mathcal{S}^1 = XZZXI, \quad \mathcal{S}^2 = IXZZX, \quad \mathcal{S}^3 = XIXZZ, \quad \mathcal{S}^4 = ZXIXZ. \tag{4.1}$$

It has transversal logical X and Z operators (i.e. the logical operator is simply physical operators applied to each individual data qubit): $\bar{X} = XXXXX$, $\bar{Z} = ZZZZZ$. Note that both the stabilisers and the logical operators are not unique. Any set in which any two stabiliser operators commute with each other and all of them commute with the codestates is valid, any product of the logical operators with one of the stabiliser operators generates new logical operators. The four stabiliser checks generate in total $2^4 = 16$ different outcomes, corresponding exactly to the 16 situations that can happen (X or Y or Z flip occurs to one of the five data qubits plus the case with no error happening), provided that there is at most one error occurring to the logical qubit. Therefore, by referring to the error syndrome, a single error can be located and fixed simply by applying one gate to the data qubit to flip it back.

As any codeword state is the +1 eigenstate of each stabiliser, the encoding of a logical state from an arbitrary physical state requires the projection of the initial state of the data qubits into eigenstates of stabiliser operators. A general method is to conduct a whole set of stabiliser checks followed by error correction based on the outcome of the measurement results [24]. However, one needs to apply a large

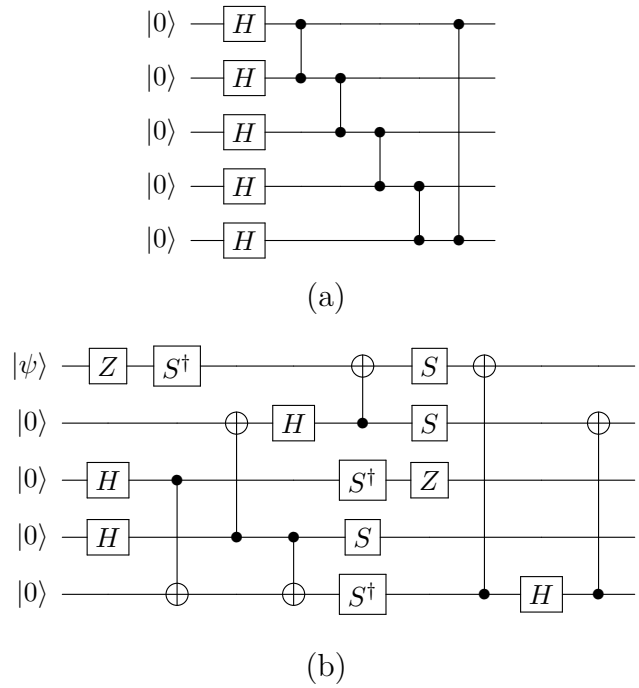


Figure 4.1: Encoding circuits for the five-qubit code. (a) Circuit to encode a logical minus state $|-\rangle_L$ [126]. (b) Circuit to encode an arbitrary logical state [127].

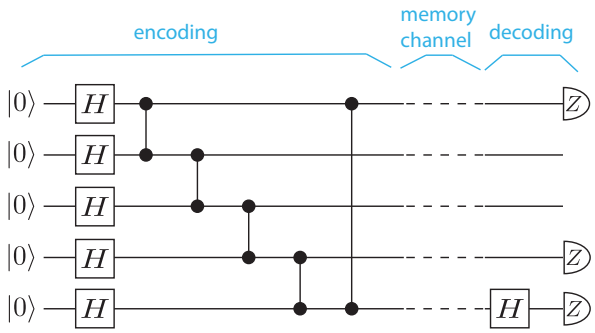


Figure 4.2: The circuit encodes and decodes a logical $|-\rangle_L$ with the five-qubit code. The decoding of the logical qubit is by measuring three data qubits and comparing the parity.

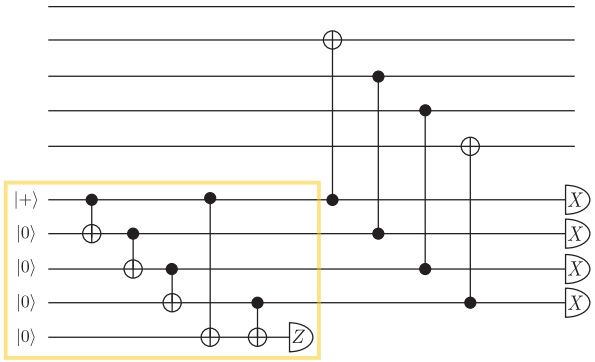
number of gates through the whole set of stabiliser checks, making the approach less favourable. Alternatively, there are other encoding circuits with fewer gates. Fig. 4.1(a) encodes a logical minus state $|-\rangle_L$ with five single-qubit gates and five CPhase gates [126]. To encode an arbitrary logical state, an extra two-qubit gate is required; a suitable circuit is shown in Fig. 4.1(b), which was recently reported in Ref. [127]. These circuits are compact, but not fault-tolerant.

To decode a logical state back to a physical state, one can simply apply the

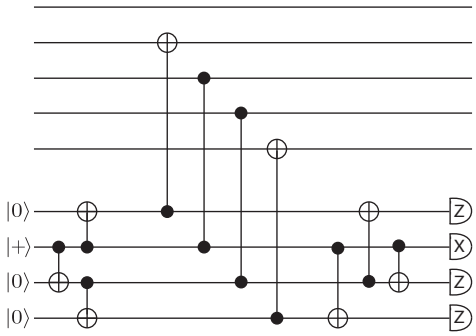
same circuit as the encoding circuit but in the reverse order. If we simply wish to measure the encoded qubit, an alternative fault-tolerant method for logical states $|0\rangle_L, |1\rangle_L, |+\rangle_L, |-\rangle_L, |+i\rangle_L, |-i\rangle_L$ is to measure the transversal logical operator (which makes no change to the logical state) and apply post-processing on the measurement outcomes. One example is shown in Fig. 4.2, which encodes and decodes a logical minus state $|-\rangle_L$, where the parity of the measurement outcomes determines whether a harmful error has occurred.

The stabiliser checks can be performed by applying the corresponding controlled-Pauli operators from the ancilla to the data qubits, as described in Section 2.2.1. However, with only one ancilla, this approach is not fault-tolerant, as any error occurring to the ancilla qubit during the stabiliser check process may propagate to the data qubits and corrupt the code. In order to avoid error propagation, the two-qubit gates can be applied transversally, with more ancilla qubits required. DiVincenzo and Shor proposed the first fault-tolerant approach for stabiliser checks [128]. As shown in Fig. 4.3(a), in total five ancilla qubits are needed. The first four prepare a cat state, and the fifth one determines whether any harmful error has occurred during the ancilla preparation process: if so, the qubit will be measured '1' and the ancilla state will be re-prepared until the fifth ancilla qubit measures to be '0'. After applying the two-qubit gates transversally between the data and ancilla qubits, all the first four ancillae are measured in the X basis: the parity of the sum of the measured outcomes gives the same result as in the non-fault-tolerant case with only one ancilla qubit, if with perfect measurement. Note that in order to avoid introduction of new errors with faulty measurement result, at least two rounds of the whole stabiliser check cycles are to be performed and compared, and if the results don't agree, a third round is required.

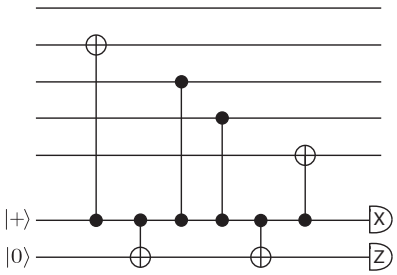
Followed by the Shor's approach, DiVincenzo and Aliferis proposed an alternative method which requires only four ancilla qubits [129], as shown in the Fig. 4.3(b). The four ancilla qubits are also prepared to be a cat state with no error check, which is however done through post selection process. One can easily find that bit flip errors propagating to three data qubits is equivalent to just a single error



(a)



(b)



(c)

Figure 4.3: Different schemes for fault-tolerant stabiliser checks with the five-qubit code. (a) DiVincenzo and Shor's approach with five ancilla qubits [128]. The yellow rectangle depicts the area where a cat state is prepared and verified to be with no harmful errors. (b) DiVincenzo and Aliferis's approach with four ancilla qubits and post processing [129]. The same as (a), the four ancilla qubits are prepared to be a cat state, while there is no extra qubit used for verification, which is instead realised by checking the measurement outcomes of the four ancilla qubits in the end. (c) A recent new approach proposed by Chao and Reichardt [130]. Only two ancilla qubits are required, with the first one functions the same as in the non-fault-tolerant case, and the second used for determining whether any harmful error happens.

propagating to the fourth qubit, as errors occurring to all four of them is equivalent to a stabiliser operator. Therefore, only errors propagating to two of the data qubits are harmful, and if that happens, the measured outcomes of the first, third and fourth of the ancilla qubits will all be 1, thus the harmful errors can be fixed by applying two gates to the corresponding qubits.

Recently, a lighter approach was introduced by Chao and Reichardt [130]. Contrary to the methods above, it requires only one more qubit, called the 'flag' qubit, to achieve fault tolerance. As shown in Fig. 4.3(c), the first ancilla qubit acts as the normal ancilla, the same as in the non-fault-tolerant case, and the second ancilla qubit is used to detect any weight-2 error, if happening to the data qubits. In the case with no weight-2 error happening, the flag qubit will be measured to be '0', while any such error will turn the flag qubit into '1'. Upon detection, a whole set of stabiliser measurements will be performed and the weight-2 error can be located and corrected based on the unique error syndrome. With fewer physical resources required, this approach is a very nice step forward.

4.1.2 The seven-qubit Steane code

The [[7,1,3]] seven-qubit code, also known as the Steane code, is a widely used error correction code. Its stabiliser set is:

$$\begin{aligned} &\mathcal{S}^1 = IIIXXXX, \quad \mathcal{S}^2 = XIXIXIX, \quad \mathcal{S}^3 = IXXIIXX, \\ &\mathcal{S}^4 = IIIZZZZ, \quad \mathcal{S}^5 = ZIZIZIZ, \quad \mathcal{S}^6 = IZZIIZZ. \end{aligned} \tag{4.2}$$

One can find that the generators of this stabiliser set contain only X or Z operators, enabling the code to correct X and Z errors separately. Therefore, the Steane code is maximally capable of correcting one bit flip error plus one phase flip error. Besides, the Steane code belongs to the group of Calderbank-Shor-Steane (CSS) codes [18, 116], which allow transversal operations applicable for the whole Clifford group, offering the Steane code an additional advantage.

Fig. 4.4(a) shows the encoding circuit to prepare a logical zero state $|0\rangle_L$ [131]. This circuit can also be fault-tolerant given three additional two-qubit gates and one ancilla qubit [132]. As shown in Fig. 4.4(b), the circuit is fault-tolerant if the

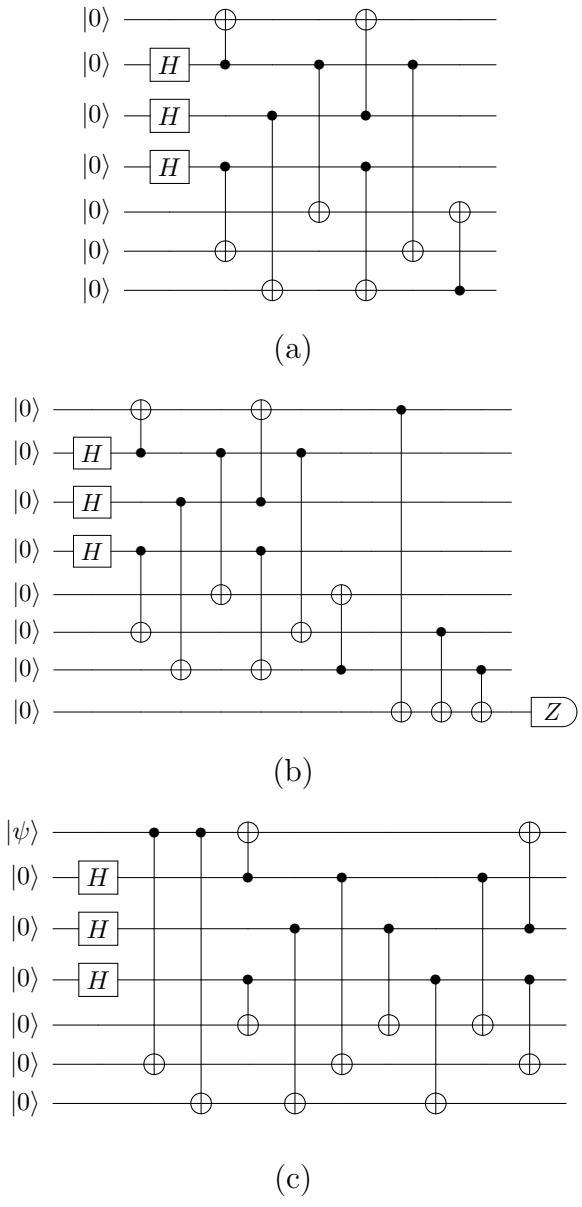


Figure 4.4: Encoding circuits for the seven-qubit code. (a) Circuit to encode a logical zero state $|0\rangle_L$[131]. (b) Circuit to fault-tolerantly encode a logical zero state $|0\rangle_L$ [132]. Three CNOT gates are applied to the ancilla qubit, which is then measured in $\{0, 1\}$ basis. If the measurement result is 1, indicating more than one bit-flip error occurring, the whole circuit is abandoned and restarted from the beginning, until the ancilla qubit is measured to be 0. (c) Circuit to encode an arbitrary logical state [132].

ancilla is measured to be 0, or otherwise, the circuit can be abandoned. To encode an arbitrary logical state, one may use the (non-fault-tolerant) circuit shown in Fig. 4.4(c), which has 11 CNOT gates [132].

The fault-tolerant approaches for stabiliser checks with the five-qubit code, as

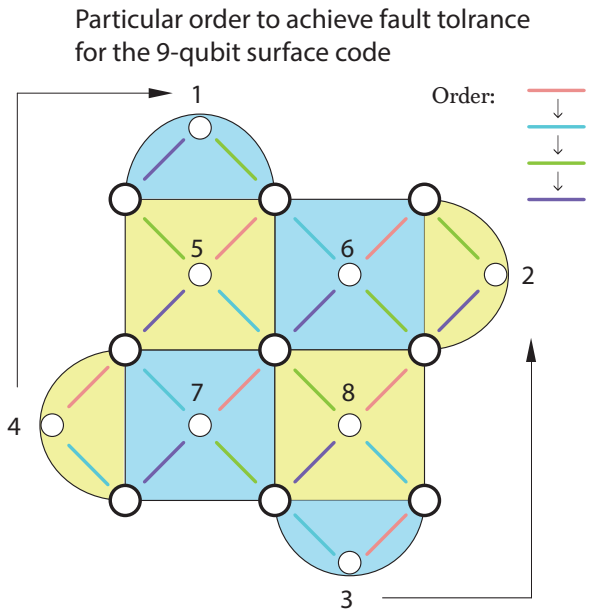


Figure 4.5: The nine-qubit rotated surface code. Stabiliser measurements of ancillae follow a particular order to achieve fault tolerance. The large circles stand for the data qubits and the small circles are ancillae. The stabiliser measurement should follow the order denoted by the colour orange, blue, green and finally purple. Since the ancilla labelled 3 can physically act as the ancilla labelled 2 after finishing the measurement in the blue half-circle and that also works for ancilla 4, which can act as ancilla 1 after the measurement in the yellow half-circle, in total six ancillae are required to demonstrate fault tolerance.

described before, can also be applied to the seven-qubit code. Similarly, we can also realise fault-tolerant measurement by measuring all the seven data qubits in the end. The parity of subsets $\{4,5,6,7\}$, $\{1,3,5,7\}$, $\{2,3,6,7\}$, where 'n' refers to the n-th qubit in Fig. 4.4, should be the same if no harmful physical errors have occurred.

4.1.3 The nine-qubit surface code

An important instance of the surface code is the rotated 9-qubit code, as shown in Fig. 4.5 [105]. It is the smallest surface code that is capable of correcting an arbitrary single qubit error. Compared with the standard surface code structure introduced in Chapter 3, it is 90° rotated, with the data qubits (the big circles) at the edge and the measurement qubits (the small circles) at the centre of the plaquettes. The eight stabilisers of this code include four $XXXX$ (yellow) and $ZZZZ$ (blue) operators and four XX and ZZ operators applied to the data qubits

in the boundary. Note here that though eight ancilla qubits are drawn in the figure, one can use one ancilla qubit to perform all stabiliser checks by reusing the same qubit after each measurement. However, by doing so a single error can propagate through the two-qubit gates. A fault-tolerant manner can be realised by applying stabiliser checks following a particular order, as illustrated in the figure. All the measurements are separated into four steps, as coloured and ordered in red, blue, green and purple. CNOT or CPhase gates of the same colour are to be performed simultaneously, thus at least six ancilla qubits are needed. The qubits that are related with arrows shown in the figure are qubits that can be reused to save the total number of qubits: after stabiliser checks labelled in red and blue, the qubits numbered 4 and 3 are to be measured and reset before they act as the qubits 1 and 2 respectively. By doing so, harmful errors that can spread to data qubits will be naturally eliminated [105, 133].

4.2 An integrity to benchmark the quantum channel

Having introduced the most widely-adopted small error correction codes, in this section we introduce the integrity as a measure to benchmark performance of the codes on quantum memories. This section, and the remaining sections of this chapter, therefore report original research. We motivate the notion of integrity through its intuitive relation with the trace distance, logical fidelity and pseudo threshold. Four milestones are introduced, with an increasing difficulty to realise. Numerical simulations are performed to show how these milestones can be achieved by quantum memories encoded with the small error correction codes introduced in the last section. Comparison of the codes are also made. The section concludes with discussions on applying integrity in a realistic scenario.

4.2.1 Definition of integrity

One can think of any memory as a channel for communicating information from the present ($t = 0$) to a specified future time ($t = \tau$). We could compare the state

at $t = 0$ with the state at $t = \tau$, using either the fidelity or the trace distance. Let us briefly review these two quantities.

There are two definitions of fidelity commonly used in the literature; one is the square of the other. Here we use the squared quantity, formally defining fidelity as

$$\mathcal{F}(\rho_0, \rho_1) = \left\| \sqrt{\rho_0}\sqrt{\rho_1} \right\|_{\text{tr}}^2. \tag{4.3}$$

This definition uses the trace norm, itself defined as

$$\|\sigma\|_{\text{tr}} = \text{Tr}\left(\sqrt{\sigma^\dagger \sigma}\right), \tag{4.4}$$

and this is also the sum of the singular values of σ.

In the case that ρ_0 is a pure state $|\psi_0\rangle \langle\psi_0|$, the fidelity then has a simple physical interpretation: if we measure state ρ_1 in a basis where one of the possible outcomes is $|\psi_0\rangle$, the fidelity is precisely the probability of this outcome. When both states are pure, we have simply

$$\mathcal{F}(\psi_0, \psi_1) = |\langle\psi_0|\psi_1\rangle|^2.$$

While the fidelity measures the similarity of two states, the trace distance measures the degree to which two states differ. It is defined as

$$\mathcal{D}(\rho_0, \rho_1) = \frac{1}{2}\left\| \rho_1 - \rho_0 \right\|_{\text{tr}}. \tag{4.5}$$

Ranging from 0 to 1, the trace distance tells us the probability that two states ρ_0 and ρ_1 could be told apart by an ideal experimentalist. Suppose that we present to an experimentalist, Bob, a theoretical description of both ρ_0 and ρ_1, and we also prepare a physical system in one of these two states (with a 50/50 prior probability) and present this to the Bob. He must guess whether the physical state is ρ_0 or ρ_1. Using his optimal strategy, his probability p_g of guessing correctly is simply

$$p_g = \frac{1}{2} + \frac{1}{2}\mathcal{D}(\rho_0, \rho_1). \tag{4.6}$$

We will make extensive use of this idea presently.

Suppose that some qubit with density matrix ρ is to be stored in a code-based memory channel for a specified period of time. We will use the symbol Φ to denote the memory channel. The initial state ρ maps to the final state $\tilde{\rho}$ through this process:

- **Setup:** At $t = 0$ Alice (taken to be perfect) encodes the single qubit ρ into an n-qubit logical code: $\rho_n = \mathsf{E}(\rho)$ where E is the encoding map.
- **The memory channel:** Evolution and degradation of the logical qubit occurs while it is stored. This may include the effects of actively applied error correction cycles (involving a non-ideal agent, whom we label 'Igor' and discuss presently). We have $\rho'_n = \mathsf{N}(\rho_n)$ where N is the noise map.
- **Conclusion:** At $t = \tau$, Bob (taken to be perfect) performs an error correction cycle, and then reverses Alice's encoding process to obtain a single physical qubit: $\tilde{\rho} = \mathsf{D}(\rho'_n)$ where D is the decoding map.

We can write the entire channel as $\Phi = \mathsf{D} \circ \mathsf{N} \circ \mathsf{E}$, thus incorporating Alice's encoding E, the noise N, and Bob's decoding D. For an initial concept of an ideal memory as one causing no change at all, we would desire $\Phi(\rho) = \rho$, and so (for example) $1 - \mathcal{D}(\rho, \Phi(\rho))$ could suffice as a good metric for the performance of our memory. However, certain changes are in fact harmless and do not practically reduce the quality of a memory (e.g. $|+\rangle_L$ is immune to X errors.). Therefore, instead of focusing on the changes suffered by the state of a single logical qubit between $t = 0$ and $t = \tau$, we can instead focus on the idea that a memory should preserve the distinguishability of different states.

Consider two pure states $\psi = |\psi\rangle\langle\psi|$ and $\psi_\perp = |\psi_\perp\rangle\langle\psi_\perp|$ which are orthogonal to one another. Orthogonal states have trace distance of unity, since they can certainly be told apart. Let Alice choose ψ at random, uniformly from all possible single-qubit states, and then opt to encode either ψ or instead the antipodal state $\psi_\perp$. Then $\Phi(\psi)$ or $\Phi(\psi_\perp)$ will describe the state after it has passed through the memory channel and been decoded by Bob. If the channel is such that it causes the same fixed evolution to occur to any (logical) state, or indeed if it has introduced errors but they are correctable, then these states will still be completely distinguishable – they will still have trace distance equal to unity. Therefore we define the integrity of the memory as

$$\boxed{\mathcal{R}(\Phi) = \min_{\psi} \mathcal{D}\Big(\Phi(\psi), \Phi(\psi_\perp)\Big).} \tag{4.7}$$

Note that we take the minimum over all possible choices of ψ made by Alice. We do this to account for the fact that certain memory channels may have no detrimental effect on special choices of the state, as for example a dephasing channel leaves $|0\rangle$ and $|1\rangle$ unchanged. To provide a measure which guarantees at least some quality of storage for all states, we consider the performance in the worst case.

The integrity of the memory has a highly intuitive and natural meaning through the following scenario: We suppose that Bob initially knows nothing about Alice's choice of qubit to encode, but after Bob has completed his decode process to obtain the single qubit we then describe to him two choices: either Alice's initial qubit was ψ or it was $\psi_\perp$. Bob then makes a measurement of his choice to try to determine whether it is $\Phi(\psi)$ or $\Phi(\psi_\perp)$ that he has received. The integrity $\mathcal{R}(\Phi)$ tells us Bob's probability p_g of guessing successfully according to

$$p_{g,\text{worst}} = \tfrac{1}{2} + \tfrac{1}{2}\mathcal{R}(\Phi) \quad \text{and so} \quad \mathcal{R}(\Phi) = 2p_{g,\text{worst}} - 1. \tag{4.8}$$

Here the label 'worst' reminds us that this is the lowest success probability, i.e. when the options ψ, $\psi_\perp$ are the ones least well preserved by the memory (if indeed there is any variation). The integrity therefore describes the best possible guarantee that can be made on how well a memory preserves the distinctiveness of different states.

For some memories Φ (although not for all conceivable memories) Bob's correct strategy for his final step is the obvious one: just measure in the basis $\{|\psi\rangle, |\psi_\perp\rangle\}$ and make the guess correspondingly. Then Bob's success probability is simply the fidelity of the state $\Phi(\psi)$ with respect to Alice's initial state ψ, since the fidelity of any state with respect to a pure state is the probability of obtaining that outcome in a measurement (as we remarked following Eq. (4.3)). So in this case Bob's probability of guessing correctly is simply $p_g = \mathcal{F}(\psi, \Phi(\psi))$. Moreover his worst performance is

$$p_{g,\text{worst}} = \min_{\psi} \mathcal{F}(\psi, \Phi(\psi)). \tag{4.9}$$

But given Eq. (4.8) we can now offer a precise meaning to the idea of "the (worst case) fidelity of a logical qubit stored in the memory" for any channel where Bob would opt to measure in the basis $\{|\psi\rangle, |\psi_\perp\rangle\}$. For such a channel,

$$F_{\text{logic}} = \tfrac{1}{2} + \tfrac{1}{2}\mathcal{R}(\Phi). \tag{4.10}$$

Loosely, F_{logic} is the fidelity after we project into the logical subspace of the code with a perfect round of error correction. For memory channels with sufficiently complex noise maps $\mathcal{N}$ that Bob's choice of measurement basis would not be $\{|\psi\rangle, |\psi_\perp\rangle\}$, the very idea of the "fidelity of a logical qubit stored in the memory" becomes ill defined. Thus, integrity is a general measure which relates to the notion of logical fidelity when the latter notion makes sense. However integrity remains well-defined and meaningful even when the logical fidelity does not: it is the more general and robust concept.

Importantly, it is eminently practical to directly measure integrity in an experimental setting. Notice that although the definition Eq. (4.7) refers to two different states, we would evaluate the integrity $\mathcal{R}$ through a series of single uses of the memory – we simply follow our scenario described above involving Bob guessing between options and employ Eq. (4.8). Consequently the costly process of performing full state tomography is not required.

4.2.2 Milestones towards successful error correction

Armed with this notion of the integrity $\mathcal{R}$ of a memory channel Φ, we now identify milestones towards the goal of superior code-based quantum memories. For convenience of exposition we may imagine that a third party, besides Alice and Bob, is responsible for the cycle(s) of error correction performed during the memory period: we use the name Igor after the famous fictional lab assistant. We initially focus on the case where at most one error correction cycle is used during the entire period τ where memory operates, i.e. in between Alice ($t = 0$) and Bob ($t = \tau$). We therefore now specify Step 2 of Table 4.1 in more detail, setting it out in Table 4.2. The key idea will be to compare the integrity of the memory channel without error correction (no Igor) to the case with EC (Igor participates) and determine whether the latter is superior.

Let us use the symbol Φ^m to label the memory channel when Igor performs m rounds of error correction, so that Φ^0 labels the channel when no QEC is performed (i.e. noise sources are purely environmental). Then we say that a round of error

	Theoretical protocol for measuring integrity
1a 1b	Alice (perfect) prepares a single qubit state $\|\psi\rangle$ or $\|\psi_\perp\rangle$. Alice perfectly encodes it into the n physical qubits.
2	From $t = 0$ to τ the n qubits are in the memory; noise occurs from environment and possibly error correction.
3a	Bob (perfect) performs error correction on the n qubits, then decodes (inverse of 1b) the state to a single qubit.
3b	Bob is told Alice's qubit was *either* $\|\psi\rangle$ *or* $\|\psi_\perp\rangle$. He measures his qubit and guesses, success probability p_g.

Table 4.1: Evaluating the integrity of a memory system.

	Without Error Correction: Memory Φ^0
2a	The n physical qubits are subjected to environmental noise for a time τ.

	With Error Correction: Memory Φ^1
2a	The n physical qubits are subjected to environmental noise for a time $(\tau - \delta)/2$.
2b	Igor is asked to apply a full round of *imperfect* error correction, taking time δ.
2c	The n physical qubits are subjected to environmental noise for a further time $(\tau - \delta)/2$.

Table 4.2: Expanding on Step 2 of Table 4.1 when we wish to assess the benefits of error correction.

correction is beneficial if Bob's probability of subsequently discriminating the state correctly is higher when Igor indeed performs that round, i.e. when

$$\mathcal{R}(\Phi^1) > \mathcal{R}(\Phi^0). \tag{4.11}$$

This seems entirely straightforward but there is a subtlety: the question of whether Eq. (4.11) is satisfied will depend on the duration τ of the memory channels (here we would naturally set the same τ for both Φ^1 and Φ^0 for a fair comparison). Since we are interested in defining a first milestone for experimental efforts, we say that error correction can be beneficial if

$$\mathcal{R}(\Phi^1) > \mathcal{R}(\Phi^0) \quad \text{for some value of } \tau. \tag{4.12}$$

We will refer to the challenge of satisfying Eq. (4.12) as milestone **M1: Beneficial error correction**.

One might wonder if we should consider a stronger condition: $\mathcal{R}(\Phi^1) > \mathcal{R}(\Phi^0)$ for all values of the common duration τ. It is interesting and important to note that this condition will be impossible to satisfy in physical devices. Assuming environmental decoherence is a continuous process, we will have $\mathcal{R}(\Phi^0) \to 1$ as $\tau \to 0$ while $\mathcal{R}(\Phi^1) \to \epsilon$ as $\tau \to 0$, where ϵ is non-zero and is related to the inevitable imperfections in the cycle of error correction.

While Eq. (4.12) is an important first milestone for an error-correcting quantum memory, further milestones can also be identified. For a sufficiently high performing Igor, and a long memory duration τ, it will be beneficial to have multiple rounds of correction. This will be a signature of further progress toward a practical quantum memory. We would then find that

$$\mathcal{R}(\Phi^m) > \mathcal{R}(\Phi^{m-1}) \quad \text{for some value of } \tau. \tag{4.13}$$

Here we understand this need only be satisfied for some particular $m > 1$. We will refer to the challenge of meeting the condition in Eq. (4.13) as milestone **M2: Beneficial multi-round error correction**.

Equations (4.12) and (4.13) involve comparisons between memories which both employ encoded qubits. There is of course another type of comparison we can make, one which directly addresses the question of whether it is 'worth' using encoded memories at all: we should contrast such a memory channel to a simple, single qubit memory. Let us use the symbol Θ for that memory channel. We can consider its integrity $\mathcal{R}(\Theta)$ easily enough. Alice prepares a single qubit, again choosing between ψ and $\psi_\perp$, but does not encode it into multiple qubits. It exists as a memory from $t = 0$ to $t = \tau$, and finally Bob receives it but of course he has no decoding to do. The qubit he receives, $\Theta(\psi)$ or $\Theta(\psi_\perp)$, differs from Alice's qubit only because of environmental noise. But as before Bob must measure it to guess between the two possible states, and as before his probability of success is simply $\frac{1}{2} + \frac{1}{2}\mathcal{R}(\Theta)$. For our actively-corrected encoded memory to 'beat' the simple single-qubit memory, we require

$$\mathcal{R}(\Phi^m) > \mathcal{R}(\Theta) \;\; \text{for } \textbf{some } \tau_\Theta, \text{ while using } \tau_\Phi = \alpha\tau_\Theta. \tag{4.14}$$

Here we require only that this is true for some specific value of $m > 0$ (it seems likely that $m = 1$ would be the first demonstration). Note the more complex condition on the channel durations. In Eq. (4.12) and Eq. (4.13) it was clear that the duration τ of the two memory channels should be the same for a fair comparison. This is not necessarily true of the Eq. (4.14) since one can argue that the meaningful time scale for a quantum memory is not 'wall clock' time but rather the time required to perform an active gate operation. Depending on the hardware platform and architecture, the time required to perform a gate operation on an encoded qubit may be longer than the time to perform the equivalent operation on an unencoded qubit. This would then suggest that τ_Φ should be longer than τ_Θ, and the factor $\alpha \geq 1$ is included in Eq. (4.14) to reflect this. We will refer to the challenge of satisfying Eq. (4.14) as milestone **M3: Beneficial encoded memory**.

Here we can make contact with the pseudo-threshold, which is typically used in the context of concatenated codes, where there may be several levels of concatenation required before error rates fall sufficiently for deep quantum algorithms. In the present context, we restrict our interest to the lowest level of concatenation where a process involving unencoded qubit(s) is compared to a process with a single level of encoding. The pseudo-threshold has been surpassed if a circuit performs to a higher standard with the encoded qubits, i.e. logical qubits, versus using the physical qubits directly. One might demand that for a complete universal set of operations, each operation at the encoded level outperforms the analogous operation using unencoded qubits. Alternatively one might speak of the pseudo-threshold for a specific operation, such as a single-qubit gate, a CNOT operation, a measurement or indeed a memory. In essence Eq. (4.14) represents the (lowest tier) pseudo-threshold for memory, i.e. for the identity operation in a circuit.

Assuming that Eq. (4.14) can be satisfied, there is a higher goal which might be achieved namely

$$\max_m \mathcal{R}(\Phi^m_{\mathrm{QEC}}) > \mathcal{R}(\Theta) \quad \text{for } \mathbf{all}\ \tau_\Theta, \text{ and using } \tau_\Phi = \alpha\tau_\Theta. \tag{4.15}$$

Here the maximum is over a family of memory channels having the same duration τ_Φ but with differing numbers of error correction cycles m. Importantly, we permit $m = 0$. If this condition is satisfied, it means that for any desired duration we can sustain our encoded quantum memory at a higher integrity than a single physical qubit memory. We do so by applying a suitable number of error correction cycles. Moreover this is true even allowing for the factor α discussed above. This is therefore the 'gold standard' for demonstrating a quantum memory and it is the most challenging of the criteria we have presented in this section. We will refer to the task of satisfying Eq. (4.15) as milestone **M4: Strictly superior encoded memory**.

We have presented four milestones in an order which we expect may represent an increasing degree of challenge. It is not necessarily the case that each is more difficult than the last – for example, conceivably M3 may be achieved before M2 in a given physical device. However the fourth milestone is clearly the most demanding and we should expect that the inequalities in Eqs. (4.12), (4.13) and (4.14) must all be satisfied before Eq. (4.15) can be achievable.

4.2.3 Numerical results

In this section we present our simulation results under the Alice-Igor-Bob framework described in Tables 4.1 and 4.2. We model the gate noise using the simple canonical Pauli depolarising noise model as described in Section 3.2.4 on all elements including state preparations, gates and measurements where noise occurs with equal probability. Environmental decoherence is modelled as a depolarising process that occurs independently for each physical qubit. Specifically, when any memory qubits are exposed to the environment for some time t then the probabilities of an error is given by

$$p = \tfrac{1}{2}\left(1 - \exp(-t/T)\right),$$

where T is the decoherence time of an isolated single physical qubit. Given that an error occurs, it is assigned as one of the three Pauli operators X, Y,

Z selected uniformly at random. This occurs independently and in parallel for each physical qubit.

The simulation technique is based on the Monte Carlo method. We aggregate a large number of individual runs, in each of which a pure state undergoes a specific trajectory: after every circuit element is applied, a classical random number is generated and compared with the error rate for that circuit element in order to decide whether an error is applied and if so its type. Each data point presented in this chapter is a result of at least one million runs, and in order to make a smooth curve, at least 50 data points are generated for a single curve.

For all the simulations, we take Igor's action to be instantaneous although it is of course straightforward to assign it a finite time δ as indicated in Table 4.2. If Igor's analysis indicates that a correction is necessary, then the appropriate correction is applied perfectly – this is a proxy for the reality that one can simply note the need for correction and update future operations to allow for it, thus never needing to apply an imperfect physical operation to the identified qubit.

Presently we evaluate the integrity metric for the three error correction codes introduced in Section 4.1: the five-qubit code, seven-qubit code and the rotated nine-qubit surface code.

Achieving the milestones

We begin by explaining the nature of the graphs shown in this section. Typically they are of the general form exemplified by Fig. 4.6(a). On the vertical axis we show the integrity, as defined earlier, which of course is equal to unity for an ideal memory. The horizontal axis shows the duration τ of the memory channel(s) in question; the duration is shown as a ratio with respect to the environmental decoherence rate T. Each point along a curve in the figure is thus the integrity of a specific kind of memory when operating for the specific duration indicated by the horizontal axis.

There are four types of memory channels shown in Fig. 4.6(a): The simple one-qubit memory Θ (shown in blue), an encoded channels without Igor's error correction, Φ^0 (red), and two channels where Igor does perform one round of

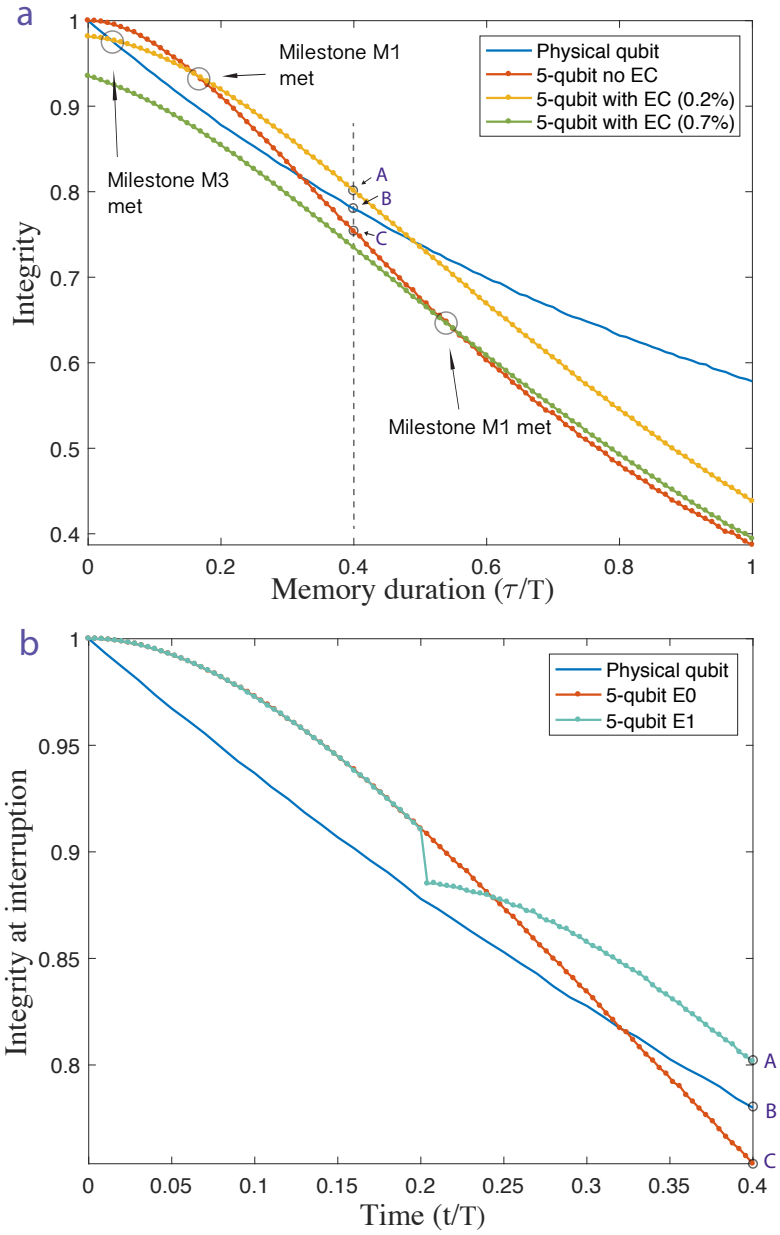


Figure 4.6: The integrity of different types of memory channel (upper), and the integrity change during a given memory channel (lower). (a) Memory integrity assessed over many different durations τ ranging from zero to T, the single-qubit decoherence time. Blue line: A single physical qubit, i.e. no encoding. Red: a memory using the five-qubit code for the stored qubit, but without active correction during the channel. Orange: the same five-qubit code, but now with a round of error correction performed mid-way through the memory duration, i.e. at time $t = \tau/2$. Error rate in operations during the correction cycle is 0.2%. Green: As for orange, but error rate 0.7%. (b) We plot the 'integrity at interruption' in order to look inside three specific memory channels during their operation. The three channels all have duration $\tau = 0.4\,T$. As explained in the text, the interesting feature is the step-like decline occurring at $t = \tau/2 = 0.2\,T$ when Igor performs imperfect error correction.

correction Φ^1 (yellow, green). The last two differ only in that Igor's error correction circuits have error rates 0.2% and 0.7%, respectively. In all cases the encoded channels are using the five-qubit code. Encoding and decoding tasks, performed by Alice and Bob, are perfect as per the definition of integrity. Later in Sec. 4.2.3, we discuss how these 'perfect' agents can be consistent with an experimental test.

Igor's error rate of 0.2% is low enough for him to perform well and consequently we observe two desirable line crossings in the figure. For all durations $\tau > 0.16\,T$ we see that the error corrected memory Φ^1 (yellow) is superior to the un-corrected memory Φ^0 (red). Thus for any $\tau > 0.16\,T$ we meet the **M1: beneficial error correction** milestone specified earlier in Eq. (4.12). Furthermore, for all durations τ between $\tau \simeq 0.035\,T$ and $\tau \simeq 0.49\,T$ the corrected memory Φ^1 (yellow) has superior integrity to the single-qubit memory Θ (blue). Note however that in comparing the single-qubit channel Θ to the encoded channels, we have not introduced any scaling factor to adjust their relative durations (i.e. we have set $\alpha = 1$ in Eq. (4.14)). This might be considered unreasonable unless the physical platform embodying the memory system is capable, in principle, of performing transversal gates in one step so that operations on logical qubits are on the same timescale as operations on physical qubits. With this important caveat, we can say that for any duration in the range $0.035\,T < \tau < 0.49\,T$ we can meet the **M3: beneficial encoded memory** milestone, Eq. (4.14).

The green line, corresponding to the higher per-gate error rate of 0.7% during Igor's error correction, never surpasses the integrity of the single physical qubit memory; therefore this channel does not meet milestone **M3: beneficial encoded memory**. However when the duration $\tau > 0.55\,T$ it does (barely) surpass the integrity of the Φ^0 channel. Therefore milestone **M1: beneficial error correction** is met.

Three points labelled A, B, and C have been highlighted in Fig. 4.6(a). They lie at a value of the duration, $\tau = 0.4\,T$ for which the high-fidelity corrected channel is superior to the single physical qubit memory Θ, which in turn is superior to the encoded-but-uncorrected channel Φ^0. One might wish to understand how the

integrity varies *over the course* of the duration of those memory channels. In fact, the question is not entirely proper since integrity is only defined as a property of the entire channel; but we can ask what would happen if Bob were to 'step in early' at any time between $t = 0$ and $t = \tau = 0.4\,T$. We suppose that Bob would perform his usual decoding, measurement and guess using the state of the memory system at that premature point. From his performance we can infer an 'integrity so far', so to speak, which we might also call the 'integrity at interruption'. Fig. 4.6(b) shows this quantity. The blue and red lines, which correspond to the single-qubit memory Θ and the encoded memory without correction Φ^0, do not reveal anything interesting. Indeed they precisely correspond to the same lines in the region $0 < \tau < 0.4\,T$ in the upper panel.

The green line in Fig. 4.6(b) corresponding to the error-corrected channel Φ^1 is far more interesting. It is exactly coincident with the Φ^0 line until $t = 0.2\,T$ because in these cases Bob interrupts before Igor performs his error correction cycle. But then the 'integrity at interruption' falls sharply, i.e. there is a significant difference between Bob interrupting immediately before Igor's effort, versus doing so immediately afterwards. The reason is that Bob's process of decoding the memory begins with a round of error correction and *his* error correction is perfect; thus it can only be worse to have Igor apply his own flawed effort at correction immediately beforehand. However despite the sharp step down, the eventual integrity at full duration is higher. This is because Igor's efforts have reset the accumulation of errors, lowering the overall chance of an uncorrectable set of errors (i.e. 2 or more errors, for the five-qubit code) over the course of the complete memory channel. We see this evidenced by the inverted parabolic curve immediately after Igor's action: in effect the environment must 'start again' to build up significant probability of weight-2 errors.

In preceding figures we have focused on cases where a single round of error correction is applied during a memory channel, and we have identified points where our milestones M1 and M3, would be satisfied. In Figure 4.7 we show how the use of multiple rounds of error correction (equispaced within the duration of the memory

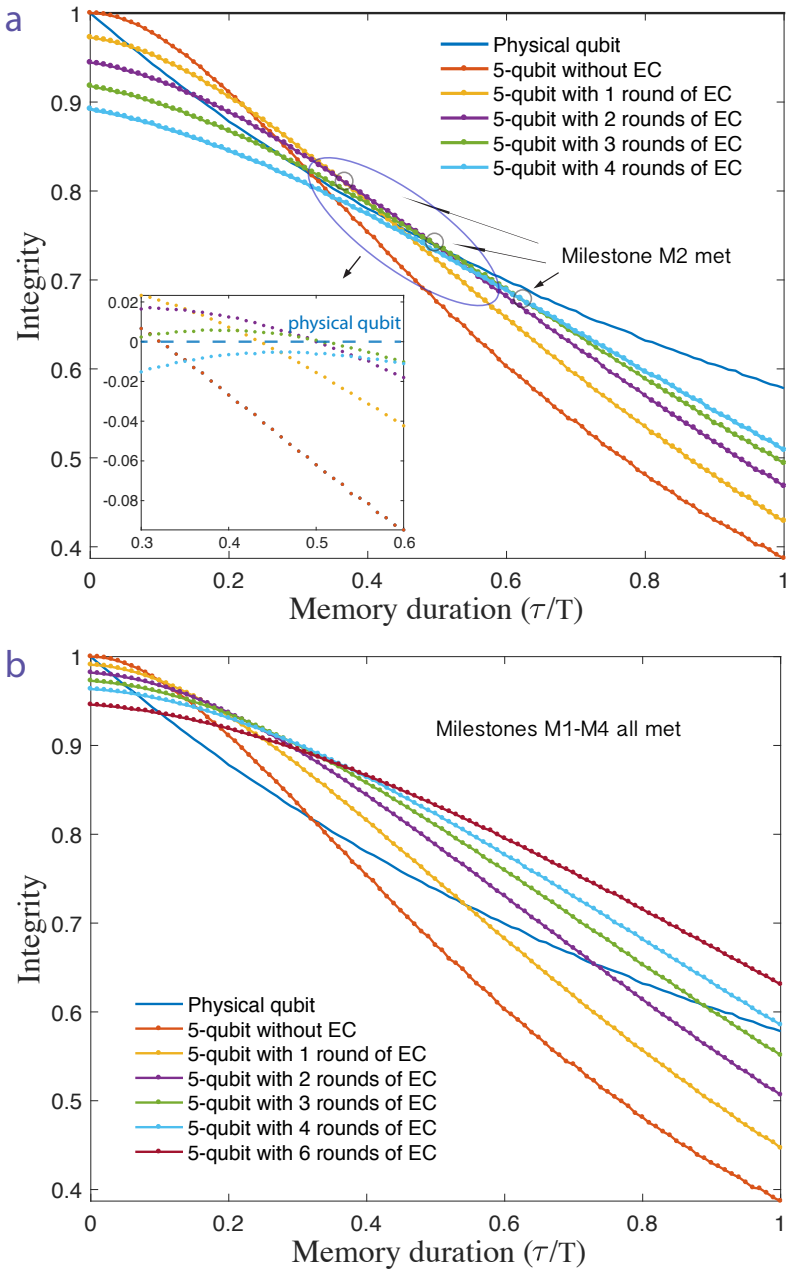


Figure 4.7: Multiple rounds of quantum error correction(EC). The integrity of a family of memory channels all employing the five-qubit code but differing in the number n of rounds of error-correction performed during the memory, where $n = 0, 1, 2, 3, 4$ or 6. Our imperfect agent Igor performs error correction cycles at times $t = m\tau/(n+1)$ for $m = 1..n$. In the upper panel (a) Igor's gate-level error rate is 0.3%. As explained in the text, the system meets milestones M1, M2 and M3 but fails to meet M4. In the lower panel (b) Igor's error rate is now 0.1% and we see that by choosing a suitable n we can select a five-qubit encoded memory that will beat the single-qubit memory for any desired duration τ, so meeting milestone M4: Strictly superior encoded memory.

channel) may allow us to meet milestone **M2: Beneficial multi-round error correction** associated with Eq. (4.13), or even milestone **M4: Strictly superior encoded memory** associated with Eq. (4.15). In the upper panel, Igor's error rate suffices for the former but not the latter. Three rounds of error correction (the green curve) is marginally superior in a finite range of τ, however four rounds (the light blue curve) is never beneficial. In the lower panel Igor's error rate is set to 0.1% which proves to be sufficient to achieve the fourth milestone.

Comparison of different codes

We now present a series of simulations which contrast different codes, and also compare fault-tolerant versus non-fault-tolerant implementations of error correction circuits.

The various encoding, decoding, and error correction circuits which we use in the simulations have been described in Section 4.1. As a first step toward comparing the efficacy of different codes, we begin by reporting a special case which is achieved by setting the memory duration to zero, and simply investigating the impact of the error correction process itself. Thus, we take a perfectly encoded qubit prepared by Alice and present it directly to Igor who performs an (entirely unnecessary) error correction cycle before passing the encoded qubit directly to Bob for his analysis. The reduction in integrity is thus purely due to Igor's action.

The results are shown in Fig. 4.8, which includes eight different options for the encoding and correction of a logical qubit. Three different codes are considered: the five-qubit code, the seven-qubit Steane code, and the nine-qubit surface code. For each of these, the performance of a non-fault-tolerant (non-FT) Igor is plotted. For the nine-qubit code, a second curve shows the performance when Igor employs a specific FT error correction circuit (Fig. 4.5). For each of the other two codes, we display the performance of two different FT circuits: The 'DiVincenzo-Shor' approach using five ancilla qubits (Fig. 4.3(a)), and the recent approach using two ancilla qubits (Fig. 4.3(c)).

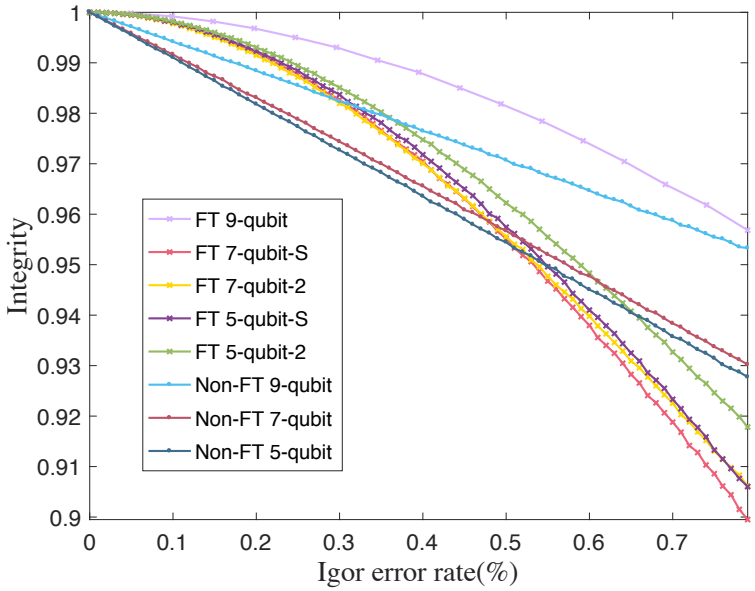


Figure 4.8: Integrity change with increasing gate error rate. Here the duration of our memory is set to zero, in order to directly inspect the negative impact of an imperfect error correction performed by Igor. The horizontal axis shows the level of noise associated with each circuit element of Igor's circuits. We analyse memories based on the five-qubit, the Steane, and the nine-qubit codes. Igor's error correction is performed either in a simple, non-fault-tolerant fashion or with full fault tolerance. As explained in the text, the various line shapes and the relative levels of performance are straightforward to understand qualitatively.

There are several interesting general observations to be made from Fig. 4.8. Firstly, it is reassuring to note that two logically-necessary features are indeed present: One observes that all the cases which employ non-FT error correction for Igor have the expected linear decay as Igor's error probability p increases from zero: integrity goes as $1 - c\,p$ for some constant c because non-FT circuits are vulnerable to single errors. Meanwhile the scenarios featuring FT error correction all have the expected inverted-parabolic shape: the integrity goes approximately as $1 - k\,p^2$ when Igor's error probability p is small. Circuits of this kind are 'immune' to single errors and vulnerable only to weight two (or higher) errors. Note that for higher (but still sub-1%) error rates for Igor, the fault-tolerant circuits become inferior to the simpler non-FT circuits. The reason is essentially combinatorial scaling: the FT-circuits are generally considerably more complex with far more gates, thus as gate failure probability p increases the risk of a double error in these complex

circuits eventually outweighs the risk of a single error in the simple non-FT circuits.

The different gradients in the various linear and parabolic curves can be qualitatively understood by considering two desiderata. The first is the portion of all possible weight-2 errors that actually prove to be correctable. For example, the five-qubit code is corrupted by all weight-2 errors, but the seven-qubit Steane code can correct any pair of errors if (and only if) one is of type X and one of type Z. The nine-qubit surface code has the highest portion of 'harmless' weight-2 errors in this sense. The relative ordering of the non-FT codes can be explained by this feature alone. However for the FT codes, there is another competing feature: as noted above the complexity of the FT error correction circuits is what 'kills' their performance, so simpler circuits are superior. Consistent with this principle, we see wherever an appreciable performance gap exists between the 'two-ancilla' variant of a FT code versus the 'DiVincenzo-Shor' variant of the same code, the former is always superior. Moreover the FT circuits for the five-qubit code are more simple than those the seven-qubit Steane, thus among the FT curves the five-qubit outperforms the Steane. Remarkably FT circuits for the nine-qubit code are very simple, thus the FT surface code benefits from both desirable features described here, and is unconditionally superior to all other codes in the plotted error range.

It is important to remember that the comparison made in Fig. 4.8 is for zero environmental error. The relative performance of different codes will change once we deploy them properly into a memory channel where environmental noise is degrading the encoded qubit. Figure 4.9 shows the integrity change of the memory under our standard memory channel scenario Φ^1 i.e. 'one use of Igor's error correction midway'. All curves in this figure correspond to an internal error rate for Igor's operations of 0.5%. For small memory duration, we see that the Steane code is marginally superior to the five-qubit code, but both are markedly inferior to the nine-qubit code. However, as we move away from the hard left of Fig. 4.9 to consider increasing duration of the memory, we find that the five-qubit code surpasses first the Steane and then even the nine-qubit code. The reason is that as more environmental error accrues, a code with a larger number of physical qubits

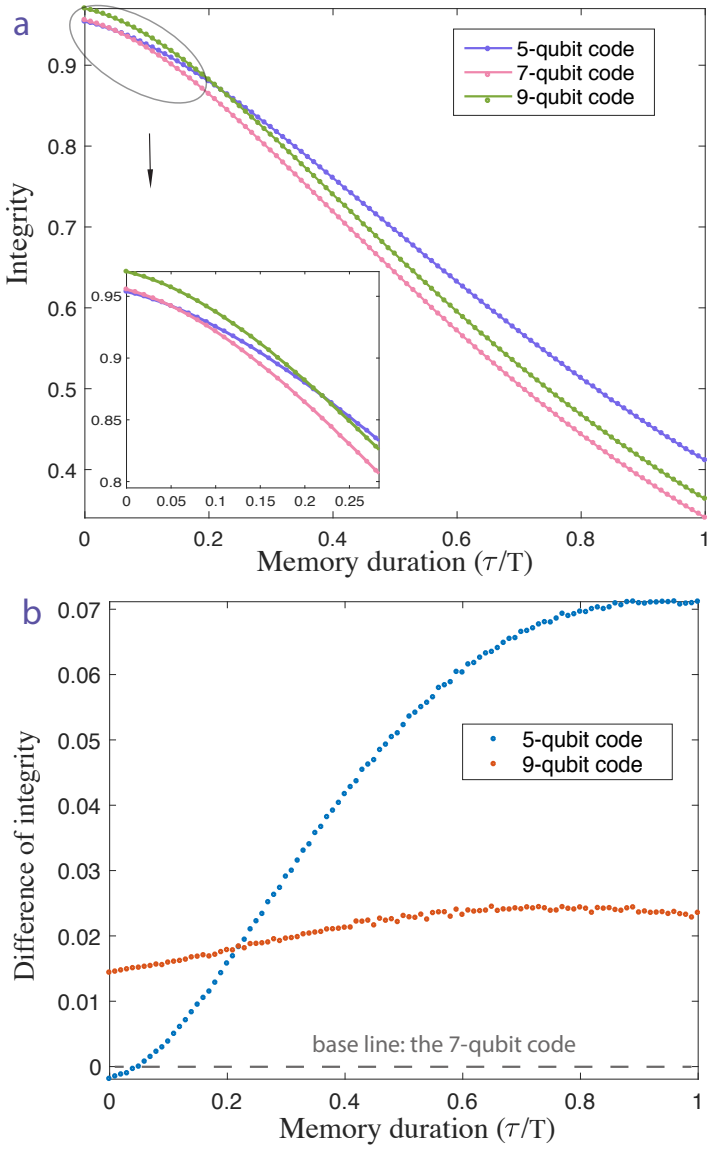


Figure 4.9: Comparison between the 5/7/9 qubit codes. The data shown are for our canonical Alice-Igor-Bob scenario where total memory duration τ during which pure environmental decoherence occurs continuously and a single (imperfect) round of error correction occurs midway at $t = \tau/2$. The error rates used for all the gate operations during the error correction procedure are 0.5%. The lower panel (b) presents the same data but now with respect to the the Steane code performance, so that the integrity of that channel now lies along the horizontal axis.

will reach the point where two-or-more errors are present, i.e. the situation where the logical qubit may be corrupted, at an earlier time.

Before concluding this comparison of different codes, we should stress that our intention is not to identify "best and worst" codes but rather to show the circumstances in which various codes can be the better choice. We also recognise that there are other merits beyond the question of how well a code preserves channel integrity – for example, the Steane code (which is also the smallest instance of the 2D colour code) has the significant merit versus the smaller five-qubit that all Clifford operations can be applied transversally.

Realistic error correction

In all the theory and the numerical simulations described above, Alice and Bob are perfect agents: they provide the framework within which we assess the memory channel. However, if we are to assess integrity in an experiment – i.e. if it is to be a practical measure for benchmarking quantum memories – then we must tackle the reality that Alice and Bob are merely phases of an experiment within which all operations are imperfect. In the simulations reported in this section, we discuss the more realistic scenario where the same error model and error severity are applied to the actions of Alice and Bob, as we apply to Igor's error correction cycle. For typical noise types, it can be shown that it suffices to consider Pauli eigenstates[1]

In the idealised case, Alice can prepare any encoded qubit she wishes, while the definition of integrity corresponds to the case when Alice opts for the worst possible choice of qubit state to encode (or rather, when she picks between the two states $|\psi\rangle$ and $|\psi_\perp\rangle$, which Bob has the most difficulty differentiating post-memory). If we have foreknowledge of which states these are, we need only find circuits for Alice and Bob to use which perform *equivalently* to their general purpose circuits in these special cases. Fortunately we know that the worst case choice Alice can make will correspond to Pauli basis states, i.e. $\{|\psi\rangle, |\psi_\perp\rangle\}$ will be either

[1]Niel de Beaudrap, one of the co-authors of Ref. [126], was able to show this and an explanation appears in the paper; however this observation did not involve the present author and so is not reproduced within this thesis.

$\{|0\rangle, |1\rangle\}$ or $\{|+\rangle, |-\rangle\}$ or $\{|y+\rangle, |y-\rangle\}$ ($|y\pm\rangle = \sqrt{\frac{1}{2}}(|0\rangle \pm i\,|1\rangle)$). Our challenge is therefore to find specific encoder circuits for Alice and analysis circuits for Bob for these special cases. To obtain best-possible performance for Alice and Bob when they are error-burdened, we apply the fault-tolerant encoding and decoding circuits described in Section 4.1.

We emphasise that once we equip Alice and Bob with suitable circuits, we have a full prescription for an experimental test of integrity: the experimental protocol simply corresponds to the steps listed in Tables 4.1 and 4.2, with the sole modification that Alice randomly picks between Pauli eigenstates, and given this pick both her encoding circuit and Bob's decoder are selected accordingly from optimised circuits such as the one in Fig. 4.2. Thus integrity is evaluated without state tomography.

The data plotted in Fig. 4.10 and Fig. 4.11 show the effect of allowing Alice and Bob to become noisy, for the Steane code and the five-qubit code respectively.

For Steane code (Fig. 4.10) we observe an excellent agreement between the 'true' integrity that would be measured if one were able to use ideal agents Alice and Bob, and the estimate of the integrity that results from using imperfect agents. We note that there is only a slight variation in the location of the crossing point, and that if a crossing occurs in the true integrity (as for the case when Igor's error is 0.5%) then a crossing also occurs in the estimate; the specific crossing shown here is that which would show milestone M1 has been met. Conversely when a crossing does not occur in the true integrity, it also fails in the estimate (here, for the case when Igor's error is 1%). The reason for the excellent agreement is that both Alice and Bob's circuits are robust against errors.

In our second example of noisy Alice and Bob, shown in Fig. 4.11 we employ the five-qubit code and, crucially, we employ non-fault-tolerant procedures for Alice. Unfortunately, when the tasks performed by Alice and Bob become vulnerable to single gate failures, the resulting memory integrity estimates become very poor approximations to the true integrity. In the lower panel of Fig. 4.11 we see that the line shapes have changed, losing the inverse-parabola shape for short memory durations. We do still see line crossings, but they occur at significantly different

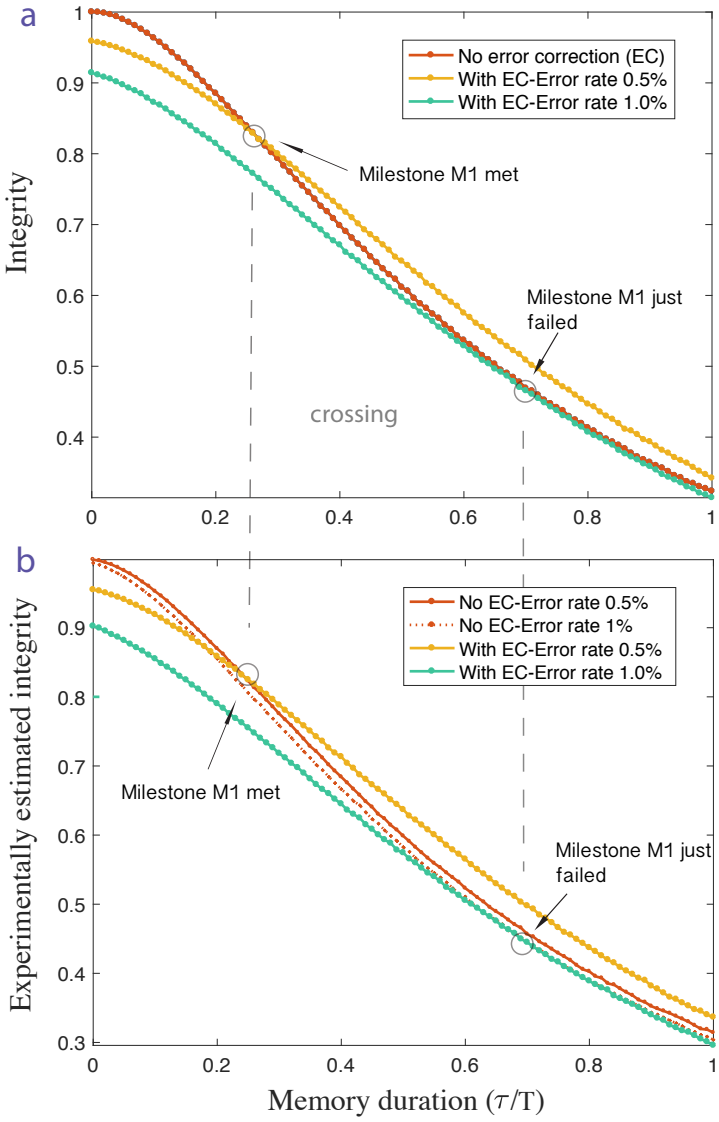


Figure 4.10: Imperfect, but fault-tolerant, Alice and Bob. The scenario in the upper panel (a) corresponds to three different memory channels each using the Steane code to protect information. The encoder Alice and the analyser Bob are both ideal, as required in the definition of integrity. We mark the meaningful line crossing which corresponds to meeting milestone M1 (for the orange line) or just failing to do so (blue line). In the lower panel (b) we present the same analysis but now with errors during Alice and Bob's circuits at the same level as Igor's. Specifically, Alice uses the circuits shown in Fig. 4.4 to encode her qubits into $|0\rangle_L$. Bob differentiates between $|0\rangle_L$ and $|1\rangle_L$ by simply measuring all qubits in the z-basis, performing classical error correction, and checking the parity of a certain subset. Note that the key crossing (and failure to cross) from the upper panel are well approximated in the lower, indicating that experimental evaluation of integrity is achievable.

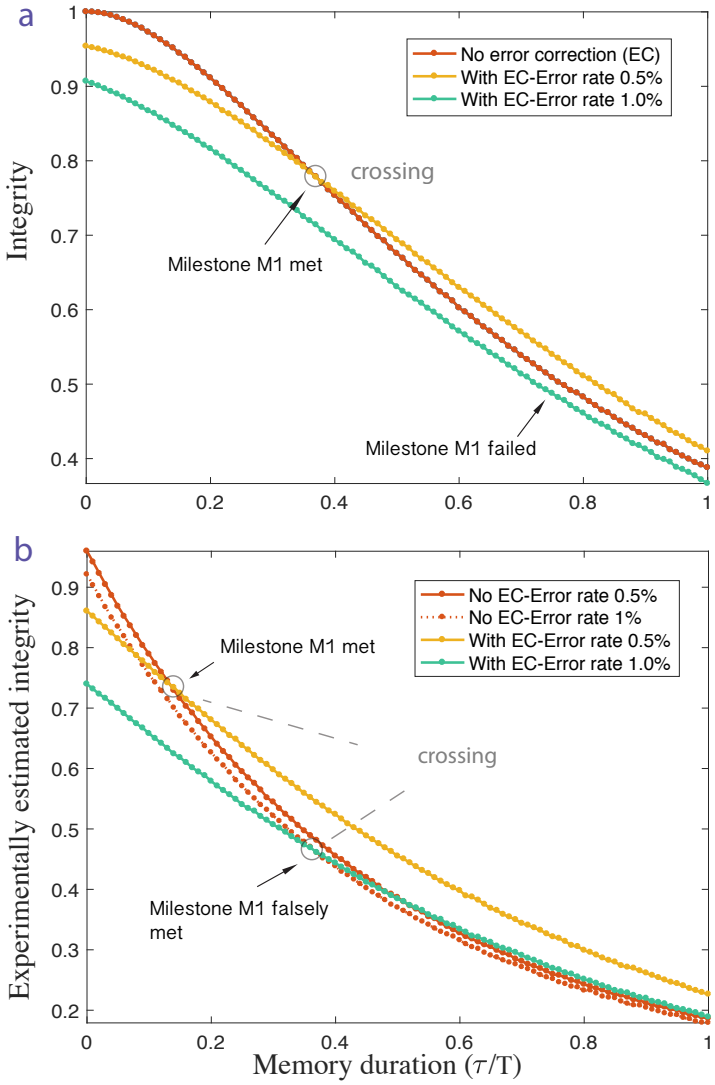


Figure 4.11: Imperfect, and non-fault-tolerant, Alice and Bob. The comparison made here is similar to that in Fig. 4.10 except that now the memory channels employ the five-qubit code and moreover in the lower panel (b) the circuits used by Alice and Bob are not fault-tolerant. Here, Alice uses the circuits specified in Fig. 4.2 to encode qubits into $|-\rangle_L$ and Bob uses the circuit shown in that figure to differentiate between $|+\rangle_L$ and $|-\rangle_L$. In contrast to Fig. 4.10 there is now a profound difference between two panels and one could not directly assert that a line crossing in the lower panel implies a crossing would exist in the upper panel.

locations. Most troublingly, a line crossing can occur in the experimental data when no such crossing would occur if Alice and Bob were ideal. Thus, the observation of a crossing in the data is not, in of itself, strong evidence that the actual memory channel has met a meaningful milestone (such as M1 in this case).

Despite these issues, it can be possible to make use of data such as that in Fig. 4.11. One would need to perform additional theoretical analysis in order to justify the claim that any observed crossing is indeed meaningful. For example, if the errors in the various circuit elements are well characterised then one could perform simulations equivalent to those presented in this chapter. Essentially one would produce a version of Fig. 4.11(b) calculated with an accurate error model in order to compare with the observed data; if the match proved to be good, one could use further simulation to discover the integrity that would have been observed with ideal Alice and Bob. In order words, if the data closely matches a simulation such as Fig. 4.11(b), one might fairly state that this is strong evidence that the integrity is as shown in Fig. 4.11(a).

In summary, we can say that integrity can be assessed experimentally in a straightforward protocol: Acting as Alice we choose a qubit state, then we perform a series of experimental runs where each run ends in a measurement from which, as Bob, we 'guess' the original state with the binary outcome 'succeeded' or 'failed'. We continue until we have a good estimate of Bob's probability of success p_g; if the system is such that p_g depends on Alice's choice, then we find the least-favourable choice. The integrity of the memory is then simply $\mathcal{R}(\Phi) = 2p_g - 1$. In this section we have shown that the creation of the logical qubit, i.e. Alice's circuit, as well as Bob's analysis circuit, can both be noisy and yet we can obtain an excellent estimate of the integrity of the memory channel itself (factoring out Alice and Bob).

4.3 Conclusion

This chapter gives an introduction to error correction with small codes. We take the example of the most commonly-used codes including the five-qubit code, the seven-qubit code and the nine-qubit surface code. The small codes can

be implemented with a small number of physical qubits and relatively shallow circuits. However, given environmental noise and imperfect operations, to make the fidelity of an encoded logical qubit higher than a raw physical qubit is very challenging. It thus motivates us to introduce 'integrity' as a means to benchmark the performance of a code-based quantum memories. Integrity measures how well a memory preserves the distinctiveness of different states. We show that integrity can be assessed experimentally in a straightforward manner without the need for full state tomography.

5

Modelling quantum error correction on an ion-trap processor

Contents

In Chapter 4, we gave an overview of the most commonly-adopted small error correction codes. Since invention, the small codes have received wide attention and have been thoroughly studied. In experiment, quantum error detecting and correcting codes have been implemented in different platforms ranging from superconducting circuits [110–113, 127], trapped ions [14, 108, 109, 134], NV centres in a diamond [135], optics [136–139], and others [114, 140]. However, those experiments are limited to handling only a certain type of errors or with particular logical states. Comprehensive demonstration of full error correction

remains a challenge.

Realising beneficial error correction requires both theoretical and experimental advances. From the experimental perspective, the manipulation of qubits should be improved to the highest accuracy. Theoretically, an essential contribution could be development of resource-efficient error correction protocols that are specifically designed for particular hardware systems. Performing the task requires realistic modelling of the physical process and error sources to identify the dominating factors and give faithful estimations of the hardware performance.

In this chapter, we present a theoretical error correction protocol designed for an ion trap quantum processor. A detailed description and thorough analysis of an experimental toolbox for error correction with the Steane code is given. We develop sequences of crystal-reconfiguration operations and stabiliser mappings to perform a full error correction cycle on a single logical qubit. Realistic error sources are considered in each stage of the protocol, realised with a microscopic modelling of the ion crystals to derive the particular expressions or values of the corresponding error rates for each operation. We apply the 'integrity' introduced in Chapter 4 to assess the performance of the quantum memory with the state-of-the-art and anticipated near-future statistics, aiming to identify the requirements and parameter regimes for beneficial error correction in the underlying ion-trap quantum processors.

This chapter is structured as follows. Sec. 5.1 gives a brief introduction to the hardware studied in this chapter. The protocol to perform a full round of error correction with the Steane code on the aforementioned hardware is described in Sec. 5.2. Sec. 5.3 gives the error model for different operations. Numerical simulations based on the Alice-Igor-Bob framework as introduced in Chapter 4 are given in Sec. 5.4. This chapter is concluded in Sec. 5.5.

The materials of this chapter are based on papers [141, 142]. The author contributed to conception and simulations of the protocol. Furthermore she jointly developed the error correction protocol and the error models with other coauthors. The numerical results were the author's own work.

5.1 An ion-trap quantum processor

The ion-trap quantum computer utilises ions arranged in a linear trap, which is controlled by radio frequency pulses, laser beams and DC electric fields. The physical demonstration of using ions to manipulate quantum information was firstly reported by Wineland's group in 1995 [144]. Among all potential hardware platforms, the ion-trap quantum memory has one of the longest coherence times in the order of seconds, and a lowest gate infidelity: the lowest reported single-qubit gate infidelity is $1.0(3) \times 10^{-6}$ [119] and the lowest two-qubit gate infidelity is $8(4) \times 10^{-4}$ [117].

The hardware considered in this chapter is a high-optical-access segmented ion trap run by our experimental partner based in Innsbruck. The trap allows one to manipulate dual-species ion crystals [145] under cryogenic conditions. This architecture is scalable, as 1D trapping zones can be coupled via junctions into larger potentially 2D trap array structures. As shown in Fig. 5.1, we consider $^{40}Ca^{+}$ ions for hosting the qubits, $^{88}Sr^{+}$ ions for providing the capabilities for sympathetic cooling, and mixed-species readout for syndrome extraction. We consider that ions undergoing the quantum logic operations can be separated and shuttled across the segmented trap array by using high-speed, low-excitation protocols in order to minimise cross-talk on neighbouring qubits.

The quantum information is encoded in $|0\rangle$ and $|1\rangle$ states, which are represented by $4S_{1/2}(m_f = -1/2)$ ground state and $3D_{5/2}(m_f = -1/2)$ metastable excited state. The excited state has a lifetime of $1.1s$, which sets the upper limit on the qubit storage time [146]. Quantum operations are performed with a laser that is nearly resonant to this transition at a wavelength of about 729 nm.

5.2 Protocols for performing quantum error correction

The protocol developed for quantum error correction (QEC) is a shuttling based approach via the ion-crystal configuration techniques. As described above, ions are confined above a planar segmented trap divided into manipulation and storage

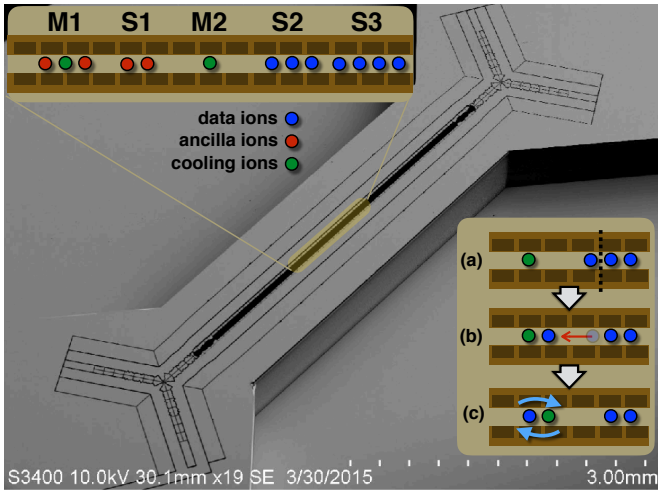


Figure 5.1: The Sandia high-optical-access segmented trap. The blue and red ions are $^{40}Ca^{+}$ ions, and the green ones are $^{88}Sr^{+}$ ions. The quantum zone has three storage regions S_1, S_2, S_3 and two manipulation regions M_1, M_2. The data ions are used to encode quantum information, while the ancilla ions are for syndrome extraction. The cooling ions are sympathetic coolants to reduce the number of phonons prior to the entangling gates. Possible crystal-reconfiguration operations are shown in the panel in the lower-right corner: (a) splitting of an ion crystal, (b) shuttling of an ion and subsequent merging with another ion to form a crystal, and (c) rotation (swapping) of a mixed species crystal. This graph is taken from Ref. [141].

regions, where various elementary operations can be performed. All the operations required for the whole QEC cycle include the following: (a) two-qubit entangling gate, (b) single-qubit gate, (c) projective measurement, (d) qubit initialisation/reset, (e) qubit recooling, (f) ion shuttling, (g) ion splitting and merging, (h) ion swapping and rotation, (i) leakage repumping. We discuss those operations below.

The experimentally available two-qubit entangling gate is the Mølmer-Sørensen (MS) gate [147, 148], which is driven by a global laser pulse and can be expressed as:

$$U_{MS,\phi}(\theta) = e^{-i\frac{\theta}{2}S_\phi^2}, S_\phi = cos(\phi)X + sin(\phi)Y, \tag{5.1}$$

where ϕ is controlled by the laser phase, and θ by laser intensity and pulse duration. By setting ϕ to be either 0 or $\frac{\pi}{2}$, we can have X-type or Y-type MS gates. Here we adopt the X-type MS gates. Furthermore, the MS gates are 'fully-entangling' when $\theta = \frac{\pi}{2}$.

The single qubit gates can either be realised with a global laser pulse to have

global rotations around the Bloch sphere:

$$U_{R,\phi}(\theta) = e^{-i\frac{\theta}{2}S_\phi}, \tag{5.2}$$

or alternatively with local rotations around the Z-axis:

$$U_z(\theta) = e^{-i\frac{\theta}{2}R_z}, \tag{5.3}$$

The details of the gate set has been described in Ref. [146].

Measurement of a data qubit is from collection of state-dependent fluorescence due to the emitted photons from the cycling transition [146]. The qubit is then reinitialised by optical pumping. The recooling process is realised by the sympathetic laser cooling technique. In this way, one can cool the vibrational mode prior to any entangling gate. The ion crystal reconfiguration includes splitting ion crystals, shuttling ions across trap segments, merging two sets of ions into a large crystal and rotating ions to reorder the position. All those operations have been experimentally demonstrated [149–151].

In our proposed experiment, one logical qubit is encoded with the seven-qubit code, which is the smallest 2D colour code. For stabiliser checks, we consider the non-fault-tolerant (non-FT) case as well as the flag-based fault-tolerant (FT) approach as introduced in Sec. 4.1. To implement the QEC cycles, we translate the circuit-level description into detailed microscopic schedules in the trap. These schedules consist of specific sequences of elementary operations, which combine quantum gates and crystal reconfiguration operations. We apply the Alice-Igor-Bob protocol introduced in Sec. 4.2 to assess the integrity of the encoded memory.

A detailed sequence of Igor's operations is shown in Fig. 5.2, which is based on the non-FT scheme, while a flag-based FT scheme follows very similar steps. The figure shows half of the QEC cycle, including stabiliser measurements of D_1, D_2 and D_3, where $D_1 = XXXXIII$, $D_2 = IXXIXXI$ and $D_3 = IIXXIXX$. In order to reduce the complexity of all the crystal configurations required to perform the sequential gates of the QEC cycle, Igor makes use of all five regions of the segmented trap in Fig. 5.1. The data qubits of a given stabiliser are shuttled one by one from

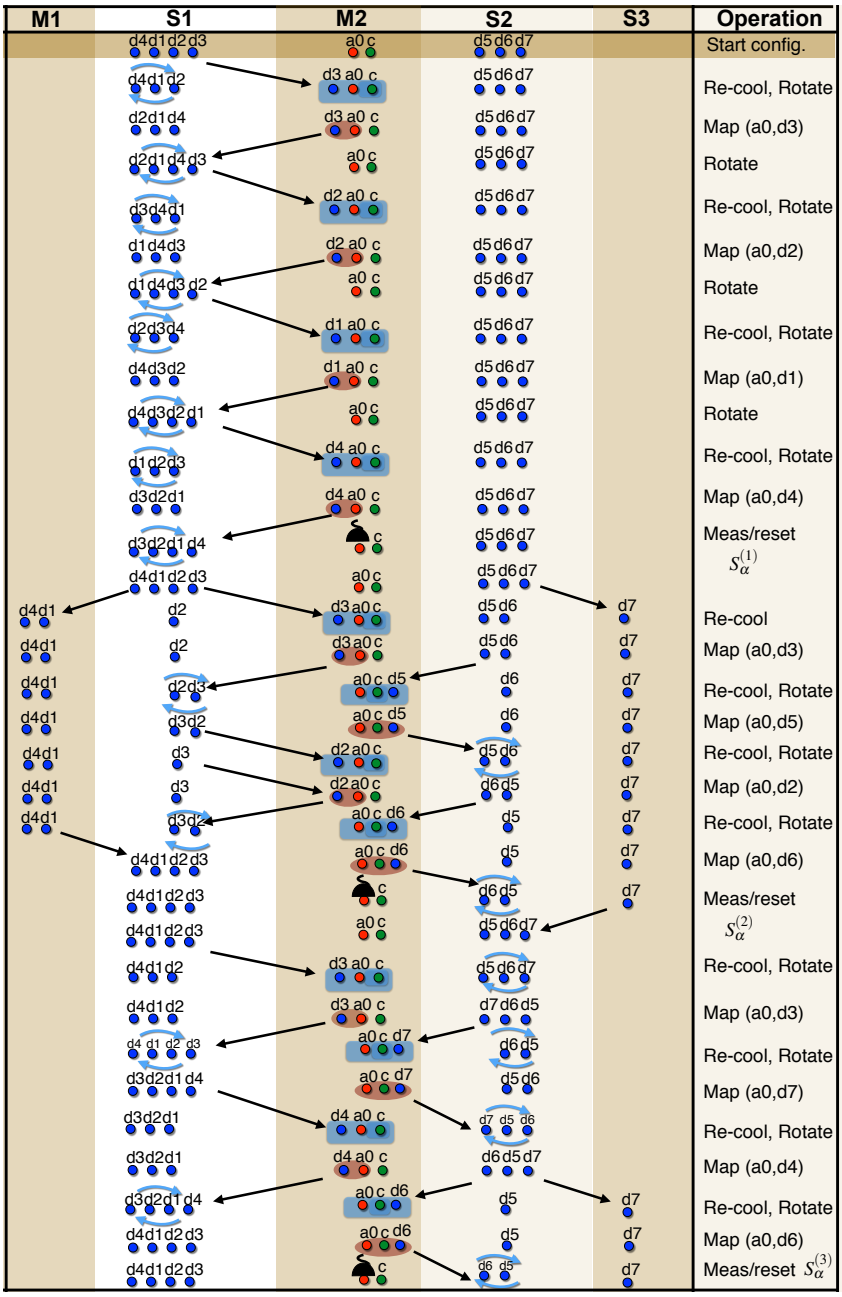


Figure 5.2: The operation sequence for the shuttling-based QEC protocol with the non-FT stabiliser measurements. The data qubits are stored in the storage zones, and each corresponding data qubit is shuttled to the manipulation zone M2 and recooled before the data-ancilla entangling gate. Each time the data qubits are rotated to reschedule the location. After all the four data qubits are entangled with the ancilla qubit, the latter is measured out to obtain the syndrome and reinitialised. The process continues until all the stabiliser checks are performed. In this graph, firstly the qubits d_3, d_2, d_1 and d_4 are mapped for one stabiliser check, followed with d_3, d_5, d_2, d_6 and then d_3, d_7, d_4, d_6. This graph is taken from Ref. [141].

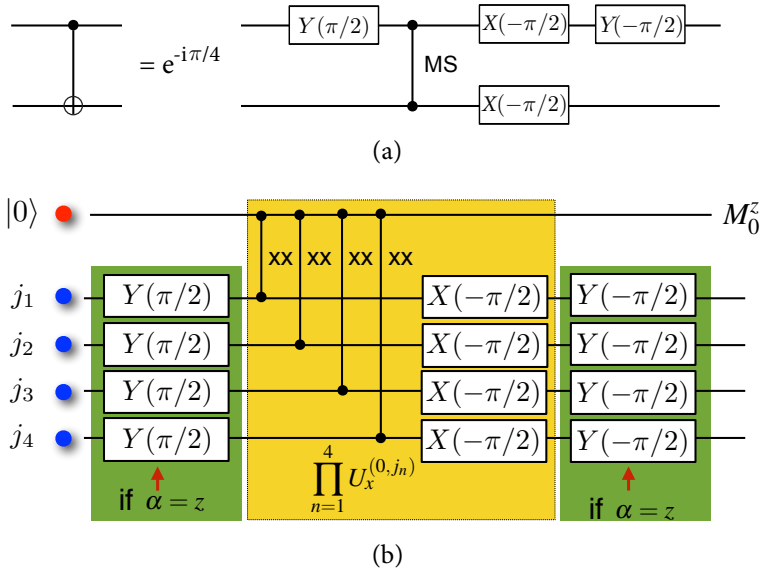


Figure 5.3: (a) the CNOT gate can be decomposed as a two-qubit X-type MS gate with several single-qubit rotations. (b) Non-fault-tolerant stabiliser readout based on the two-qubit X-type MS gate. The yellow region depicts the gates for the X-check, while for a Z-check, additional gates in the green region are required. Here $U_x^{0,j_n} = X_0(-\pi/2)U_{MS}^{0,j_n}(\pi/2)$. (b) is taken from Ref. [141].

the storage regions onto the manipulation zone M2, where sympathetic recooling represented by blue rectangles is applied prior to the data-ancilla mapping for the stabiliser check. During the process, ion rotations are performed to reorder the qubit register. After all the four data qubits of a particular stabiliser have been coupled to the ancilla ion, Igor can proceed to isolate the ancilla-cooling pair of ions in the manipulation zone and collect the state-dependent fluorescence of the ancilla ion, in this way obtaining the error syndrome. After the measurement, the ancilla qubit is then reinitialised by optical pumping. This process continues until all stabiliser checks are performed and error information can be inferred from the syndromes.

Note that the data-ancilla mapping in Fig. 5.2 is based on the two-qubit X-type MS gate. On the other hand, the stabiliser measurements are usually performed by entangling the data qubit with the ancilla qubit via CNOT gates or CPhase gates, we thus need to map the CNOT gate into the MS gate. A CNOT gate can be decomposed into a MS gate and four single-qubit rotations, as shown in Fig. 5.3(a). By permuting single-qubit rotations, we can work out the circuit for the stabiliser

Operation	Current duration	Current infidelity	Anticipated duration	Anticipated Infidelity
(a) Two-qubit MS gate	40μs	$1 \cdot 10^{-2}$	15μs	$2 \cdot 10^{-4}$
(b) One-qubit gate	5μs	$5 \cdot 10^{-5}$	1μs	$1 \cdot 10^{-5}$
(c) Measurement	400μs	$1 \cdot 10^{-3}$	30μs	$1 \cdot 10^{-4}$
(d) Qubit reset	50μs	$5 \cdot 10^{-3}$	10μs	$1 \cdot 10^{-4}$
(e) Re-cooling	400μs	$\bar{n} < 0.1$	100μs	$\bar{n} < 0.1$
(f) Ion shuttling	5μs	$\bar{n} < 0.1$	5μs	$\bar{n} < 0.1$
(g) Ion split/merge	80μs	$\bar{n} < 6$	30μs	$\bar{n} < 1$
(h) Ion rotation	18μs	$\bar{n} < 0.3$	20μs	$\bar{n} < 0.2$
(i) Leakage repumping	60μs	$5 \cdot 10^{-3}$	20μs	$1 \cdot 10^{-4}$
(j) T_2	0.2 s	–	2s	–

Table 5.1: Current and anticipated operation infidelities and durations. $\bar{n}$ is the mean phonon number. This table is taken from Ref. [142]. For the origin of the 'current' figures, see Ref. [142]; the 'anticipated' figures are the consensus view of the experimentalist authors.

readout based on the MS gates. Fig. 5.3(b) shows a non-FT circuit for one stabiliser mapping, where the yellow region highlights the gates for the X-check; and for the Z-check, additional gates are required as depicted in the green region.

5.3 Realistic noise models

The QEC protocol described in the last section allows us to perform a detailed study that goes beyond standard assumptions. We consider that different operations do not take the same amount of time. Besides, distinct error channels are used to describe different stages of the protocol with different error probabilities. Based on microscopic calculations and experimental results, the current and anticipated statistics of the all operations is summarised in Table 5.1. Error models for the different operations are described in the following sections. In each case, the choice was made in consultation with the authors of Ref. [142], as were the choices of certain constants (energies, timescales, etc.).

5.3.1 Dephasing

During the QEC cycles, there are several operations where the internal states of a subset of qubits is not affected, including (a) ancilla qubit measurement and recooling where the data qubits remain idle, (b) single-qubit and two-qubit gates which leave the spectator qubits unchanged and (c) crystal reconfigurations leaving all the qubit states unchanged. Specifically, for (c), with a simplification, we assume each operation contributes a fixed amount of energy to each ion involved in the operation, leading to momentum kicks and heating up the ions. The mean motional energies of each ion is kept track of.

Although we consider that the operations above do not lead to change of qubit state, the idle qubits do suffer from dephasing which results from interactions with the environment, e.g. fluctuating magnetic field. The idle ions can be organised in different groups residing in the same zone of the trap. Within each group, the qubits are subjected to a common noisy magnetic field

$$H(t) = \sum_{i \in \mathrm{I}} \frac{1}{2} F(t) \sigma_i^z, \tag{5.4}$$

where $F(t) = -g\mu_{\mathrm{B}} B(t)$, μ_{B} is the Bohr magneton, and g is the hyperfine g-factor.

We assume the fluctuation of $F(t)$ is stochastic. To model it, we describe it as the Ornstein-Uhlenbeck (OU) process [152]. The OU process $O(t)$ evolves following the stochastic differential equation

$$\frac{\mathrm{d}F(t)}{\mathrm{d}t} = -\frac{1}{\tau} F(t) + \sqrt{c}\Gamma(t), \tag{5.5}$$

where τ is the correlation time, c is the diffusion constant and $\Gamma(t)$ is the Gaussian white noise with $\overline{\Gamma(t)} = 0$ and $\overline{\Gamma(t)\Gamma(0)} = \delta(t)$. τ determines the time-scale over which the noise is correlated. To numerically solve the equation, we discretise the time interval in steps of $\Delta t = t_I/N$, and update the formula as,

$$F(t + \Delta t) = F(t)\mathrm{e}^{-\Delta t/\tau} + \left[\frac{c\tau}{2}\left(1 - \mathrm{e}^{-2\Delta t/\tau}\right)\right]^{1/2} n, \tag{5.6}$$

where n is a Gaussian random variable from -1 to 1, and t_I is the total idle time period.

For the numerical simulation of the phase noise, we consider short correlation time $\tau = T_2/1000$, and $c = 1/(2T_2 * \tau^2)$.

5.3.2 Amplitude damping, leakage and repumping

With the expectation that it will be possible to improve the coherence time close to the limit of $T_2 = 2T_1 \approx 2.2s$, amplitude damping and leakage should also be considered. The amplitude damping process results from spontaneously decay of the metastable qubit state $|1\rangle = \left|3D_{5/2}, -1/2\right\rangle$ into the ground-state qubit state $|0\rangle = \left|4S_{1/2}, -1/2\right\rangle$. This process can be modelled with the following channel,

$$\begin{aligned} &\rho(t) = \sum_n L_n \rho L_n^\dagger \text{ with} \\ &L_0 = |0\rangle\langle 0| + \sqrt{1 - p_{\text{ad}}(t)}|1\rangle\langle 1|, L_1 = \sqrt{p_{\text{ad}}(t)}|0\rangle\langle 1|, \end{aligned} \tag{5.7}$$

where $p_{\text{ad}}(t)$ is the error rate for amplitude damping.

Apart from amplitude damping, there is a also a finite branching ratio for the decay into a different ground-state level $|l\rangle = \left|4S_{1/2}, +1/2\right\rangle$, leading to leakage of the ions out of the computational subspace. The leakage channel can be described as

$$\begin{aligned} &\rho(t) = \sum_n L_n \rho L_n^\dagger \text{ with} \\ &L_0 = |0\rangle\langle 0| + |\ell\rangle\langle \ell| + \sqrt{1 - p_{\text{ad}}(t) + p_\ell(t)}|1\rangle\langle 1|, \\ &L_1 = \sqrt{p_{\text{ad}}(t)}|0\rangle\langle 1|, L_2 = \sqrt{p_\ell(t)}|\ell\rangle\langle 1|, \end{aligned} \tag{5.8}$$

where $p_{\text{l}}(t)$ is the error rate for leakage.

To incorporate both channels, we work out the error rates for the two processes as

$$\begin{aligned} p_{\text{l}}(t) &= \frac{\Gamma'\left(1 - \mathrm{e}^{-(\Gamma+\Gamma')t}\right)}{\Gamma + \Gamma'} \\ p_{\text{ad}}(t) &= \frac{\Gamma\left(1 - \mathrm{e}^{-(\Gamma+\Gamma')t}\right)}{\Gamma + \Gamma'\mathrm{e}^{-(\Gamma+\Gamma')t}}, \end{aligned} \tag{5.9}$$

where the branching ratio of the spontaneous decay into the leaked level Γ'/Γ is typically small. For the $^{40}\text{Ca}^+$ optical qubits, we consider $\Gamma'/\Gamma \approx 4/9$ [153].

The leakage cannot be counteracted by error correction, therefore without some additional intervention, the leaked population will accumulate as the protocol proceeds, compromising the usefulness of the QEC. Therefore, to reduce the damage

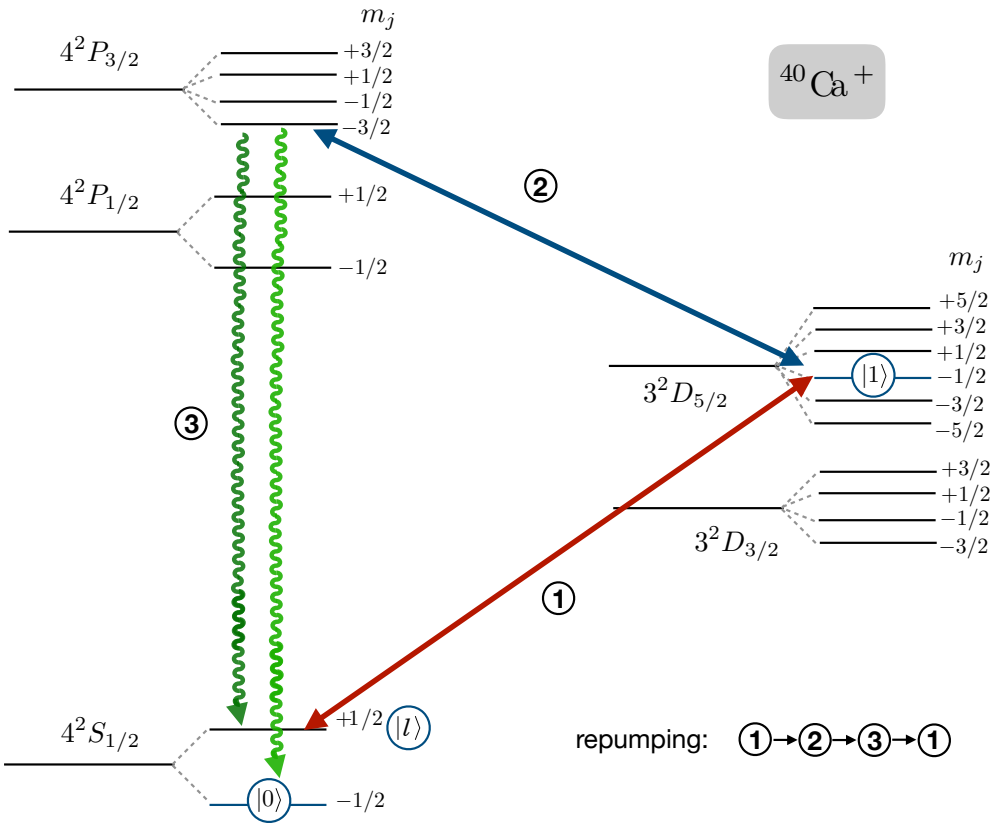


Figure 5.4: Repumping scheme for the leakage errors. The repumping cycle consists of (1) applying a π-pulse that brings the population of the leaked level to the metastable state, or otherwise hides the qubit in the $|0\rangle$ state. Then, (2) one applies a laser driving to the excited $P_{3/2}$ level, such that the population of the leaked state will (3) spontaneously decay into the ground state. By finally applying (1) another π-pulse, the population is repumped back into the computational subspace. This graph is taken from Ref. [142].

from leakage, we include a repumping pulse sequence that can bring the population of the electronic state back to the computational subspace. Fig. 5.4 shows the process. Here, (1) corresponds to a π-pulse between $|1\rangle$ and the leaked state $|l\rangle$. If the state of the qubit has not leaked, the quantum information is protected as the remaining operations (2)-(3) do not take place (the applied pulses have no effect), and the final (1) π-pulse brings the population back to $|1\rangle$. On the other hand, if the qubit had indeed leaked, (2) the lasers will bring this population up to an excited P level, which (3) will decay very fast into the ground state $|0\rangle$. At this point, the leakage has become a sort of amplitude damping, while the original coherences of the qubit state hidden in the ground state are still present. Then, a final (1) π-pulse brings the qubit back to the computational space.

5.3.3 Single-qubit rotations

As described in the last section, the single-qubit rotations are described by parameters θ and ϕ,

$$R(\theta,\phi) = e^{i\frac{\theta}{2}(\cos\phi\hat{X}+\sin\phi\hat{Y})} = \cos\frac{\theta}{2}\mathbb{I} + i\sin\frac{\theta}{2}(\cos\phi\hat{X}+\sin\phi\hat{Y}). \tag{5.10}$$

In contrast to the typical Krauss-map modelling of noise in QEC, our faulty gates are not characterised by a single error rate, but instead by two fluctuating functions $\delta\theta$ and $\delta\phi$ described by a random process with parameters fixed by experimental considerations.

The incorrect pulse area θ results from fluctuations of laser intensity, which gives the effective θ_{eff}

$$\theta_{\text{eff}}(t) = \theta\cdot\sqrt{\frac{I(t)}{\langle I(t)\rangle}}, \tag{5.11}$$

while the fluctuations of the laser phase lead to an error in the orientation of the rotation axis with,

$$\begin{aligned}\hat{X}_{\text{eff}}(t) &= \cos\delta\phi(t)\hat{X} + \sin\delta\phi(t)\hat{Y}\\ \hat{Y}_{\text{eff}}(t) &= \sin\delta\phi(t)\hat{X} + \cos\delta\phi(t)\hat{Y}.\end{aligned} \tag{5.12}$$

So the effective rotation is given by

$$R_{\text{eff}}(\theta_{\text{eff}},\phi) = \cos\frac{\theta_{\text{eff}}}{2}\mathbb{I} + i\sin\frac{\theta_{\text{eff}}}{2}(\cos\phi\hat{X}_{\text{eff}}+\sin\phi\hat{Y}_{\text{eff}}). \tag{5.13}$$

We use the Ornstein-Uhlenbeck random process to model the intensity fluctuation. As described in the modelling of the dephasing process, we could discretise the gate time into short periods of time and update the formula as,

$$F(t+\Delta t) = F(t)\mathrm{e}^{-\Delta t/\tau} + \left[\frac{c\tau}{2}\left(1-\mathrm{e}^{-2\Delta t/\tau}\right)\right]^{1/2} n. \tag{5.14}$$

Here $F(t) = \sqrt{\frac{I(t)-\langle I(t)\rangle}{\langle I(t)\rangle}}$. We take a short correlation time $\tau = T_2/10$ and $c = 20/T_2^2$. These parameters are chosen such that the fidelity of the various gates coincides with the values listed in Table 5.1.

For the phase fluctuation, we assume it occurs on a much slower timescale than the intensity fluctuation, as is typically the case in experiments. In the limit of very large correlation times, the Wiener process [154] is adopted to model the phase fluctuation. The standard form Langevin equation of the Wiener process $W(t)$ is

$$\frac{dW(t)}{dt} = \sqrt{c}\Gamma(t), \tag{5.15}$$

which can be solved by

$$W(t + \Delta t) = W(t) + \sqrt{c\Delta t}n, \tag{5.16}$$

where n is a random Gaussian variable. Therefore, we can write the fluctuation of phase by $\delta\phi = W(t + t_g) - W(t)$, where t_g is the gate duration time. We take $c = 0.01$ to fit the fidelity of the gate listed in Table 5.1.

5.3.4 Mølmer-Sørensen (MS) gate

We now consider the error model for the two-qubit MS gate[1]. The MS gates are mediated by the longitudinal centre-of-mass (CoM) mode of a crystal of two $^{40}Ca^{+}$ ions. As a starting point, we consider the MS gate suffers from motional errors from both the CoM and stretch modes, and collective dephasing (i.e. decoherence due to fluctuations of global magnetic fields). The time-evolution of the two qubits subjected to the laser-ion interaction is

$$\rho(t_g) = \mathrm{Tr}_{\mathrm{ph}}(U_g \rho_0 U_g^{\dagger}), \tag{5.17}$$

where t_g is the gate time, $\mathrm{Tr}_{\mathrm{ph}}$ represents partial trace over the phonons, and ρ_0 contains the initial qubit state and a thermal state for the vibrations with typical phonon numbers given in Table 5.1. U_g is the operation with the form of

$$U_g = e^{-it_g H_0} \mathsf{T} \left\{ e^{-i\int_0^{t_g} dt' H_c(t')} \right\}, \tag{5.18}$$

[1]This error model is solely developed by Alejandro Bermudez and evaluated with his own code, while the present author includes it for completeness of the thesis. The key elements of the error model are given in brief, and the details are explained in the paper [142]

where H_0 is the Hamiltonian for the uncoupled qubits and longitudinal phonons. T is the time-ordering operator. $H_c(t')$ describes the qubit-phonon coupling with the above sources of noise.

We solve numerically the equations above describing qubit-phonon dynamics, and perform process tomography [155] to express the final qubit state as the result of a generic quantum channel. Then, we can describe the noisy MS gate directly in terms of a set of Kraus operators with certain probabilities

$$\rho(t_g) = \sum_n p_n K_n \rho_0 K_n^\dagger, \qquad \sum_n p_n K_n^\dagger K_n = \mathbb{I}, \tag{5.19}$$

where K_n are the two-qubit Kraus operators to be found, and p_n are their corresponding probabilities. All the different sources of noise would result in a set $\{p_n\}_{n=1}^{16}$, where $p_1 \approx 1$ corresponds to the nearly ideal MS gate.

5.3.5 Measurement error

The dominant source of readout infidelity in both $^{40}Ca^+$ and $^{88}Sr^+$ optical qubits will likely be background counts for short readout times, which increase the dark and bright histogram overlap. To model it, we use a bit-flip error channel described as:

$$\epsilon(\rho) = (1-p)\rho + pX\rho X, \tag{5.20}$$

where p is the probability for the error to occur.

5.4 Numerical simulation results

Having introduced the QEC protocol and the error models for each operation, in this section we present our numerical results from exact wave function simulations for the performance of the seven-qubit code. Our simulation is based on the Monte Carlo method to perform exact simulation of the physical system using pure states. Once sufficiently many results are aggregated, one obtains the same data as would result from a single numerical run using a density matrix.

We use the Alice-Igor-Bob framework as introduced in Sec. 4.2 for assessment of the beneficial character of QEC. Both the non-FT and the flag-based FT protocols

are modelled, with the realistic error models describing the experimental architecture. Depending on the choice of trapped-ion parameters, two sets of figures are presented based on the reported-to-date and the anticipated parameters listed in Table 5.1.

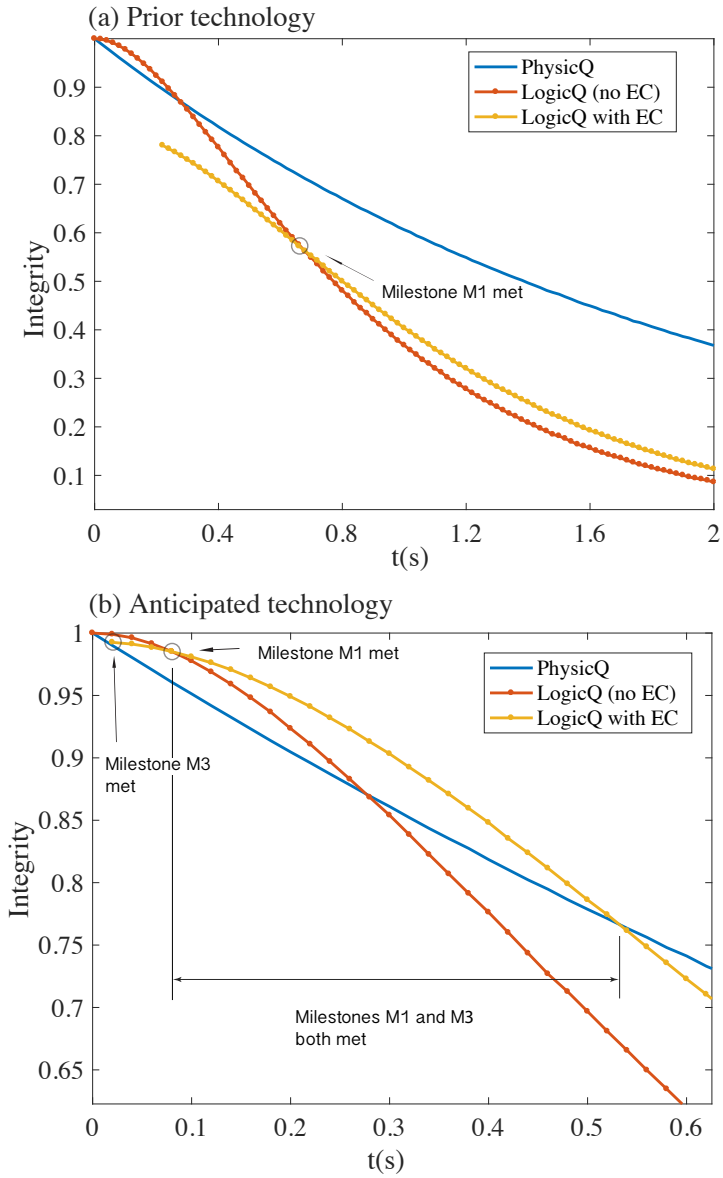


Figure 5.5: (a) The standard Alice-Igor-Bob protocol with the parameters underlying the simulation corresponding to the current values from Table 5.1. The red curve shown here denotes the integrity change of a logical qubit (noted as 'LogicQ') in a purely dephasing environment. For reference, the integrity change of a raw physical qubit (noted as 'PhysicQ') with the same environmental noise is also shown (the blue curve). We see that the non-FT QEC can produce a small positive effect versus strong environmental noise (i.e. there is a yellow-red line crossing). (b) The same model, but with parameters now matched to the expected fidelities as in Table 5.1. Performance is profoundly improved.

The first set of numerical results, shown in Fig. 5.5(a), shows the integrity change

with only dephasing noise. The Figure shows two reference lines: the blue line is the integrity of a raw physical qubit. The red line shows the integrity of an encoded quantum memory when no error correction is performed. As one would expect, for very short periods of environmental exposure the red line is superior to the blue since the probability of two (or more) errors is negligible. The yellow line shows the performance of the simple non-FT error correcting protocol; while it can never beat the integrity of a raw physical qubit, it does exceed the integrity of the encoded quantum memory once the environmental exposure is severe. Note that the yellow line does not start from $t = 0$, as the error correction takes finite amount of time, and we wait until Igor has enough time to perform a whole round of error correction. Fig. 5.5(b) shows the results of repeating the simulations in Fig. 5.5(a) but now with the future improved values from Table 5.1. One can see that the performance is profoundly improved. Now, the lines for the non-FT Igor outperforms both the raw physical qubit (blue) and the un-corrected quantum memory (red) over a substantial range. As the anticipated performance of the quantum memory is more interesting and significant, we now only show the numerical results with the future parameters for the following studies.

We now move on to the question of how the flag-based FT scheme (introduced in Sec. 4.1) performs. We take the non-FT curve as in Fig. 5.5(b) and redraw it, now with the FT scheme. As shown in Fig. 5.6, Igor indeed derives a positive benefit from the FT protocol on the left of the graph when the total environmental noise is weak. However, as noise accumulates, the FT curve gradually becomes less advantageous as compared to the non-FT one and crosses the latter at around 0.5 s. This is intuitively reasonable since at low error rates (below the threshold), the extra gates of the FT scheme pay off and it outperforms the non-FT one. At too high error rates (above the threshold), the extra gates to ensure fault-tolerance become relatively useless but do still give rise to gate errors. It is worth noting that for the first (leftmost) plotted points in the green and purple curves, the green has about an order of magnitude smaller loss of integrity, indicating the efficiency of the flag-based FT protocol.

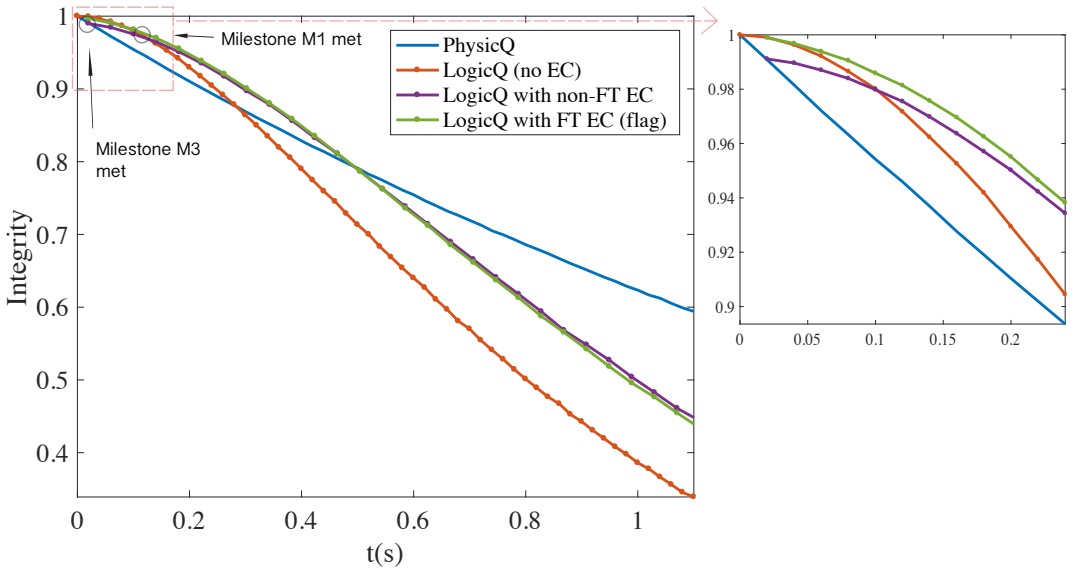


Figure 5.6: Comparison between the non-FT protocol and the flag-based FT protocol. The purple curve is the same as the yellow curve in Fig. 5.5(b). The green curve is the same as the purple except that Igor's protocol is upgraded to the flag-based FT approach as described in Sec. 4.1.

We now explore numerically the effects of amplitude damping and leakage noise. Environmental decoherence including dephasing, amplitude damping and leakage are all applied to the quantum channel. A branching ratio of 50/50 for damping/leakage is firstly considered in order to clearly assess the dangerous effects of leakage, and how to combat it with repumping. Fig. 5.7 shows the result. The black curve represents the integrity of a raw physical qubit and only serves as a reference. The dashed red curve stands for the integrity of the encoded memory, after a single round of flag-based FT QEC, where the spontaneous emission from the metastable state only contributes with amplitude damping (i.e. we artificially switch off the leakage). In this way, this line represents the optimal beneficial effects of Igor's error correction. With these limiting cases, we can now understand the effects of leakage noise. The green curve refers to the case where the leakage noise has been applied. As the figure shows, the integrity gets degraded considerably faster with respect to the noise model without leakage, leading to a much smaller region where QEC is beneficial in comparison to the raw single-qubit memory. However, as shown by the yellow curve, when repumping is applied prior to gates

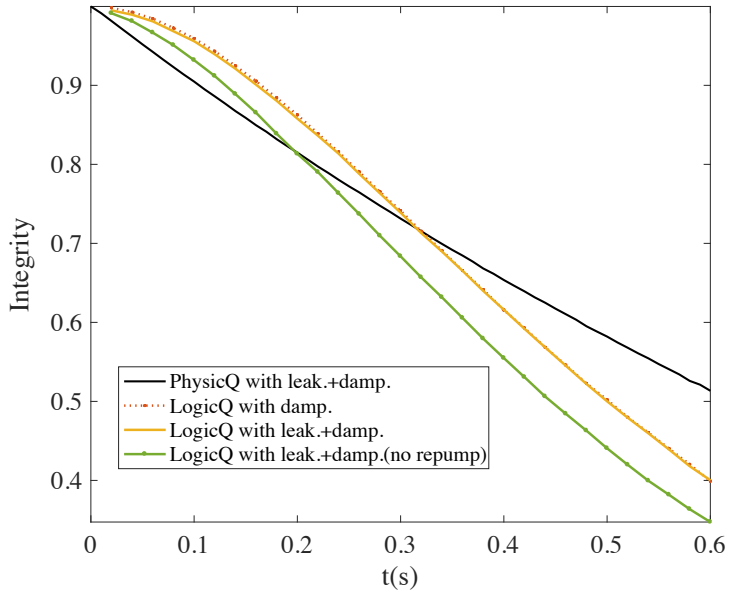


Figure 5.7: Integrity of raw and encoded quantum memories as a function of the memory time. The black curve represents the integrity degradation for a raw physical qubit due to environmental noise, including dephasing, spontaneous emission and leakage with a 50/50 damping/leakage ratio. The green curve represents the integrity for an encoded memory when Igor performs a single round of QEC. When the repumping sequence is applied prior to gates of Igor and Bob, the memory integrity increases considerably (yellow curve), almost reaching the maximum set by a noise model where spontaneous emission only occurs in the damping channel without any leakage (red dashed line).

of Igor and Bob, the degradation of the integrity almost reaches the optimal case where all the spontaneous decay occurs in the amplitude damping channel (the dashed red curve). These results also confirm that the repumping turns leakage into a sort of amplitude damping, which is correctable by the QEC code.

Finally, we incorporate all the elements together, i.e. the environmental noise is employed including dephasing, amplitude damping and leakage (with the damping/leakage branching ratio of 9/4), with repumping applied right before the gates of Igor and Bob. We show the effect of applying one, two, three or four rounds of flag-based FT QEC in Fig. 5.8. We see that we can sustain the encoded quantum memory at a (far) higher integrity than the physical qubit by selecting an appropriate number of error correction cycles according to the duration of the memory channel. This result compellingly meets and surpasses our 'milestone M4', as defined earlier in Sec. 4.2.2.

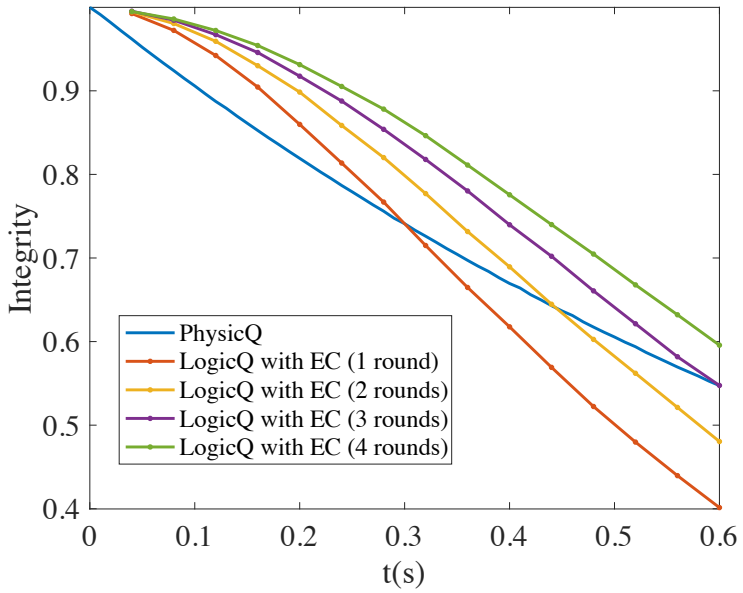


Figure 5.8: Integrity of the encoded quantum memory with multiple cycles of flag-based FT QEC. The blue curve represents the integrity degradation for a raw physical qubit with dephasing, spontaneous emission and leakage including a 9/4 damping/leakage ratio. We see that more cycles of error correction extend the life of the quantum memory significantly.

5.5 Conclusion

In this chapter we have presented a detailed QEC protocol based on a high-optical-access segmented ion trap. The operations of the crystal configurations and gate sets are designed considering the architecture of the hardware system. Realistic error models are developed for each stage of the QEC protocol. Numerical simulations are performed to assess the integrity of the quantum memory, incorporating the detailed protocol and error models, with both non-FT and FT schemes. The result suggests that with the currently reported experimental data, there is only a small advantage of the error-corrected logical qubit versus an unencoded logical qubit, while both are inferior to a raw physical qubit. However, beneficial QEC is demonstrated with the anticipated parameters. We also show that the damage from leakage can also be cancelled out by repumping. Finally, the flag-based approach is found to be superior for the expected trapped-ion technologies, and a much longer characteristic lifetime of the logical qubit can be realised by applying one or more rounds of error

correction. The results indicate the potential of the ion-trap quantum processor to achieve various milestones in an experimentally relevant scenario.

The error models considered in this chapter are beyond the standard depolarising model, but are not fully realistic with many simplifications. Further work may involve continuous improvement of error models, e.g. taking into consideration of spatial correlations for ions within the same zone and introducing the effect of cross talk.

6

Variational circuit compiler for quantum error correction

Contents

In Chapter 5 we presented a realistic quantum error correction protocol to be implemented on an ion-trap quantum processor. Based on the protocol, the exact sequences for control pulses and operations are derived to perform a complete error correction cycle. Incorporating all the realistic error models, we saw that beneficial error correction can not be achieved using parameters corresponding to certain 'current' devices (Table 5.1) but would require 'anticipated' progress.

The fact is, it is currently challenging for the whole quantum computing community to achieve proof-of-principle realisations using near-term noisy quantum devices. All the encoding, parity checks and decoding processes may include several dozens of imperfect gates as well as non-trivial environmental decoherence, so that

the error burden may go beyond the capability that a code can correct and therefore lead to a logical error that cannot be detected and corrected. From the experimental perspective, the manipulation of qubits should be improved to the highest accuracy. However, different physical systems have different hardware arrangements, control pulses, native types of multi-qubit gates, etc. For example, the natural entangling operation between two qubits with certain superconducting circuits and NV centres is a form of CNOT or CPhase gate [156–159], while it is a Mølmer-Sørensen gate for many trapped ion systems [160, 161], and sqrt-SWAP gate may be the native operation in quantum dot systems [162, 163]. Meanwhile, conventional methods for the realisation of error correcting codes do not necessarily involve rigorous analytic or numerical optimisation and therefore may have an unnecessarily large number of gates. It is thus theoretically important to optimally compile quantum error correcting protocols against a specific hardware target.

In this chapter, we introduce a quantum compiler as an approach to optimally support any given noisy quantum hardware where formal circuit equivalence may be difficult or impossible to determine. We construct a variational compiler to automatically search for the optimal circuit that encodes a target logical state of an error correcting code. The main idea is to make use of the variational quantum eigensolver to search for the ground state energy. Our compiler first maps the problem into a ground state searching problem where the desired logical state is the ground state of the given (synthetic) Hamiltonian. Next, we consider a set of parameterised ansatz circuits and make use of the variational imaginary time evolution method [65, 73] to find the ground state, thus discovering the encoding circuit. The ansatz circuit can be tailored to meet specific requirements of any hardware system and any optimisation target.

The structure of this chapter is as follows. In Sec. 6.1, we introduce the variational compiling algorithm including the construction of the Hamiltonian, the design of ansatz, and the variational imaginary time evolution. In Sec. 6.2, we show numerical realisations of the compiling algorithm for different logical states of the five-qubit and seven-qubit codes. We compare our results to existing

circuits and discuss its applicability in NISQ computing. We discuss potential applications and summarise in Sec. 6.3.

The concept of the variational compiling algorithm was developed collectively by all the authors in Ref. [164]. All numerical implementation and modelling were performed by the present author.

6.1 Variational circuit compiling for quantum error correction

In this section, we introduce the variational circuit compiler for preparing a logical state of an error correcting code. The key idea is to construct a Hamiltonian so that the target logical state is its ground state. Then with a parameterised ansatz circuit, we optimise the parameters in order to find the ground state of the Hamiltonian and hence the encoding circuit of the target logical state.

6.1.1 Variational circuit compiler

We first show that any logical state of a stabiliser code can be straightforwardly described as the ground state of a Hamiltonian. Suppose an error correcting code is stabilised by the generator set $\{g_i\}$. By definition we have

$$g_i \left|\Phi\right\rangle_L = \left|\Phi\right\rangle_L, \quad \forall i \tag{6.1}$$

where $\left|\Phi\right\rangle_L$ is an arbitrary logical state. The logical state can be uniquely determined by the whole stabiliser set plus an additional logical operator

$$O_L = \left|\Phi\right\rangle_L \left\langle\Phi\right|_L - \left|\Phi\right\rangle_L^{\perp} \left\langle\Phi\right|_L^{\perp}, \tag{6.2}$$

which satisfies $O_L \left|\Phi\right\rangle_L = \left|\Phi\right\rangle_L$. Suppose $\left|\Phi\right\rangle_L = \alpha \left|0\right\rangle_L + \beta \left|1\right\rangle_L$, O_L can be further decomposed as a linear sum of logical Pauli operators X_L, Y_L, and Z_L as

$$O_L = (\alpha\beta^* + \alpha^*\beta)X_L - i(\alpha^*\beta - \alpha\beta^*)Y_L - (\beta\beta^* - \alpha\alpha^*)Z_L. \tag{6.3}$$

In general, one can always construct a set of logical operators that determines a particular logical state. Therefore, we can construct a Hamiltonian,

$$H = -\sum_i c_i g_i - c_o O_L, \tag{6.4}$$

such as the target state $|\Phi\rangle_L$ be its unique ground state

$$H|\Phi\rangle_L = -\left(\sum_i c_i + c_o\right)|\Phi\rangle_L, \tag{6.5}$$

with energy $E_0 = -(\sum_i c_i + c_o)$. Note that since we choose c_i and c_o, we know the correct E_0. As the terms of the Hamiltonian commute with each other, the eigenstate of the Hamiltonian should be the eigenstate of each term. Thus, it is not hard to see that the first excited state has an energy $E_1 = E_0 + 2\min\{c_i, c_o\}$.

To find the encoding circuit that prepares the target logical state $|\Phi\rangle_L$, we employ the variational method for determining the Hamiltonian's ground state. The general approach was outlined in Sec. 2.3.1. We first prepare a trial state via a parameterised quantum circuit, the so-called ansatz, $\psi(\vec{\theta}) = V(\vec{\theta})|\bar{0}\rangle$, where $|\bar{0}\rangle$ refers to a quantum register of which all the data qubits are initialised at $|0\rangle$, and $V(\vec{\theta})$ is described with m parameters, $V(\vec{\theta}) = V_m(\theta_m)...V_2(\theta_2)V_1(\theta_1)$. Suppose the ground state of the Hamiltonian H can be represented by the circuit ansatz, the problem is then rephrased as finding an set of parameters $\vec{\theta}_{min}$ which minimises the energy

$$E_{\min} = \min\left\{\langle\psi(\vec{\theta})|H|\psi(\vec{\theta})\rangle\right\}. \tag{6.6}$$

The minimisation can be accomplished by any optimisation algorithm, such as simple gradient descent or the imaginary time evolution [65, 73]. For this work, we adopt the imaginary time evolution approach introduced in Sec. 2.3.2, which can either be realised with a classical emulator for small error correcting codes, or can be implemented on a quantum computer for general codes, with the circuits described in Sec. 2.3.3.

In practice, we may not be able to find the exact ground state, for example because it lies outside the set of states reachable from the ansatz, or because of the existence of gate noise, etc. Suppose the minimal energy we can find is $E_{\min}$, then we can lower bound the fidelity between the state ρ we find and the target logical state $|\Phi\rangle_L$ according to

$$F = \langle\Phi|\rho|\Phi\rangle_L \geq 1 - (E_{\min} - E_0)/c, \tag{6.7}$$

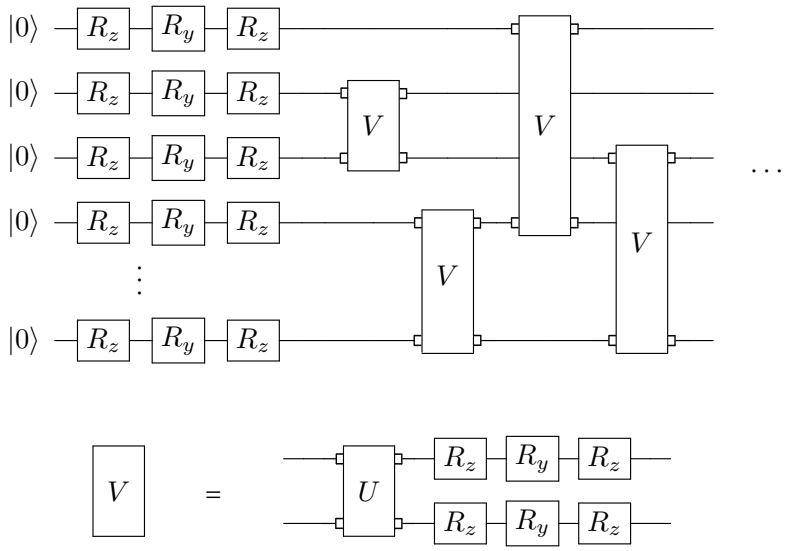


Figure 6.1: Building block of the ansatz. The circuit starts with three single-qubit gates applied to each of the data qubits. Then elementary blocks are randomly inserted into the circuit. Each elementary block consists of one two-qubit gate followed with three single-qubit gates on both of the two data qubits. R_y and R_z represent a single-qubit rotation over the Y and Z axes of the Bloch sphere respectively, where the rotation angle is the parameter to be updated over time.

where we denote $c = 2\min\{c_i, c_o\}$ and assume $E_{\min} \in [E_0, E_1]$. The fidelity was defined in Eq. (4.3) in Chapter 4. The proof of Eq. (6.7) can be found in Appendix B. Therefore, when observing an energy that is close to the ground state energy, we are assured that the state is indeed close to the exact ground state. We thus now only focus on minimising the energy of the Hamiltonian.

6.1.2 The ansatz

Our variational circuit compiler assumes the logical state can be prepared by a parameterised ansatz. For different quantum hardware, the ansatz can have different structures. The general structure of the ansatz used here is shown in Fig. 6.1. We consider parameterised single-qubit gates rotating along the Pauli basis. For example, the gate R_x is defined as $R_x = \exp\left(-i\frac{\theta}{2}X\right)$ with the rotation angle being a variable parameter θ. The definitions of R_y and R_z are similar. We also consider general two-qubit gates composed by a fixed two-qubit gate followed by six parameterised single-qubit gates. As shown in Fig. 6.1, where three single-qubit

rotations with different parameters are firstly applied to each of the data qubits following the order of $R_z R_y R_z$. The gate set is chosen such that an arbitrary single-qubit rotation can be realised. With given constraints in the connectivity of the qubits and the type of two-qubit gates that can be realised in a given quantum hardware, a certain number of multi-qubit sets are then inserted into the circuit, where one gate set consists of a multi-qubit gate followed with three single-qubit gates applied to each of the data qubits.

For a given ansatz, which is either randomly generated or following a certain procedure, we update the parameters of the single-qubit gates in order to minimise the energy of the Hamiltonian. Based on different ansätze or different initial values of parameters, we could either reach the ground state verified by the ground state energy, or we may reach a local minimum. In the latter case, it may indicate that the ansatz cannot represent the target state, so the experiment is abandoned and restarted until the energy reaches close to the ground state energy. In order to minimise the number of parameters or single-qubit gates, circuit compilation is applied with the following rule: after each several steps, gates with small parameters are removed; this process continues until the energy starts to increase. The technique cannot guarantee to find a circuit with minimum number of parameters, though many equivalent circuits can be found and selected.

In practice, one may also need to change the structure of the ansatz when the previously selected ansatz is not powerful enough to represent the target state. Different strategies can be applied here by either adding more single and two-qubit gates or fully adopting a different ansatz structure. Trying all possible ansatz structures is in general impossible for a large error correcting code. Therefore one may either systematically explore a given family of ansatz structures, or alternatively we may apply some kind of circuit morphing algorithm to the ansatz (as recently discussed in Ref. [55, 165]). In the following, we focus on the five- and seven-qubit codes, and show numerical emulation of the variational compiler for these two codes with different ansatz structures.

6.2 Numerical simulations

In this section we present numerical simulations for the variational circuit compiler. The simulation is performed on a classical computer with the Quantum Exact Simulation Toolkit (QuEST) package [166], which is a high performance classical simulator written in C/C++. We focus on the five- and seven-qubit codes and consider two scenarios respectively with noiseless and noisy gates. For the noiseless case, the pure state is modelled, while the mixed state is applied for noisy circuits. For both cases, the imaginary time evolution is adopted as the optimisation algorithm, with the methods described in Sec. 2.3.2. Several particular states are considered in the simulation, including eigenstates of the Pauli basis $|0\rangle_L$, $|-\rangle_L$ and the magic state $|T\rangle_L = (|0\rangle + e^{\pi i/4}|1\rangle)/\sqrt{2}$. For the Hamiltonian, we set the coefficients of all the terms to be the same and normalise them so that the ground state energy is -1. In particular, the Hamiltonian corresponding to a logical state $|\Phi\rangle_L$ of the five- or seven-qubit code is

$$H = -\frac{1}{n}\left(\sum_i g_i + O_L\right), \tag{6.8}$$

where g_i are the stabilisers for the five- or seven-qubit code (described in Sec. 4.1), O_L is the logical operator defined in Eq. (6.2), $n = 5$ for the five-qubit code, and $n = 7$ for the seven-qubit code. We can verify that $H|\Phi\rangle_L = E_0|\Phi\rangle_L$ with $E_0 = -1$ and $E_1 = -(n-2)/n$ for the first excited state. Following Eq. (6.7), when we find a state ρ with energy $E_{\min}$, its fidelity to the target state $|\Phi\rangle_L$ is lower bounded by

$$F \geq 1 - \frac{n}{2}(E_{\min} + 1), \tag{6.9}$$

when $E_{\min} \in [-1, -(n-2)/n]$.

We also consider different constraints of the circuit topology for different hardware structures, as revealed in the choice of ansätze. The constraints could be from practical experimental limitations or inferred from preferences in a future experimental design. In this chapter, we take three constraints as examples to illustrate our method:

1. Minimise the number of two-qubit gates, as most quantum hardware has a lower fidelity for two-qubit gates;

2. Only use a single type of two-qubit gate, reflecting the fact that hardware typically supports one entangling process at the physical level;

3. Only apply nearest-neighbour interactions, such as in superconducting qubit systems.

We apply one constraint at a time, while when searching for circuits satisfying (2) or (3), (1) is also applied by default, as more simplified circuits are usually preferred. At the start of one experiment, an ansatz is generated based on the constraints, with all parameters initialised from a small value around zero. The parameters are then updated through the variational imaginary time algorithm, until the energy becomes static in a local minimum, in which case the ansatz is abandoned, or goes to the ground state energy, in which case we consider the current ansatz is successfully configured. The number of parameters is also gradually reduced in the simulation process: if a certain parameter is found to be around zero, a further simulation is attempted with that gate omitted. This procedure continues until the energy starts to increase. With this trick, we manage to largely reduce the number of parameters and accelerate the searching process.

In the following, we give several examples of applying our compiler for certain logical states prepared with the five- and seven-qubit code.

6.2.1 Compiling with ideal gates

We first consider the case where gates are assumed to be perfect. In this case, we can focus on the optimisation with pure state imaginary time evolution as described in Sec. 2.3.2.

We first consider the five-qubit code. By considering the circuit with the minimum number of two-qubit gates, we first rediscover the circuits for encoding the $|-\rangle_L$ state, which is consistent with the latest conventional circuit as shown in Fig. 4.1(a). With five Controlled phase gates, the circuit is also capable of encoding

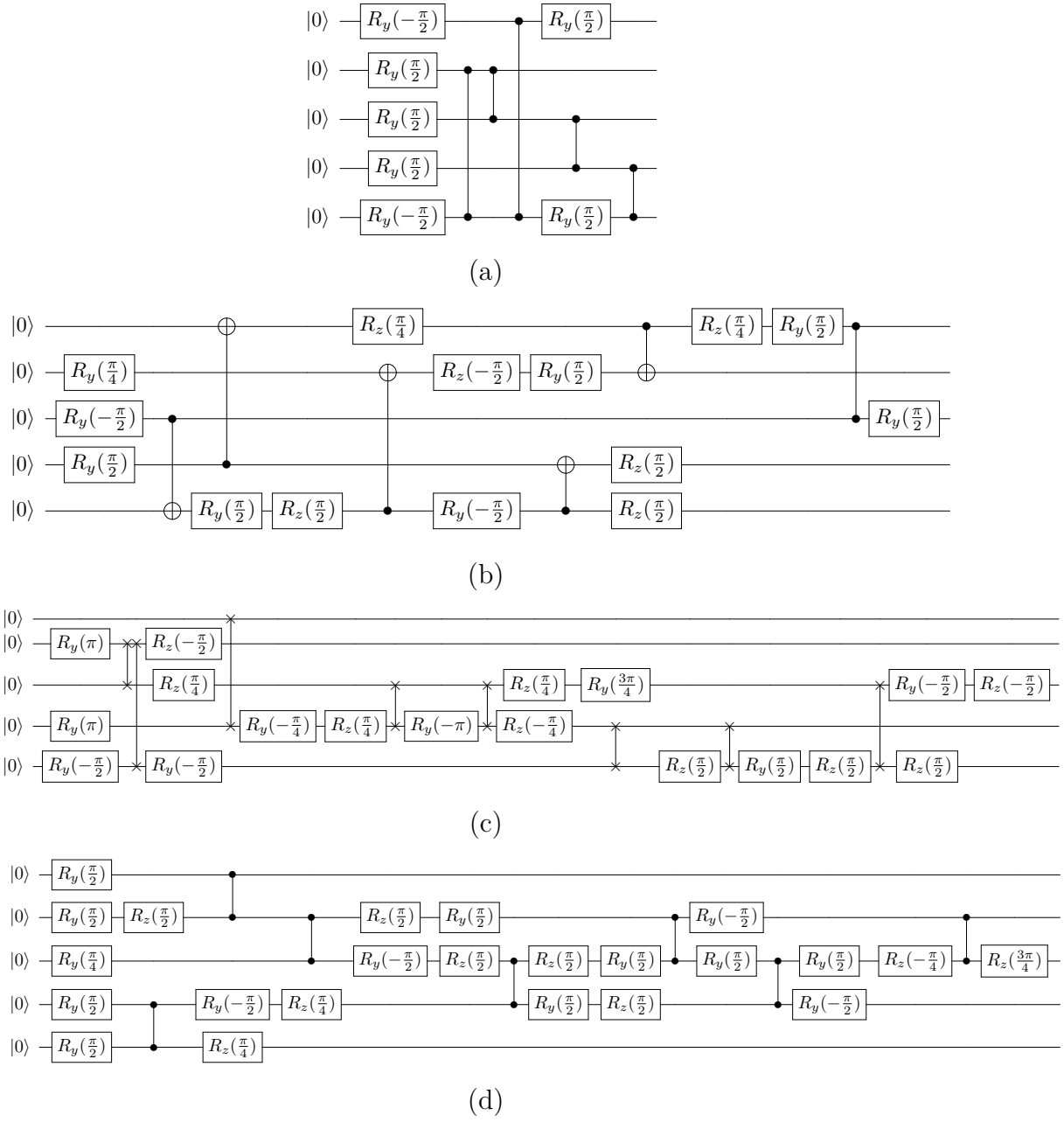


Figure 6.2: (a) Circuit to encode a logical minus state $|-\rangle_L$, with the five-qubit code. This circuit is equivalent to Fig. 4.1(a). (b) Circuit to encode a logical magic state $|T\rangle_L = \frac{\sqrt{2}}{2}(|0\rangle_L + e^{\frac{\pi}{4}i}|1\rangle_L)$, with the five-qubit code. Our method found at minimum six two-qubit gates are required to prepare the state. (c) Circuit to encode a logical minus state $|-\rangle_L$ with the five-qubit code. The two-qubit gate is restricted to be a sqrt-SWAP gate. (d) Circuit to encode a logical magic state $|T\rangle_L = \frac{\sqrt{2}}{2}(|0\rangle_L + e^{\frac{\pi}{4}i}|1\rangle_L)$. Only nearest-neighbour CPhase gates are permitted in this case.

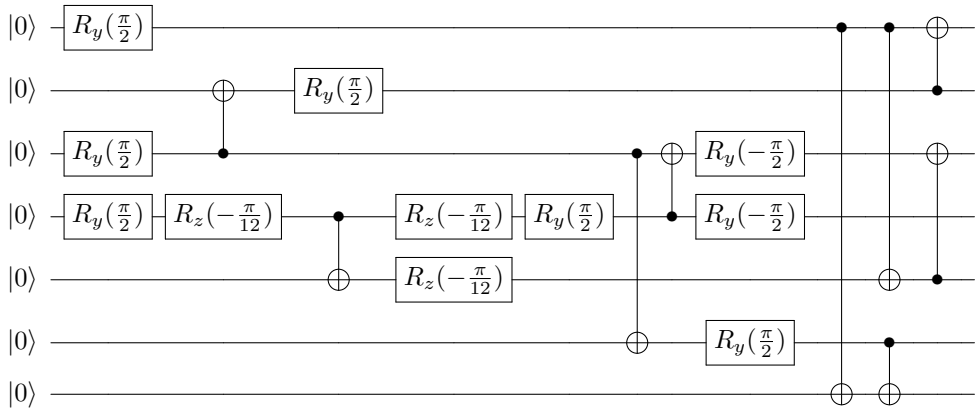


Figure 6.3: Circuit to encode a logical magic state $|T\rangle_L = \frac{\sqrt{2}}{2}(|0\rangle_L + e^{\frac{\pi}{4}i}|1\rangle_L)$ with the seven-qubit code. Minimally 9 two-qubit gates are required to prepare the state.

the $|0\rangle_L$ state with additional transversal single-qubit gates. As our method is capable of discovering different but equivalent circuits, we show one example in Fig. 6.2(a), which encodes a logical minus state $|-\rangle_L$.

Next, we apply our compiler for the magic state $|T\rangle_L$ and we find the encoding circuit as shown in Fig. 6.2(b). The encoding circuit for the magic state is also consistent with the circuit for encoding an arbitrary logical state as shown in Fig. 4.1(b). Therefore, we didn't find a more efficient circuit to encode the magic state.

In addition to rediscovering existing encoding circuits, our compiler can also find efficient encoding methods when considering a variety of constraints on the circuit structure. First, we consider the case where sqrt-SWAP gates are considered as the only type of two-qubit gates in the ansatz. The sqrt-SWAP gate is a non-Clifford gate, so it is not often seen in conventional error correction encoding circuits. On the other hand, it may be a natural two-qubit gate [167]. The canonical approach would be to convert this gate into a CNOT/CPhase gate in a circuit design. As one CNOT gate is decomposed into two sqrt-SWAP gates and several single-qubit rotations, at minimum 10 sqrt-SWAP gates are then required to encode a $|-\rangle_L$ state. We show here that by applying the sqrt-SWAP gate into the ansatz directly, the number of the sqrt-SWAP gates can be reduced to eight as shown in Fig. 6.2(c). Next, we consider the case where the ansatz is restricted to allow only nearest-neighbour interactions, which is common for solid state qubits. We present in Fig. 6.2(d) a

circuit satisfying the constraint, which prepares the magic state $|T\rangle_L$ with seven nearest-neighbour CPhase gates. (Note that here we specified that CPhase gates should be the only two-qubit gates; if we relax this and subsume certain single-qubit and CPhase gates into CNOT gates, then this circuit simplifies further.)

For the seven-qubit Steane code, we also first rediscover the encoding circuits for the $|0\rangle_L$ state as shown in Fig. 4.4(a), which has 8 CNOT gates. For the magic state, we find a circuit that only uses 9 CNOT gates, in contrast to the encoding circuit for an arbitrary state, which requires 11 CNOT gates as shown in Fig. 4.4(c).

6.2.2 Noise-robust circuits

In this section, we consider the practical scenario with noisy gates. The aim is to find the most noise-robust encoding circuit for preparing a target logical state. For each gate, we apply a small probability of depolarising noise as follows,

$$\rho' \to (1-r)\rho + \frac{r}{3}(X\rho X + Y\rho Y + Z\rho Z). \tag{6.10}$$

The depolarising noise is considered here in order to more readily compare to known methods and results. Here, an additional ancilla qubit is introduced to the circuit and post-selected in a similar way to fault-tolerant state preparation. Note that when the ancilla is not entangled with the physical qubits, it reduces to the previous case where there is no ancilla. As the encoded state is a mixed state, we make use of the mixed state mode of the QuEST and the imaginary time evolution for mixed states described in Sec. 2.3.2. We search over a large number (order 10000) of different ansätze, and obtain the circuit corresponding to the lowest energy. We then verify that this circuit performs better in the presence of small gate noise than the circuits found in the earlier section, which were found under the zero-noise assumption.

For the five-qubit code, a noise-robust circuit is found for encoding the logical minus state $|-\rangle_L$ as shown in Fig. 6.4. The circuit contains two parts, with the first part (gates on the data qubits) preparing a logical minus state, while the second part (gates between the data qubits and the ancilla) detecting and post-selecting errors. Note that the second part is a logical $\bar{X}$ operator which does not change the logical state.

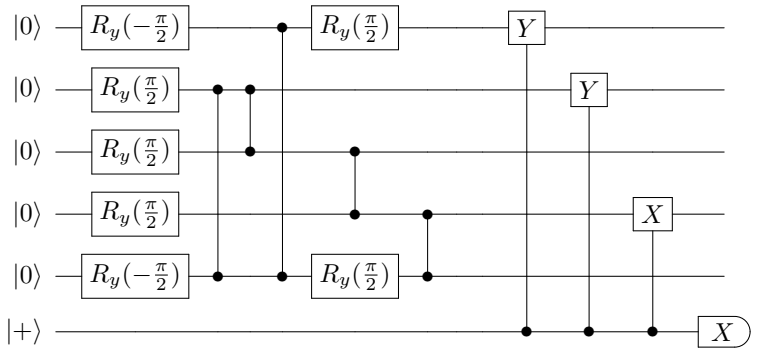


Figure 6.4: A noise-robust circuit to encode a logical minus state $|-\rangle_L$ with the five-qubit code. In addition to the first part of the circuit (Fig. 6.2(a)) which prepares $|-\rangle_L$, three two-qubit gates are applied from the ancilla qubit, which is then measured in the $\{+,-\}$ basis. If measured to be $-$, the output is abandoned and re-prepared until the measurement result is $+$.

It is interesting to reflect further on the principle of the circuit which our automated method has found. We could generalise the rule to any error correction codes with transversal Pauli gates. For example, a noise-robust circuit to encode any of the logical states $|0\rangle_L, |1\rangle_L, |+\rangle_L, |-\rangle_L, |+i\rangle_L, |-i\rangle_L$ could be realised by preparing the states with the non-fault-tolerant (non-FT) circuits first and applying a transversal logical operator (which does not change the logical state) for detecting and post-selecting errors. We notice that by combining the transversal logical operator with the stabiliser operator, some Pauli terms can be cancelled out. If replaced with this new operator for error detection, the circuit is noise-robust. Note that the circuit shown in Fig. 6.4 is not fully fault fault-tolerant, as the logical state prepared with the five-qubit code can be corrupted by any single error, while measuring out one logical operator and applying post selection cannot guarantee FT detection of any single error. However, circuits to prepare some logical states with the Steane code applied with error detection can be realised fault-tolerantly, as the error detection needs only to detect a certain type of noise, while the logical states are immune to the other type of noise. For example, a logical $|0\rangle_L$ encoded with the Steane code is immune to any phase noise. The circuit discovered by Goto [132] to prepare a FT logical zero state $|0\rangle_L$ with the Steane code (as shown in Fig. 4.4(b)) is an example.

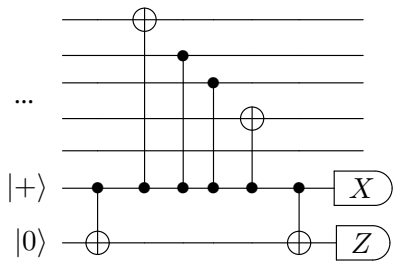


Figure 6.5: Circuit to fault-tolerantly measure one stabiliser operator with the five-qubit code. The first and second ancillae need to be measured to be + and 0, or otherwise the output is abandoned.

For the Steane code, one can implement transversal Clifford gates, so one may feel that the same trick can be applied. That is, to test the fault tolerance of the circuit, we measure the logical operator P, when it belongs to the Clifford group and satisfies $P\left|\Phi\right\rangle_L = \left|\Phi\right\rangle_L$, with $\left|\Phi\right\rangle_L$ being the target logical state. Unfortunately, if an error occurs to a data qubit, an extra gate needs to be applied to the data qubit even if the ancilla is measured to be +. Therefore, post error detection or error correction procedure is still required to remove the single error, as pointed out in Ref. [132, 168].

In general, there are no universal transversal logical operators for small error correcting codes. Thus we cannot apply the same trick to fault-tolerantly prepare an arbitrary logical state, such as the magic state of the five- and seven-qubit code. However, a FT circuit can be realised by measuring all the stabiliser operators and applying post selections: the ancilla should always be measured to be 1, otherwise the circuit is abandoned and restarted from the beginning. Note that measuring all the stabiliser operators cannot guarantee fault tolerance of the circuit – a logical error created before the error detection cannot be found. As it takes a relatively long time to prepare a logical state, this approach may not be ideal for systems with short coherence time. On the other hand, one can measure one of the stabiliser operators and still benefit partially.

We also note that measuring the stabiliser operators can also be fault-tolerant with an extra flag qubit, as shown in Fig. 6.5. After a logical state is prepared non-fault-tolerantly with the five-qubit code, we measure one of the stabiliser

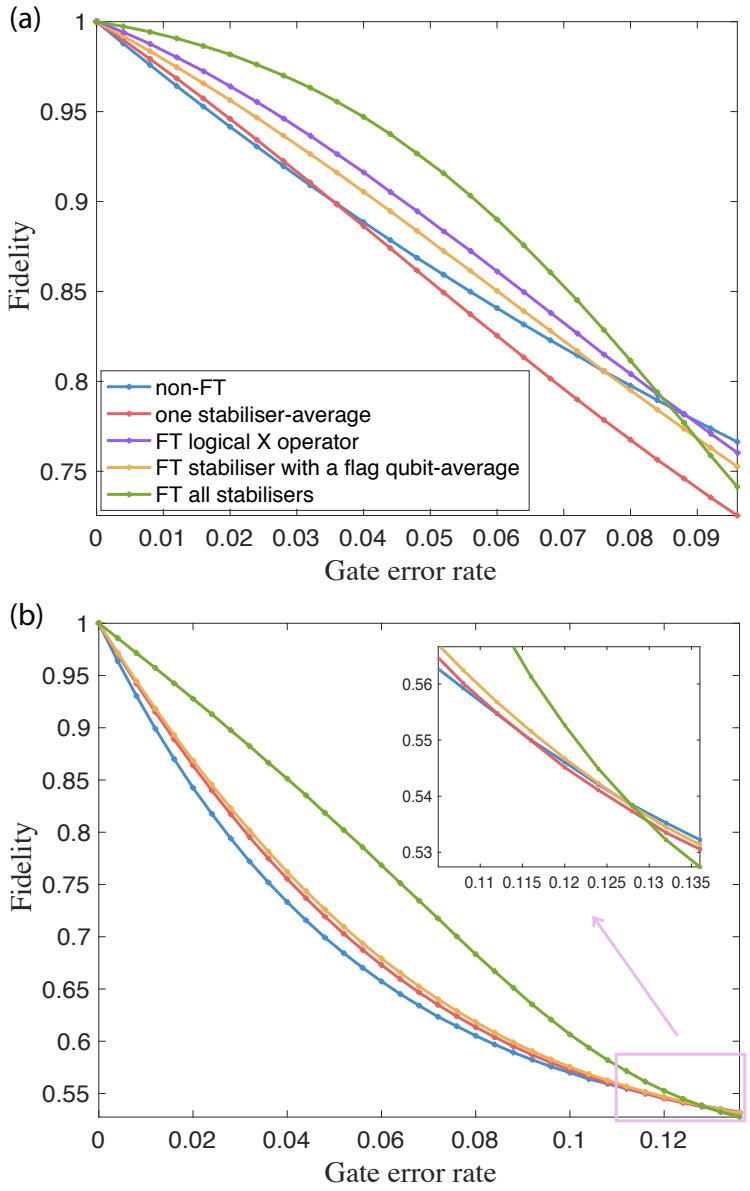


Figure 6.6: The fidelity change with the increasing gate error rate in preparation of (a) a logical minus state $|-\rangle_L$ with the five-qubit code and (b) a logical magic state $|T\rangle_L = \frac{\sqrt{2}}{2}(|0\rangle_L + e^{\frac{\pi}{4}i}|1\rangle_L)$ with the seven-qubit code. The error rate for single-qubit gates, two-qubit gates and measurements is the same. The blue curves represent the case where the logical state is prepared with the non-FT circuits in Fig. 6.2(a) and Fig. 6.3 for the two codes respectively. If one of the four/six stabiliser operators is measured with one ancilla and post selection is applied, the average behaviour of the four/six cases is shown as the red curves, while the orange ones refer to the same scenario but the operators are measured fault-tolerantly with two ancillae. The purple curve in (a) refers to the case where a logical X operator is measured, with the circuit shown in Fig. 6.4, except that the error detection is conducted fault-tolerantly with an additional ancilla. The greens refer to the case where all the stabiliser operators are measured with two ancillae.

operators. The flag qubit is to detect errors occurring between one of the two-qubit gates as shown in the figure.

Finally, we compare the fidelity of the prepared logical state with different encoding circuits as discussed above. In Fig. 6.6, we show the fidelity change with respect to an increasing gate error rate. The error rate is the same for single-qubit gates, two-qubit gates and measurement. A logical minus state is prepared with the five-qubit code with circuit shown in Fig. 6.2(a). We see that with small noise, applying error detection always leads to a higher fidelity. However, as the gate error rate gradually increases, the extra noise introduced with the extra gates negates the advantage of error detection. In Fig. 6.6(a), there is a small gap between the blue and the red curves, indicating that measuring one stabiliser operator non-fault-tolerantly only gains a small benefit over the case where no error detection is performed. However, such an advantage increases if the measurement is conducted fault-tolerantly with one more ancilla. The purple curve, representing post selection by measuring the X operator, has an overall higher fidelity than the red and orange ones probably due to one fewer two-qubit gate applied. If all the stabiliser operators are measured fault-tolerantly with post selection applied, as demonstrated by the green curve, the fidelity is higher than the non-FT circuit when the gate error rate is smaller than 8.6%. Note we have verified that in this case, the circuit involving measuring all the stabilisers fault-tolerantly is fault-tolerant (i.e. in the limit of very small gate error rate, a quadratic curve can be observed if a round of error correction is applied right after the state preparation.). In Fig. 6.6(b), a logical magic state $|T\rangle_L = \frac{1}{\sqrt{2}}(|0\rangle_L + e^{\frac{\pi}{4}i}|1\rangle_L)$ is prepared with the seven-qubit code. Compared with (a), we see a group of curves with different shapes but a similar trend, that the curves applied with post selection have a higher fidelity given a small gate error rate. The advantage starts to vanish when the gate error rate is larger than 11%. In this case, we found the circuit involving measuring the full stabiliser set is not fault-tolerant but still leads to a notably higher state fidelity. The result suggests the noise-robustness of our method.

6.3 Conclusion

In this chapter, we introduce the variational circuit compiler for efficiently encoding the logical state of an error correcting code. A Hamiltonian is constructed so that the target logical state is its ground state and it can be found with the variational imaginary time evolution method. The five- and seven-qubit codes are considered as examples. When having noiseless operations, we show the encoding circuit to prepare different logical states with the minimal number of gates for different hardware structures. When considering noisy gates, we discover the encoding circuit to prepare a target state with the highest fidelity. By introducing ancillary qubits and applying post selection, noise-robust circuits are found for preparing logical states. The fidelity of logical states prepared with different encoding circuits with respect to different gate error rates are also compared.

Future studies may focus on the design and searching of ansätze for different codes and the realisation of the compiler with a real quantum computer. Also, as the approach described is not limited to error models or type of codes – through the automated discovery process it will be possible to find optimal circuits relevant to newly emerging codes, or for bespoke error models that are matched to specific hardware implementations.

7

Variational Algorithms for linear algebra

Contents

In the last chapter we described an automatic circuit compiler based on the variational algorithm. Here we continue to exploit the potential of the variational algorithm and apply it to solve linear algebra problems. Linear algebra problems are essential components in science and technology and have many applications in a wide spectrum of fields. With the fastest known classical approaches, the complexity scales roughly polynomially with N and c, where N is the matrix size, and c is the condition number, the ratio of the largest eigenvalue to the smallest eigenvalue of a given matrix. In matrix inversion problems, the condition number is an important measure that quantifies the sensitivity of the solution to perturbations of the input data. The solution is less stable with a larger condition number.

In recent years, people have sought to use quantum computers to solve linear algebra problems. A prominent example is the quantum algorithm for solving linear systems of equations by Harrow, Hassidim, and Lloyd (HHL) in Ref. [169]. Given an N by N sparse matrix M and a state vector $|v_0\rangle$, the HHL quantum algorithm can prepare a state that is proportional to $|v_M\rangle = M^{-1}|v_0\rangle$, with a complexity polynomial in $\log_2 N$ and condition number c under certain conditions. Therefore, the HHL algorithm suggests exponential speed-ups of quantum computers. Recent developments of quantum algorithms for linear systems of equations can be found in Refs. [170–176].

Another common linear algebra task is the matrix-vector multiplication by applying a sparse matrix M to a vector $|v_0\rangle$ such as $|v_{M^{-1}}\rangle = M|v_0\rangle$. The quantum algorithms for linear equations can be similarly applied for matrix-vector multiplications. Furthermore, they can be applied for quantum optimisations and machine learning [177–179].

Those conventional quantum algorithms generally require a long-depth circuit and thus are challenging to be realised with the current technology. In this chapter, we propose variational algorithms for linear algebra problems, including linear systems of equations and matrix-vector multiplications, implemented with NISQ hardware. The main idea is to construct a many-body Hamiltonian so that its ground state corresponds to the solution of the linear algebra problem. We present the Hamiltonian construction, circuit implementation and solution verification in Sec. 7.1. To address the 'barren plateau' problem [67] when searching for the solution, we introduce Hamiltonian morphing with an adaptive ansatz in Sec. 7.2. In Sec. 7.3 we show that the variational matrix-vector multiplication algorithm can be applied for Hamiltonian simulation as an alternative to the one proposed by Li and Benjamin [60]. Numerical simulation results are presented for solving linear systems of equations in Sec. 7.4. We conclude the chapter in Sec. 7.5.

All the authors in Ref. [180] collectively conceived the idea of this work. The analytical result was generated jointly by the Xiao Yuan and the present author. Numerical implementation and modelling were performed by the present author.

7.1 Variational algorithms for linear algebra

7.1.1 Hamiltonian construction

Matrix-vector multiplication

We first introduce our variational algorithms for matrix-vector multiplication. Given a sparse N by N matrix M and an initial state vector $|v_0\rangle$, our task is to calculate the normalised state

$$|v_M\rangle = \frac{M|v_0\rangle}{\|M|v_0\rangle\|}, \tag{7.1}$$

with $\|M|v_0\rangle\| = \langle v_0|M^\dagger M|v_0\rangle$ and $M|v_0\rangle \neq 0$. Here, we consider the case that M can be a general (non-Hermitian) matrix. We find that $|v_M\rangle$ is the ground state of the Hamiltonian

$$H_M = I - \frac{M|v_0\rangle\langle v_0|M^\dagger}{\|M|v_0\rangle\|^2}, \tag{7.2}$$

with energy 0. Using universal quantum computers, one may apply the conventional techniques [181, 182], such as adiabatic state preparation and phase estimation to find the ground state of H_M. With NISQ devices, we can instead consider the variational method with a parametrised state. As has been described in Chapter 6, the main idea is to first consider a state or ansatz, $|\phi(\vec{\theta})\rangle = U(\vec{\theta})|0\rangle$, with $U(\vec{\theta}) = U_L(\theta_L)\dots U_2(\theta_2)U_1(\theta_1)$, $\vec{\theta} = (\theta_1, \theta_2, ..., \theta_L)$. Suppose the ground state of H_M can be represented by the ansatz with certain parameters, the ground state finding problem is then converted to a minimisation problem,

$$\vec{\theta}_{\min} = \arg\min_{\vec{\theta}} \langle\phi(\vec{\theta})|H_M|\phi(\vec{\theta})\rangle, \tag{7.3}$$

with the solution given by $|v_M\rangle = |\phi(\vec{\theta}_{\min})\rangle$.

The search for the optimal set of parameters based on the cost function can be generalised as an optimisation problem, which can be handled with many methods, such as gradient based classical optimisation [36], global search algorithms [72], imaginary time evolution [65, 73], etc. For this work, we apply the gradient descent method, which we introduced in Chapter 2 (Sec. 2.3.2).

In the gradient descent method [36], we start with a guess of the solution, $\vec{\theta}_0$ and update the parameters along the negative gradient of the energy $E_M(\vec{\theta}) = \langle\phi(\vec{\theta})|H_M|\phi(\vec{\theta})\rangle$,

$$\vec{\theta}_{i+1} = \vec{\theta}_i - a\nabla E_M(\vec{\theta}_i),\ \forall i = 0, 1, \ldots, T \tag{7.4}$$

where a is the time step, and T is the total number of steps. As the energy E_M monotonically decreases with sufficiently small a, the solution via VQE always at least corresponds to a local minimum. Starting from several random initial positions, one may find the true solution, i.e., the global minimum.

Linear system of equations

The second problem considered here is solving linear equations. Given a sparse N by N matrix M and an initial state vector $|v_0\rangle$, we want to calculate

$$|v_{M^{-1}}\rangle = \frac{M^{-1}|v_0\rangle}{\|M^{-1}|v_0\rangle\|}, \tag{7.5}$$

where $\|M^{-1}|v_0\rangle\| = \sqrt{\langle v_0|M^{-1}M^{-1}|v_0\rangle}$, $M^{-1}|v_0\rangle \neq 0$. When the matrix M is not Hermitian, it can always be converted into an equivalent problem with a Hermitian matrix by doubling the number of qubits [169]. We therefore focus on the case where M is Hermitian and invertible.

It is not hard to see that the solution $|v_{M^{-1}}\rangle$ is the ground state of the Hamiltonian

$$H_{M^{-1}} = M^\dagger(I - |v_0\rangle\langle v_0|)M \tag{7.6}$$

with energy 0. Again, one can apply adiabatic algorithms to find the ground state with a universal quantum computer [176]. Here, we focus on the variational algorithm and consider the following optimisation problem

$$\vec{\theta}_{\min} = \arg\min_{\vec{\theta}} \langle\phi(\vec{\theta})|H_{M^{-1}}|\phi(\vec{\theta})\rangle, \tag{7.7}$$

with $|v_{M^{-1}}\rangle = |\phi(\vec{\theta}_{\min})\rangle$. The optimisation could then be realised with the same approaches as introduced above.

7.1.2 Implementation with quantum circuits

Now, we consider the implementation of our algorithms with quantum circuits that has been described in Sec. 2.3.3 in Chapter 2. To realise the VQE optimisation, we need to obtain $\nabla E_{M^{-1}}$ or ∇E_M, which can be estimated either with the finite difference formulae or quantum gradient finding methods recently considered in Ref. [60, 71, 183].

Matrix-vector multiplication

With the finite difference formulae, each component $\partial E_M/\partial\theta_i$ of the gradient $\nabla E_M = (\partial E_M/\partial\theta_i)$ around $\vec{\theta}$ can be calculated as

$$\frac{\partial E_M}{\partial\theta_i} \approx \frac{E_M(\vec{\theta}+\delta\theta_i) - E_M(\vec{\theta}-\delta\theta_i)}{2\delta\theta_i}, \tag{7.8}$$

with $\delta\theta_i \ll 1$. Note that from Eq. (7.2), it follows that $E_M(\vec{\theta}) = 1 - |\langle\phi(\vec{\theta})|M|v_0\rangle|^2$ with an arbitrary angle $\vec{\theta}$ (Here we take normalised $M|v_0\rangle$ such as $\|M|v_0\rangle\| = 1$). As M is sparse, suppose M can be decomposed as $M = \sum_j \lambda_j\sigma_j$, we can then obtain $E_M(\vec{\theta})$ as

$$E_M(\vec{\theta}) = 1 - \Big|\sum_i \lambda_i\langle\phi(\vec{\theta})|\sigma_i|v_0\rangle\Big|^2. \tag{7.9}$$

For each term, we denote $\langle\phi(\vec{\theta})|\sigma_i|v_0\rangle = \langle 0|U|0\rangle$ with $U = U(\vec{\theta})^\dagger\sigma_i U_{v_0}$ and $|v_0\rangle = U_{v_0}|0\rangle$. The real and imaginary part of $\langle\phi(\vec{\theta})|\sigma_i|v_0\rangle$ can be evaluated with the circuit shown in Figure 7.1(a). Suppose M consists of a polynomial number (with respect to the number of qubits) of tensor products of local operators σ_j, we can then measure each term and efficiently calculate $E_M(\vec{\theta})$. Meanwhile, even if $M = \sum_j \lambda_j\sigma_j$ has exponential number of terms with respect to the number of qubits, the algorithm is still efficient if we can efficiently sample the probability distribution $|\lambda_i|/\sum_i|\lambda_i|$ with a polynomial dependence of $\sum_i|\lambda_i|$ on the number of qubits. This includes cases where the coefficients have analytical expressions, which enables efficient sampling.

With the quantum gradient finding method, we first express the gradient as

$$\frac{\partial E_M}{\partial\theta_i} = -2\,\mathrm{Re}\left(\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}M|v_0\rangle\langle v_0|M^\dagger|\phi(\vec{\theta})\rangle\right). \tag{7.10}$$

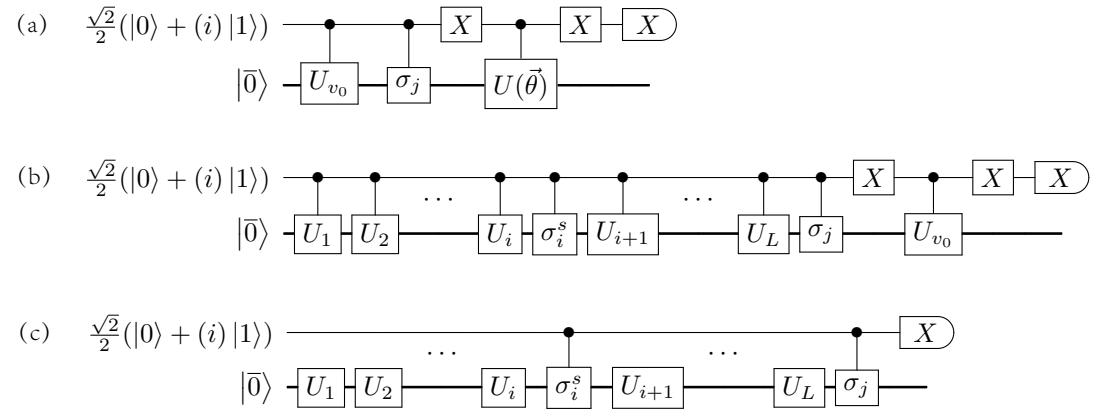


Figure 7.1: The circuits to implement ∇E_M or $\nabla E_{M^{-1}}$. The first qubit is the ancilla, and the second refers to the data qubit register. The real part can be obtained from measuring $\langle X \rangle$ when the ancilla is initialised at $\frac{\sqrt{2}}{2}(|0\rangle + |1\rangle)$, while the imaginary part is evaluated if the ancilla is initialised at $\frac{\sqrt{2}}{2}(|0\rangle + i\,|1\rangle)$.

The term $\langle v_0|\, M^\dagger |\phi(\vec{\theta})\rangle$ can be efficiently measured with the aforementioned method. To measure $\frac{\partial \langle \phi(\vec{\theta})|}{\partial \theta_i} M\,|v_0\rangle$, we have to first decompose the derivative of the state as

$$\frac{\partial |\phi(\vec{\theta})\rangle}{\partial \theta_i} = \sum_s f_i^s |\phi_i^s(\vec{\theta})\rangle. \tag{7.11}$$

Here $|\phi_i^s(\vec{\theta})\rangle = U_L(\theta_L)\ldots\sigma_i^s U_i(\theta_i)\ldots U_2(\theta_2)U_1(\theta_1)\,|0\rangle$, as the derivative of the unitary is decomposed as $\partial U_i(\theta_i)/\partial\theta_i = \sum_s f_i^s \sigma_i^s U_i(\theta_i)$ with tensor products of local operators σ_i^s and coefficients f_i^s. Then we have

$$\begin{aligned}\frac{\partial \langle \phi(\vec{\theta})|}{\partial \theta_i} M\,|v_0\rangle &= \sum_s f_i^{s*} \langle \phi_i^s(\vec{\theta})| M U_{v_0}\,|0\rangle \\ &= \sum_j \sum_s d_i^{s,j}\, \langle 0|\, U_1^\dagger(\theta_1) U_2^\dagger(\theta_2)\ldots U_i^\dagger(\theta_i)\sigma_i^s \ldots U_L^\dagger(\theta_L)\sigma_j U_{v_0}\,|0\rangle\,,\end{aligned} \tag{7.12}$$

where each term can be evaluated with the circuit shown in Figure 7.1(b).

Linear system of equations

The circuits to implement $\nabla E_{M^{-1}}$ can be found similarly as we do for ∇E_M. Using Eq. (7.6), the partial derivative of $\nabla E_{M^{-1}}$ is expressed as

$$\begin{aligned}\frac{\partial E_{M^{-1}}}{\partial \theta_i} &= \frac{\partial}{\partial \theta_i}(\langle\phi(\vec{\theta})|H_{M^{-1}}|\phi(\vec{\theta})\rangle) = 2\,\mathrm{Re}\left(\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}H|\phi(\vec{\theta})\rangle\right) \\ &= 2\,\mathrm{Re}\left(\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}M^\dagger M|\phi(\vec{\theta})\rangle\right) - 2\,\mathrm{Re}\left(\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}M^\dagger\,|v_0\rangle\,\langle v_0|\,M|\phi(\vec{\theta})\rangle\right).\end{aligned} \tag{7.13}$$

Suppose M can be decomposed into combinations of Pauli terms $M = \sum_j \alpha_j\sigma_j$, $M^\dagger M$ can then be expressed as $M^\dagger M = \sum_j \beta_j\sigma_j$, where the coefficients α and β for each term are different. We decompose the derivative of the state as

$$\frac{\partial|\phi(\vec{\theta})\rangle}{\partial\theta_i} = \sum_s f_i^s|\phi_i^s(\vec{\theta})\rangle, \tag{7.14}$$

the same way as described above. Then the first term in Eq. (7.13) is expressed as

$$\begin{aligned}&\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}M^\dagger M|\phi(\vec{\theta})\rangle = \sum_s f_i^{s*}\langle\phi_i^s(\vec{\theta})|M^\dagger M|\phi(\vec{\theta})\rangle \\ &= \sum_j\sum_s c_i^{s,j}\,\langle 0|\,U_1^\dagger(\theta_1)U_2^\dagger(\theta_2)\ldots U_i^\dagger(\theta_i)\sigma_i^s\ldots U_L^\dagger(\theta_L)\sigma_j U_L(\theta_L)\ldots U_2(\theta_2)U_1(\theta_1)\,|0\rangle\,.\end{aligned} \tag{7.15}$$

The real and imaginary value of each term

$$\langle 0|\,U_1^\dagger(\theta_1)U_2^\dagger(\theta_2)\ldots U_i^\dagger(\theta_i)\sigma_i^s\ldots U_L^\dagger(\theta_L)\sigma_j U_L(\theta_L)\ldots U_2(\theta_2)U_1(\theta_1)\,|0\rangle$$

can be evaluated with the circuit shown in Figure 7.1(c), by initialising the ancilla qubit to be at $\frac{\sqrt{2}}{2}(|0\rangle + |1\rangle)$ and $\frac{\sqrt{2}}{2}(|0\rangle + i\,|1\rangle)$ respectively.

The second term can be separated as two parts, $\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}M^\dagger\,|v_0\rangle$ and $\langle v_0|\,M|\phi(\vec{\theta})\rangle$. The first part is decomposed into $\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}M^\dagger\,|v_0\rangle = \sum_s f_i^{s*}\langle\phi_i^s(\vec{\theta})|M^\dagger\,|v_0\rangle$, which can be efficiently measured with the aforementioned method. The second part is written as

$$\langle v_0|\,M|\phi(\vec{\theta})\rangle = \langle 0|\,U_{v_0}^\dagger MU(\vec{\theta})\,|0\rangle = \sum_j k_j\,\langle 0|\,U_{v_0}^\dagger\sigma_j U(\vec{\theta})\,|0\rangle \tag{7.16}$$

where each term can also be evaluated with the same method as above.

7.1.3 Solution verification

After finding the solution, it is also practically important to verify whether it is the correct one. This is in general impossible for conventional VQE as neither the ground state nor the ground state energy is known. However, we can verify the solution of the linear algebra problems as the exact solution should always have zero energy. We first consider the matrix-vector multiplication problem. With a solution $|\phi(\vec{\theta}_{\min})\rangle$, we can estimate $|\langle\phi(\vec{\theta}_{\min})|v_M\rangle|^2$ via

$$|\langle\phi(\vec{\theta}_{\min})|v_M\rangle|^2 = 1 - E_M(\vec{\theta}_{\min}). \tag{7.17}$$

Therefore, we can also have the fidelity of $|\phi(\vec{\theta}_{\min})\rangle$ to the exact solution $|v\rangle$ from the energy $E_M(\vec{\theta}_{\min})$ and the solution can be verified as correct whenever $|\langle\phi(\vec{\theta}_{\min})|v\rangle|^2$ is close to 1. Similarly, the solution of the linear equation can be verified by measuring the energy $E_{M^{-1}}(\vec{\theta}_{\min})$ to be close to 0. Note that the fidelity $|\langle\phi(\vec{\theta}_{\min})|v_{M^{-1}}\rangle|^2$ cannot be determined as we cannot directly measure $|\langle\phi(\vec{\theta}_{\min})|M^{-1}\,|v_0\rangle\,|^2$ or $\langle v_0|\,M^{-1}M^{-1}\,|v_0\rangle$. Alternatively, we can verify the solution by checking whether $|v_0\rangle \propto M|\phi(\vec{\theta}_{\min})\rangle$ is satisfied. Therefore, we can measure $|\,\langle v_0|\,M|\phi(\vec{\theta}_{\min})\rangle|^2/\langle\phi(\vec{\theta}_{\min})|M^\dagger M|\phi(\vec{\theta}_{\min})\rangle$ and the solution is verified as correct when the value is close to 1.

7.2 Optimisation via Hamiltonian morphing

The VQE method tries to find the global minimum of a large multi-parameter space. For each trial, one starts from a randomly initialised parameter set and obtains a local minimum after several optimisation steps. One can then verify whether the solution is correct by measuring the energy and restart from another random parameter set until the energy is close to 0. This may need to be repeated many times until a satisfactory solution can be found. There are several potential issues that can affect the practical performance. First, as the parameter space is large, it may require a large number of repetitions starting from different initial random parameters. This is a common issue in optimisation and machine learning, where a

combination of different algorithms may help to speed up the search time. A more serious problem is that the gradient may vanish for randomly initialised parameters in some random quantum circuits of large systems [67]. In quantum computational chemistry, physically motivated ansätze and a good guess of initial parameters are considered to avoid vanishing gradient, as discussed in Refs. [184, 185]. However, this is not straightforward for the linear algebra tasks.

Here we employ a Hamiltonian morphing optimisation method to avoid these problems. Similar methods have been introduced in Refs. [186, 187] for variational quantum simulation. The morphing method is an analog to adiabatic state preparation, where one slowly varies the Hamiltonian from H_i to H_f, so as to gradually evolve the corresponding ground state from $|\psi_i\rangle$ to $|\psi_f\rangle$. For our morphing method, we take the linear equation problem as an example and it works similarly for the matrix-vector multiplication problem. Denote a time-dependent Hamiltonian $H_{M^{-1}}(t) = M(t)(I - |v_0\rangle\langle v_0|)M(t)$ with $M(t)$ satisfying $M(t=0) = I$ and $M(t=T) = M$ respectively. The morphing technique starts from the ground state $|v_0\rangle$ of $H_{M^{-1}}(0)$ and reaches the ground state of the target Hamiltonian $H_{M^{-1}}(T)$ as follows. Consider discretised time step δt and $M(t) = (1 - t/T)I + t/T \cdot M$.

1. At the $n = 0$ step with time $t = 0$, we denote the parameters that represent $|v_0\rangle$ as $\vec{\theta}(0)$, i.e., $|v_0\rangle = |\phi(\vec{\theta}(0))\rangle$.

2. At the $n^{\text{th}} \in [1, T/\delta t)$ step, we use parameters $\vec{\theta}((n-1)\delta t)$ found in the last step for $H_{M^{-1}}((n-1)\delta t)$ as the initial position to find parameters $\vec{\theta}(n\delta t)$ that correspond to the ground state of $H_{M^{-1}}(n\delta t)$.

For any positive matrix M, when we use a sufficiently small δt and a powerful enough ansatz, the method is guaranteed to find the exact solution according to the adiabatic theorem [188]. For a general M, one can introduce an extra qubit so that the gap $H_{M^{-1}}(t)$ is non-vanishing, as discussed in Ref. [176]. In practice, the initially specified ansatz may not be powerful enough to represent the states at all time t. Fortunately, when the ansatz is insufficient at any time t, one can detect it by measuring a high averaged energy. Then one can either simply choose a more powerful ansatz or adaptively vary the ansatz by changing its structure and adding

more gates [55]. Note that this is in general not possible for conventional many-body ground state problems with VQE, owing to the unknown ground state energy.

7.3 Hamiltonian simulation

The matrix-vector multiplication algorithm we introduced in this chapter can also be applied for variational Hamiltonian simulation. Starting from $|\psi_0\rangle$, the Hamiltonian simulation task is to prepare the state $|\psi_t\rangle = e^{-iHt}|\psi_0\rangle$ at time t with sparse Hamiltonian H. For instance, when the Hamiltonian can be decomposed as a linear sum of polynomial terms $H = \sum_j \lambda_j \sigma_j$, we can make use of the Trotterisation method to have

$$|\psi_t\rangle \approx \left(\prod_i e^{-i\lambda_j\sigma_j t/K}\right)^K |\psi_0\rangle + O(t^2/K), \tag{7.18}$$

with K Trotter steps. As the circuit depth increases with the evolution time, the circuit depth can be large for long evolution time. The variational quantum algorithm for simulating real-time dynamics with a constant circuit depth has been proposed by Li and Benjamin [60]. Here, we present an alternative algorithm based on matrix-vector that can be more robust than the one in Ref. [60]. The key idea of Ref. [60] is to prepare the state at any time t via a parameterised state $|\phi(\vec{\theta}(t))\rangle$ and update the parameters from $\vec{\theta}(t)$ to $\vec{\theta}(t+\delta t))$ that effectively realise $e^{-iH\delta t}|\phi(\vec{\theta}(t))\rangle$ via a deterministic process $\vec{\theta}(t+\delta t)) = \vec{\theta}(t) + \dot{\vec{\theta}}(t)\delta t$ with $\dot{\vec{\theta}}(t) = A^{-1}C$, where $A_{i,j}(t) = -\mathrm{Im}\left(\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}\frac{\partial|\phi(\vec{\theta})\rangle}{\partial\theta_j}\right)$ and $C_i(t) = \mathrm{Re}\left(\frac{\partial\langle\phi(\vec{\theta})|}{\partial\theta_i}H|\phi(\vec{\theta})\rangle\right)$. Each term of A and C can be efficiently measured via a quantum circuit.

Instead of deterministically obtaining $\vec{\theta}(t+\delta t))$ via $\dot{\vec{\theta}}(t)$, we make use of the algorithm introduced in this chapter. That is, to realise $e^{-iH\delta t}|\phi(\vec{\theta}(t))\rangle$, we can consider $M = e^{-iH\delta t} \approx 1 - iH\delta t$ and update the parameters as $|\phi(\vec{\theta}(t+\delta t))\rangle = M|\phi(\vec{\theta}(t))\rangle$. Therefore, for each time step, we can update the parameters according to the matrix-vector algorithm.

7.4 Numerical simulation

We numerically test the variational algorithm for solving linear systems of equations. The simulation is based on the Quantum Exact Simulation Toolkit (QuEST) package [166] as has been mentioned in Chapter 6. In our simulation, we consider 2^n by 2^n positive complex Hermitian matrices M with n qubits. Note that even though the matrix size is assumed to be an exponential of 2, an m by m matrix with $2^{n-1} < m \leq 2^n$ can be also handled with an n-qubit circuit. The matrix is randomly generated with a given condition number c and the input state is also randomly generated.

As we use no prior information regarding the matrix, we consider the ansatz as shown in Fig. 7.2, where each green blocks V_i has a fixed structure. We also vary the number of blocks, called the circuit depth m, to endow the ansatz with different powers. In a more practical scenario, we can assume that we have a sparse decomposition of the matrix and the input state is simple to prepare or it is obtained from a previous calculation.

We also implement the optimisation based on Hamiltonian morphing with an adaptive ansatz. We divide the total time T into 10 intervals with an equal duration, where T ranges from 20 to 100 depending on the matrix size. At $t = 0$, we set $H_{M^{-1}} = I - |v_0\rangle\langle v_0|$ with the ground state being the input state $|v_0\rangle$ and start with all single-qubit gates parameterised with a small random rotation angle. In the i^{th} variational cycle, the gradient descent method is used to search for the ground state energy of a time-dependent Hamiltonian $H_{M^{-1}}(t) = M(t)(I - |v_0\rangle \langle v_0|)M(t)$ with $M(t) = (1 - i/10)I + i/10 \cdot M$, and a step size $\delta t = 0.1$. We choose initial parameters to be the ones obtained from the previous cycle. A small circuit depth m is tried at the beginning and is increased gradually until the fidelity of the qubit state to the target state is higher than 99%, which is regarded as a success.

We study the complexity of our algorithm with respect to the matrix size and the condition number of the matrix. We first fix the matrix size to be $2^6 * 2^6$ with 6 qubits and consider random matrices with different condition numbers. For different condition numbers from 10 to 100, we test our algorithm with different

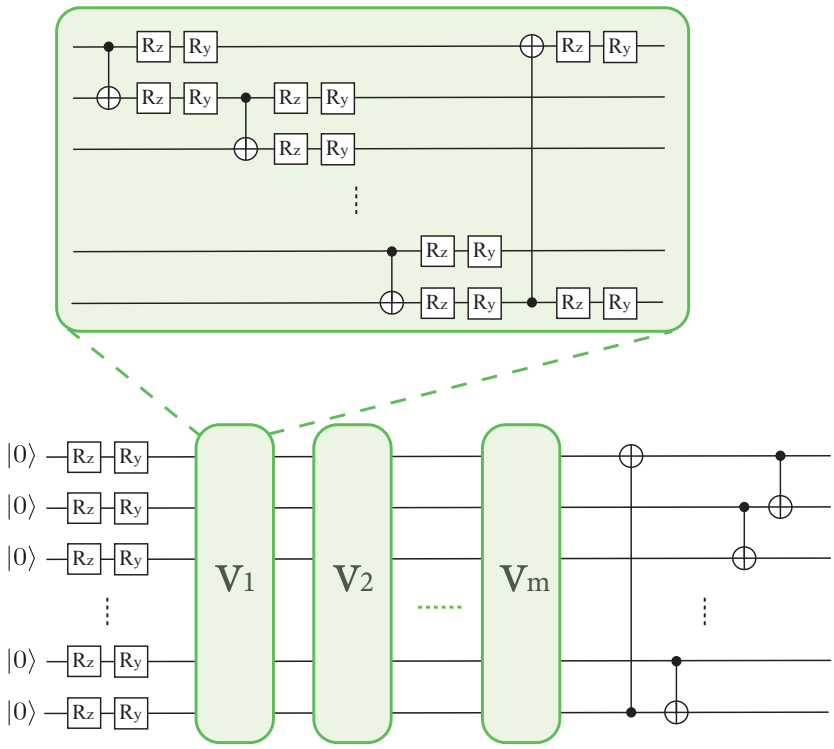


Figure 7.2: The circuit ansatz. In total n qubits are required to solve a problem with a $2^n * 2^n$ matrix. R_y and R_z represent single qubit rotations around the Y and Z axes, respectively. The rotation angle (parameter) for each single-qubit gate is initialised from a small random value, which is updated in each of the variational cycle. At the beginning, two single-qubit gates are applied to each of the data qubits, followed with sets of CNOT gates, as depicted as the green block V. Within each set, a CNOT gate is followed with two single-qubit gates. The number of sets represents the depth of the circuit, which is gradually increased until a desired target state is found. In the end, a set of CNOT gates is applied in reverse order as in V. The aim for this set is to ensure that the eigenstate can be found at $t = 0$, when the Hamiltonian morphing technique is applied.

circuit depths. For each case, we run 4940 random trials to calculate the averaged success probability with results shown in Fig. 7.3. Not surprisingly, we found that a larger circuit depth ansatz generally leads to a higher success probability. We also obtain the minimum depth that is required to guarantee the success of finding $|v_{M^{-1}}\rangle$, plotted as an insert. Overall, we find that the minimally required depth varies linearly with the condition number.

Next, we vary the size $2^n * 2^n$ of the matrix M, by changing the number of data qubits n in the circuit ansatz. For an n-qubit system, the condition number is adopted as $c = 10n$. As shown in Fig. 7.4, the circuit depth leading to a

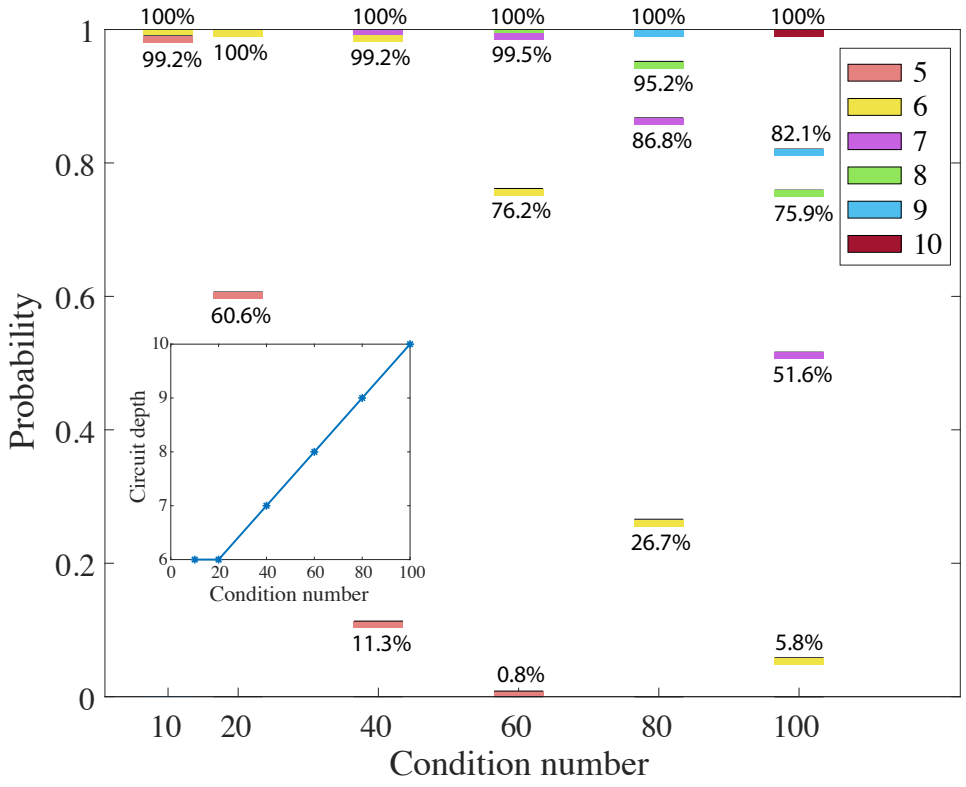


Figure 7.3: The probability to find the target state with increasing condition number and circuit depths. This experiment is based on a 6-qubit circuit, thus the matrix size is fixed to be 64 *64. For a given condition number, the success (having a solution with a fidelity higher than 99%) probability is shown for the circuit with increasing depth, until it reaches 100% (i.e., all the 4940 runs). The minimum depth required to find the target state with 100% success probability is plotted in the inserted graph.

non-zero success probability also increases with respect to the number of qubits. The inserted plot shows the minimum circuit depth for achieving a 100% success probability. Overall, we find a super-linear increase of the circuit depth with respect to the number of qubits.

This is not surprising as we consider random matrices with random input states. Even though we consider fixed condition numbers, it only determines properties of eigenvalues of the matrix. For any matrix M, all other matrices $UMU^\dagger$ with unitary U have the same eigenvalues and hence the same condition number. As the unitary U is an arbitrary unitary, it can lead to a general solution state, which may be impossible to prepare with shallow circuits. This explains the super-linear dependence of the circuit depth with respect to the number of qubits. One possible

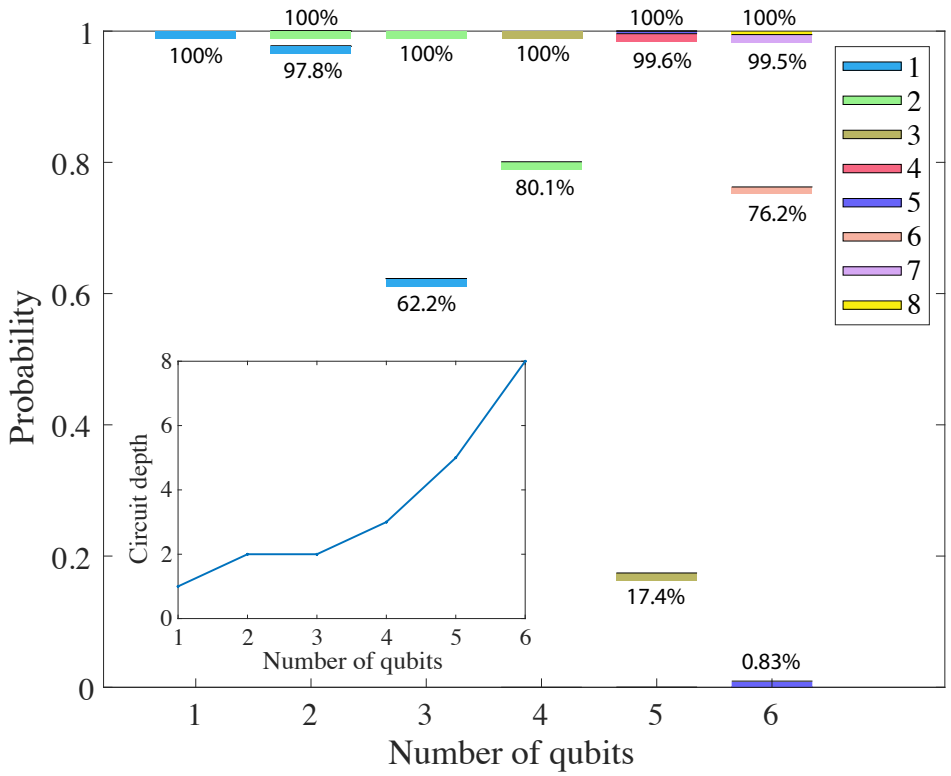


Figure 7.4: The probability to find the target state with increasing number of qubits and circuit depths. For a given number of qubits, circuit depth is increased gradually until the probability reaches 1. The inserted graph shows the minimum depth found in the simulation when the target state is found with 100% probability.

solution to this problem is to consider sparse matrices as in Ref. [169]. Meanwhile, a better ansatz whose design is informed by the matrix can be designed. For example, we note that the chosen ansatz is not optimal for the problem, and a much more compact one can be obtained from circuit compiling [55] as shown in the Appendix C. As our algorithm has the capability of verifying the solution, it also enables us to dynamically vary the ansatz until the solution is found. In our simulation, we dynamically increase the circuit depth and we expect more profound circuit morphing techniques can be designed and exploited.

7.5 Conclusion

In this chapter we present variational algorithms for solving linear algebra problems including linear system of equations and matrix-vector multiplications with NISQ

devices. A Hamiltonian is designed such the ground state is the solution of the target problem, which is then solved with the variational algorithms. As the ground state has zero energy, it also enables us to verify the solution and adjust the ansatz.

A novel Hamiltonian morphing technique is also introduced to avoid local minima and to accelerate the search progress. Owing to computation limitations, we consider randomly chosen matrices with a small range of matrix size. Within this range and with the non-optimal ansatz, the result suggests that the circuit depth scales linearly with the condition number and super-linearly with the number of qubits. Further improvements of the performance of the algorithm are expected for sparse matrices together with more sophisticated ansätze.

8 Conclusion

Since the concept has been proposed by Feynman in 1982, quantum computing has experienced rapid development over the last several decades. While it has the potential to bring revolutionary change to science and technology, a well-functioning quantum machine is difficult to make.

The foremost factor is that the unavoidable noise occurring in interactions with the environment and associated with every attempt to control a physical system, prohibits the quantum hardware from behaving as it ideally should. Quantum error correction offers one potential solution, suggesting that the encoded quantum information could preserve a longer life time with the benefit of (repeated) error correction operations.

The most promising error correction code for large-scale quantum computing is currently the surface code that maintains excellent practicality and a high threshold. While the standard error model has theoretical significance, it is also vital to consider more realistic models for real devices. We addressed this problem in Chapter 3 by considering biased noise that is common for many hardware systems. We introduced a concatenated code with the surface code as the upper level and a two-qubit error detection code beneath it. A new surface code decoder is proposed to fully utilise the extra information from the error-detecting cycle as to identify where the errors are. High thresholds are found with biased noise models and additionally if a

modular structure of the device is considered where the error-detecting process can be performed within each module.

Promising as the code may be, realisation of such a large-scale code is hard to achieve in the near future. Alternatively, small error correction codes can be the choice to demonstrate beneficial error correction in near-term devices. In Chapter 4 we introduced several popular small codes, and we also considered practical problems in implementation of the small codes. A crucial one could be, how to compare an encoded logical qubit operated upon with dozens of noisy gates, with an untouched raw physical qubit? To address this question, we proposed a measure called 'integrity' to benchmark the performance of quantum memories. We introduced the Alice-Igor-Bob framework as a straightforward method for experimentalists to assess the performance of the hardware devices, and we showed that beneficial error correction can be realised through four milestones, each with increasing difficulty.

Under this framework, in Chapter 5 we considered a sophisticated error correction protocol implemented on an ion-trap quantum processor. To perform a full error correction cycle with the seven-qubit code, detailed sequences of crystal-reconfiguration operations were derived. Realistic error models based on the underlying physical system were considered, associated with two groups of parameters that represent the current and anticipated hardware performance respectively. Incorporating the method introduced in Chapter 4, we showed that while beneficial error correction can not be demonstrated with the current technology, the advantage of error correction can be realised in the anticipated near future.

Designing the protocol in Chapter 5 was assisted with circuit compilations to minimise the resources required. As the standard error correction circuits do not consider the hardware requirements for a particular system, it is necessary to have an automatic quantum compiler, as we presented in Chapter 6. Constructing the Hamiltonian such that the ground state is the ideal logical state to be prepared, we applied the variational quantum eigensolver, a hybrid algorithm to search for the ground state. A couple of example circuits discovered are shown, with specific

requirements satisfied. This quantum compilation technique is generally applicable for quantum hardware to optimise any such circuit in near-term experiments.

While current quantum computers are not protected by error correction codes, the community is interested in potential applications with those noisy devices in the NISQ regime. The variational quantum eigensolver can operate with shallow circuits and thus is widely used in quantum simulations. In Chapter 7 we presented one example of such applications. We showed that, by mapping the ground state as the ideal target state, one could solve linear algebra problems by searching the ground state energy. The algorithm shows a linear and super-linear dependence of the circuit depth on the condition number and number of qubits, making it potentially an algorithm applicable for NISQ devices.

This thesis covers a wide range of topics in quantum information processing from large-scale quantum computing to applications of near-term algorithms implemented with devices that are, or will soon be, available. Full-fledged large-scale quantum computers might take years to build, in the mean time, continuous experimental progress offers space for improvement of current theories and opens possibilities for new algorithms. The author hopes the materials of this thesis could contribute to the quantum computing community in development of today's machine towards long-term universal quantum computations.

Appendices

A

Derivation of equations for imaginary time evolution

This appendix gives derivations of the equations for imaginary time evolution in Sec. 2.3.2. The materials are based on Ref. [73].

A.1 Pure state

The McLachlan's variational principle applied to the variational algorithm for real time evolution can be written,

$$\delta \left\| (d/dt + iH)|\phi(\vec{\theta}(t))\rangle \right\| = 0, \tag{A.1}$$

where $|\phi(\vec{\theta}(t))\rangle$ is the parameterised trial state. As $\delta \left\| (d/dt + iH)|\phi(\vec{\theta}(t))\rangle \right\| = 0$ and $\delta \left\| (d/dt + iH)|\phi(\vec{\theta}(t))\rangle \right\|^2 = 0$ are equivalent, we focus on $\delta \left\| (d/dt + iH)|\phi(\vec{\theta}(t))\rangle \right\|^2 = 0$. The expression can be expanded as

$$\begin{aligned} &\|(d/dt + iH)|\phi(\vec{\theta}(t))\rangle\|^2 = ((d/dt + iH)|\phi(\vec{\theta}(t))\rangle)^\dagger (d/dt + iH)|\phi(\vec{\theta}(t))\rangle \\ =&\sum_{i,j} \frac{\partial \langle\phi(\vec{\theta}(t))|}{\partial\theta_i} \frac{\partial |\phi(\vec{\theta}(t))\rangle}{\partial\theta_j} \dot{\theta}_i^* \dot{\theta}_j + i \sum_i \frac{\partial \langle\phi(\vec{\theta}(t))|}{\partial\theta_i} H |\phi(\vec{\theta}(t))\rangle \dot{\theta}_i^* \\ &- i \sum_i \langle\phi(\vec{\theta}(t))| H \frac{\partial |\phi(\vec{\theta}(t))\rangle}{\partial\theta_i} \dot{\theta}_i + \langle\phi(\vec{\theta}(t))\left| H^2 \right|\phi(\vec{\theta}(t))\rangle \end{aligned} \tag{A.2}$$

We assume the parameters are real, so the expression above leads to

$$\begin{aligned}\delta\|(d/dt+iH)|\phi(\vec{\theta}(t))\rangle\|^2 =& \sum_j \left(\frac{\partial\langle\phi(\vec{\theta}(t))|}{\partial\theta_i}\frac{\partial|\phi(\vec{\theta}(t))\rangle}{\partial\theta_j}+\frac{\partial\langle\phi(\vec{\theta}(t))|}{\partial\theta_j}\frac{\partial|\phi(\vec{\theta}(t))\rangle}{\partial\theta_i}\right)\dot{\theta}_j\delta\dot{\theta}_i \\ &+ i\left(\frac{\partial\langle\phi(\vec{\theta}(t))|}{\partial\theta_i}H|\phi(\vec{\theta}(t))\rangle-\langle\phi(\vec{\theta}(t))|H\frac{\partial|\phi(\vec{\theta}(t))\rangle}{\partial\theta_i}\right)\delta\dot{\theta}_i.\end{aligned} \tag{A.3}$$

Making the equation equal to zero, we have

$$\begin{aligned}&\sum_j \left(\frac{\partial\langle\phi(\vec{\theta}(t))|}{\partial\theta_i}\frac{\partial|\phi(\vec{\theta}(t))\rangle}{\partial\theta_j}+\frac{\partial\langle\phi(\vec{\theta}(t))|}{\partial\theta_j}\frac{\partial|\phi(\vec{\theta}(t))\rangle}{\partial\theta_i}\right)\dot{\theta}_j \\ =&-i\left(\frac{\partial\langle\phi(\vec{\theta}(t))|}{\partial\theta_i}H|\phi(\vec{\theta}(t))\rangle-\langle\phi(\vec{\theta}(t))|H\frac{\partial|\phi(\vec{\theta}(t))\rangle}{\partial\theta_i}\right),\end{aligned} \tag{A.4}$$

which corresponds to the equation $\sum_j M_{i,j}\dot{\theta}_j = V_i$ in the main text.

Similarly, the McLachlan's variational principle for imaginary evolution generates $\delta\|(d/d\tau+H-E_\tau)|\phi(\vec{\theta}(\tau))\rangle\| = 0$. Expanding the expression, we have

$$\begin{aligned}&\|(d/d\tau+H-E_\tau)|\phi(\vec{\theta}(\tau))\rangle\|^2 \\ =&\left((d/d\tau+H-E_\tau)\,|\phi(\vec{\theta}(\tau))\rangle\right)^\dagger(d/d\tau+H-E_\tau)\,|\phi(\vec{\theta}(\tau))\rangle \\ =&\sum_{i,j}\frac{\partial\langle\phi(\vec{\theta}(\tau))|}{\partial\theta_i}\frac{\partial|\phi(\vec{\theta}(\tau))\rangle}{\partial\theta_j}\dot{\theta}_i^*\dot{\theta}_j+\sum_i\frac{\partial\langle\phi(\vec{\theta}(\tau))|}{\partial\theta_i}(H-E_\tau)\,|\phi(\vec{\theta}(\tau))\rangle\dot{\theta}_i^*+ \\ &\sum_i\langle\phi(\vec{\theta}(\tau))|(H-E_\tau)\frac{\partial|\phi(\vec{\theta}(\tau))\rangle}{\partial\theta_i}\dot{\theta}_i+\langle\phi(\vec{\theta}(\tau))|(H-E_\tau)^2|\phi(\vec{\theta}(\tau))\rangle.\end{aligned} \tag{A.5}$$

If the parameters are real, we then have

$$\begin{aligned}&\delta\|(d/d\tau+H-E_\tau)|\phi(\vec{\theta}(\tau)))\rangle\|^2 \\ =&\sum_j\left(\frac{\partial\langle\phi(\vec{\theta}(\tau))|}{\partial\theta_i}\frac{\partial|\phi(\vec{\theta}(\tau))\rangle}{\partial\theta_j}+\frac{\partial\langle\phi(\vec{\theta}(\tau))|}{\partial\theta_j}\frac{\partial|\phi(\vec{\theta}(\tau))\rangle}{\partial\theta_i}\right)\dot{\theta}_j\delta\dot{\theta}_i \\ &+\left(\frac{\partial\langle\phi(\vec{\theta}(\tau))|}{\partial\theta_i}(H-E_\tau)\,|\phi(\vec{\theta}(\tau))\rangle+\langle\phi(\vec{\theta}(\tau))|\,(H-E_\tau)\frac{\partial|\phi(\vec{\theta}(\tau))\rangle}{\partial\theta_i}\right)\delta\dot{\theta}_i\end{aligned}, \tag{A.6}$$

and by making the equation zero, we obtain

$$\begin{aligned}
&\sum_j \left(\frac{\partial \langle \phi(\vec{\theta}(\tau))|}{\partial \theta_i} \frac{\partial |\phi(\vec{\theta}(\tau))\rangle}{\partial \theta_j} + \frac{\partial \langle \phi(\vec{\theta}(\tau))|}{\partial \theta_j} \frac{\partial |\phi(\vec{\theta}(\tau))\rangle}{\partial \theta_i} \right) \dot{\theta}_j \\
&= - \left(\frac{\partial \langle \phi(\vec{\theta}(\tau))|}{\partial \theta_i} (H - E_\tau) |\phi(\vec{\theta}(\tau))\rangle + \langle \phi(\vec{\theta}(\tau))| (H - E_\tau) \frac{\partial |\phi(\vec{\theta}(\tau))\rangle}{\partial \theta_i} \right) \\
&= - \left(\frac{\partial \langle \phi(\vec{\theta}(\tau))|}{\partial \theta_i} H |\phi(\vec{\theta}(\tau))\rangle + \langle \phi(\vec{\theta}(\tau))| H \frac{\partial |\phi(\vec{\theta}(\tau))\rangle}{\partial \theta_i} \right) + E_\tau \frac{\partial}{\partial \theta_i} (\langle \phi(\vec{\theta}(\tau)|\phi(\vec{\theta}(\tau))\rangle)
\end{aligned} \tag{A.7}$$

As $\langle \phi(\vec{\theta}(\tau)|\phi(\vec{\theta}(\tau)\rangle = 1$, $\frac{\partial}{\partial \theta_i}(\langle \phi(\vec{\theta}(\tau)|\phi(\vec{\theta}(\tau))\rangle) = 0$. The remaining part corresponds to the equation $\sum_j A_{i,j}\dot{\theta}_i = -B_i$ in the main text.

A.2 Mixed state

With the mixed state, as described in the main text, the McLachlan's variational principle with the imaginary time leads to $\delta \| d\rho/d\tau + (\{H, \rho\} - 2\langle H \rangle \rho) \| = 0$. With the same method as applied above, we expand it as

$$\begin{aligned}
&\| d\rho/d\tau + (\{H, \rho\} - 2\langle H \rangle \rho) \|^2 \\
&= \mathrm{Tr} \left[(d\rho/d\tau + (\{H, \rho\} - 2\langle H \rangle \rho))^\dagger (d\rho/d\tau + (\{H, \rho\} - 2\langle H \rangle \rho)) \right] \\
&= \mathrm{Tr} \left[\left(\sum_i \frac{\partial \rho}{\partial \theta_i} \dot{\theta}_i \right)^\dagger \sum_j \frac{\partial \rho}{\partial \theta_j} \dot{\theta}_j + \left(\sum_i \frac{\partial \rho}{\partial \theta_i} \dot{\theta}_i \right)^\dagger (\{H, \rho\} - 2\langle H \rangle \rho) \right] \\
&+ \mathrm{Tr} \left[(\{H, \rho\} - 2\langle H \rangle \rho) \sum_j \frac{\partial \rho}{\partial \theta_j} \dot{\theta}_j + (\{H, \rho\} - 2\langle H \rangle \rho)^2 \right].
\end{aligned} \tag{A.8}$$

Assuming the parameters are real, then we have

$$\begin{aligned}
&\delta \| d\rho/d\tau + (\{H, \rho\} - 2\langle H \rangle \rho) \|^2 \\
&= \sum_i \left(\mathrm{Tr} \left[\frac{\partial \rho(\vec{\theta})}{\partial \theta_i} \sum_j \frac{\partial \rho(\vec{\theta})}{\partial \theta_j} \dot{\theta}_j + \frac{\partial \rho(\vec{\theta})}{\partial \theta_i} (\{H, \rho(\vec{\theta}(\tau))\} - 2\langle H \rangle \rho(\vec{\theta})) \right] \delta \dot{\theta}_i \right. \\
&\left. + \mathrm{Tr} \left[\left(\sum_j \frac{\partial \rho(\vec{\theta})}{\partial \theta_j} \dot{\theta}_j \right) \frac{\partial \rho(\vec{\theta})}{\partial \theta_i} + (\{H, \rho(\vec{\theta}(\tau))\} - 2\langle H \rangle \rho(\vec{\theta})) \frac{\partial \rho(\vec{\theta})}{\partial \theta_i} \right] \delta \dot{\theta}_i \right).
\end{aligned} \tag{A.9}$$

By letting the equation to be zero, we obtain

$$\begin{aligned}\sum_j \mathrm{Tr}\left[\frac{\partial\rho(\vec{\theta}(\tau))}{\partial\theta_i}\frac{\partial\rho(\vec{\theta}(\tau))}{\partial\theta_j}\right]\dot{\theta}_j &= -\mathrm{Tr}\left[\frac{\partial\rho(\vec{\theta}(\tau))}{\partial\theta_i}(\{H,\rho(\vec{\theta}(\tau))\}-2\langle H\rangle\rho(\vec{\theta}(\tau)))\right] \\ &= -\mathrm{Tr}\left[\frac{\partial\rho(\vec{\theta}(\tau))}{\partial\theta_i}\{H,\rho(\vec{\theta}(\tau))\}\right]\end{aligned}, \tag{A.10}$$

which is equivalent to the equation $\sum_i C_{j,i}\dot{\theta}_i = -D_j$ in the main text.

B
Proof of Eq. (6.7)

In this Appendix we present proof for Eq. (6.7) in Chapter 6. Suppose we have a Hamiltonian H with ground and first excited energy denoted by E_0 and E_1, respectively. Given a quantum state ρ with averaged energy $E = \mathrm{tr}[\rho H]$ satisfying $E \in [E_0, E_1]$, we want to prove

$$F = \langle\psi_0|\rho|\psi_0\rangle \geq 1 - (E - E_0)/c, \tag{B.1}$$

where we denote $c = E_1 - E_0$ and $|\psi_0\rangle$ being the ground state of H.

Denote the eigenstates of H by $|\psi_i\rangle$ with corresponding eigenvalues E_i satisfying $E_i \leq E_j$ when $i \leq j$. Then we have

$$H = \sum_i E_i \left|\psi_i\right\rangle \left\langle\psi_i\right|,$$

and

$$\begin{aligned} E =& \mathrm{tr}[\rho H], \\ =& \sum_i E_i \left\langle\psi_i\right| \rho \left|\psi_i\right\rangle, \\ =& E_0 F + \sum_{i\geq 1} E_i \left\langle\psi_i\right| \rho \left|\psi_i\right\rangle, \\ \geq& E_0 F + \sum_{i\geq 1} E_1 \left\langle\psi_i\right| \rho \left|\psi_i\right\rangle, \\ =& E_0 F + E_1(1 - F), \\ =& (E_0 + c) - cF. \end{aligned} \tag{B.2}$$

In the third line, we make use of the definition of F in Eq. (B.1); In the fourth line, we make use of the ordering of the eigenvalues; In the fifth line, we make use of $\sum_{i\geq 1} \langle\psi_i| \rho |\psi_i\rangle = 1 - \langle\psi_0| \rho |\psi_0\rangle = 1 - F$. Solving the equation, we thus get the lower bound of the fidelity F based on the energy E as in Eq. (B.1).

C

An example with a more compact ansätz

This Appendix provides the supplementary material for Chapter 7. For numerical simulations, we apply the hardware-efficient ansätz for practical soundness, however, this is not the optimal ansätz for most cases. Here we take one case as an example, where the solution is found difficult to reach in our numerical simulations.

The matrix in this case is a complex matrix with size of 64*64, and condition number $c = 60$. $|v_0\rangle$ is a complex vector, and the target state is $|v_{M^{-1}}\rangle = \frac{M^{-1}|v_0\rangle}{\|M^{-1}|v_0\rangle\|}$. It is found in the numerical simulation that the target state can not be represented by a circuit with depth 7 but can be by a circuit with depth 8. On the other hand, applying quantum compiling, it is found that the target state can be realised with a circuit that requires only half the number of parameters. Figure C.1 is one example of the circuits found to prepare a state with fidelity higher than 99.9% compared with the target state. A limited topology with only nearest-neighbour interactions is considered here as it is more hardware efficient. The detailed description and the parameters can be found at https://questlink.qtechtheory.org/. This compact circuit was found by Simon Benjamin using the depth 8 initial circuit provided by the author. It is included here for completeness.

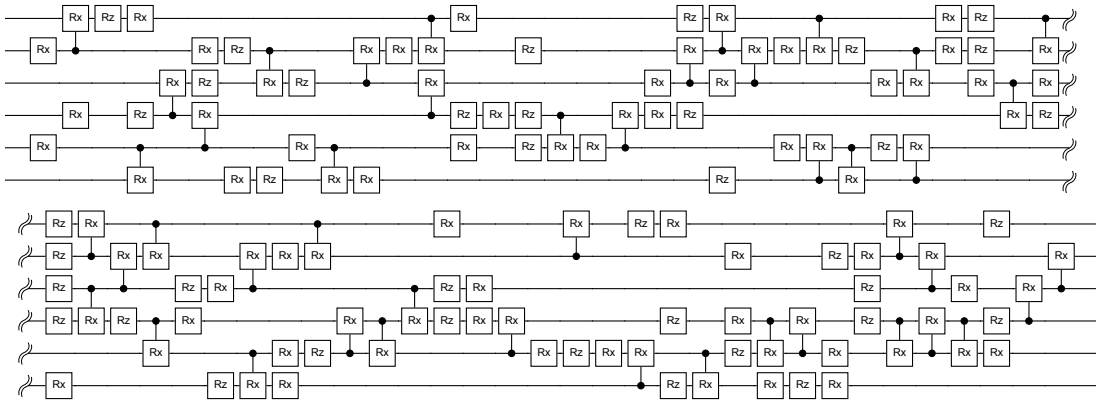


Figure C.1: A more compact circuit found with quantum compilation to realise the target state. Each single- and multi-qubit gate is controlled with one parameter. In total 126 parameters are included in the circuit, which realises the state with a fidelity higher than 99.9%.

References

[1] Richard P Feynman. "Simulating physics with computers". In: *International journal of theoretical physics* 21.6 (1982), pp. 467–488.

[2] David Deutsch. "Quantum theory, the Church-Turing principle and the universal quantum computer". In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 400. 1818. The Royal Society. 1985, pp. 97–117.

[3] David Deutsch and Richard Jozsa. "Rapid solution of problems by quantum computation". In: *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439.1907 (1992), pp. 553–558.

[4] Peter W Shor. "Algorithms for quantum computation: Discrete logarithms and factoring". In: *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*. Ieee. 1994, pp. 124–134.

[5] Lov K Grover. "A fast quantum mechanical algorithm for database search". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM. 1996, pp. 212–219.

[6] William K Wootters and Wojciech H Zurek. "A single quantum cannot be cloned". In: *Nature* 299.5886 (1982), pp. 802–803.

[7] Peter W Shor. "Scheme for reducing decoherence in quantum computer memory". In: *Physical review A* 52.4 (1995), R2493.

[8] A Robert Calderbank and Peter W Shor. "Good quantum error-correcting codes exist". In: *Physical Review A* 54.2 (1996), p. 1098.

[9] Andrew Steane. "Multiple-particle interference and quantum error correction". In: *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 452.1954 (1996), pp. 2551–2577.

[10] Lev Vaidman, Lior Goldenberg, and Stephen Wiesner. "Error prevention scheme with four particles". In: *Physical Review A* 54.3 (1996), R1745.

[11] Markus Grassl, Th Beth, and Thomas Pellizzari. "Codes for the quantum erasure channel". In: *Physical Review A* 56.1 (1997), p. 33.

[12] Emanuel Knill. "Quantum computing with realistically noisy devices". In: *Nature* 434.7029 (2005), p. 39.

[13] Daniel Gottesman. "Quantum fault tolerance in small experiments". In: *arXiv preprint arXiv:1610.03507* (2016).

[14] Norbert M Linke et al. "Fault-tolerant quantum error detection". In: *Science advances* 3.10 (2017), e1701074.

[15] Daniel Gottesman. "Stabilizer codes and quantum error correction". In: *arXiv preprint quant-ph/9705052* (1997).

[16] John Von Neumann. “Probabilistic logics and the synthesis of reliable organisms from unreliable components”. In: *Automata studies* 34 (1956), pp. 43–98.

[17] Peter Gács. “Reliable computation with cellular automata”. In: *Proceedings of the fifteenth annual ACM symposium on Theory of computing*. ACM. 1983, pp. 32–41.

[18] Peter W Shor. “Fault-tolerant quantum computation”. In: *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*. IEEE. 1996, pp. 56–65.

[19] John Preskill. “Fault-tolerant quantum computation”. In: *Introduction to quantum computation and information* 213 (1998).

[20] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. 2002.

[21] Emanuel Knill, Raymond Laflamme, and W Zurek. “Threshold accuracy for quantum computation”. In: *arXiv preprint quant-ph/9610011* (1996).

[22] Emanuel Knill, Raymond Laflamme, and Wojciech H Zurek. “Resilient quantum computation”. In: *Science* 279.5349 (1998), pp. 342–345.

[23] Dorit Aharonov and Michael Ben-Or. “Fault-tolerant quantum computation with constant error”. In: *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. ACM. 1997, pp. 176–188.

[24] Simon J Devitt, William J Munro, and Kae Nemoto. “Quantum error correction for beginners”. In: *Reports on Progress in Physics* 76.7 (2013), p. 076001.

[25] Dorit Aharonov and Michael Ben-Or. “Fault-tolerant quantum computation with constant error rate”. In: *SIAM Journal on Computing* 38.4 (2008), pp. 1207–1282.

[26] Adam Paetznick and Ben W Reichardt. “Fault-tolerant ancilla preparation and noise threshold lower boudds for the 23-qubit Golay code”. In: *Quantum Information & Computation* 12.11-12 (2012), pp. 1034–1080.

[27] Tzvetan Metodiev et al. “Preliminary results on simulating a scalable fault tolerant ion-trap system for quantum computation”. In: *3rd workshop on Non-Silicon Computing*. 2004.

[28] Robert Raussendorf, Jim Harrington, and Kovid Goyal. “Topological fault-tolerance in cluster state quantum computation”. In: *New Journal of Physics* 9.6 (2007), p. 199.

[29] Austin G Fowler, Ashley M Stephens, and Peter Groszkowski. “High-threshold universal quantum computation on the surface code”. In: *Physical Review A* 80.5 (2009), p. 052312.

[30] David S Wang, Austin G Fowler, and Lloyd CL Hollenberg. “Surface code quantum computing with error rates over 1%”. In: *Physical Review A* 83.2 (2011), p. 020302.

[31] A Yu Kitaev. “Quantum computations: algorithms and error correction”. In: *Russian Mathematical Surveys* 52.6 (1997), pp. 1191–1249.

[32] Eric Dennis et al. “Topological quantum memory”. In: *Journal of Mathematical Physics* 43.9 (2002), pp. 4452–4505.

[33] Markus Reiher et al. “Elucidating reaction mechanisms on quantum computers”. In: *Proceedings of the National Academy of Sciences* 114.29 (2017), pp. 7555–7560.

[34] Ryan Babbush et al. “Encoding electronic spectra in quantum circuits with linear T complexity”. In: *Physical Review X* 8.4 (2018), p. 041015.

[35] C Neill et al. “A blueprint for demonstrating quantum supremacy with superconducting qubits”. In: *Science* 360.6385 (2018), pp. 195–199.

[36] Alberto Peruzzo et al. “A variational eigenvalue solver on a photonic quantum processor”. In: *Nature communications* 5 (2014).

[37] Jarrod R McClean et al. “The theory of variational hybrid quantum-classical algorithms”. In: *New Journal of Physics* 18.2 (2016), p. 023023.

[38] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. “A quantum approximate optimization algorithm”. In: *arXiv preprint arXiv:1411.4028* (2014).

[39] Aram W Harrow and Ashley Montanaro. “Quantum computational supremacy”. In: *Nature* 549.7671 (2017), p. 203.

[40] Sergio Boixo et al. “Characterizing quantum supremacy in near-term devices”. In: *Nature Physics* 14.6 (2018), p. 595.

[41] Ya Wang et al. “Quantum simulation of helium hydride cation in a solid-state spin register”. In: *ACS nano* 9.8 (2015), pp. 7769–7774.

[42] Peter JJ O?Malley et al. “Scalable quantum simulation of molecular energies”. In: *Physical Review X* 6.3 (2016), p. 031007.

[43] James I Colless et al. “Computation of molecular spectra on a quantum processor with an error-resilient algorithm”. In: *Physical Review X* 8.1 (2018), p. 011021.

[44] Raffaele Santagati et al. “Witnessing eigenstates for quantum simulation of Hamiltonian spectra”. In: *Science advances* 4.1 (2018), eaap9646.

[45] Abhinav Kandala et al. “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets”. In: *Nature* 549.7671 (2017), p. 242.

[46] Cornelius Hempel et al. “Quantum chemistry calculations on a trapped-ion quantum simulator”. In: *Physical Review X* 8.3 (2018), p. 031022.

[47] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik. “Quantum autoencoders for efficient compression of quantum data”. In: *Quantum Science and Technology* 2.4 (2017), p. 045001.

[48] Alejandro Perdomo-Ortiz et al. “Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers”. In: *Quantum Science and Technology* 3.3 (2018), p. 030502.

[49] Marcello Benedetti et al. “A generative modeling approach for benchmarking and training shallow quantum circuits”. In: *npj Quantum Information* 5.1 (2019), p. 45.

[50] Lucas Lamata et al. “Quantum autoencoders via quantum adders with genetic algorithms”. In: *Quantum Science and Technology* 4.1 (2018), p. 014007.

[51] Amir Khoshaman et al. “Quantum variational autoencoder”. In: *Quantum Science and Technology* 4.1 (2018), p. 014001.

[52] Seth Lloyd and Christian Weedbrook. “Quantum generative adversarial learning”. In: *Physical review letters* 121.4 (2018), p. 040502.

[53] Kosuke Mitarai et al. “Quantum circuit learning”. In: *Physical Review A* 98.3 (2018), p. 032309.

[54] Lukasz Cincio et al. “Learning the quantum algorithm for state overlap”. In: *New Journal of Physics* 20.11 (2018), p. 113022.

[55] Tyson Jones and Simon C Benjamin. “Quantum compilation and circuit optimisation via energy dissipation”. In: *arXiv preprint arXiv:1811.03147* (2018).

[56] Kunal Sharma et al. “Noise Resilience of Variational Quantum Compiling”. In: *arXiv preprint arXiv:1908.04416* (2019).

[57] Michael Lubasch et al. “Variational Quantum Algorithms for Nonlinear Problems”. In: *arXiv preprint arXiv:1907.09032* (2019).

[58] Yonghae Lee, Jaewoo Joo, and Soojoon Lee. “Hybrid quantum linear equation algorithm and its experimental test on IBM Quantum Experience”. In: *Scientific reports* 9.1 (2019), p. 4778.

[59] Jarrod R McClean et al. “Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states”. In: *Physical Review A* 95.4 (2017), p. 042308.

[60] Ying Li and Simon C Benjamin. “Efficient variational quantum simulator incorporating active error minimization”. In: *Physical Review X* 7.2 (2017), p. 021050.

[61] Kristan Temme, Sergey Bravyi, and Jay M Gambetta. “Error mitigation for short-depth quantum circuits”. In: *Physical review letters* 119.18 (2017), p. 180509.

[62] Suguru Endo, Simon C Benjamin, and Ying Li. “Practical quantum error mitigation for near-future applications”. In: *Physical Review X* 8.3 (2018), p. 031027.

[63] Matthew Otten and Stephen K Gray. “Recovering noise-free quantum observables”. In: *Physical Review A* 99.1 (2019), p. 012338.

[64] Xavi Bonet-Monroig et al. “Low-cost error mitigation by symmetry verification”. In: *Physical Review A* 98.6 (2018), p. 062339.

[65] Sam McArdle et al. “Variational ansatz-based quantum simulation of imaginary time evolution”. In: *npj Quantum Information* 5.1 (2019), pp. 1–6.

[66] Abhinav Kandala et al. “Extending the computational reach of a noisy superconducting quantum processor”. In: *arXiv preprint arXiv:1805.04492* (2018).

[67] Jarrod R McClean et al. “Barren plateaus in quantum neural network training landscapes”. In: *Nature communications* 9.1 (2018), p. 4812.

[68] Rodney J Bartlett, Stanislaw A Kucharski, and Jozef Noga. “Alternative coupled-cluster ansätze II. The unitary coupled-cluster method”. In: *Chemical physics letters* 155.1 (1989), pp. 133–140.

[69] Mark R Hoffmann and Jack Simons. “A unitary multiconfigurational coupled-cluster method: Theory and applications”. In: *The Journal of chemical physics* 88.2 (1988), pp. 993–1002.

[70] M-H Yung et al. “From transistor to trapped-ion computers for quantum chemistry”. In: *Scientific reports* 4 (2014), p. 3589.

[71] Jonathan Romero et al. “Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz”. In: *Quantum Science and Technology* 4.1 (2018), p. 014008.

[72] C Kokail et al. “Self-verifying variational quantum simulation of lattice models”. In: *Nature* 569.7756 (2019), p. 355.

[73] Xiao Yuan et al. “Theory of variational quantum simulation”. In: *Quantum* 3 (2019), p. 191.

[74] AD McLachlan. “A variational solution of the time-dependent Schrodinger equation”. In: *Molecular Physics* 8.1 (1964), pp. 39–44.

[75] Artur K Ekert et al. “Direct estimations of linear and nonlinear functionals of a quantum state”. In: *Physical review letters* 88.21 (2002), p. 217901.

[76] A Yu Kitaev. “Fault-tolerant quantum computation by anyons”. In: *Annals of Physics* 303.1 (2003), pp. 2–30.

[77] Austin G Fowler et al. “Surface codes: Towards practical large-scale quantum computation”. In: *Physical Review A* 86.3 (2012), p. 032324.

[78] Panos Aliferis et al. “Fault-tolerant computing with biased-noise superconducting qubits: a case study”. In: *New Journal of Physics* 11.1 (2009), p. 013061.

[79] Ben P Lanyon et al. “Universal digital quantum simulation with trapped ions”. In: *Science* 334.6052 (2011), pp. 57–61.

[80] Kamyar Saeedi et al. “Room-temperature quantum bit storage exceeding 39 minutes using ionized donors in silicon-28”. In: *Science* 342.6160 (2013), pp. 830–833.

[81] David K Tuckett, Stephen D Bartlett, and Steven T Flammia. “Ultrahigh error threshold for surface codes with biased noise”. In: *Physical review letters* 120.5 (2018), p. 050505.

[82] Muyuan Li et al. “2D Compass Codes”. In: *Physical Review X* 9.2 (2019), p. 021041.

[83] David K Tuckett et al. “Fault-tolerant thresholds for the surface code in excess of 5% under biased noise”. In: *arXiv preprint arXiv:1907.02554* (2019).

[84] Ben Criger and Barbara Terhal. “Noise thresholds for the [4, 2, 2]-concatenated toric code”. In: *Quantum Information & Computation* 16.15-16 (2016), pp. 1261–1281.

[85] Ashley M Stephens, William J Munro, and Kae Nemoto. “High-threshold topological quantum error correction against biased noise”. In: *Physical Review A* 88.6 (2013), p. 060301.

[86] Ferdinand Schmidt-Kaler et al. “Realization of the Cirac–Zoller controlled-NOT quantum gate”. In: *Nature* 422.6930 (2003), p. 408.

[87] Johannes Zeiher et al. “Microscopic characterization of scalable coherent Rydberg superatoms”. In: *Physical Review X* 5.3 (2015), p. 031015.

[88] Jun Yoneda et al. “A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%”. In: *Nature nanotechnology* 13.2 (2018), p. 102.

[89] Xiaosi Xu et al. “High-Threshold Code for Modular Hardware With Asymmetric Noise”. In: *Physical Review Applied* 12.6 (2019), p. 064006.

[90] DS Wang et al. “Threshold error rates for the toric and planar codes”. In: *Quantum Information & Computation* 10.5 (2010), pp. 456–469.

[91] Jack Edmonds. “Maximum matching and a polyhedron with 0, 1-vertices”. In: *Journal of research of the National Bureau of Standards B* 69.125-130 (1965), pp. 55–56.

[92] William Cook and Andre Rohe. “Computing minimum-weight perfect matchings”. In: *INFORMS journal on computing* 11.2 (1999), pp. 138–148.

[93] Austin G Fowler. “Minimum weight perfect matching of fault-tolerant topological quantum error correction in average O (1) parallel time”. In: *Quantum Information & Computation* 15.1-2 (2015), pp. 145–158.

[94] Sergey Bravyi, Martin Suchara, and Alexander Vargo. “Efficient algorithms for maximum likelihood decoding in the surface code”. In: *Physical Review A* 90.3 (2014), p. 032326.

[95] Andrew S Darmawan and David Poulin. “Tensor-network simulations of the surface code under realistic noise”. In: *Physical review letters* 119.4 (2017), p. 040502.

[96] Austin G Fowler, Adam C Whiteside, and Lloyd CL Hollenberg. “Towards practical classical processing for the surface code”. In: *Physical review letters* 108.18 (2012), p. 180501.

[97] Robert Raussendorf and Jim Harrington. “Fault-tolerant quantum computation with high threshold in two dimensions”. In: *Physical review letters* 98.19 (2007), p. 190504.

[98] Sergey Bravyi and Jeongwan Haah. “Quantum self-correction in the 3d cubic code model”. In: *Physical review letters* 111.20 (2013), p. 200501.

[99] Guillaume Duclos-Cianci and David Poulin. “Fast decoders for topological quantum codes”. In: *Physical review letters* 104.5 (2010), p. 050504.

[100] Fern HE Watson, Hussain Anwar, and Dan E Browne. “Fast fault-tolerant decoder for qubit and qudit surface codes”. In: *Physical Review A* 92.3 (2015), p. 032309.

[101] Jack Edmonds. “Paths, trees, and flowers”. In: *Canadian Journal of mathematics* 17 (1965), pp. 449–467.

[102] MEJ Newman and RM Ziff. “Efficient Monte Carlo algorithm and high-precision results for percolation”. In: *Physical Review Letters* 85.19 (2000), p. 4104.

[103] Jesper Lykke Jacobsen. “High-precision percolation thresholds and Potts-model critical manifolds from graph polynomials”. In: *Journal of Physics A: Mathematical and Theoretical* 47.13 (2014), p. 135001.

[104] Edsger W Dijkstra. “A note on two problems in connexion with graphs”. In: *Numerische mathematik* 1.1 (1959), pp. 269–271.

[105] Yu Tomita and Krysta M Svore. “Low-distance surface codes under realistic quantum noise”. In: *Physical Review A* 90.6 (2014), p. 062320.

[106] Naomi H Nickerson, Joseph F Fitzsimons, and Simon C Benjamin. "Freely scalable quantum technologies using cells of 5-to-50 qubits with very lossy and noisy photonic links". In: *Physical Review X* 4.4 (2014), p. 041041.

[107] Ying Li and Simon C Benjamin. "Hierarchical surface code for network quantum computing with modules of arbitrary size". In: *Physical Review A* 94.4 (2016), p. 042303.

[108] Philipp Schindler et al. "Experimental repetitive quantum error correction". In: *Science* 332.6033 (2011), pp. 1059–1061.

[109] Daniel Nigg et al. "Quantum computations on a topologically encoded qubit". In: *Science* 345.6194 (2014), pp. 302–305.

[110] Matthew D Reed et al. "Realization of three-qubit quantum error correction with superconducting circuits". In: *Nature* 482.7385 (2012), p. 382.

[111] Julian Kelly et al. "State preservation by repetitive error detection in a superconducting quantum circuit". In: *Nature* 519.7541 (2015), p. 66.

[112] Antonio D Córcoles et al. "Demonstration of a quantum error detection code using a square lattice of four superconducting qubits". In: *Nature communications* 6 (2015), p. 6979.

[113] Diego Riste et al. "Detecting bit-flip errors in a logical qubit using stabilizer measurements". In: *Nature communications* 6 (2015), p. 6983.

[114] Julia Cramer et al. "Repeated quantum error correction on a continuously encoded qubit by real-time feedback". In: *Nature communications* 7 (2016), p. 11526.

[115] Raymond Laflamme et al. "Perfect quantum error correcting code". In: *Physical Review Letters* 77.1 (1996), p. 198.

[116] Andrew M Steane. "Error correcting codes in quantum theory". In: *Physical Review Letters* 77.5 (1996), p. 793.

[117] John P Gaebler et al. "High-fidelity universal gate set for be 9+ ion qubits". In: *Physical review letters* 117.6 (2016), p. 060505.

[118] CJ Ballance et al. "High-fidelity quantum logic gates using trapped-ion hyperfine qubits". In: *Physical review letters* 117.6 (2016), p. 060504.

[119] TP Harty et al. "High-fidelity preparation, gates, memory, and readout of a trapped-ion quantum bit". In: *Physical review letters* 113.22 (2014), p. 220501.

[120] Ye Wang et al. "Single-qubit quantum memory exceeding ten-minute coherence time". In: *Nature Photonics* 11.10 (2017), p. 646.

[121] Heinz-Peter Breuer, Elsi-Mari Laine, and Jyrki Piilo. "Measure for the degree of non-Markovian behavior of quantum processes in open systems". In: *Physical review letters* 103.21 (2009), p. 210401.

[122] Benjamin Aaronson et al. "Hierarchy and dynamics of trace distance correlations". In: *New Journal of Physics* 15.9 (2013), p. 093022.

[123] Jacob R West et al. "High fidelity quantum gates via dynamical decoupling". In: *Physical review letters* 105.23 (2010), p. 230503.

[124] K. M. Svore et al. “A flow-map model for analyzing pseudothresholds in fault-tolerant quantum computing”. In: *Quant. Inf. Comp.* 6 (2006), p. 193.

[125] Andrew W. Cross, David P. DiVincenzo, and Barbara M. Terhal. “Quantum error correction with only two extra qubits”. In: *Quant. Inf. Comp.* 9 (2009), p. 0541.

[126] Xiaosi Xu et al. “An integrity measure to benchmark quantum error correcting memories”. In: *New Journal of Physics* 20.2 (2018), p. 023009.

[127] Ming Gong et al. “Experimental verification of five-qubit quantum error correction with superconducting qubits”. In: *arXiv preprint arXiv:1907.04507* (2019).

[128] David P DiVincenzo and Peter W Shor. “Fault-tolerant error correction with efficient quantum codes”. In: *Physical review letters* 77.15 (1996), p. 3260.

[129] David P DiVincenzo and Panos Aliferis. “Effective fault-tolerant quantum computation with slow measurements”. In: *Physical review letters* 98.2 (2007), p. 020501.

[130] Rui Chao and Ben W Reichardt. “Quantum error correction with only two extra qubits”. In: *Physical review letters* 121.5 (2018), p. 050502.

[131] Adam Paetznick and Ben W Reichardt. “Fault-tolerant ancilla preparation and noise threshold lower bounds for the 23-qubit Golay code”. In: *arXiv preprint arXiv:1106.2190* (2011).

[132] Hayato Goto. “Minimizing resource overheads for fault-tolerant preparation of encoded states of the steane code”. In: *Scientific reports* 6 (2016), p. 19578.

[133] James R Wootton et al. “Proposal for a minimal surface code experiment”. In: *Physical Review A* 96.3 (2017), p. 032338.

[134] John Chiaverini et al. “Realization of quantum error correction”. In: *Nature* 432.7017 (2004), p. 602.

[135] Tim Hugo Taminiau et al. “Universal control and error correction in multi-qubit spin registers in diamond”. In: *Nature nanotechnology* 9.3 (2014), p. 171.

[136] TB Pittman, BC Jacobs, and JD Franson. “Demonstration of quantum error correction using linear optics”. In: *Physical Review A* 71.5 (2005), p. 052332.

[137] Chao-Yang Lu et al. “Experimental quantum coding against qubit loss error”. In: *Proceedings of the National Academy of Sciences* 105.32 (2008), pp. 11050–11054.

[138] Xing-Can Yao et al. “Experimental demonstration of topological error correction”. In: *Nature* 482.7386 (2012), p. 489.

[139] BA Bell et al. “Experimental demonstration of a graph state quantum error-correction code”. In: *Nature communications* 5 (2014), p. 3658.

[140] G Waldherr et al. “Quantum error correction in a solid-state hybrid spin register”. In: *Nature* 506.7487 (2014), p. 204.

[141] Alejandro Bermudez et al. “Assessing the progress of trapped-ion processors towards fault-tolerant quantum computation”. In: *Physical Review X* 7.4 (2017), p. 041061.

[142] A Bermudez et al. “Fault-tolerant protection of near-term trapped-ion topological qubits under realistic noise sources”. In: *Physical Review A* 100.6 (2019), p. 062307.

[143] Juan I Cirac and Peter Zoller. "Quantum computations with cold trapped ions". In: *Physical review letters* 74.20 (1995), p. 4091.

[144] Chris Monroe et al. "Demonstration of a fundamental quantum logic gate". In: *Physical review letters* 75.25 (1995), p. 4714.

[145] Jonathon P Home. "Quantum science and metrology with mixed-species ion chains". In: *Advances in Atomic, Molecular, and Optical Physics*. Vol. 62. Elsevier, 2013, pp. 231–277.

[146] Philipp Schindler et al. "A quantum information processor with trapped ions". In: *New Journal of Physics* 15.12 (2013), p. 123012.

[147] Klaus Mølmer and Anders Sørensen. "Multiparticle entanglement of hot trapped ions". In: *Physical Review Letters* 82.9 (1999), p. 1835.

[148] Anders Sørensen and Klaus Mølmer. "Entanglement and quantum computation with ions in thermal motion". In: *Physical Review A* 62.2 (2000), p. 022311.

[149] Thomas Ruster et al. "Experimental realization of fast ion separation in segmented Paul traps". In: *Physical Review A* 90.3 (2014), p. 033410.

[150] Ryan Bowler et al. "Coherent diabatic ion transport and separation in a multizone trap array". In: *Physical review letters* 109.8 (2012), p. 080502.

[151] Henning Kaufmann et al. "Fast ion swapping for quantum-information processing". In: *Physical Review A* 95.5 (2017), p. 052319.

[152] Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*. Vol. 1. Elsevier, 1992.

[153] Christian Roos. "Controlling the quantum state of trapped ions". In: (2000).

[154] Daniel T Gillespie. "The mathematics of Brownian motion and Johnson noise". In: *American Journal of Physics* 64.3 (1996), pp. 225–240.

[155] Isaac L Chuang and Michael A Nielsen. "Prescription for experimental determination of the dynamics of a quantum black box". In: *Journal of Modern Optics* 44.11-12 (1997), pp. 2455–2467.

[156] Rami Barends et al. "Superconducting quantum circuits at the surface code threshold for fault tolerance". In: *Nature* 508.7497 (2014), p. 500.

[157] Leonardo DiCarlo et al. "Preparation and measurement of three-qubit entanglement in a superconducting circuit". In: *Nature* 467.7315 (2010), p. 574.

[158] Chuan Wang et al. "Universal quantum controlled phase gate on photonic qubits based on nitrogen vacancy centers and microcavity resonators". In: *Optics express* 21.16 (2013), pp. 19252–19260.

[159] Wolfgang Pfaff et al. "Unconditional quantum teleportation between distant solid-state quantum bits". In: *Science* 345.6196 (2014), pp. 532–535.

[160] T Monz et al. "Realization of universal ion-trap quantum computation with decoherence-free qubits". In: *Physical review letters* 103.20 (2009), p. 200503.

[161] Ting Rei Tan et al. "Multi-element logic gates for trapped-ion qubits". In: *Nature* 528.7582 (2015), p. 380.

[162] Zhenyu Cai et al. "A Silicon Surface Code Architecture Resilient Against Leakage Errors". In: *arXiv preprint arXiv:1904.10378* (2019).

[163] Brandon Buonacorsi et al. “Network architecture for a topological quantum computer in silicon”. In: *Quantum Science and Technology* (2018).

[164] Xiaosi Xu, Simon C Benjamin, and Xiao Yuan. “Variational circuit compiler for quantum error correction”. In: *arXiv preprint arXiv:1911.05759* (2019).

[165] Harper R Grimsley et al. “An adaptive variational algorithm for exact molecular simulations on a quantum computer”. In: *Nature communications* 10.1 (2019), pp. 1–9.

[166] Tyson Jones et al. “Quest and high performance simulation of quantum computers”. In: *Scientific Reports* 9.1 (2019), p. 10736.

[167] I Alonso Calafell et al. “Quantum computing with graphene plasmons”. In: *npj Quantum Information* 5.1 (2019), p. 37.

[168] Christopher Chamberland and Andrew W Cross. “Fault-tolerant magic state preparation with flag qubits”. In: *Quantum* 3 (2019), p. 143.

[169] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. “Quantum algorithm for linear systems of equations”. In: *Physical review letters* 103.15 (2009), p. 150502.

[170] Andrew M Childs, Robin Kothari, and Rolando D Somma. “Quantum algorithm for systems of linear equations with exponentially improved dependence on precision”. In: *SIAM Journal on Computing* 46.6 (2017), pp. 1920–1950.

[171] Andris Ambainis. “Variable time amplitude amplification and quantum algorithms for linear algebra problems”. In: *STACS’12 (29th Symposium on Theoretical Aspects of Computer Science)*. Vol. 14. LIPIcs. 2012, pp. 636–647.

[172] B David Clader, Bryan C Jacobs, and Chad R Sprouse. “Preconditioned quantum linear system algorithm”. In: *Physical review letters* 110.25 (2013), p. 250504.

[173] Leonard Wossnig, Zhikuan Zhao, and Anupam Prakash. “Quantum linear system algorithm for dense matrices”. In: *Physical review letters* 120.5 (2018), p. 050502.

[174] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. “The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation”. In: *arXiv preprint arXiv:1804.01973* (2018).

[175] András Gilyén et al. “Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics”. In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*. ACM. 2019, pp. 193–204.

[176] Yiğit Subaşı, Rolando D Somma, and Davide Orsucci. “Quantum algorithms for systems of linear equations inspired by adiabatic quantum computing”. In: *Physical review letters* 122.6 (2019), p. 060504.

[177] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. “Quantum support vector machine for big data classification”. In: *Physical review letters* 113.13 (2014), p. 130503.

[178] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. “Quantum principal component analysis”. In: *Nature Physics* 10.9 (2014), p. 631.

[179] Jacob Biamonte et al. “Quantum machine learning”. In: *Nature* 549.7671 (2017), p. 195.

[180] Xiaosi Xu et al. “Variational algorithms for linear algebra”. In: *arXiv preprint arXiv:1909.03898* (2019).

[181] Daniel S Abrams and Seth Lloyd. “Simulation of many-body Fermi systems on a universal quantum computer”. In: *Physical Review Letters* 79.13 (1997), p. 2586.

[182] Alán Aspuru-Guzik et al. “Simulated quantum computation of molecular energies”. In: *Science* 309.5741 (2005), pp. 1704–1707.

[183] Pierre-Luc Dallaire-Demers et al. “Low-depth circuit ansatz for preparing correlated fermionic states on a quantum computer”. In: *arXiv preprint arXiv:1801.01053* (2018).

[184] Sam McArdle et al. “Quantum computational chemistry”. In: *arXiv preprint arXiv:1808.10402* (2018).

[185] Yudong Cao et al. “Quantum chemistry in the age of quantum computing”. In: *Chemical reviews* 119.19 (2019), pp. 10856–10915.

[186] Dave Wecker, Matthew B Hastings, and Matthias Troyer. “Progress towards practical quantum variational algorithms”. In: *Physical Review A* 92.4 (2015), p. 042303.

[187] A. Garcia-Saez and J. I. Latorre. “Addressing hard classical problems with Adiabatically Assisted Variational Quantum Eigensolvers”. In: *arXiv:1806.02287* (2018).

[188] Tameem Albash and Daniel A Lidar. “Adiabatic quantum computation”. In: *Reviews of Modern Physics* 90.1 (2018), p. 015002.