

Keisuke Fujii

Quantum
Computation with
Topological Codes
From Qubit
to Topological
Fault-Tolerance



Springer

SpringerBriefs in Mathematical Physics

Volume 8

Series editors

Nathanaël Berestycki, Cambridge, UK

Mihalis Dafermos, Cambridge, UK

Tohru Eguchi, Tokyo, Japan

Atsuo Kuniba, Tokyo, Japan

Matilde Marcolli, Pasadena, USA

Bruno Nachtergaele, Davis, USA

More information about this series at <http://www.springer.com/series/11953>

Keisuke Fujii

Quantum Computation with Topological Codes

From Qubit to Topological Fault-Tolerance

 Springer

Keisuke Fujii
Graduate School of Science
Kyoto University
Kyoto
Japan

ISSN 2197-1757 ISSN 2197-1765 (electronic)
SpringerBriefs in Mathematical Physics
ISBN 978-981-287-995-0 ISBN 978-981-287-996-7 (eBook)
DOI 10.1007/978-981-287-996-7

Library of Congress Control Number: 2015952538

Springer Singapore Heidelberg New York Dordrecht London
© The Author(s) 2015

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

Springer Science+Business Media Singapore Pte Ltd. is part of Springer Science+Business Media
(www.springer.com)

Preface

In 1982, Richard Feynman pointed out that a simulation of quantum systems on classical computers is generally inefficient because the dimension of the state space increases exponentially with the number of particles [1]. Instead, quantum systems could be simulated efficiently by other quantum systems. David Deutsch put this idea forward by formulating a quantum version of a Turing machine [2]. Quantum computation enables us to solve certain kinds of problems that are thought to be intractable with classical computers such as the prime factoring problem and an approximation of the Jones polynomial. It has a great possibility to disprove the extended (strong) Church–Turing thesis, i.e., that any computational process on realistic devices can be simulated efficiently on a probabilistic Turing machine.

However, for this statement to make sense, we need to determine whether or not quantum computation is a realistic model of computation. Rolf Landauer criticized it (encouragingly) by suggesting to put a footnote: “This proposal, like all proposals for quantum computation, relies on speculative technology, does not in its current form take into account all possible sources of noise, unreliability and manufacturing error, and probably will not work.” [3]. Actually, quantum coherence, which is essential for quantum computation is quite fragile against noise. If we cannot handle the effect of noise, quantum computation is of a limiting interest, like classical analog computers, as a realistic model of computation. To solve this, many researchers have investigated the fault-tolerance of quantum computation with developing quantum error correction techniques. One of the greatest achievements of this approach is topological fault-tolerant quantum computation using the surface code proposed by R. Raussendorf et al. [4–6]. It says that nearest-neighbor two-qubit gates and single-qubit operations on a two-dimensional array of qubits can perform universal quantum computation fault-tolerantly as long as the error rate per operation is less than $\sim 1\%$.

In this book, I present a self-consistent review of topological fault-tolerant quantum computation using the surface code. The book covers everything required to understand topological fault-tolerant quantum computation, ranging from the definition of the surface code to topological quantum error correction and

topological operations on the surface code. The basic concepts and powerful tools for understanding topological fault-tolerant quantum computation, such as universal quantum computation, quantum algorithms, stabilizer formalism, and measurement-based quantum computation, are also introduced in the first part (Chaps. 1 and 2) of the book. In particular, in Chap. 1, I also mentioned a quantum algorithm for approximating the Jones polynomials, which is also related to topological quantum computation with braiding non-Abelian anyons. In Chap. 3, the definition of the surface code and topological quantum error correction on it is explained. In Chap. 4, topological quantum computation on the surface code is described in the circuit-based model, where topological diagrams are introduced to understand the logical operations on the surface code diagrammatically. In Chap. 5, I explained the same thing in the measurement-based model, as done in the original proposal [4]. Hopefully, it would be easy to see how these two viewpoints are related.

Throughout the book, I have tried to explain the quantum operations using circuit and topological diagrams so that the readers can get a graphical understanding of the operations. The graphical understanding should be helpful to study the subjects more efficiently. Topological quantum error correction codes are a nice playground for studying the interdisciplinary connections between quantum information and other fields of physics, such as condensed matter physics and statistical physics. Actually, there is a nice correspondence between topological quantum error correction codes and topologically ordered systems in condensed matter physics. Furthermore, if we consider a decoding problem of a quantum error correction code, a partition function of a random statistical mechanical model is naturally appeared as a posterior probability for the decoding. These interdisciplinary topics are also included in Chap. 3.

Almost all topics, except for the basic concepts in the first part, are based on the results achieved after the appearance of the standard textbook of quantum information science entitled “Quantum Computation and Quantum Information” (Cambridge University Press 2000) by M.A. Nielsen and I.L. Chuang. In this sense, the present comprehensive review on these topics would be helpful to learn and update the recent progress efficiently. In this book, I concentrated on the quantum information aspect of topological quantum computation. Unfortunately, I cannot cover the more physical and condensed matter aspects of topological quantum computation, such as non-Abelian anyons and topological quantum field theory. In this sense, this book is complemented by the book “Introduction to Topological Quantum Computation” (Cambridge University Press 2012) written by J.K. Pachos. Readers who are interested in the more physical aspects of topological quantum computation are recommended to read it.

Hopefully, this review will encourage both theoretical and experimental researchers to find a more feasible way of quantum computation. It will also bring me great pleasure if this review provides an opportunity to reunify and refine various subdivided fields of modern physics in terms of quantum information.

Acknowledgments

First of all, I would like to express gratitude to Prof. Katsuji Yamamoto, who was my supervisor when I was a Ph.D. student and who has always supported me both heartily and critically. Even though quantum information science is not Prof. Katsuji Yamamoto's specialty, I was fortunate to have an opportunity to work freely on this subject. Unfortunately, he passed away in 2013, and I would like to express my deepest condolences and dedicate this book to him. I would like to thank Dr. Shuhei Tamate for his detailed reading of the manuscript and providing comments on it, which definitely helps to improve this book. Almost all my understanding of statistical mechanics, especially on spin glass models, was obtained through the discussions with Dr. Yoshifumi Nakata, Prof. Masayuki Ohzeki, and Prof. Shu Tanaka, and I am grateful for their help. A part of my own findings mentioned in this book is based on the collaborations with Dr. Yuuki Tokunaga and Prof. Tomoyuki Morimae. I would like to thank Rie Fujii for her dedicated work on splendid graphic images during her childcare holiday. Finally, I would like to thank my family, Rie Fujii, Mutsumi Fujii, and Manami Fujii for their constant support and everlasting enjoyable moments.

Contents

1	Introduction to Quantum Computation	1
1.1	Quantum Bit and Elementary Operations	1
1.2	The Solovay–Kitaev Algorithm	4
1.3	Multi-Qubit Gates	6
1.4	Universal Quantum Computation	10
1.5	Quantum Algorithms	12
1.5.1	Indirect Measurement and the Hadamard Test	12
1.5.2	Phase Estimation, Quantum Fourier Transformation, and Factorization	14
1.5.3	A Quantum Algorithm to Approximate Jones Polynomial	16
1.6	Quantum Noise	21
1.7	Summary and Discussion	23
2	Stabilizer Formalism and Its Applications	24
2.1	Stabilizer Formalism	24
2.2	Clifford Operations	26
2.3	Pauli Basis Measurements	29
2.4	Gottesman–Knill Theorem	30
2.5	Graph States	32
2.6	Measurement-Based Quantum Computation	35
2.7	Quantum Error Correction Codes	43
2.8	Magic State Distillation	49
2.8.1	Knill-Laflamme-Zurek Protocol	49
2.8.2	Bravyi-Kitaev Protocol	51
2.9	Summary and Discussion	55
3	Topological Stabilizer Codes	56
3.1	Z_2 Chain Complex	57
3.2	A Bit-Flip Code: Exercise	60

3.3	Definition of Surface Codes	61
3.3.1	Surface Code on a Torus: Toric Code	61
3.3.2	Planar Surface Code	64
3.4	Topological Quantum Error Correction	65
3.5	Error Correction and Spin Glass Model	71
3.6	Other Topological Codes	76
3.7	Connection to Topological Order in Condensed Matter Physics	79
3.8	Summary and Discussion	84
4	Topological Quantum Computation with Surface Codes	86
4.1	Defect Pair Qubits	86
4.2	Defect Creation, Annihilation, and Movement	88
4.3	Logical Controlled-NOT Gate by Braiding	91
4.4	Magic State Injections and Distillation	93
4.5	Topological Calculus	96
4.6	Faulty Syndrome Measurements and Noise Thresholds	102
4.7	Summary and Discussion	106
5	Topologically Protected Measurement-Based Quantum Computation	107
5.1	Topological Cluster State in Three Dimensions	107
5.2	Vacuum, Defect, and Singular Qubit Regions	109
5.3	Elementary Operations in Topological Measurement-Based Quantum Computation	110
5.4	Topological Quantum Error Correction in Three Dimensions	115
5.5	Applications for Measurement-Based Quantum Computation on Thermal States	116
5.6	Summary and Discussion	121
	Appendix A: Fault-Tolerant Quantum Computation	122
	Appendix B: Decoding Stabilizer Codes	127
	References	131
	Index	137

Chapter 1

Introduction to Quantum Computation

In this chapter, we introduce the basic concepts of quantum computation. We first describe the minimum unit of quantum information, the qubit, and define several gate operations for it. Then, we explain the Solovay–Kitaev algorithm, which provides a way to decompose an arbitrary single-qubit unitary operation into an elementary set of single-qubit gates. Using multi-qubit gates, we construct an arbitrary n -qubit unitary operation from an elementary universal set of gates, which we call universal quantum computation. Quantum algorithms, which run on a universal quantum computer, are also presented. One example of this is Shor’s prime factoring algorithm based on the phase estimation algorithm. Another is an approximation of the Jones polynomial. Finally, we will introduce quantum noise and see how we can describe a quantum system coupled with an environment.

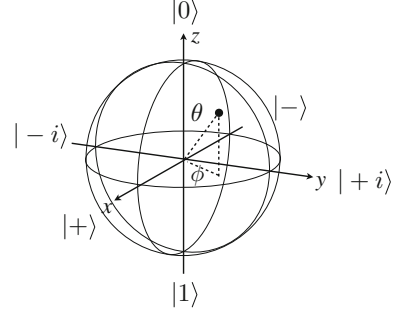
1.1 Quantum Bit and Elementary Operations

In classical information science, the minimum unit of information is described by a binary digit or *bit*, which takes the value 0 or 1. Its quantum counterpart is a quantum bit, the so-called *qubit*. The qubit is defined as a linear superposition of two orthogonal quantum states $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad (1.1)$$

where α and β are arbitrary complex values satisfying $|\alpha|^2 + |\beta|^2 = 1$. The complex amplitudes can be expressed as

$$\alpha = \cos \frac{\theta}{2}, \quad \beta = e^{i\phi} \sin \frac{\theta}{2}, \quad (1.2)$$

Fig. 1.1 The Bloch sphere

up to an unimportant global phase. By using the angles θ and ϕ , the qubit can be mapped onto a point on a three-dimensional (3D) sphere, the so-called *Bloch sphere*, as shown in Fig. 1.1.

The time evolution, which maps a quantum state into another, is given as a unitary operator in quantum mechanics. The most important operators are the Pauli operators

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (1.3)$$

The computational basis states $\{|0\rangle, |1\rangle\}$ are eigenstates of the Pauli Z operator. The Pauli X operator flips the computational basis state:

$$|1\rangle = X|0\rangle, \quad |0\rangle = X|1\rangle. \quad (1.4)$$

We define the eigenstates of the Pauli X operator as

$$|+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad |-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \quad (1.5)$$

which we call the X-basis states. Similarly, the eigenstates of the Pauli Y operator are defined as

$$|+i\rangle = \frac{|0\rangle + i|1\rangle}{\sqrt{2}}, \quad |-i\rangle = \frac{|0\rangle - i|1\rangle}{\sqrt{2}}, \quad (1.6)$$

which we call the Y-basis states.

The second most important operators are the Hadamard H and phase S operators

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad \text{and} \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}.$$

These gates transform between the different Pauli-basis states, i.e., $H : \{|0\rangle, |1\rangle\} \leftrightarrow \{|+\rangle, |-\rangle\}$ and $S : \{|+\rangle, |-\rangle\} \leftrightarrow \{|+i\rangle, |-i\rangle\}$. Equivalently, we may say that

these operators transform a Pauli operator into another Pauli operator under their conjugations:

$$X = HZH, \quad Y = SXS^\dagger. \quad (1.7)$$

From this property, the H and S gates are called Clifford gates. These Clifford gates are depicted as circuit diagrams as follows:



If we measure the qubit $|\psi\rangle$ in the Z -basis $\{|0\rangle, |1\rangle\}$, we obtain the measurement outcomes 0 and 1 with probabilities

$$p_0 = |\langle 0|\psi\rangle|^2 = \text{Tr} [|0\rangle\langle 0|\psi\rangle\langle\psi|], \quad (1.8)$$

$$p_1 = |\langle 1|\psi\rangle|^2 = \text{Tr} [|1\rangle\langle 1|\psi\rangle\langle\psi|], \quad (1.9)$$

respectively. More generally, we may use the measurement operators $\{M_i\}$, satisfying that $E_i \equiv M_i^\dagger M_i$ is positive semidefinite and that $\sum_i E_i = I$. (If an operator A satisfies $\forall |u\rangle, \langle u|A|u\rangle \geq 0$, it is said to be positive semidefinite.) The probability of obtaining the measurement outcome i is given by

$$p_i = \langle\psi|M_i^\dagger M_i|\psi\rangle = \text{Tr} [M_i^\dagger M_i|\psi\rangle\langle\psi|] = \text{Tr} [E_i|\psi\rangle\langle\psi|]. \quad (1.10)$$

Such a measurement and the set of positive operators $\{E_i = M_i^\dagger M_i\}$, are called a positive-operator-valued measure (POVM) measurement and POVM elements, respectively. The post-measurement state conditioned on the measurement outcome i is

$$\frac{M_i|\psi\rangle}{\sqrt{\text{Tr}[M_i^\dagger M_i|\psi\rangle\langle\psi|]}}. \quad (1.11)$$

Suppose we have quantum states $|\psi\rangle$ and $|\phi\rangle$ with probability p_ψ and p_ϕ , respectively. We perform a measurement with the measurement operator $\{M_i\}$. If we assume that the measurement outcome i is obtained with probability

$$p_i = p_\psi \text{Tr} [M_i^\dagger M_i|\psi\rangle\langle\psi|] + p_\phi \text{Tr} [M_i^\dagger M_i|\phi\rangle\langle\phi|], \quad (1.12)$$

we can express the classical mixture of $|\psi\rangle$ and $|\phi\rangle$ as an operator

$$\rho = p_\psi |\psi\rangle\langle\psi| + p_\phi |\phi\rangle\langle\phi|, \quad (1.13)$$

which is called the *density matrix* or *density operator*. More generally, if we have pure quantum states $\{|\psi_k\rangle\}$ with probability $\{p_k\}$, the density matrix is given by

$\rho = \sum_k p_k |\psi\rangle\langle\psi|$. A density matrix, which has information of a quantum state including its statistical property, has to be a positive hermitian (self-adjoint) operator ρ satisfying $\text{Tr}[\rho] = 1$. Note that for a given density matrix ρ , the pure state decomposition of it is not uniquely determined, and any decomposition has the same statistical property.

A mixed state of a qubit ρ can be represented as a point inside the Bloch sphere through the three coordinates calculated from the density matrix

$$(r_x, r_y, r_z) = (\text{Tr}[X\rho], \text{Tr}[Y\rho], \text{Tr}[Z\rho]). \quad (1.14)$$

If the state is a pure state $|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle$, the coordinates can be calculated to be

$$(r_x, r_y, r_z) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta), \quad (1.15)$$

which is consistent with the previous definition. The completely mixed state $I/2$ corresponds to the origin of the coordinate system.

1.2 The Solovay–Kitaev Algorithm

The Pauli operators $\{I, X, Y, Z\}$ and single-qubit Clifford operators $\{H, S\}$ form a finite group, and hence cannot cover all unitary operations for a qubit. If a non-Clifford operation exists, e.g., $e^{-i(\pi/8)Z}$, we can generate an arbitrary single-qubit unitary operation using the Solovay–Kitaev algorithm [16]. The underlying Solovay–Kitaev theorem states that if a set of single-qubit operations generates a dense subset of $SU(2)$, then that set is guaranteed to fill $SU(2)$ quickly.

Suppose we have a basic (0th order) approximation U_0 of an arbitrary unitary operator U , and that it approximates U with a certain constant error ε_0 :

$$\|U_0 - U\| \leq \varepsilon_0, \quad (1.16)$$

where $\|\dots\|$ indicates an operator norm. The Solovay–Kitaev algorithm takes the $(n-1)$ th order approximation U_{n-1} with an error ε_{n-1} and returns the n th order approximation U_n with an error ε_n as follows.

First, UU_{n-1}^\dagger is decomposed in terms of the unitary operators V and W as a group commutator:

$$UU_{n-1}^\dagger = VWV^\dagger W^\dagger, \quad (1.17)$$

where V and W are chosen such that

$$\|V - I\| < c\sqrt{\varepsilon_{n-1}}, \quad \|W - I\| < c\sqrt{\varepsilon_{n-1}}, \quad (1.18)$$

with a constant c . The above decomposition always exists from the following argument. Let V and W be rotations with an angle ϕ about the x - and y -axes, respectively. Then $VWV^\dagger W^\dagger$ is a rotation with an angle θ about some axis, where

$$\sin(\theta/2) = 2 \sin^2(\phi/2) \sqrt{1 - \sin^4(\phi/2)}. \quad (1.19)$$

We assume that $\phi, \theta \ll 1$, and hence $\theta \simeq \phi^2$. By inverting the above argument, if

$$\|I - VWV^\dagger W^\dagger\| \simeq \theta/2 + O(\theta^3) \quad (1.20)$$

we can find V and W such that

$$\|I - V\|, \|I - W\| \simeq \phi/2 + O(\phi^3). \quad (1.21)$$

Because

$$\|I - UU_{n-1}^\dagger\| = \|I - VWV^\dagger W^\dagger\| < \varepsilon_{n-1}, \quad (1.22)$$

we can find V and W such that $\|I - V\|, \|I - W\| < c\sqrt{\varepsilon_{n-1}}$.

Second, we calculate the $(n-1)$ th order approximations V_{n-1} and W_{n-1} of V and W , respectively. Then the Solovay–Kitaev algorithm returns the n th order approximation

$$U_n = V_{n-1} W_{n-1} V_{n-1}^\dagger W_{n-1}^\dagger U_{n-1}, \quad (1.23)$$

which satisfies

$$\|U_n - U\| \leq \varepsilon_n. \quad (1.24)$$

Next, we will calculate ε_n as a function of ε_{n-1} . By using the property of the operator norm, we obtain

$$\|U_n - U\| \leq \|V_{n-1} W_{n-1} V_{n-1}^\dagger W_{n-1}^\dagger - UU_{n-1}^\dagger\| \|U_{n-1}\| \quad (1.25)$$

$$\leq \|V_{n-1} W_{n-1} V_{n-1}^\dagger W_{n-1}^\dagger - VWV^\dagger W^\dagger\| \|U_{n-1}\| \quad (1.26)$$

By denoting $V_{n-1} = V + \Delta_V$, $W_{n-1} = W + \Delta_W$, we obtain

$$\|U_n - U\| \leq \|V_{n-1} W_{n-1} V_{n-1}^\dagger W_{n-1}^\dagger - VWV^\dagger W^\dagger\| \quad (1.27)$$

$$\leq \|\Delta_V W V^\dagger W^\dagger + V W \Delta_V^\dagger W^\dagger\| + \|V \Delta_W V^\dagger W^\dagger + V W V^\dagger \Delta_W^\dagger\| + O(\Delta^2). \quad (1.28)$$

Moreover, in terms of δ_V and δ_W defined by $V = I + \delta_V$ and $W = W + \delta_W$ respectively, we obtain

$$\|U_n - U\| \leq \|\Delta_V V^\dagger + V \Delta_V^\dagger\| + \|\Delta_W W^\dagger + W \Delta_W^\dagger\| \quad (1.29)$$

$$+ \|\Delta_V \delta_W\| + \|\Delta_V^\dagger \delta_W^\dagger\| + \|\delta_V \Delta_W\| + \|\delta_V^\dagger \Delta_W^\dagger\| + O(\Delta^2). \quad (1.30)$$

$$\leq 4c\varepsilon_{n-1}^{3/2} + O(\Delta^2, \delta^2 \Delta). \quad (1.31)$$

Here we have used that

$$\Delta_V V^\dagger + V \Delta_V^\dagger = \Delta_V \Delta_V^\dagger, \quad (1.32)$$

which can be derived from the unitarity of V and $V_{n-1} = V + \Delta_V$. To leading order, the error is given by $\varepsilon_n \leq c'\varepsilon_{n-1}^{3/2}$ with a certain constant c' and is calculated to be

$$\varepsilon_n \leq (2c'\varepsilon_0)^{(3/2)^n} / (2c'). \quad (1.33)$$

If $\varepsilon_0 < 1/(2c')$, the error decreases super-exponentially in n . On the other hand, the n th order approximation calls the $(n-1)$ th order approximation three times, i.e., through U_{n-1} , V_{n-1} , and W_{n-1} . Including the resource R_D required for the decomposition (1.17), the overhead R_n for the n th order approximation is given by

$$R_n = 3R_{n-1} + R_D \quad (1.34)$$

$$\Leftrightarrow R_n = O(3^n). \quad (1.35)$$

Similarly, the number of unitary operations employed in the n th order approximation is calculated to be $M_n = O(5^n)$. By using (1.33) and (1.35), the overhead R_ε and the number M_ε of gates required to obtain an approximation of U with an error ε can be estimated:

$$R_\varepsilon = O(\ln^{\ln 3 / \ln(3/2)}(1/\varepsilon)), \quad (1.36)$$

$$M_\varepsilon = O(\ln^{\ln 5 / \ln(3/2)}(1/\varepsilon)). \quad (1.37)$$

Thus, both R_ε and M_ε scale as polylogarithmic functions of $1/\varepsilon$.

1.3 Multi-Qubit Gates

An n -qubit state is given by a superposition of tensor product states

$$|\Psi\rangle = \sum_{i_1, i_2, \dots, i_n} C_{i_1 i_2 \dots i_n} |i_1 i_2 \dots i_n\rangle,$$

where $i_k = 0, 1$ and $|i_1 i_2 \dots i_n\rangle \equiv |i_1\rangle \otimes |i_2\rangle \otimes \dots \otimes |i_n\rangle$. A single qubit gate A acting on the k th qubit is denoted by

$$A_k = \overbrace{I \otimes \dots \otimes I}^{k-1} \otimes A \otimes \overbrace{I \otimes \dots \otimes I}^{n-k-1}. \tag{1.38}$$

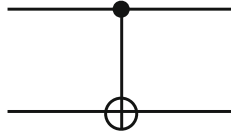
An important two-qubit gate is the controlled-NOT (CNOT) gate,

$$\Lambda_{c,t}(X) = |0\rangle\langle 0|_c I_t + |1\rangle\langle 1|_c X_t.$$

For a computational basis input state $|i\rangle_c |j\rangle_t$, the CNOT gate acts as $\Lambda_{c,t}(X)|i\rangle_c |j\rangle_t = |i \oplus j\rangle$. In this sense, the CNOT gate is a quantum generalization of the XOR operation in classical computation. If the input state is $|+\rangle_c |0\rangle_t$, the output of the CNOT gate is a maximally entangled state:

$$\Lambda_{c,t}(X)|+\rangle_c |0\rangle_t = (|00\rangle + |11\rangle)/\sqrt{2}. \tag{1.39}$$

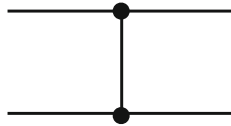
The CNOT gate is depicted by a circuit diagram as follows:



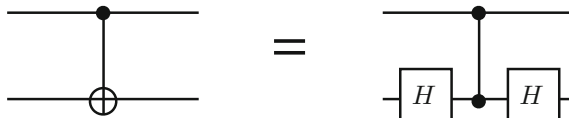
We may also define the controlled-Z (CZ) gate,

$$\Lambda_{c,t}(Z) = |0\rangle\langle 0|_c I_t + |1\rangle\langle 1|_c Z_t.$$

The CZ operation is symmetric; $\Lambda_{c,t}(Z) = \Lambda_{t,c}(Z)$, and is represented by the circuit diagram



Because the Pauli Z operator is transformed into the Pauli X operator by the Hadamard gate, we have the following relation between the CZ and CNOT gates:



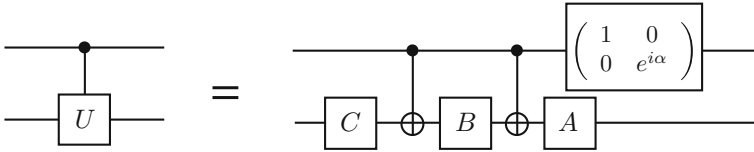
These CNOT and CZ gates are both Clifford gates, i.e., $\Lambda(A)$ ($A = X, Z$) transforms the two-qubit Pauli group onto itself under the conjugation $\Lambda(A)[\dots]\Lambda(A)^\dagger$.

For example, $\Lambda(X)_{c,t} X_c \otimes I_t \Lambda(X)_{c,t}^\dagger = X_c \otimes X_t$, $\Lambda(X)_{c,t} I_c \otimes Z_t \Lambda(X)_{c,t}^\dagger = Z_c \otimes Z_t$, $\Lambda(Z)_{c,t} X_c \otimes I_t \Lambda(Z)_{c,t}^\dagger = X_c \otimes Z_t$, etc.

For an arbitrary unitary operator U , the controlled- U gate is denoted by

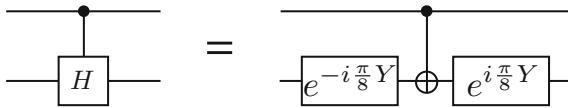
$$\Lambda_{c,t}(U) = |0\rangle\langle 0|_c I_t + |1\rangle\langle 1|_c U_t, \tag{1.40}$$

where the qubits c and t are called the control and target qubits, respectively. By decomposing a single-qubit unitary gate into $U = e^{i\alpha}AXBXC$ with the unitary operators A, B, C satisfying $ABC = I$, the controlled- U operation $\Lambda(U)$ can be implemented as follows:



(Note that we always have such a decomposition for an arbitrary single-qubit gate [8, 9].)

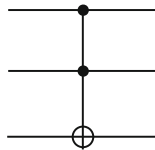
For example, the controlled-Hadamard gate $\Lambda(H)$ can be represented by



Next, we will discuss one of the most important multi-qubit gates, the Toffoli gate:

$$\Lambda_{c_1,c_2,t}^2(X) = (I_{c_1} I_{c_2} - |1\rangle\langle 1|_{c_1} |1\rangle\langle 1|_{c_2}) I_t + |1\rangle\langle 1|_{c_1} |1\rangle\langle 1|_{c_2} X_t, \tag{1.41}$$

which is depicted as a circuit diagram:



For a computational basis input state $|i_1\rangle_{c_1} |i_2\rangle_{c_2} |j\rangle_t$, the Toffoli gate acts as

$$\Lambda_{c_1,c_2,t}^2(X) |i_1\rangle_{c_1} |i_2\rangle_{c_2} |j\rangle_t = |i_1\rangle_{c_1} |i_2\rangle_{c_2} |j \oplus (i_1 \cdot i_2)\rangle_t. \tag{1.42}$$

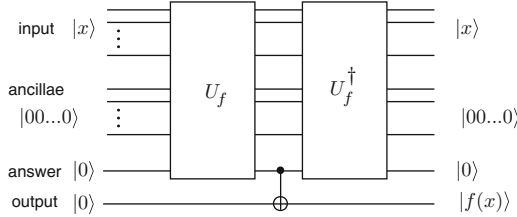
The state of the third qubit is equivalent to the output of the NAND operation in classical computation. In this sense, the Toffoli operation can be regarded as a quantum extension of the NAND operation. The NAND operations are known to be universal in classical computation in the sense that any logic gate (boolean function) can be constructed from them. This implies that quantum computation trivially includes classical computation. More importantly, because unitary operations are reversible,

quantum computation can simulate classical computation in a reversible way. Suppose we want to calculate a boolean function $f(x)$ for an input state x . Then, we can construct a quantum circuit U_f consisting of Toffoli and Pauli X gates:

$$U_f |x\rangle_{\text{input}} |0\dots 0\rangle_{\text{ancilla}} |0\rangle_{\text{answer}} = |x\rangle_{\text{input}} |g(x)\rangle_{\text{ancilla}} |f(x)\rangle_{\text{answer}}, \quad (1.43)$$

where the qubits $|\cdot\rangle_{\text{input}}$, $|\cdot\rangle_{\text{ancilla}}$, and $|\cdot\rangle_{\text{answer}}$ indicate the registers for the input state, the ancillae for the Toffoli operations, and the answer of the calculation, respectively. The output state of the ancilla register $|g(x)\rangle_{\text{ancilla}}$ is the garbage of the computation. However, the garbage can be uncomputed as follows (see also the circuit diagram below):

$$\begin{aligned} U_f^\dagger A_{\text{answer,out}}(X) U_f |x\rangle_{\text{input}} |0\dots 0\rangle_{\text{ancilla}} |0\rangle_{\text{answer}} |0\rangle_{\text{out}} \\ = U_f^\dagger |x\rangle_{\text{input}} |g(x)\rangle_{\text{ancilla}} |f(x)\rangle_{\text{answer}} |f(x)\rangle_{\text{out}} \\ = |x\rangle_{\text{input}} |0\dots 0\rangle_{\text{ancilla}} |0\rangle_{\text{answer}} |f(x)\rangle_{\text{out}}. \end{aligned} \quad (1.44)$$

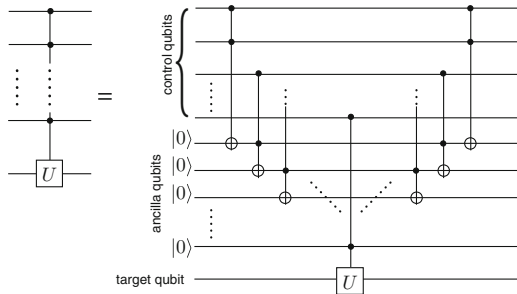


In this way, we can calculate an arbitrary boolean function in a reversible way.

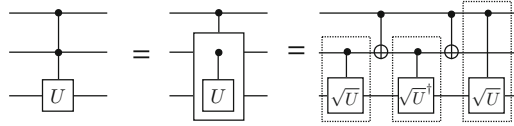
Finally, we introduce the multi-controlled unitary gate

$$A^k(U) = (I^{\otimes k} - |1\rangle\langle 1|^{\otimes k}) \otimes I + |1\rangle\langle 1|^{\otimes k} \otimes U, \quad (1.45)$$

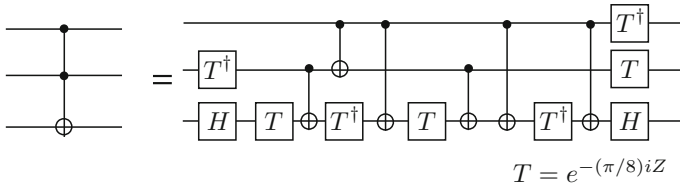
where U is applied to the target qubit if all k control qubits are $|1\rangle$. The multi-controlled gate can be implemented by using the Toffoli gates and k ancilla qubits as follow:



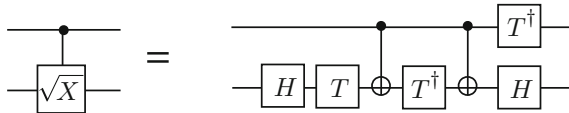
The $\Lambda^2(U)$ gate can be decomposed into CNOT and single-qubit gates by using an idea similar to the decomposition of the $\Lambda(U)$ gate:



For example, the Toffoli gate can be constructed from the CNOT, Hadamard, and $\pi/8$ operations as follows:



where we have used the fact that that the controlled- \sqrt{X} gate is decomposed into the CNOT, Hadamard, and $\pi/8$ ($T = e^{-i(\pi/8)Z}$) gates as follows:



It should be noted that a multi-controlled gate $\Lambda^k(U)$ can also be constructed without any ancilla qubit from multi-controlled gates $\Lambda^{(k-1)}(U)$ with lower controlled qubits by using the trick employed for decomposing $\Lambda^2(U)$ into $\Lambda(U)$ [8].

1.4 Universal Quantum Computation

Here we will show that an arbitrary unitary operation can be decomposed into single-qubit and CNOT gates. To this end, we will first show how to decompose an arbitrary unitary operation into a product of two-level unitary gates. Second, the two-level gates are decomposed into the multi-controlled gates, which can be constructed from single-qubit and CNOT gates as seen in the previous section.

Let U be an arbitrary n -qubit unitary operator, represented by an $m \times m$ unitary matrix with $m \equiv 2^n$. Let T_{ij} be a unitary operator such that $(T_{ij})_{kl} = \delta_{kl}$ if $k, l \neq i, j$, which we call a two-level unitary gate. (The kl element of a matrix A is denoted by $(A)_{kl}$.) The $ii, ij, ji,$ and jj elements define a unitary operation on the two-level

subsystem spanned by $|j\rangle$ and $|i\rangle$. By choosing $T_{m\ m-1}$ appropriately, we have

$$UT_{m\ m-1} = \begin{pmatrix} u_{11} & \cdots & u_{1\ m-1} & u_{1\ m} \\ \vdots & \ddots & \vdots & \vdots \\ u_{m-1\ 1} & \cdots & u'_{m-1\ m-1} & u'_{m-1\ m} \\ u_{m\ 1} & \cdots & 0 & u'_{m\ m} \end{pmatrix}, \quad (1.46)$$

where $(U)_{kl} = u_{kl}$. By repeating this procedure, we obtain

$$UT_{m\ m-1}T_{m\ m-2}\cdots T_{m\ 1} = \begin{pmatrix} u''_{11} & \cdots & u''_{1\ m-1} & u''_{1\ m} \\ \vdots & \ddots & \vdots & \vdots \\ u''_{m-1\ 1} & \cdots & u''_{m-1\ m-1} & u''_{m-1\ m} \\ 0 & \cdots & 0 & u''_{m\ m} \end{pmatrix}. \quad (1.47)$$

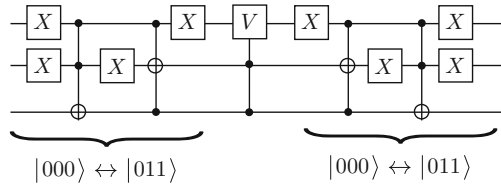
Due to unitarity, $u''_{1\ m} = \cdots = u''_{m-1\ m} = 0$ and $|u''_{mm}| = 1$. Defining $R_m \equiv T_{m\ m-1}T_{m\ m-2}\cdots T_{m\ 1}$, we can decompose U into a product of R_k and a diagonal unitary operator D :

$$U = D(R_m \cdots R_1)^\dagger. \quad (1.48)$$

It is obvious that D can be decomposed into two-level unitary gates. Thus, an arbitrary unitary operator U can be decomposed into two-level unitary gates.

Next, we show that any two-level unitary operator T_{ij} can be implemented by using CNOT and single-qubit gates. Let us rewrite i and j ($i, j = 0, \dots, m-1$) by using the n -bit strings $\mathbf{s} = s_1s_2\dots s_n$ and $\mathbf{t} = t_1t_2\dots t_n$, respectively. It is easy to find a sequence of n -bit strings $\{\mathbf{g}_k\}_{k=1}^d$ such that $\mathbf{s} = \mathbf{g}_1$, $\mathbf{t} = \mathbf{g}_d$, and \mathbf{g}_k and \mathbf{g}_{k+1} differ by only one bit. By using the Pauli X gate and the multi-controlled-NOT gate controlled by the same $n-1$ bits and targeting the one different bit, we can transform the basis $|\mathbf{g}_k\rangle$ to $|\mathbf{g}_{k+1}\rangle$. In this way, the basis $|i\rangle = |\mathbf{s}\rangle = |\mathbf{g}_1\rangle$ is transformed into $|\mathbf{g}_{d-1}\rangle$. After this basis transformation, we now want to apply a two-level unitary gate between $|\mathbf{g}_{d-1}\rangle$ and $|j\rangle = |\mathbf{t}\rangle = |\mathbf{g}_d\rangle$. Because \mathbf{g}_{d-1} and \mathbf{g}_d differ by only one bit, we can perform such a two-level unitary gate by the multi-conditional gate, using the Pauli X gate for bit flips. Finally, the basis $|\mathbf{g}_{d-1}\rangle$ is returned to $|\mathbf{s}\rangle$ by applying the inverse of the basis transformation.

For example, a two-level unitary operator acting on a subspace spanned by $\{|000\rangle, |111\rangle\}$ can be implemented as follows:



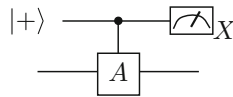
As seen previously, the multi-conditional gate can be decomposed into CNOT and single-qubit gates. Moreover, an arbitrary single-qubit unitary operation can be approximated by using the Hadamard and $\pi/8$ operations by virtue of the Solovay–Kitaev algorithm. Thus, the CNOT, Hadamard, and $\pi/8$ operations form a universal set of operations for quantum computations. The Toffoli and Hadamard operations also form a universal set as shown in [8–12].

1.5 Quantum Algorithms

In this section, we explain two representative quantum algorithms; Shor’s prime factorization algorithm [13, 14] and the Aharonov-Jones-Landau algorithm for an additive approximation of the Jones polynomial [15, 16]. We do not go deep into the mathematically rigorous details, but we aim to understand how they work. The readers who are interested in more details should read Refs. [9, 13–17].

1.5.1 Indirect Measurement and the Hadamard Test

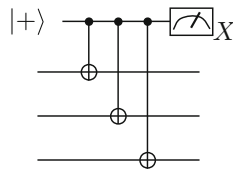
An indirect measurement of an observable (hermitian) A with eigenvalues ± 1 can be performed by using $\Lambda(A)$:



By denoting the input state $|\psi\rangle$, the post-measurement state is given by

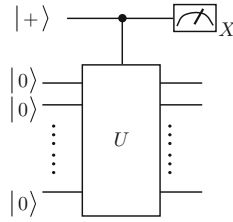
$$\frac{I + (-1)^s A}{2} |\psi\rangle / \sqrt{\text{Tr}[(I + (-1)^s A)/2 |\psi\rangle\langle\psi|]}, \tag{1.49}$$

where $s = 0, 1$ is the measurement outcome. For example, a circuit measuring the eigenvalue of the operator $X_1 X_2 X_3$ is given by



According to the measurement outcome $s = 0, 1$, the post-measurement state is projected by $\frac{I + (-1)^s X_1 X_2 X_3}{2}$. The indirect measurement will be employed frequently in quantum error correction to measure the eigenvalues of the stabilizer operators.

The Hadamard test of an arbitrary unitary operator U is defined by the following circuit:

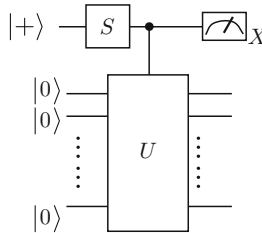


The probabilities of the measurement outcomes 0, 1 of the X-basis measurement are calculated to be

$$p_0 = \frac{1}{2} (1 + \text{Re}\langle 0|^{\otimes n} U |0\rangle^{\otimes n}), \tag{1.50}$$

$$p_1 = \frac{1}{2} (1 - \text{Re}\langle 0|^{\otimes n} U |0\rangle^{\otimes n}). \tag{1.51}$$

Similarly, the Hadamard test for the imaginary part is defined:

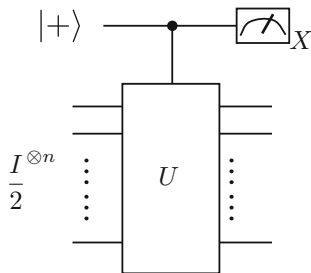


Suppose we perform the Hadamard test N times and obtain the measurement outcome 0, N_0 times. By using the Chernoff-Hoeffding bound,

$$\text{Prob} \left(\left| \frac{N_0}{N} - p_0 \right| > \varepsilon \right) < 2e^{-2\varepsilon^2 N}, \tag{1.52}$$

we can estimate the matrix element $\langle 0|^{\otimes n} U |0\rangle^{\otimes n}$ with an error ε by repeating the Hadamard test $N = \text{poly}(1/\varepsilon)$ times. The Hadamard test is employed in various quantum algorithms such as approximations of the Jones and Tutte polynomials [15, 16, 18] and the partition functions of statistical mechanical models [19–21].

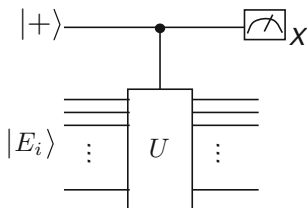
Specifically, if we choose the input state to be a completely mixed n -qubit state,



then, the Hadamard test provides the trace $\text{Tr}[U]/2^n$ of the unitary operator U . Such a restricted type of quantum computation is called a deterministic quantum computation with one clean qubit (DQC1) [22]. DQC1 seems to be less powerful than universal quantum computation, because only one qubit is a pure state. However, DQC1 can evaluate functions, which would be intractable on a classical computer, such as the Jones and Homefly polynomials [23, 24], spectral density function [22], and fidelity decay [25]. Recently, classical sampling of the output of DQC1 with a few qubits measurements has been shown to be intractable unless the polynomial hierarchy collapses to the third level [26, 27].

1.5.2 Phase Estimation, Quantum Fourier Transformation, and Factorization

If we have an eigenstate $|E_i\rangle$ of a unitary operator U , for which $\Lambda(U)$ is described by a polynomial number of gates, we can estimate the eigenvalue λ_i of U with a polynomial accuracy using the Hadamard test as follows:

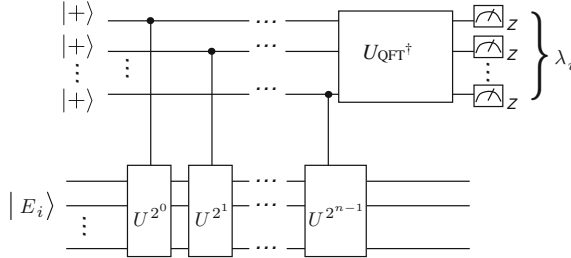


When we do not have the eigenstate, the input state is projected onto one of the eigenstates by the Hadamard test. Thus, by repeating the Hadamard test, we can obtain one of the eigenvalues with polynomial accuracy.

Moreover, if a controlled- U^{2^k} gate $\Lambda(U^{2^k})$ can be described by a polynomial number of gates, we can efficiently estimate the eigenvalue with exponential accuracy [28]. Suppose the eigenvalue is given by $\lambda_i = e^{i\phi} = e^{2\pi i 0.j_1j_2\dots j_n}$, where we employ a decimal of a binary number,

$$0.j_1j_2\dots j_n = \sum_{k=1}^n j_k(1/2)^k. \tag{1.53}$$

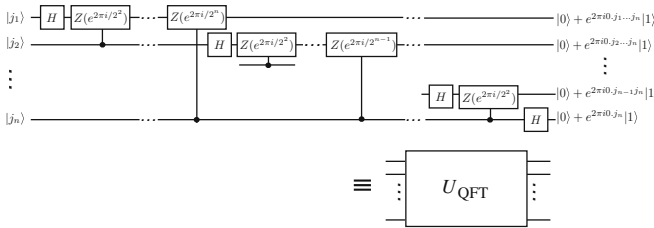
Then, the Kitaev’s phase estimation algorithm is given by



where U_{QFT} stands for the quantum Fourier transform. The controlled gate $\Lambda(U^{2^k})$ kicks back the phase $e^{2^k\phi} = e^{2\pi i j_k j_{k+1} \dots j_n}$ to the ancilla state $|+\rangle$. The phase information is transformed into a computational basis state by using the inverse quantum Fourier transformation:

$$U_{QFT}^\dagger(|0\rangle + e^{2\pi i 0.j_1j_2\dots j_n}|1\rangle)(|0\rangle + e^{2\pi i 0.j_2\dots j_n}|1\rangle)\dots(|0\rangle + e^{2\pi i 0.j_n}|1\rangle) = |j_1j_2\dots j_n\rangle, \tag{1.54}$$

and given by



Then, we can obtain the phase $\phi = (2\pi i)0.j_1j_2\dots j_n$ through the Z-basis measurements. Note that the accuracy of the estimate is improved exponentially compared to the Hadamard test (provided that we have a polynomial size description of $\Lambda(U^{2^k})$).

Let N and x be co-prime integers. The order-finding problem is about finding an order r such that $x^r = 1 \pmod N$. This problem can be solved by using the phase estimation against a unitary operator

$$U_x = \sum_y |xy \pmod N\rangle \langle y|, \tag{1.55}$$

because we have

$$U_x|u_s\rangle = e^{2\pi i(s/r)}|u_s\rangle, \quad (1.56)$$

for a state

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i(s/r)k} |x^k \bmod N\rangle. \quad (1.57)$$

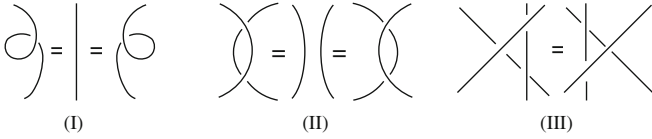
Note that we can prepare the initial state $|u_s\rangle$ randomly by using the fact that $|1\rangle = \sum_{s=0}^{r-1} |u_s\rangle$. After estimating the phase $2\pi i(s/r)$, the continued fraction provides r with high probability. The modular exponentiation $x^{2^k} \bmod N$ can be calculated by using the square k times. Thus we can implement controlled- $U_x^{2^k}$, $\Lambda(U_x^{2^k})$, by a polynomial number of gates.

If we randomly choose s , we obtain an even order r with high probability. Thus, we have $(x^{r/2} - 1)(x^{r/2} + 1) = 0 \bmod N$. Finally, the euclidean algorithm gives us the greatest common divisor of $x^{r/2} - 1$ and N or $x^{r/2} + 1$ and N , and it is a factor of N . This is the so-called Shor's prime factoring algorithm. Kitaev, who formulated the phase estimation algorithm, generalized this idea for the more general Abelian stabilizer problem [17].

1.5.3 A Quantum Algorithm to Approximate Jones Polynomial

Next we will explain Aharonov-Jones-Landau algorithm for approximating the Jones polynomial [15, 16], which is an algorithmic version of the original proposal based on topological quantum field theory [29, 30]. We will show that the approximation of the Jones polynomial is BQP-complete; the approximation of the Jones polynomial can be done efficiently by using a universal quantum computer, and inversely it is as hard as any problems (BQP) solvable by a universal quantum computation. Below we will introduce the braid diagram and the braid group, which are related to link diagrams with an appropriate closure. A unitary representation of the braid group is constructed by embedding the braid group to the Temperley-Lieb (TL) algebra. Using the constructed representation, the Jones polynomial and quantum algorithm are connected.

Let us first define the Jones polynomial. The Jones polynomial is an invariant of a link, that is, closed loops in a 3D space, which are tangled in general [31]. It is convenient to describe the link on a 2D projected space as shown in Fig. 1.2a, which we call a link diagram. Specifically two links are equivalent if two link diagrams can be transformed into each others under the Reidemeister moves:



which appropriately reflect the continuous deformations in the 3D space. The Jones polynomial is calculated from a directed link diagram L as follows: (i) Smooth each crossing $\nearrow \nwarrow$ in two ways $\{ \nearrow \nearrow, \nwarrow \nwarrow \}$. Let s be the resultant diagram consisting of closed loops with no crossing, which we call a state. (ii) For each state s , we assign a weight

$$W(s) = A^{s^+ - s^-} d^{|s| - 1}, \tag{1.58}$$

where $d = -(A^2 + A^{-2})$ is a complex constant called a loop value, $|s|$ is the number of the crossing, and s^+ and s^- are the number of smoothings by $\nearrow \nearrow$ and $\nwarrow \nwarrow$, respectively. By taking the summation over all states, the Kauffman bracket $\langle L \rangle$ of the link L is defined:

$$\langle L \rangle = \sum_s W(s). \tag{1.59}$$

The Jones polynomial is defined as a function of $t = A^{-4}$ by multiplying a factor to the Kauffman bracket:

$$V_L(t) = (-A)^{3\omega(L)} \langle L \rangle, \tag{1.60}$$

where the writhe $\omega(L)$ of the link L is defined as the number of $\nearrow \nwarrow$ type crossings minus that of $\nwarrow \nearrow$ type. It is easy and a good exercise to confirm that the Jones polynomial is invariant under the Reidemeister moves based on the above definition [31].

Let B_n be a braid group consisting of the braid diagrams of n strands, where two endpoints of each strand are tied at the top and bottom, respectively as shown in Fig. 1.2b. The multiplication of two braids b_1 and b_2 are defined by connecting the

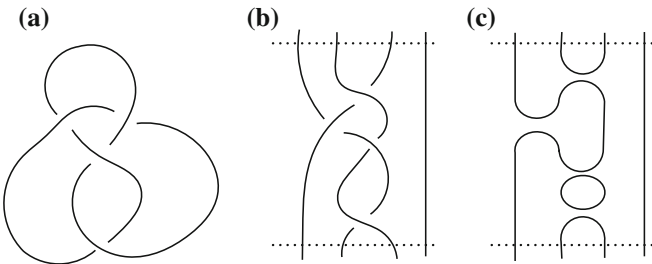


Fig. 1.2 **a** A link diagram. **b** A braid diagram. **c** A tangle diagram

bottom and top endpoints of two braid diagrams as shown in Fig. 1.3a. The braid group B_n is generated by $n - 1$ generators $\{\sigma_i\}$ subject to

$$\sigma_i \sigma_j = \sigma_j \sigma_i \quad \text{for } |i - j| \geq 2 \quad (1.61)$$

$$\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}. \quad (1.62)$$

The generator σ_i corresponds to the braid diagram with only one crossing of i th and $i + 1$ th strands as shown in Fig. 1.3b. Then, the latter equality is nothing but the Reidemeister move III. By introducing an appropriate closure, which closes the endpoints of the strands, a braid diagram is related to a link diagram as we will see later.

In order to construct a unitary representation of B_n , we embed it into the Temperley-Lieb (TL) algebra $TL_n(d)$ [32] on tangle diagrams with no crossing as shown in Fig. 1.2c. Similarly to the previous case, $TL_n(d)$ is generated by a set of generators $\{E_1, \dots, E_{n-1}\}$ subject to

$$E_i E_j = E_j E_i \quad \text{for } |i - j| \geq 2, \quad (1.63)$$

$$E_i E_{i \pm 1} E_i = E_i, \quad (1.64)$$

$$E_i^2 = d E_i. \quad (1.65)$$

The generator E_i corresponds to \cup and \cap for the i th and $(i + 1)$ th strands as shown in Fig. 1.3c. The braid group is embedded into the TL algebra by

$$\sigma_i = A E_i + A^{-1} \mathbf{1}, \quad (1.66)$$

where $\mathbf{1}$ indicates the identity element.

In order to relate the braid diagram and the Jones polynomial, we employ a Markov trace closure on the tangle diagram as shown in Fig. 1.4a. The Markov trace closure denoted by mtr has the following property:

$$\text{mtr}(\mathbf{1}) = 1, \quad (1.67)$$

$$\text{mtr}(AB) = \text{mtr}(BA) \quad \text{for } A, B \in TL_n(d), \quad (1.68)$$

$$\text{mtr}(A) = d \text{mtr}(A E_{n-1}) \quad \text{for } A \in TL_{n-1}(d). \quad (1.69)$$

Fig. 1.3 **a** A multiplication for two braid diagrams b_1 and b_2 . **b** The generator σ_i of the braid group. **c** The generator E_i of the TL algebra

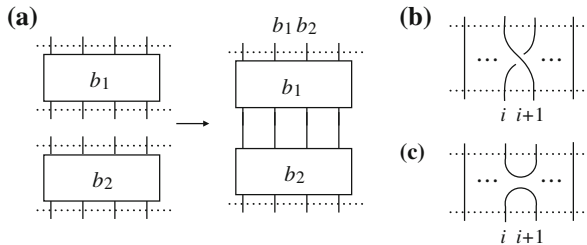
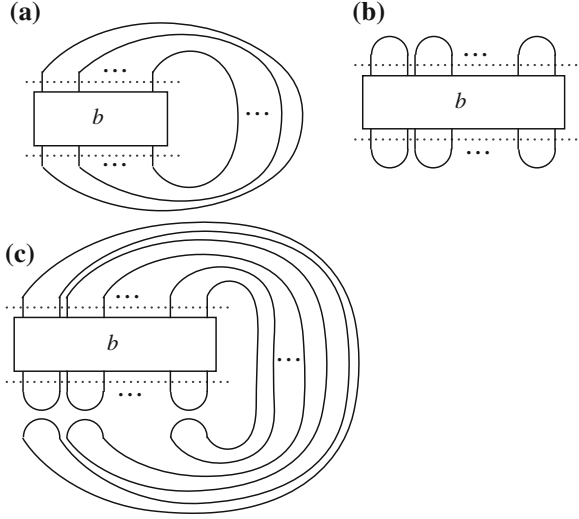


Fig. 1.4 **a** The Markov trace closure for the braid b . **b** The plat closure for the braid b . **c** The plat closure is deformed into the Markov trace closure



Now the representation of the braid group and the Jones polynomial are related. Equation (1.66) corresponds to superposition of two smoothings $\{ \nearrow, \searrow \}$ of a crossing \times . More precisely, let ρ and $\tilde{\rho}$ be a representation of the TL algebra and an induced representation of the braid group, respectively. Then we have $\tilde{\rho}(\sigma_i) \equiv A\rho(E_i) + A^{-1}I$. Moreover, the matrix trace has the same property as the Markov trace closure. Thus, for a given braid diagram b , we have

$$V_{b^{\text{mtr}}}(A^{-4}) = \Delta \text{Tr}[\tilde{\rho}(b)], \tag{1.70}$$

where b^{mtr} is a link diagram generated from the braid diagram b with the Markov trace closure, and $\Delta_{\text{mtr}} = (-A)^{3\omega(b^{\text{mtr}})} d^{n-1}$.

Let us construct a representation of the TL algebra, the so-called path-model representation. Suppose G is a one-dimensional graph with k vertices labeled by $1, \dots, k$ from left to right and $k - 1$ edges. We consider an n step walk on G starting from the left endpoint. Let $p = (v_1, \dots, v_n)$ be a path of such an n step walk, where $v_1 = 1$ and $v_j = 0$ and $= 1$ mean moving to the left and right neighboring vertices, respectively. Then we define a Hilbert space $\mathcal{H}_{n,k}$ spanned by all possible paths as a basis $\{|p\rangle\}$. For each generator E_i , we define its representation $\Phi_i = \rho(E_i)$ on $\mathcal{H}_{n,k}$ as follows:

$$\Phi_i | \dots v_{i-1} 0 0 v_{i+1} \dots \rangle = 0, \tag{1.71}$$

$$\Phi_i | \dots v_{i-1} 0 1 v_{i+1} \dots \rangle = \frac{\lambda_{z_i-1}}{\lambda_{z_i}} | \dots v_{i-1} 0 1 v_{i+1} \dots \rangle + \frac{\sqrt{\lambda_{z_i+1} \lambda_{z_i-1}}}{\lambda_{z_i}} | \dots v_{i-1} 1 0 v_{i+1} \dots \rangle, \tag{1.72}$$

$$\Phi_i | \dots v_{i-1} 10 v_{i+1} \dots \rangle = \frac{\lambda_{z_i+1}}{\lambda_{z_i}} | \dots v_{i-1} 10 v_{i+1} \dots \rangle + \frac{\sqrt{\lambda_{z_i+1} \lambda_{z_i-1}}}{\lambda_{z_i}} | \dots v_{i-1} 01 v_{i+1} \dots \rangle, \quad (1.73)$$

$$\Phi_i | \dots v_{i-1} 11 v_{i+1} \dots \rangle = 0, \quad (1.74)$$

where $z_i \in \{1, \dots, k\}$ is the label of the vertex at the i th step and $\lambda_j = \sin(j\theta)$ with $\theta = \pi/k$. The representation $\tilde{\rho}(\sigma_i) = A\Phi_i + A^{-1}I$ for the braid group B_n is induced by ρ . It is easy to confirm that, with $A = ie^{-i\theta/2}$ and $d = 2 \cos \theta$, the representation ρ is hermitian for all E_i , and hence the induced representation $\tilde{\rho}$ is unitary. Since $\rho(\sigma_i)$ is a unitary operator acting on the neighboring two-qubit, a multiplication of such unitary operators can be implemented by using universal quantum computer. Since, by using the Hadamard test in Sect. 1.5.1, we can evaluate the Jones polynomial $V_{b^{\text{mir}}}(A^{-4})$ with an additive error $\Delta 2^n \varepsilon$ taking $\text{poly}(n, m, k, 1/\varepsilon)$ overhead, where m is the number of the crossing.

The approximation scale is further improved by considering another closure, the so-called plat closure as shown in Fig. 1.4b. We deform the link diagram obtained by the plat closure to another link diagram obtained by the Markov trace closure as shown in Fig. 1.4c. Then, we have \cup and \cap for all strands. In this case, the support of $\rho(b)$ is only on $|10101\dots\rangle$, which allows us to replace the matrix trace as follows:

$$V_{b^{\text{plt}}}(A^{-4}) = \Delta_{\text{plt}} \langle 10101\dots | \rho(b) | 10101\dots \rangle, \quad (1.75)$$

where b^{plt} is a link generated from the braiding b with the plat closure and $\Delta_{\text{plt}} = (-A)^{3\omega(b^{\text{plt}})} d^{n/2-1}$. (Note that for each $\rho(E_i)$ of $n/2$ \cup s and \cap s, a factor $\lambda_2/\lambda_1 = d$ is substituted.) Again using the Hadamard test, we can estimate $V_{b^{\text{plt}}}(A^{-4})$ with an additive error $\Delta_{\text{plt}} \varepsilon$ taking $\text{poly}(n, m, k, 1/\varepsilon)$ overhead.

Finally, we show that the approximation of the Jones polynomial with an additive error $\Delta_{\text{plt}} \varepsilon$ is BQP-hard. Unfortunately, the computational basis defined by the path $p = (v_1, \dots, v_n)$ is inappropriate for this purpose, since the n -step walks cannot span the whole 2^n -dimensional Hilbert space. Instead, we employ the four-step encoding [33], where the computational basis states are defined by the following two four-steps:

$$|\bar{0}\rangle \equiv |1100\rangle, \quad |\bar{1}\rangle \equiv |1010\rangle. \quad (1.76)$$

The space $\mathcal{H}_{4n,k}$ spanned by the $4n$ -step walks can support the whole 2^n -dimensional space for $\{|\bar{0}\rangle, |\bar{1}\rangle\}^{\otimes n}$. The dimension of the space spanned by the 8-step walks starting from vertex 1 ending at vertex 1 is 14. Thus the unitary representation ρ of the braid group B_8 results in unitary operators in $SU(14)$. If they are dense in $SU(14)$, we can approximate an arbitrary unitary operator in $SU(4)$ spanned by two four-step encoded qubits by using the Solovay–Kitaev algorithm [16], which are enough to implement universal quantum computation. The density is achieved by $k > 4$ and $k \neq 6$. For such a parameter, the approximation of the Jones polynomial with an additive error $\Delta_{\text{plt}} \varepsilon$ is enough to solve a BQP-complete problem.

The approximation of the Jones polynomial can be done by universal quantum computer and also is enough to solve the problems solvable by universal quantum computer. Thus the approximation of the Jones polynomial is a BQP-complete problem [15, 16]. The AJL algorithm for approximation of the Jones polynomial was extended for the Tutte polynomial in Ref. [18], where the Solovay–Kitaev algorithm for non-unitary linear operators was developed.

1.6 Quantum Noise

Quantum coherence, one of the essential properties of quantum systems, is quite fragile against noise, due to interactions between the system and the environment. Suppose that the system S of interest interacts via a unitary operation U with the environment E , where the system and environment are initially uncorrelated. The reduced density matrix ρ'_S of the system after the interaction is calculated to be

$$\rho'_S = \text{Tr}_E [U(\rho_S \otimes \rho_E)U^\dagger]. \quad (1.77)$$

Using a spectral decomposition of the initial state in the environment, $\rho_E = \sum_k p_k |e_k\rangle\langle e_k|$, we obtain a map of the system S :

$$\rho'_S = \sum_{(k',k)} K_{(k',k)} \rho_S K_{(k',k)}^\dagger, \quad (1.78)$$

where

$$K_{(k,k')} = \sqrt{p_k} \langle e'_k | U | e_k \rangle. \quad (1.79)$$

This map satisfies

$$\begin{aligned} \sum_k K_{(k',k)}^\dagger K_{(k',k)} &= p_k \langle e_k | U^\dagger | e'_k \rangle \langle e'_k | U^\dagger | e_k \rangle \\ &= I_S, \end{aligned} \quad (1.80)$$

where I_S is the identity operator in the system S .

In general, a map of a quantum state is given as a completely-positive-trace-preserving (CPTP) map \mathcal{E} , subject to

- $\text{Tr}[\mathcal{E}\rho] = 1$ for any density matrix ρ (preservation of probability)
- $\mathcal{E}(\sum_i q_i \rho_i) = \sum_i q_i \mathcal{E}\rho_i$ (convex linear map)
- $\mathcal{I}_A \otimes \mathcal{E}_S(\rho_{AS}) \geq 0$ (complete positivity) for an arbitrary ancilla system A and density matrix ρ_{AS} on the composite system AS .

Such a CPTP map can always be written by Kraus operators $\{K_k\}$ [9]:

$$\mathcal{E}\rho = \sum_{k=1}^M K_k \rho K_k^\dagger. \quad (1.81)$$

Under the Born-Markov (with rotating-wave) approximation, the time evolution of a two-level system coupled with an environment is given by a master equation of the Lindblad form [34]:

$$\begin{aligned} \dot{\rho}(t) = & -\frac{\gamma_+}{2} [\sigma_- \sigma_+ \rho(t) + \rho(t) \sigma_- \sigma_+ - 2\sigma_+ \rho(t) \sigma_-] \\ & -\frac{\gamma_-}{2} [\sigma_+ \sigma_- \rho(t) + \rho(t) \sigma_+ \sigma_- - 2\sigma_- \rho(t) \sigma_+] \\ & -\frac{\gamma_0}{2} [\sigma_z \rho(t) + \rho(t) \sigma_z - 2\sigma_z \rho(t) \sigma_z] \equiv \mathcal{L}\rho(t), \end{aligned} \quad (1.82)$$

where $\sigma_+ = |0\rangle\langle 1|$, $\sigma_- = |1\rangle\langle 0|$ and γ_α ($\alpha = 0, +, -$) are the decay rates of the decay channels. One can easily find the eigenoperators of the Lindblad superoperator. These eigenoperators form the damping basis [35]:

$$\mathcal{L}\sigma_1 = \frac{\gamma_+ + \gamma_- + 2\gamma_0}{2} \sigma_1 \equiv \lambda_1 \sigma_1, \quad (1.83)$$

$$\mathcal{L}\sigma_2 = \frac{\gamma_+ + \gamma_- + 2\gamma_0}{2} \sigma_2 \equiv \lambda_2 \sigma_2, \quad (1.84)$$

$$\mathcal{L}\sigma_3 = (\gamma_+ + \gamma_-) \sigma_3 \equiv \lambda_3 \sigma_3, \quad (1.85)$$

$$\mathcal{L}\rho_{eq} = 0, \quad (1.86)$$

where $\rho_{eq} = (\gamma_+ |0\rangle\langle 0| + \gamma_- |1\rangle\langle 1|) / (\gamma_+ + \gamma_-) \equiv (\sigma_0 + a\sigma_3) / 2$ with $a = (\gamma_+ - \gamma_-) / (2\gamma_+ + 2\gamma_-)$. The solution of this master equation is given by the CPTP map $\mathcal{E}(t)$:

$$\mathcal{E}(t)\rho = p_0(t)\rho + \sum_{i=1,2,3} p_i(t)\sigma_i \rho \sigma_i + f(t)(\sigma_3 \rho + \rho \sigma_3 - i\sigma_1 \rho \sigma_2 + i\sigma_2 \rho \sigma_1), \quad (1.87)$$

where

$$p_0(t) = \frac{1}{4}(1 + e^{-\lambda_1 t} + e^{-\lambda_2 t} + e^{-\lambda_3 t}), \quad (1.88)$$

$$p_1(t) = \frac{1}{4}(1 + e^{-\lambda_1 t} - e^{-\lambda_2 t} - e^{-\lambda_3 t}), \quad (1.89)$$

$$p_2(t) = \frac{1}{4}(1 - e^{-\lambda_1 t} + e^{-\lambda_2 t} - e^{-\lambda_3 t}), \quad (1.90)$$

$$p_3(t) = \frac{1}{4}(1 - e^{-\lambda_1 t} - e^{-\lambda_2 t} + e^{-\lambda_3 t}), \quad (1.91)$$

$$f(t) = \frac{a}{4}(1 - e^{-\lambda_3 t}). \quad (1.92)$$

If we consider a high temperature case (i.e., $a \rightarrow 0$), Eq. (1.87) can be rewritten as

$$\mathcal{E}(t)\rho = [1 - p_1(t) - p_2(t) - p_3(t)]\rho + \sum_{i=1}^3 p_i(t)\sigma_i\rho\sigma_i. \quad (1.93)$$

Hence, the CPTP map can be viewed as a stochastic Pauli error with probabilities $p_i(t)$. In general, noise cannot be written by a stochastic Pauli error. However, by performing an appropriate operation, one can depolarize the CPTP map into a stochastic Pauli error as a standard form, where the noiseless part of the evolution is not altered [36]. Otherwise, the Pauli basis measurements can collapse the CPTP map into a stochastic Pauli error, as we will see later. In the rest of this book, we assume that the noise is given by a Markovian stochastic Pauli error, for simplicity.

1.7 Summary and Discussion

In this chapter, we have seen key ingredients behind quantum computation, such as universal quantum computation, quantum algorithms, and quantum noise. More detailed descriptions of universal quantum computation including foundations of quantum mechanics and computer scientific concepts can be found in Chaps. 2–4 of the book [9]. While I cannot mention, the quantum search algorithm [37] is one of the most important quantum algorithms, which provides quantum speedup in quite generic situations and hence has many applications (see for example [9]). More physical aspects of the quantum algorithm for the Jones polynomials can be found in the book [38]. In Ref. [18], the quantum algorithm for the Jones polynomials is extended to the Tutte polynomials.

The stochastic Pauli noise substantially simplifies the argument for quantum error correction and a fault-tolerant design. However, in actual experiments, the noise would not be given by the stochastic Pauli noise. Even in such a case, we can analyze fault-tolerance without changing the hardware design, while we have to take more care to evaluate the amount of noise per gate. The fault-tolerance against more general CPTP noise has been discussed in Refs. [39–43], where the noise is characterized by a kind of distance, super-operator norm, from the ideal operation. Moreover, fault-tolerance against the non-Markovian noise has been analyzed in Refs. [41, 44, 45], where the noise was characterized by the operator norm of the system-environment interaction Hamiltonian. If the decoherence is non-Markovian, we might take a hardware approach to suppress such an effect; the dynamical decoupling or quantum Zeno effect can also be used [46–51]. Besides, if the decoherence is spatially correlated, one can utilize a passive error-prevention scheme, the so-called *decoherence free subspace* (DFS), which is immune to collective noise [52–54].

Chapter 2

Stabilizer Formalism and Its Applications

In general, the description of quantum states is a difficult task because it requires exponentially many parameters in the number of qubits as shown in Eq. (1.38). To understand these complex quantum systems, it is essential to have efficient tools. The stabilizer formalism is one such powerful tool to describe an important class of entangled states. It also provides a diagrammatic understanding of quantum states and operations. The stabilizer states, described by the stabilizer formalism, play important roles in quantum computation, such as for quantum error correction codes and resource states in MBQC. In this chapter, we introduce the stabilizer formalism, especially focusing on its diagrammatic understanding. Based on the stabilizer formalism, we explain quantum error correction, magic state distillation, and MBQC.

2.1 Stabilizer Formalism

We first define an n -qubit Pauli group \mathcal{P}_n :

$$\mathcal{P}_n := \{\pm 1, \pm i\} \times \{I, X, Y, Z\}^{\otimes n}. \quad (2.1)$$

An element of the Pauli group is called a Pauli product. For example, the two-qubit Pauli group is given by

$$\begin{aligned} \mathcal{P}_2 := & \{\pm 1, \pm i\} \\ & \times \{II, IX, IY, IZ, XI, XX, XY, XZ, YI, YX, YY, YZ, ZI, ZX, ZY, ZZ\}, \end{aligned} \quad (2.2)$$

where $A \otimes B$ is denoted by AB for simplicity. (We will frequently use this notation when there is no possibility for confusion.) Next, we define an n -qubit stabilizer

group \mathcal{S} as an Abelian (commutative) subgroup of the n -qubit Pauli group:

$$\mathcal{S} := \{S_i\} \text{ s.t. } -I \notin \mathcal{S} \text{ and } \forall S_i, S_j \in \mathcal{S}, [S_i, S_j] = 0. \quad (2.3)$$

Because $-I$ is not included in the stabilizer group, all elements are hermitian $S_i = S_i^\dagger$, which guarantees that the eigenvalues $= \pm 1$. An element of the stabilizer group is called a stabilizer operator. The maximum independent subset \mathcal{S}_g of the stabilizer group is called stabilizer generators. Here, independence means that any element of \mathcal{S}_g cannot be expressed as a product of other elements in \mathcal{S}_g . Any element of the stabilizer group can be generated as a product of the stabilizer generators. The stabilizer group \mathcal{S} generated by the generators \mathcal{S}_g is denoted by $\mathcal{S} = \langle \mathcal{S}_g \rangle$.

Let us, for example, consider a two-qubit stabilizer group:

$$\mathcal{S}_{\text{Bell}} = \{II, XX, ZZ, -YY\}. \quad (2.4)$$

Because they contain two anticommuting Pauli operators, XX and ZZ commutes. The stabilizer group $\mathcal{S}_{\text{Bell}}$ is generated by $\{XX, ZZ\}$, because $-YY$ can be expressed as a product of XX and ZZ . Thus, we can write $\mathcal{S}_{\text{Bell}} = \langle \{XX, ZZ\} \rangle$.

For a given stabilizer group \mathcal{S} , the stabilizer state is defined as a simultaneous eigenstate of all stabilizer elements $S_i \in \mathcal{S}$ with the eigenvalue $+1$:

$$\forall S_i \in \mathcal{S}, \quad S_i |\psi\rangle = |\psi\rangle. \quad (2.5)$$

It is sufficient that the state is an eigenstate of all stabilizer generators:

$$\forall S_i \in \mathcal{S}_g, \quad S_i |\psi\rangle = |\psi\rangle. \quad (2.6)$$

Let k be the number of elements in the stabilizer generator \mathcal{S}_g . Each stabilizer generator divides an n -qubit system (Hilbert space) into two orthogonal subspaces associated with the eigenvalues ± 1 . Because all stabilizer operators commute with each other, the k stabilizer generators divide the n -qubit system into 2^k orthogonal subspaces. Thus, the dimension of the space spanned by the stabilizer states, which we call a *stabilizer subspace*, is $2^d = 2^{n-k}$. When $n = k$, we can define the quantum state uniquely. The number of stabilizer generators is at most n for an n -qubit stabilizer group. In the case of $k < n$, the degrees of freedom in the stabilizer subspace can be addressed by using *logical operators*, which commute with all stabilizer generators and also are independent of them.

Let us consider the stabilizer group $\mathcal{S}_{\text{Bell}}$ again. The stabilizer state is the eigenstate of XX and ZZ with eigenvalue $+1$, and hence given by the Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$ [55]. If XX is removed from the generators, the two-dimensional subspace spanned by $|00\rangle$ and $|11\rangle$ is stabilized. By choosing logical operators $L_X = XX$ and $L_Z = ZZ$, we can specify the state in the subspace. For example, the eigenstate of L_X with the eigenvalue $+1$ is the Bell state. The eigenstate of L_Z

with the eigenvalue $+1$ is $|00\rangle$. Another representative example of the stabilizer states is an n -qubit cat state,

$$|\text{cat}\rangle = \frac{1}{\sqrt{2}}(|00\dots 0\rangle + |11\dots 1\rangle), \quad (2.7)$$

whose stabilizer group is given by

$$\left\langle Z_1 Z_2, \dots, Z_{n-1} Z_n, \prod_{i=1}^n X_i \right\rangle. \quad (2.8)$$

The cat state is a representative example of a macroscopically entangled state. If it is determined whether a particle is $|0\rangle$ or $|1\rangle$, the superposition is completely destroyed. If an element $\prod_{i=1}^n X_i$ is removed from the stabilizer generator, it defines a stabilizer subspace spanned by $|00\dots 0\rangle$ and $|11\dots 1\rangle$. We can choose $L_X = \prod_{i=1}^n X_i$ and $L_Z = Z_i$ as logical operators, which anti-commute with each other and behave as logical Pauli operators.

2.2 Clifford Operations

In the stabilizer formalism, we can describe a restricted class of unitary operations, the so-called Clifford operations, acting on the stabilizer states quite efficiently. The Clifford operation is defined as an operation U that transforms a Pauli product into another Pauli product under its conjugation, $[\dots] \rightarrow U[\dots]U^\dagger$. Let us consider the action of a Clifford operation U on the stabilizer state $|\psi\rangle$ defined by a stabilizer group $\mathcal{S} = \langle \{S_i\} \rangle$:

$$U|\psi\rangle = US_i|\psi\rangle = US_iU^\dagger U|\psi\rangle = S'_i U|\psi\rangle, \quad (2.9)$$

where we define $S'_i \equiv US_iU^\dagger$. The above equality indicates that the state $U|\psi\rangle$ is an eigenstate of the operator S'_i with an eigenvalue $+1$ for all S'_i . Because U is a Clifford (unitary) operation, the group $\{S'_i\}$ is also an Abelian subgroup of the Pauli group. Accordingly, the state $U|\psi\rangle$ is a stabilizer state with respect to the stabilizer group $\{S'_i\}$. In this way, the action of U on the stabilizer state can be represented as a transformation of the stabilizer groups under the conjugation of U as shown in Fig. 2.1. For example, the stabilizer state stabilized by $\langle X_1 I_2, I_1 Z_2 \rangle$ is $|+\rangle_1 |0\rangle_2$. The stabilizer group is transformed by $\Lambda(X)_{1,2}$ into $\langle X_1 X_2, Z_1 Z_2 \rangle$, whose stabilizer state is $(|00\rangle + |11\rangle)/\sqrt{2}$.

The stabilizer formalism corresponds to the Heisenberg picture of quantum computation, where a minimum number of operators are employed to describe a restricted type of quantum states and operations [56, 57]. This representation is powerful because it requires us to keep a time evolution of at most n operators, while a straightforward state-based approach needs exponentially many states. For example, let us consider the following quantum circuit:

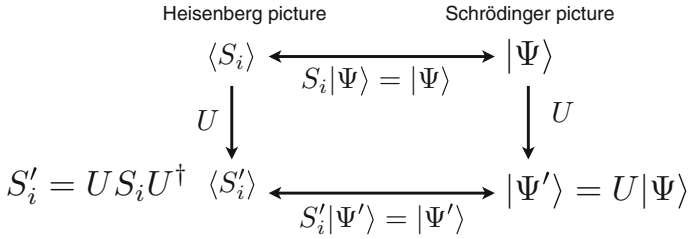
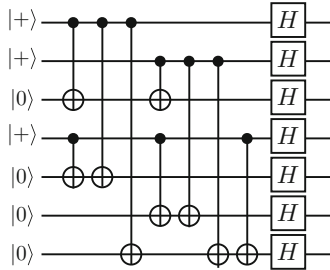


Fig. 2.1 The stabilizer formalism: a Heisenberg picture of quantum computation. A Clifford operation is represented as a transformation of the stabilizer group by the conjugation of U



A straightforward calculation yields the output state $|\psi\rangle$,

$$\begin{aligned}
 |\psi\rangle = & (|0000000\rangle + |1010101\rangle + |0110011\rangle + |1100110\rangle \\
 & + |0001111\rangle + |1011010\rangle + |0111100\rangle + |1101001\rangle \\
 & + |1111111\rangle + |0101010\rangle + |1001100\rangle + |0011001\rangle \\
 & + |1110000\rangle + |0100101\rangle + |1000011\rangle + |0010110\rangle)/4.
 \end{aligned}
 \tag{2.10}$$

It is rather cumbersome to write down the above state. Instead, we can understand the output state as a stabilizer state whose stabilizer generators are

$$\{ZIZIZIZ, IZZIIZZ, IIIZZZZ, XXXIIII, \\
 XXIXXII, IXIXIXI, XIIXIIX\}.
 \tag{2.11}$$

Equivalently, we may also choose the following stabilizer generators because they generate the same stabilizer group:

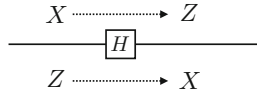
$$\{ZIZIZIZ, IZZIIZZ, IIIZZZZ, XXXXXXX, \\
 IIIXXXX, XIXIXIX, IXXIXXI\}.
 \tag{2.12}$$

Actually, these stabilizer generators are enough to understand the properties of the quantum state $|\psi\rangle$. If an explicit description of the state is required, we can systematically write it down as follows:

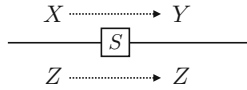
$$|\psi\rangle = 4 \frac{I + S_4}{2} \frac{I + S_3}{2} \frac{I + S_2}{2} \frac{I + S_1}{2} |0000000\rangle, \tag{2.13}$$

where $S_1 = XIXIXIX$, $S_2 = IXXIIXX$, $S_3 = IIIXXXX$, and $S_4 = XXXXXXX$. The above equation means that $|0000000\rangle$ is an eigenstate for all Z 's stabilizer operators. By projecting it into the $+1$ eigenstate of the stabilizer generator S_i by the projection $\frac{I+S_i}{2}$, we obtain the stabilizer state $|\psi\rangle$.

In order for the above calculation to work, we have to obtain the stabilizer generators of the output state. This can easily be done graphically. We introduce commutation rules between the Pauli operators and Clifford operations below. In the case of the Hadamard operation, $HX = ZH$ and $ZH = HX$, and hence we have



meaning that the Pauli X operator acting before the Hadamard operation is equivalent to the Pauli Z operator acting after the Hadamard operation and so on. Similarly, for the phase operation S , we have



The CNOT operation transforms the Pauli operators under its conjugation as follows:

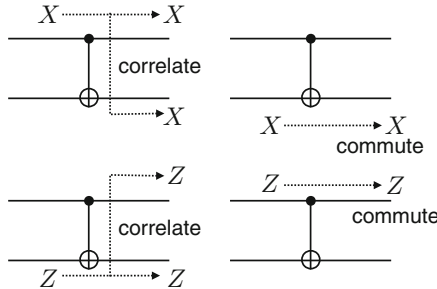
$$\Lambda_{c,t}(X)X_c\Lambda_{c,t}(X) = X_cX_t, \tag{2.14}$$

$$\Lambda_{c,t}(X)X_t\Lambda_{c,t}(X) = X_t, \tag{2.15}$$

$$\Lambda_{c,t}(X)Z_c\Lambda_{c,t}(X) = Z_c, \tag{2.16}$$

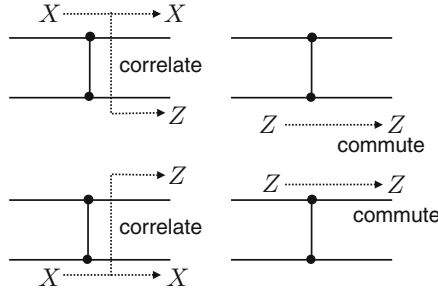
$$\Lambda_{c,t}(X)Z_t\Lambda_{c,t}(X) = Z_cZ_t. \tag{2.17}$$

The commutation relation between the CNOT operation and the Pauli operators is understood as follows:



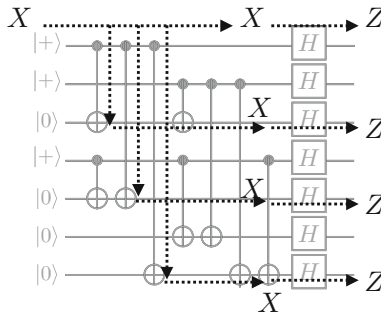
In the above circuit diagram, the solid circle commutes with the Pauli Z operator, while the Pauli X operator is propagated as the Pauli X operator on the target qubit, making a correlation. Similarly, the open circle commutes with the Pauli X operator,

while the Pauli Z operator is propagated as the Pauli Z operator on the control qubit, making a correlation. By recalling that the CNOT operation is transformed into the CZ operation by the Hadamard operations on the target qubit, the commutation relation between the CZ operation and the Pauli operators are obtained straightforwardly. This is described graphically as follows:



In this case, note that the Pauli X operation is propagated as the Pauli Z operation.

This graphical understanding allows us to calculate the stabilizer generators of the output of the Clifford circuits. For example, in the following circuit diagram, the first qubit is stabilized by X before the Clifford operation. The Pauli X operator is propagated toward the right, and we obtain the stabilizer operator $ZIZIZIZ$ for the output:



The reader should use this graphical technique to calculate the other stabilizer generators and verify Eq. (2.11).

2.3 Pauli Basis Measurements

Next, we will see how the Pauli-basis measurements on the stabilizer states are described in the stabilizer formalism. Suppose the A -basis ($A = X, Y, Z$) measurement is performed on a stabilizer state $|\psi\rangle$, whose stabilizer group is given by $\langle S_i \rangle$. (We assume that the number of stabilizer generators is equal to the number of qubits, and hence that the stabilizer state is uniquely defined.) Depending on the stabilizer group $\langle \{S_i\} \rangle$ and A , there are two possibilities:

- (i) The Pauli operator A commutes with all stabilizer generators. In that case, either A or $-A$ is an element of the stabilizer group. If A ($-A$) is an element, the eigenvalue $+1$ (-1) is obtained with probability 1. The post-measurement state is the same as the stabilizer state before measurement.
- (ii) At least one stabilizer operator does not commute with A . In this case, we can choose another set of generators $\{S'_i\}$ such that S'_1 anti-commutes with A but all other generators commute with A . The measurement outcomes $+1$ and -1 are obtained with an equal probability of $1/2$. The post-measurement state is given by $\langle (-1)^m A, S'_2, \dots, S'_k \rangle$ depending on the measurement outcomes $m = 0, 1$ corresponding to the eigenvalues $(-1)^m$.

For example, suppose we perform the Y -basis measurement on the first qubit of the Bell state stabilized by $\mathcal{S}_{\text{Bell}} = \langle XX, ZZ \rangle$. We can redefine the stabilizer generators by $\{XX, -YY\}$. The measurement outcome m is randomly given. Then the stabilizer group after the measurement is given by $\langle (-1)^m YI, -YY \rangle = \langle YI, (-1)^{m+1} IY \rangle$. Thus, we obtain $| - i \rangle$ as the post-measurement state on the second qubit.

2.4 Gottesman–Knill Theorem

Because the stabilizer states and Clifford operations are described efficiently in the stabilizer formalism, it implies that such a restricted type of quantum computation can be simulated efficiently on a classical computer. This is stated by the Gottesman–Knill theorem [9, 56, 57].

Theorem 2.1 *Any Clifford operations, applied to the input state $|0\rangle^{\otimes n}$ followed by the Z measurements, can be simulated efficiently in the strong sense.*

Here, an efficient *strong* classical simulation of a quantum circuit C is a classical polynomial-time computation that calculates the probability $P_C(x)$ for a given output x of the circuit C , including an arbitrary marginal distribution $\sum_{x'} P_C(x')$. (See, for example, Ref. [58] for the definition of a strong simulation.) Note that this theorem holds true even when the initial state is generalized to an arbitrary stabilizer state, and also any Pauli products are measured, because they are done in the above setup by modifying the Clifford operations appropriately.

Proof The stabilizer group of the input state is $\langle \{Z_i\} \rangle$ ($i = 0, 1, \dots, n - 1$). By applying the Clifford operations as mentioned, we obtain the stabilizer generators $\langle \{S_i\} \rangle$ of the quantum output before the measurements. Suppose the measurement outcome, the classical output, is given by $\{m_i = 0, 1\}$. Then the probability of obtaining the measurement outcome $\{m_i\}$ can be calculated as follows:

- (i) Set the stabilizer generators $\mathcal{S}^{(0)} = \langle \{S_i\} \rangle$ and the initial probability $p^{(0)} = 1$.
- (ii) For $k = 0, 1, \dots, n - 1$, repeat the following procedures.
 - (1) If $(-1)^{m_k} Z_k \in \mathcal{S}^{(k)}$, update the probability $p^{(k+1)} = p^{(k)}$, because the measurement outcome m_k is obtained with probability 1. The stabilizer group after the measurement is also updated to $\mathcal{S}^{(k+1)} = \mathcal{S}^{(k)}$.

- (2) Else, if $(-1)^{m_k \oplus 1} Z_k \in \mathcal{S}^{(k)}$, update the probability $p^{(k+1)} = 0$, because such a measurement outcome does not appear. (You may stop the calculation at this stage, and return the probability 0.)
- (3) Else, $\mathcal{S}^{(k)}$ is updated into $\mathcal{S}^{(k+1)}$ by removing an anticommuting generator and adding $(-1)^{m_k} Z_k$ as a new generator. Because the measurement outcome is obtained randomly with probability 1/2, the probability is taken as $p^{(k+1)} = p^{(k)}/2$.

(iii) Return $p^{(n)}$ as the probability of obtaining the measurement outcome $\{m_i\}$.

Note that, in step (ii), we can efficiently decide which of the three is the case for any k by checking the commutability of Z_k with the stabilizer generators of $\mathcal{S}^{(k)}$. ■

The statement of Theorem 2.1 can be extended by weakening the notion of the classical simulation.

Theorem 2.2 *Any Clifford operations, applied to any product states of convex mixtures of the Pauli basis states, followed by Z measurements can be efficiently simulated in the weak sense.*

Here, an efficient *weak* classical simulation of a quantum circuit C is a classical polynomial-time randomized computation that samples the output x according to the probability distribution $P_C(x)$ of the output of the circuit C . (See, for example, Ref. [58] for the definition of weak simulation.) Apparently, a strong simulation includes a weak simulation, because we sample the output by using the marginal distributions [59].

Proof Suppose that the i th input qubit is given by

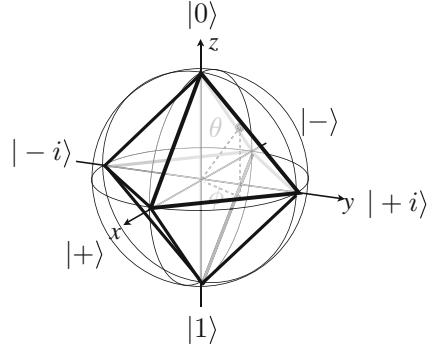
$$\begin{aligned} \rho_i = & p_{x,+}^{(i)} |+\rangle\langle +| + p_{x,-}^{(i)} |-\rangle\langle -| + p_{y,+}^{(i)} |+\rangle\langle +| + p_{y,-}^{(i)} |-\rangle\langle -| \\ & + p_{z,+}^{(i)} |0\rangle\langle 0| + p_{z,-}^{(i)} |1\rangle\langle 1|, \end{aligned} \quad (2.18)$$

where $\sum_{\alpha=x,y,z} \sum_{v=+,-} p_{\alpha,v}^{(i)} = 1$. By using the probability distribution $\{p_{\alpha,v}^{(i)}\}$, the input state of each qubit is randomly sampled. Conditioned by the sampling result, the input state is a product of the Pauli basis states, and hence the output probability distribution can be calculated as shown in Theorem 2.1. Combined with the random sampling of the input state, this provides an efficient weak simulation of the Clifford circuit with noisy input states (convex mixture of the Pauli basis states). ■

The input state can be generalized into a classical mixture of stabilizer states, when its polynomial size description of the probability distribution is provided. Similarly, the Clifford operations can be extended to stochastic Clifford operations such as the stochastic Pauli error.

The convex mixture of the Pauli basis state lies inside the octahedron of the Bloch sphere as shown in Fig. 2.2. It is natural to ask whether or not the Clifford circuit allows universal quantum computation if the input state lies outside the octahedron. If the input state is a pure non-stabilizer state such as $e^{-i(\pi/8)Z}|+\rangle$, we can implement a non-Clifford gate $e^{-i(\pi/8)Z}$ by using gate teleportation, explained in Sect. 2.6. Even

Fig. 2.2 A convex mixture of the Pauli basis states lies inside the octahedron of the Bloch sphere



some mixed states can be converted into a pure non-stabilizer state, the so-called magic state, by using only Clifford operations. Such a protocol is called magic state distillation [60] and will be explained in Sect. 2.8.

2.5 Graph States

In this section, we introduce an important class of stabilizer states, the so-called *graph states* [61], whose stabilizer generators are defined on graphs. The graph states are employed as resource states for MBQC as explained in the next section.

A graph state is defined by a graph $G = (V, E)$. Here, V and E are the sets of the vertices and edges, respectively. A qubit is located on each vertex of the graph. The stabilizer generator of the graph state $|G\rangle$ is defined as

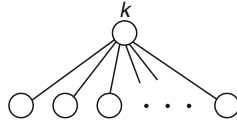
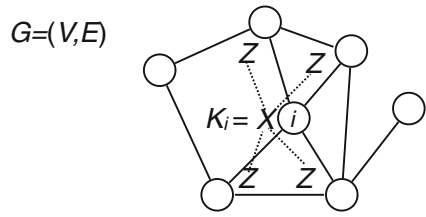
$$K_i = X_i \prod_{j \in V_i} Z_j \quad \text{for all } i \in V, \quad (2.19)$$

where we define a set of vertices $V_i := \{j | (i, j) \in E\}$, which are connected to the vertex i by an edge on the graph G (see Fig. 2.3). The graph state $|G\rangle$ is generated from a product state $|+\rangle^{\otimes |V|}$ by applying the CZ gate on each of the graphs:

$$|G\rangle = \prod_{(i,j) \in E} \Lambda(Z)_{i,j} |+\rangle^{\otimes |V|}, \quad (2.20)$$

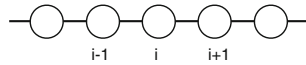
where $|V|$ indicates the number of vertices of the graph $G = (V, E)$. This can be understood that the stabilizer generator X_i for the state $|+\rangle$ is transformed into K_i by the CZ operations $U \equiv \prod_{(i,j) \in E} \Lambda(Z)_{i,j}$. Especially, when the graphs are regular lattices such as one-dimensional (1D), square, hexagonal, and cubic lattices, the corresponding graph states tend to be referred to as cluster states [62]. Any stabilizer state is equivalent to a certain graph state up to local Clifford operations [61, 63]. For example, the cat state is equivalent to the following graph state by applying the Hadamard operation on the k th qubit:

Fig. 2.3 The graph state $|G\rangle$ associated with a graph $G = (V, E)$. A stabilizer generator K_i is also shown



Unfortunately, the graph associated with a stabilizer state is not uniquely defined, because there are local Clifford operations that change the underlying graph. This property is called the local complementarity of the graph states [61, 63].

Next, we will see how the Pauli basis measurements transform the graph states. For simplicity, we assume that the state is projected into an eigenstate with eigenvalue $+1$. Let us consider a 1D graph state as follows:



whose stabilizer generator is given by

$$K_i = Z_{i-1}X_iZ_{i+1}. \tag{2.21}$$

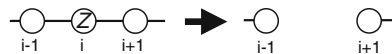
We first consider the Z basis measurement (projective measurement of the observable Z) on the i th qubit. Following the procedure seen in Sect. 2.3, K_i is removed from the stabilizer generator. By adding Z_i instead, we obtain the stabilizer group for the post-measurement state

$$\langle \dots, K_{i-1}, Z_i, K_{i+1}, \dots \rangle. \tag{2.22}$$

After the projection, the i th qubit is $|0\rangle$ and hence decoupled from the other qubits. By rewriting the stabilizer generators, we obtain three decoupled stabilizer groups

$$\langle \dots, Z_{i-2}X_{i-1} \rangle, \langle Z_i \rangle, \langle X_{i+1}Z_{i+2}, \dots \rangle. \tag{2.23}$$

This means that the graph is divided into two parts as follows:



For any graph, this property of the Z -basis measurement holds; the post-measurement state is defined by a modified graph, where the vertex corresponding to the measured qubit and the edges incident to it are removed from the original graph.

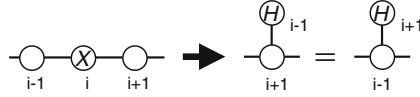
Next, we consider the X -basis measurement. The observable X_i does not commute with K_{i-1} and K_{i+1} , but does commute with $K_{i-1}K_{i+1} = Z_{i-2}X_{i-1}X_{i+1}Z_{i+2}$. Following the procedure in Sect. 2.3, the stabilizer group for the post-measurement state is calculated to be

$$\langle \dots, Z_{i-2}X_{i-1}X_{i+1}Z_{i+2}, Z_{i-1}Z_{i+1}, \dots \rangle, \langle X_i \rangle. \quad (2.24)$$

By performing the Hadamard operation H on the $(i-1)$ th qubit, we obtain a new stabilizer group

$$\langle \dots, Z_{i-2}Z_{i-1}X_{i+1}Z_{i+2}, X_{i-1}Z_{i+1}, \dots \rangle, \langle X_i \rangle, \quad (2.25)$$

which indicates that the graph is transformed into the following graph with the Hadamard operation:



Instead of the $(i-1)$ th qubit, we can obtain a similar result by performing the Hadamard operation on the $(i+1)$ th qubit as shown above.

Suppose the i th and $(i+1)$ th qubits are measured in the X -basis on a 1D graph state. This is equivalent to measuring the $(i+1)$ th qubit of the above post-measurement graph state in the Z basis, because the Hadamard operation is applied on it as a byproduct. From the previous argument, the Z -basis measurement remove the measured qubit from the graph. Thus, two neighboring X -basis measurements remove the measured qubits and connect the left and right hand sides directly:



which we call a contraction.

Finally, we consider the Y -basis measurement. The observable Y_i does not commute with either K_{i-1} , K_i , or K_{i+1} , but does commute with $K_{i-1}K_i = Z_{i-2}Y_{i-1}Y_iZ_{i+1}$ and $K_iK_{i+1} = Z_{i-1}Y_iY_{i+1}Z_{i+2}$. The stabilizer group for the post-measurement state is calculated to be

$$\langle \dots, Z_{i-2}Y_{i-1}Z_{i+1}, Z_{i-1}Y_{i+1}Z_{i+2}, \dots \rangle, \langle Y_i \rangle. \quad (2.26)$$

By performing the phase gates S on the $(i-1)$ th and $(i+1)$ th qubits, we obtain a new stabilizer group

$$\langle \dots, Z_{i-2}X_{i-1}Z_{i+1}, Z_{i-1}X_{i+1}Z_{i+2}, \dots \rangle, \langle Y_i \rangle. \quad (2.27)$$

This indicates that the graph is directly connected up to the phase operation S as a byproduct:



Suppose three neighboring qubits $(i-1)$, i , and $(i+1)$ are measured in the Y -basis. This is equivalent to measuring the i th qubit in the Y -basis, and then measuring the $(i-1)$ th and $(i+1)$ th qubits of the post-measurement graph state in the X -basis, because there is a phase operation S acting on them as a product. As seen previously, the X -basis measurements on two neighboring qubits result in a contraction of the two qubits on the graph. Thus, the Y -basis measurements on three neighboring qubits contract them from the 1D graph state.



This property is useful to change even and odd of the length of the 1D graph state.

While we have considered the Pauli-basis measurements only on the 1D graph state, we can generalize these arguments into graph states of general structures. A graph state is still mapped into another graph state up to some single-qubit Clifford operations as byproducts.

2.6 Measurement-Based Quantum Computation

Measurement-based quantum computation (MBQC) is a model of quantum computation, where quantum gates are implemented by adoptive measurements on a highly entangled resource state [64–66]. Specifically, certain graph states, the so-called cluster states, are employed as resource states in MBQC. Below we will first demonstrate quantum teleportation, a building block of MBQC. Then, we explain how adoptive measurements on a graph state enable us to emulate universal quantum computation via quantum teleportation.

Quantum teleportation is a quantum communication protocol, in which Alice sends a quantum state to Bob by using a shared entangled state and classical communication [67]. Suppose Alice and Bob share a maximally entangled state, the Bell state,

$$\frac{|0\rangle_a|0\rangle_b + |1\rangle_a|1\rangle_b}{\sqrt{2}}. \quad (2.28)$$

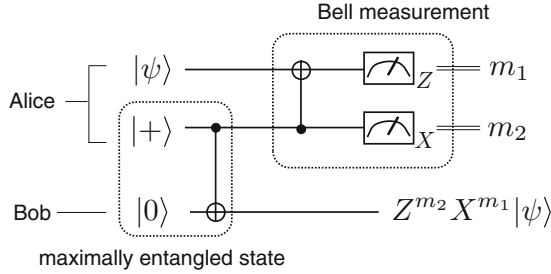
For an unknown input state $|\psi\rangle_i$ and the half of the Bell state, Alice performs a Bell measurement, which is a projection onto the Bell basis states

$$|\Psi(m_1, m_2)\rangle_{i,a} = X_i^{m_1} Z_i^{m_2} \frac{|0\rangle_i|0\rangle_a + |1\rangle_i|1\rangle_a}{\sqrt{2}}, \quad (2.29)$$

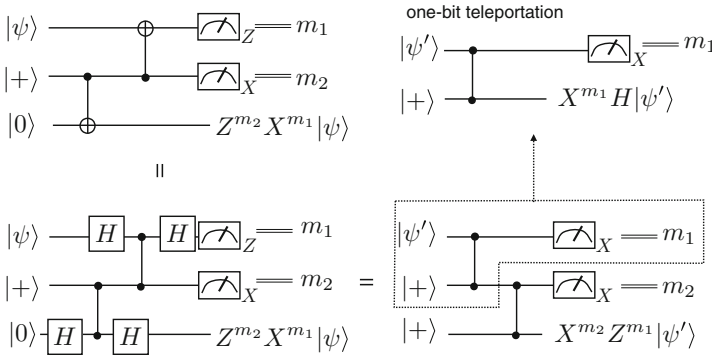
where $m_1, m_2 = 0, 1$ correspond to the measurement outcomes. A straightforward calculation provides

$$\langle\Psi(m_1, m_2)|_{i,a} \left(|\psi\rangle_i \frac{|0\rangle_a|0\rangle_b + |1\rangle_a|1\rangle_b}{\sqrt{2}} \right) = Z_b^{m_2} X_b^{m_1} |\psi\rangle_b / 2. \quad (2.30)$$

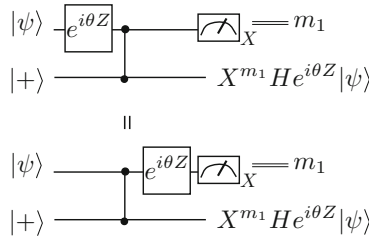
Hence, the unknown input state is teleported to Bob with a byproduct operator $X^{m_1} Z^{m_2}$. If Bob does not know the measurement outcomes (m_1, m_2) , the teleported state is a completely mixed state for Bob. However, if Alice sends the measurement outcome as a classical message, Bob can undo the byproduct and obtain the unknown quantum state at Bob's side. The circuit diagram of quantum teleportation is:



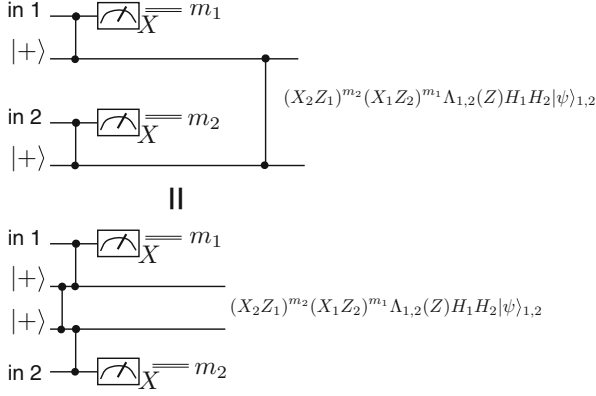
By using the following circuit equivalence, we can decompose the teleportation circuit into two elementary teleportations, the so-called *one-bit teleportations*:



One-bit teleportation is useful as a building block of the teleportation-based gates employed in MBQC. A single-qubit Z rotation $e^{i\theta Z}$ can be implemented in a teleportation-based way. Its action can be understood from the following circuit equivalence:



where we utilized the fact that $e^{i\theta Z}$ and $\Lambda(Z)$ commute. The controlled- Z operation is also implemented in a teleportation-based way as follows:



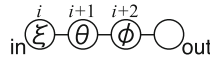
That is, instead of performing the $\Lambda(Z)$ gate after one-bit teleportations, we can prepare a special resource state, on which the $\Lambda(Z)$ gate is pre-implemented, and the $\Lambda(Z)$ gate is then performed via teleportation. These quantum operations based on quantum teleportation are called *gate teleportation* [68].

Now we are ready to formulate MBQC. An arbitrary single-qubit unitary operation U can be decomposed, up to an unimportant global phase, into

$$U = H e^{i\phi Z} e^{i\theta X} e^{i\xi Z} \tag{2.31}$$

$$= H e^{i\phi Z} H e^{i\theta Z} H e^{i\xi Z}. \tag{2.32}$$

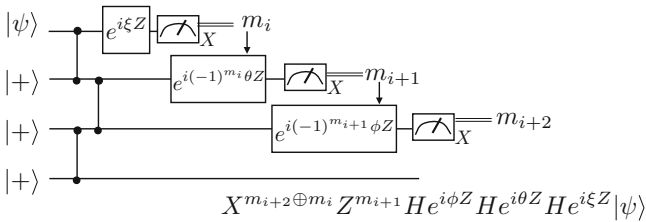
This indicates that we can perform an arbitrary single-qubit unitary operation by a sequence of one-bit teleportations. The resource state for the sequential one-bit teleportations is a 1D cluster state:



where the stabilizer generator for the left-most qubit is removed and the input state is encoded. We have to take care of the byproduct Pauli operators depending on the measurement outcomes. Fortunately, we can propagate the Pauli byproduct operators forward as follows:

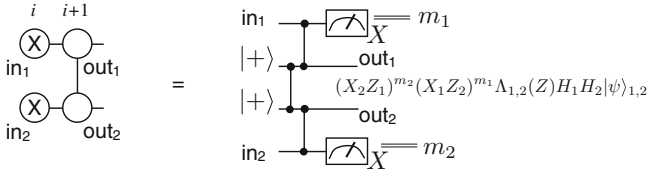
$$U = X^{m_{i+2}} H e^{i\phi' Z} X^{m_{i+1}} H e^{i\theta' Z} X^{m_i} H e^{i\xi Z}. \tag{2.33}$$

$$= X^{m_{i+2} \oplus m_i} Z^{m_{i+1}} H e^{i(-1)^{m_{i+1}} \phi' Z} H e^{i(-1)^{m_i} \theta' Z} H e^{i\xi Z}. \tag{2.34}$$



By choosing $\theta' = (-1)^{m_i}\theta$ and $\phi' = (-1)^{m_{i+1}}\phi$ adaptively, depending on the previous measurement outcomes, the random nature of the measurements can be managed. This procedure is called *feedforward*. The Pauli byproduct is propagated and updated throughout the computation. Note that the classical processing required to determine the measurement angle has only XOR (addition modulo two) operations [65].

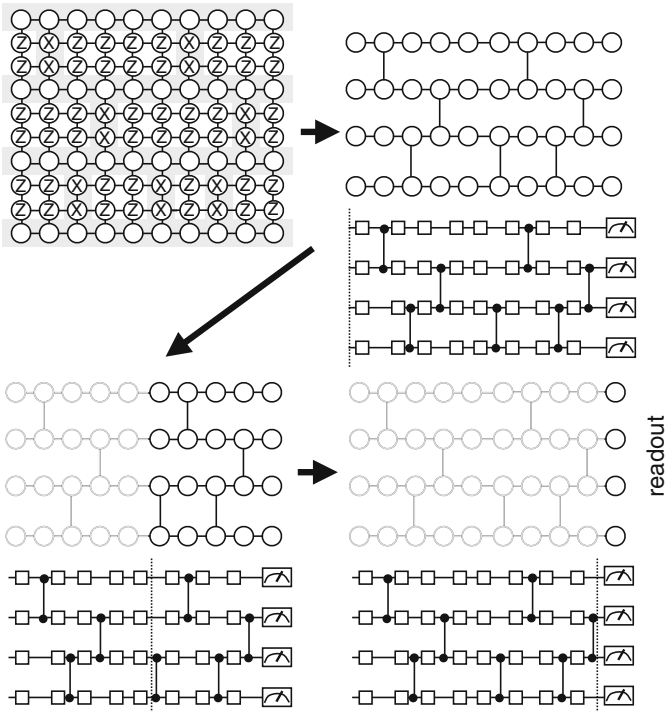
Next, we will investigate the measurement-based two-qubit gate operation. The resource state for the gate teleportation is the following cluster state:



To adjust the timing of the two-qubit operation, we can insert identity operations depending on the even and odd lengths as follows:



Without loss of generality, we can assume that all input states of the quantum computation are given by $|+\rangle$, which are automatically encoded by preparing the graph state. At the end of the computation (on the right-most qubits), measurements are performed to read out the result as follows:



In this way, universal quantum computation is simulated solely by measurements on a brickwork-like cluster state. This state can be generated from a cluster state on a square lattice by using the Pauli basis measurements as shown above. Accordingly, the square lattice cluster states are universal resources for MBQC.

The above circuit-based explanation of MBQC [69] is very intuitive and straightforward. However, for a complicated resource state, as will be shown, an operator-based understanding of MBQC [65] is quite useful. Let us reformulate MBQC from an operator viewpoint. Suppose again an MBQC on a 1D cluster state. The measurements are executed up to the $(i - 1)$ th qubits, and hence the operator K_l ($l \leq i$) is removed from the stabilizer generators. The logical degree of freedom on the remaining resource state can be specified by the i th logical operators

$$L_X^{(i)} = X_i Z_{i+1}, \quad (2.35)$$

$$L_Y^{(i)} = Y_i Z_{i+1}, \quad (2.36)$$

$$L_Z^{(i)} = Z_i. \quad (2.37)$$

These logical operators commute with all remaining stabilizer generators K_l ($l \geq i + 1$). Moreover, they anticommute with each other, satisfying the commutation relations for the Pauli operators. Thus, they specify the state encoded in the graph state. As seen above, a Z -rotation $e^{-i(\theta/2)Z_i}$ is applied before the X -basis measurement. Because $Z_i = L_Z^{(i)}$, this rotation induces a unitary transformation U of the logical operator

$$L_X^{(i)} \rightarrow \cos \theta L_X^{(i)} + \sin \theta L_Y^{(i)}, \quad (2.38)$$

$$L_Y^{(i)} \rightarrow \cos \theta L_Y^{(i)} - \sin \theta L_X^{(i)}. \quad (2.39)$$

Because $L_X^{(i)} = X_i L_Z^{(i+1)}$, the logical X operator after the X -basis measurements is given by $(-1)^{m_i} L_Z^{(i+1)}$ depending on the measurement outcome $m_i = 0, 1$. On the other hand, the logical operators $L_{Y,Z}^{(i)}$ do not commute with the X -basis measurement; they are not relevant logical operators after the measurement. If two operators are equivalent up to multiplications of the stabilizer operators, their action on the stabilizer state is also the same. By using this fact, we can replace the logical operators in (2.39) with

$$L_Z^{(i)} \sim L_Z^{(i)} K_{i+1} = X_{i+1} Z_{i+2} \equiv L_X^{(i+1)}, \quad (2.40)$$

$$L_Y^{(i)} \sim K_{i+1} = X_i Y_{i+1} Z_{i+2} \equiv X_i L_Y^{(i+1)}, \quad (2.41)$$

where \sim indicates that two operators are equivalent up to stabilizer operators. After the X -basis measurement, X_i can be replaced by its eigenvalue $(-1)^{m_i}$. Then the logical operator of the post-measurement state is given by

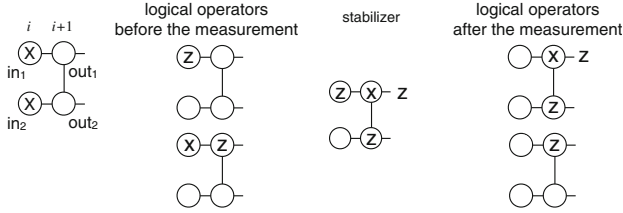
$$L_X^{(i)} \rightarrow (-1)^{m_i} (\cos \theta L_Z^{(i+1)} + \sin \theta L_Y^{(i+1)}) = UL_X^{(i+1)} U^\dagger, \quad (2.42)$$

$$L_Y^{(i)} \rightarrow (-1)^{m_i} (\cos \theta L_Y^{(i+1)} - \sin \theta L_Z^{(i+1)}) = UL_Y^{(i+1)} U^\dagger, \quad (2.43)$$

$$L_Z^{(i)} \rightarrow L_X^{(i+1)} = UL_Z^{(i+1)} U^\dagger. \quad (2.44)$$

We now realize that the logical operators for the i th step are transformed into those for the $(i+1)$ th step rotated by $U \equiv \bar{X}^{m_i} \bar{H} e^{-i(\theta/2)\bar{Z}}$, where \bar{A} indicates the operator A represented in terms of the logical basis specified by the $(i+1)$ th logical Pauli operators $\{L_X^{(i+1)}, L_Y^{(i+1)}, L_Z^{(i+1)}\}$.

Similarly, a two-qubit gate in MBQC can also be regarded as a propagation of a correlation by a projection on the stabilizer state. Consider the following graph state.



The logical operators for the i th step are given by $\{L_{X1}^{(i)}, L_{Z1}^{(i)}\}$ and $\{L_{X2}^{(i)}, L_{Z2}^{(i)}\}$. By multiplying the stabilizer operator, we obtain

$$L_{Z1}^{(i)} \sim X_{1,i+1} Z_{1,i+2} Z_{2,i+1} = L_{X1}^{(i+1)} L_{Z2}^{(i+1)}, \quad (2.45)$$

$$L_{Z2}^{(i)} \sim X_{2,i+1} Z_{2,i+2} Z_{1,i+1} = L_{X2}^{(i+1)} L_{Z1}^{(i+1)}. \quad (2.46)$$

The logical operators for the $(i+1)$ th step after the projections are calculated to be

$$\{L_{X1}^{(i)}, L_{Z1}^{(i)}\} \rightarrow \{(-1)^{m_1} L_{Z1}^{(i+1)}, L_{X1}^{(i+1)} L_{Z2}^{(i+1)}\} = \{VL_{X1}^{(i+1)} V^\dagger, VL_{Z1}^{(i+1)} V^\dagger\}, \quad (2.47)$$

$$\{L_{X2}^{(i)}, L_{Z2}^{(i)}\} \rightarrow \{(-1)^{m_2} L_{Z2}^{(i+1)}, L_{X2}^{(i+1)} L_{Z1}^{(i+1)}\} = \{VL_{X2}^{(i+1)} V^\dagger, VL_{Z2}^{(i+1)} V^\dagger\}. \quad (2.48)$$

Again, we realize that the logical operators for the i th step are transformed into those for the $(i+1)$ th step with a two-qubit unitary operation

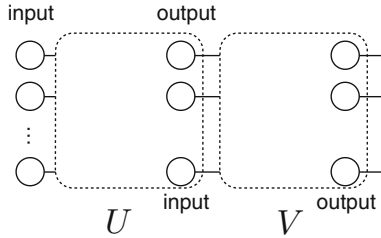
$$V \equiv (\bar{X}_1 \bar{Z}_2)^{m_1} (\bar{X}_2 \bar{Z}_1)^{m_2} \bar{\Lambda}_{1,2}(Z) \bar{H}_1 \bar{H}_2. \quad (2.49)$$

By combining single-qubit rotations $X^m H e^{i(\theta/2)Z}$ and the two-qubit operation $(X_1 Z_2)^{m_1} (X_2 Z_1)^{m_2} \Lambda_{1,2}(Z) H_1 H_2$ as seen above, we can perform a universal quantum computation. In this way, MBQC can be understood in the Heisenberg picture.

Suppose the logical Pauli operators of the k th input and output qubits are related by the measurements as follows:

$$\{L_{X,k}^{(\text{In})}, L_{Z,k}^{(\text{In})}\} \rightarrow \{UL_{X,k}^{(\text{Out})} U^\dagger, UL_{Z,k}^{(\text{Out})} U^\dagger\}. \quad (2.50)$$

The unitary operator U is performed on the input qubits. Here a Pauli byproduct, depending on the measurement outcomes, is also included in U . Moreover, if two graph states, which perform U and V , are concatenated with the appropriate feed-forwarding of the Pauli byproducts, then VU is performed:



Let us consider the example shown in Fig. 2.4a. The logical operators on the inputs are replaced by multiplying stabilizer generators so that they commute with the X -basis measurements as shown in Fig. 2.4b. Then the X operators on the measured qubits are replaced by ± 1 . The measurements transform the input logical operators as follows:

$$\begin{aligned} & \{L_{X,1}^{(\text{In})}, L_{Z,1}^{(\text{In})}, L_{X,2}^{(\text{In})}, L_{Z,2}^{(\text{In})}\} \\ & \rightarrow \left\{ (-1)^{m_1 \oplus m_5} L_{X,1}^{(\text{Out})} L_{Z,2}^{(\text{Out})}, (-1)^{m_3 \oplus m_6} L_{Z,1}^{(\text{Out})}, \right. \\ & \quad \left. (-1)^{m_2 \oplus m_6} L_{X,2}^{(\text{Out})} L_{Z,1}^{(\text{Out})}, (-1)^{m_4 \oplus m_5} L_{Z,2}^{(\text{Out})} \right\}. \end{aligned} \tag{2.51}$$

Thus, the $A_{1,2}(Z)$ gate is implemented up to a Pauli byproduct.

Using this fact and concatenation of the input-output relations, we can construct a measurement-based CNOT gate between the separated two-qubit as follows [65]:

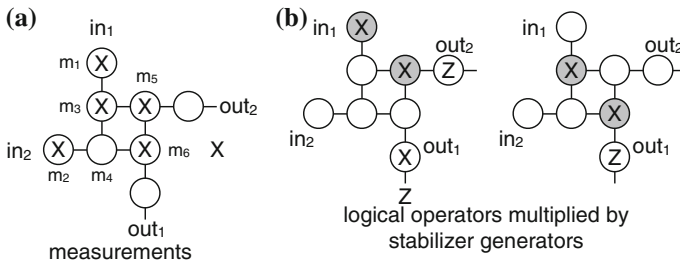
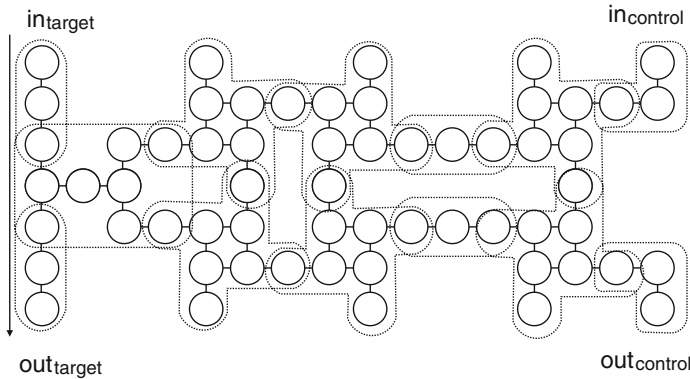
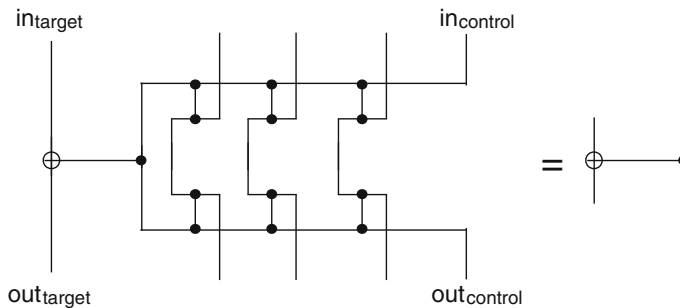


Fig. 2.4 **a** A graph state and a measurement pattern. **b** The logical X operator of input 1 is multiplied by the stabilizer generators and we obtain a correlated operator on outputs 1 and 2 (*left*). The logical Z operator of input 1 is multiplied by the stabilizer generators and we obtain the logical Z operator of output 1. The *gray colored* X operators are replaced by ± 1 depending on the measurement outcomes



The input-output relation of the above graph state is equivalent to that for the following circuit:



In this way, a CNOT gate between two arbitrary separated qubits can be implemented using a constant depth (constant width) resource state.

To conclude, we summarize the properties of MBQC and recent progress in this area. A unique feature of MBQC is that the resource state for universal quantum computation is prepared offline. Entangling operations, which would be one of the most difficult tasks in experiments, are employed only in this stage. Quantum computation is executed solely by adaptive measurements. This property is useful for experimental realization in certain physical systems. For example, a deterministic entangling operation is difficult to achieve in an optical quantum computation. In such a case, we can utilize linear optics and measurement-induced nonlinearity to generate a cluster state [70–72]. Importantly, the entangling operation can be non-deterministic, as long as the successful or non-successful outcome is heralded. By using such a probabilistic entangling operation, we can gradually expand the cluster state. After successful cluster state generation, we can start the measurements for quantum computation. Note that the probability of successful cluster state generation is not exponentially small by using a divide and conquer approach [71–76].

The clear separation between the quantum stage requiring entangling operations and the measurement stage is useful, not only for the physical implementation, but also in a quantum cryptographic scenario. Suppose that Bob (server) possesses a fully fledged quantum computer and that Alice (client), who has a less advanced quantum

device, such as a single-qubit state generator, wants to delegate quantum computation to Bob. By using the idea of MBQC, such a delegated quantum computation can be made unconditionally secure. This is called a blind quantum computation and was proposed by Broadbent, Fitzsimons, and Kashefi (BFK) [77] (see also the related earlier works [78, 79]). In the BFK protocol, Alice sends randomly rotated qubits $\{e^{-i(\theta_j/2)Z}|+\rangle\}$ to Bob, where the angle is chosen to be $\theta_j = k_j\pi/4$ ($k_j = 0, 1, \dots, 7$). Bob generates a cluster state by using the randomly rotated qubits. In the computation phase, Alice sends a classical message $\delta_j = \phi_j + \theta_j + r_j\pi$. Here, ϕ_j is the measurement angle with which Alice wants to perform a measurement. The angle θ_j is for the randomly rotated state (which is secret to Bob). The random bit $r \in \{0, 1\}$ makes the measurement angle completely random for Bob. Then Bob performs the measurement in the $\{e^{-i(\delta_j/2)Z}|\pm\rangle\}$ basis. Because the initial state is pre-rotated by θ_j (from Alice's viewpoint), Bob performs the measurement in the $\{e^{-i(\phi_j/2+r_j\pi/2)Z}|\pm\rangle\}$ basis, which is what Alice wants to do. However, from Bob's viewpoint, the state is a completely mixed state with no information about $\{\phi_j\}$. Thus, Bob is blind to any information about the input, the algorithm, and the output. Instead of the state generation, Alice, who has a measurement device, can also perform a blind quantum computation, whose security is guaranteed by the no-signaling principle [80]. A fault-tolerant blind quantum computation has been proposed, based on topologically protected MBQC [81].

2.7 Quantum Error Correction Codes

In this section, we introduce stabilizer codes, which are a class of quantum error correction (QEC) codes.

Three-Qubit Bit-Flip Code

The QEC codes can be described elegantly in the stabilizer formalism. Let us first consider the simplest one, the three-qubit bit flip code, whose stabilizer generators are given by

$$S_1 = Z_1Z_2, \quad S_2 = Z_2Z_3. \quad (2.52)$$

The stabilizer subspace is spanned by the following two logical states:

$$|0_L\rangle = |000\rangle, \quad |1_L\rangle = |111\rangle. \quad (2.53)$$

The logical Pauli- X operator is given by $L_X \equiv X_1X_2X_3$. The logical Pauli- Z operator is defined as $L_Z \equiv Z_1$. We may, equivalently, choose the logical Pauli Z operator to be Z_2 or Z_3 , because their actions on the code space are equivalent. The present code

is a quantum analogue of the classical three-bit repetition code. Consider a bit flip error with an error probability p :

$$\mathcal{E}_i \rho = (1 - p)\rho + pX_i \rho X_i. \quad (2.54)$$

If the initial state $|\psi_L\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$ undergoes the bit flip error independently, the output state is transformed in leading order as

$$\mathcal{E}_1 \circ \mathcal{E}_2 \circ \mathcal{E}_3 |\psi_L\rangle \langle \psi_L| = (1 - p)^3 |\psi_L\rangle \langle \psi_L| + p(1 - p)^2 \sum_i X_i |\psi_L\rangle \langle \psi_L| X_i + O(p^2). \quad (2.55)$$

The error X_i maps the code space to an orthogonal space. We perform a projective measurement onto the orthogonal subspaces, $P_k^\pm = (I \pm S_k)/2$, which we call a syndrome measurement, to know in which orthogonal space the state lies. Note that the encoded quantum information is not destroyed by the syndrome measurement, because it commutes with the logical operators. According to the measurement outcomes, the logical state can recover from the error as follows:

$$\mathcal{R} \circ \mathcal{E}_1 \circ \mathcal{E}_2 \circ \mathcal{E}_3 |\psi_L\rangle \langle \psi_L| = [(1 - p)^3 + 3p(1 - p)^2] |\psi_L\rangle \langle \psi_L| + O(p^2), \quad (2.56)$$

where the recovery operator is given by

$$\begin{aligned} \mathcal{R} \rho = & P_1^+ P_2^+ \rho P_2^+ P_1^+ + X_1 P_1^- P_2^+ \rho P_2^+ P_1^- X_1 + X_2 P_1^- P_2^- \rho P_2^- P_1^- X_2 \\ & + X_3 P_1^+ P_2^- \rho P_2^- P_1^+ X_3. \end{aligned} \quad (2.57)$$

The four terms in $\mathcal{R} \rho$ correspond to the measurement outcomes (eigenvalues) $(+1, +1)$, $(-1, +1)$, $(-1, -1)$, and $(+1, -1)$ of the stabilizer generators, respectively. By comparing Eqs. (2.55) and (2.56), one can understand that if p is sufficiently small, the fidelity of the logical state is improved.

Similarly, we can construct a three-qubit phase flip code, which can correct a phase flip error, by changing the basis with the Hadamard transformation:

$$\langle Z_1 Z_2, Z_2 Z_3 \rangle \rightarrow \langle X_1 X_2, X_2 X_3 \rangle. \quad (2.58)$$

9-Qubit Shor Code

The three-qubit bit-flip code cannot correct Z errors, which commute with the stabilizer generators. A QEC code that can correct all X , Y , and Z errors was developed by Shor based on a concatenation of three-qubit bit-flip and phase-flip codes [82]. The stabilizer generators of the 9-qubit Shor code are given as follows:

$$\begin{array}{cccccccc}
\hline\hline
X & X & X & X & X & X & I & I & I \\
I & I & I & X & X & X & X & X & X \\
Z & Z & I & I & I & I & I & I & I \\
I & Z & Z & I & I & I & I & I & I \\
I & I & I & Z & Z & I & I & I & I \\
I & I & I & I & Z & Z & I & I & I \\
I & I & I & I & I & I & Z & Z & I \\
I & I & I & I & I & I & I & Z & Z \\
\hline\hline
\end{array} \tag{2.59}$$

The code space is spanned by

$$\frac{|0_L\rangle + |1_L\rangle}{\sqrt{2}} = \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}, \tag{2.60}$$

$$\frac{|0_L\rangle - |1_L\rangle}{\sqrt{2}} = \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}. \tag{2.61}$$

The logical Pauli operators are given by $X_L = X^{\otimes 9}$ and $Z_L = Z^{\otimes 9}$, which are bitwise tensor products of physical Pauli operators. If the logical A operator is given by a bitwise tensor product of the physical A operators on the QEC code, we say that the operation A has *transversality*. The 9-qubit code is capable of correcting all X , Y , and Z errors for each qubit, which can be understood because the three-qubit phase flip code $\{|+++ \rangle, |---\rangle\}$ is constructed by using the three logical qubits of the three-qubit bit flip codes $\{|000\rangle, |111\rangle\}$.

Note that any single-qubit noise \mathcal{E} can be described by using the Kraus operators $\{K_j\}$:

$$\mathcal{E}\rho = \sum_j K_j \rho K_j^\dagger. \tag{2.62}$$

Any operator K_j can be decomposed into the Pauli operators $\sigma_0 = I, \sigma_1 = X, \sigma_2 = Y$, and $\sigma_3 = Z$:

$$K_j = \sum_l c_{jl} \sigma_l. \tag{2.63}$$

Thus, if the X and Z errors on a single qubit are both corrected appropriately, we can correct any single-qubit noise automatically. Specifically, because noise contains a superposition of the Pauli errors, it can be collapsed by the syndrome measurements.

Stabilizer Codes

To summarize the above examples, let us formalize the stabilizer quantum error correction codes and their properties. The code space of a stabilizer QEC code is defined by a stabilizer group $\{\{S_i\}\}$. The encoded degree of freedom is specified by the mutually independent logical operators $\{L_j^Z\}$, which commute with all stabilizer generators

and are independent of the stabilizer generators. The computational basis state of the code state is completely determined by the stabilizer group $\langle \{S_i\}, \{(-1)^{m_j} L_j^Z\} \rangle$. We can always find another set of the logical operators $\{L_j^X\}$ being subject to

$$L_j^X L_i^Z = (-1)^{\delta_{ij}} L_i^Z L_j^X, \quad (2.64)$$

where δ_{ij} is the Kronecker delta. Hence, the pair of logical operators L_i^Z and L_i^X represents the i th logical qubit. In terms of the numbers n and $(n - k)$ of qubits and stabilizer generators, respectively, the number of pairs of logical operators is k .

Let us define the *weight* $\text{wt}(S_i)$ of a Pauli product S_i as the number of qubits on which a Pauli operator (except for the identity I) is acting. The minimum weight of the logical operator over all possible logical operators is called the *code distance* d . This implies that all Pauli products whose weights are smaller than d are elements of the stabilizer group or anticommute with the stabilizer generators. Thus, they act trivially on the code state or map the code state into an orthogonal subspace. If the weight of a Pauli product as an error is less than $(d - 1)/2$, we can find a unique recovery operator that returns the erroneous state into the code space. Thus, we can correct $\lfloor (d - 1)/2 \rfloor$ errors. Such a stabilizer QEC code is called a $[[n, k, d]]$ stabilizer code. For example, the code distance of the 9-qubit code is a $[[9, 1, 3]]$ stabilizer code correcting weight-one errors.

The nine-qubit code is not the smallest QEC code that can correct all weight-one X , Y , and Z errors. The smallest code is the five-qubit code, found independently by Laflamme et al. [83] and Bennett et al. [84]. The stabilizer generators and the logical Pauli operators are given as follows:

$$\begin{aligned} \overline{\overline{S_1}} &= X Z Z X I \\ \overline{\overline{S_2}} &= I X Z Z X \\ \overline{\overline{S_3}} &= X I X Z Z \\ \overline{\overline{S_4}} &= Z X I X Z \\ \overline{\overline{X_L}} &= X X X X X \\ \overline{\overline{Z_L}} &= Z Z Z Z Z \end{aligned} \quad (2.65)$$

We see that the code distance is three, and hence an arbitrary single-qubit error can be corrected.

Calderbank-Shor-Steane Codes

The readers who are familiar with classical coding theory might already be aware of the correspondence between stabilizer codes and classical linear codes. Let us recall the 9-qubit code. The X and Z errors are detected independently through the Z -type and X -type stabilizer generators, respectively. This implies that X and Z error corrections are described by classical coding theory, where two classical error corrections are subject to a certain constraint to appropriately form a stabilizer group.

To formulate this, we briefly review classical linear codes. A $[[n, k]]$ classical linear code C is defined as a k -dimensional space V_C of n -dimensional vector space over $GF(2)$ by using an $n \times k$ generator matrix

$$G = (\mathbf{b}_1, \dots, \mathbf{b}_k), \quad (2.66)$$

where the column vectors $\{\mathbf{b}_i\}$ are the basis vectors of V_C . A k -bit classical information \mathbf{y} is encoded into the code \mathbf{c} as

$$\mathbf{a} = G\mathbf{c}. \quad (2.67)$$

To detect and analyze the errors, we define an $(n - k) \times n$ parity check matrix H such that $H\mathbf{b}_k = 0$ for all basis vectors $\{\mathbf{b}_k\}$. Suppose an error \mathbf{e} occurs on the code state, $\mathbf{a}' = \mathbf{a} \oplus \mathbf{e}$, where \oplus indicates a bitwise addition modulo two. By using the parity check matrix H , we can detect the error

$$H\mathbf{a}' = H(\mathbf{a} \oplus \mathbf{e}) = H\mathbf{e} \equiv \mathbf{s}, \quad (2.68)$$

where \mathbf{s} is called an error syndrome.

For example, the three-bit repetition code is defined by the generator

$$G = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}. \quad (2.69)$$

A classical bit 0 and 1 is encoded into $(0, 0, 0)^T$ and $(1, 1, 1)^T$, respectively. The parity check matrix is defined to be

$$H = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}. \quad (2.70)$$

Now, we realize that the positions of the 1s of the parity check matrix are exactly the same as those of the Z s in the stabilizer generators of the three-qubit bit flip code. This suggests to use the parity check matrices H_x and H_z of the two classical linear codes C_x and C_z , respectively, in the definition of the X -type and Z -type stabilizer generators:

$$S_X^{(i)} = \prod_j X_j^{(H_x)_{ij}}, \quad S_Z^{(i)} = \prod_j Z_j^{(H_z)_{ij}}. \quad (2.71)$$

For these operators to commute with each other, the two parity check matrices have to satisfy

$$H_x H_z^T = \mathbf{0}, \quad (2.72)$$

where $\mathbf{0}$ indicates a matrix with all elements = 0. To define the logical Z operators, we define a quotient space $\text{Ker}(H_x)/\text{Img}(H_z^T)$. Denoting the basis vectors of the quotient space $\text{Ker}(H_x)/\text{Img}(H_z^T)$ by $\{\mathbf{b}_k^z\}$, we define the logical Z operators

$$L_Z^{(k)} = \prod_i Z_i^{(\mathbf{b}_k^z)_i}. \quad (2.73)$$

Similarly, we can define the logical X operators

$$L_X^{(k)} = \prod_i X_i^{(\mathbf{b}_k^x)_i}, \quad (2.74)$$

using the basis vectors $\{\mathbf{b}_k^x\}$ of a quotient space $\text{Ker}(H_z)/\text{Img}(H_x^T)$, where \mathbf{b}_k^x is chosen such that $L_Z^{(i)} L_X^{(j)} = (-1)^{\delta_{ij}} L_X^{(j)} L_Z^{(i)}$. Note that dimensions of these kernel subspaces are the same, and we can easily find such pairs of anticommuting logical operators. The above stabilizer code constructed from two classical linear codes is called a Calderbank-Shor-Steane (CSS) code.

Let us see an important example of CSS codes, the 7-qubit code introduced by Steane [85]. Specifically, we utilize a classical linear code, the $[[7, 4, 3]]$ Hamming code, whose generator and parity check matrices are given by

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}, \quad H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}. \quad (2.75)$$

Because $HH^T = \mathbf{0}$, we can employ the Hamming code to define both X - and Z -type stabilizer generators:

$$\begin{aligned} \overline{\overline{S_1}} &= I I I X X X X \\ \overline{\overline{S_2}} &= I X X I I X X \\ \overline{\overline{S_3}} &= X I X I X I X \\ \overline{\overline{S_4}} &= I I I Z Z Z Z \\ \overline{\overline{S_5}} &= I Z Z I I Z Z \\ \overline{\overline{S_6}} &= Z I Z I Z I Z \end{aligned} \quad (2.76)$$

There is an element $(1, 1, 1, 1, 1, 1, 1)^T$ in the quotient space $\text{Ker}(H)/V_H$. The logical operators are given by

$$\begin{aligned} L_X &= X X X X X X X, \\ L_Z &= Z Z Z Z Z Z Z. \end{aligned} \quad (2.77)$$

The 7-qubit code is quite useful for fault-tolerant quantum computation. Both the X - and Z -type stabilizer generators are defined from the Hamming code, and the stabilizer group is invariant under the transversal Hadamard operation $\bar{H} \equiv H^{\otimes 7}$. Moreover, the logical X operator is mapped into the logical Z operator, $\bar{H}L_X\bar{H} = L_Z$. Thus, the transversal Hadamard operation acts as the logical Hadamard operation for the encoded degree of freedom. Similarly, a transversal phase operation $\bar{S} \equiv (ZS)^{\otimes 7}$ acts as a logical phase operation, $\bar{S}L_X\bar{S}^\dagger = L_Y$. Furthermore, a transversal CNOT operation $\bar{\Lambda}(X) = \Lambda(X)^{\otimes 7}$ keeps the stabilizer group of two logical qubits invariant:

$$\langle \{S_X^{(i)} \otimes I^{\otimes 7}\}, \{S_Z^{(i)} \otimes I^{\otimes 7}\}, \{I^{\otimes 7} \otimes S_X^{(i)}\}, \{I^{\otimes 7} \otimes S_Z^{(i)}\} \rangle \quad (2.78)$$

$$= \langle \{S_X^{(i)} \otimes S_X^{(i)}\}, \{S_Z^{(i)} \otimes I^{\otimes 7}\}, \{I^{\otimes 7} \otimes S_X^{(i)}\}, \{S_Z^{(i)} \otimes S_Z^{(i)}\} \rangle. \quad (2.79)$$

The logical Pauli operators are subject to the transformation rule of the CNOT gate. Accordingly, the transversal CNOT operation $\bar{\Lambda}(X)$ acts as a logical CNOT operation for the encoded degree of freedom. Because the Hadamard, phase, and CNOT operations are implemented transversally, whole Clifford group elements can be implemented by transversal operations.

The transversal implementation is fault-tolerant because the operations apparently do not increase the number of errors on a code block and there is no internal interaction between the qubits in the same code block. Combined with a fault-tolerant gadget for measuring the error syndrome as explained in Appendix A.1, we can implement Clifford operations fault-tolerantly.

Unfortunately, the non-Clifford operation does not transform a Pauli operator into another Pauli operator. For example, the $\pi/8$ operation $e^{-i(\pi/8)Z}$ transforms the Pauli X operator into a Clifford operator:

$$e^{-i(\pi/8)Z} X e^{i(\pi/8)Z} = (X + Y)/\sqrt{2}. \quad (2.80)$$

This implies that a transversal non-Clifford operation hardly results in a logical non-Clifford operation. Thus, a fault-tolerant non-Clifford gate operation is not so straightforward. To settle this, we can utilize magic state distillation, consisting of noisy non-Clifford resource states and ideal (fault-tolerant) Clifford operations, as explained in the next section.

2.8 Magic State Distillation

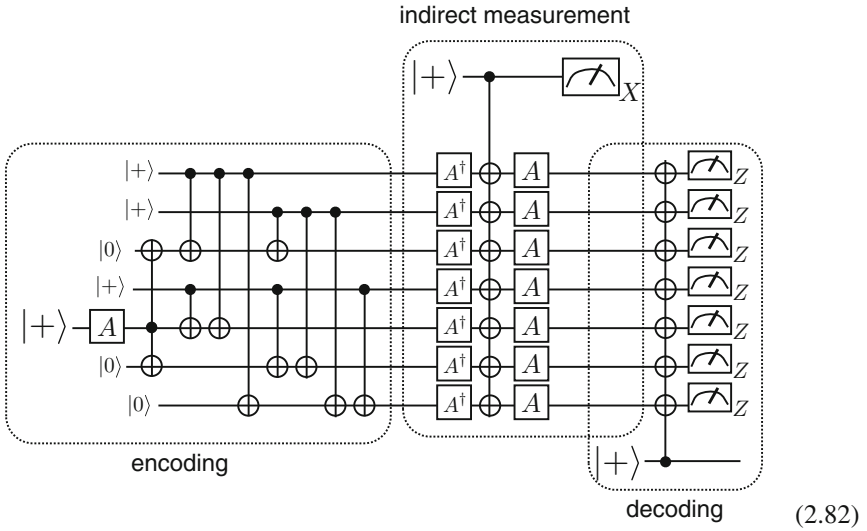
2.8.1 Knill-Laflamme-Zurek Protocol

A fault-tolerant implementation of a non-Clifford gate was first proposed in an earlier paper by Knill, Laflamme, and Zurek [86, 87]. Instead of implementing the non-Clifford gate directly, we consider a fault-tolerant preparation of the non-stabilizer state, the so-called magic state,

$$|A\rangle \equiv e^{i(\pi/8)Y}|+\rangle. \quad (2.81)$$

The magic state can be utilized to implement a non-Clifford gate $A \equiv e^{i(\pi/8)Y}$ by using one-bit teleportation consisting of Clifford gates and the Pauli basis state preparations and measurements. Thus, if we can prepare a clean magic state, we can create a fault-tolerant non-Clifford gate by using fault-tolerant Clifford gates.

The Knill-Laflamme-Zurek construction of the fault-tolerant preparation of the magic state was based on the fact that $|A\rangle$ is an eigenstate of H . The Hadamard operation has transversality, e.g., on the 7-qubit code. Hence, if we perform a projective measurement of $H^{\otimes 7}$, we obtain a clean magic state. The circuit is given as follow



where $A = e^{i(\pi/8)Y}$ and we used the fact that $A_t \Lambda_{c,t}(X) A_t^\dagger = \Lambda_{c,t}(H)$. The above circuit consists of three parts, encoding of the logical magic state into the 7-qubit code, indirect measurement of $H^{\otimes 7}$, and decoding by one-bit teleportation. Note that all Clifford operations are assumed to be ideal, because they are easily made fault-tolerant by using a stabilizer code. With an appropriate randomization operation made up by Clifford operations, a noisy magic state can be transformed into a mixture of $|A\rangle$ and $|A^\perp\rangle = Y|A\rangle$:

$$\rho_A = (1-p)|A\rangle\langle A| + p|A^\perp\rangle\langle A^\perp| \quad (2.83)$$

Hence, the noise on the magic state can be expressed as a Y error. The Y error can be detected by transversal Z measurements for the decoding (see also Knill's gadget for a fault-tolerant syndrome measurement in Appendix A.1). Assuming an ideal Clifford operation, the error probability p decreases as $O(p^3)$, when we employ the

7-qubit code with distance 3. Including the A gate for the initial encoding, we need 15 noisy magic states $|A\rangle$ to obtain a clean magic state. This distillation protocol works for all self-dual CSS codes, which are symmetric between the X - and Z -type stabilizer generators and has transversality for the Hadamard gate. Recently, improved protocols have been proposed based on this approach [88, 89].

2.8.2 Bravyi-Kitaev Protocol

Bravyi and Kitaev proposed another magic state distillation protocol based on a 15-qubit code [60]. While their and the Knill-Laflamme-Zurek protocols seem quite different, they are, interestingly, known to be equivalent [90]. In the Bravyi-Kitaev protocol, a $[[15, 1, 3]]$ quantum code is defined by the $[[15, 7, 3]]$ classical Reed-Muller code, whose parity check matrix is given by

$$H_x = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \quad (2.84)$$

The X -type stabilizer generators are defined by $S_X^{(i)} = \prod_j X_j^{(H)_{ij}}$. Then, we choose a parity check matrix of another classical code

$$H_z = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (2.85)$$

so that $H_x H_z^T = \mathbf{0}$ and the Z -type stabilizer generators are defined similarly. The logical operators are given by $L_X = X^{\otimes 15}$ and $L_Z = Z^{\otimes 15}$. The logical states are written down explicitly as

$$|0_L\rangle = \prod_{i=1}^4 (I + S_X^{(i)}) |00 \dots 0\rangle, \quad (2.86)$$

$$|1_L\rangle = \prod_{i=1}^4 (I + S_X^{(i)}) |11 \dots 1\rangle. \quad (2.87)$$

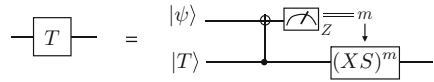
The number of 1s in each term of $|0_L\rangle$ and $|1_L\rangle$ is 8 and 7, respectively. By applying the $T = e^{-i(\pi/8)Z}$ gate transversally, we obtain

$$T^{\otimes 15}|0_L\rangle = e^{i\pi/8}|0_L\rangle, \tag{2.88}$$

$$T^{\otimes 15}|1_L\rangle = e^{-i\pi/8}|0_L\rangle. \tag{2.89}$$

Thus, the transversal T gate acts as a logical T^\dagger gate. Note that this transversality does not hold in the orthogonal (erroneous) subspace, e.g., spanned by $\{X_k|0_L\rangle, X_k|1_L\rangle\}$. However, we can show that this is enough to perform a fault-tolerant logical T gate.

Instead of applying the T gate directly, we implement it using a one-bit teleportation:



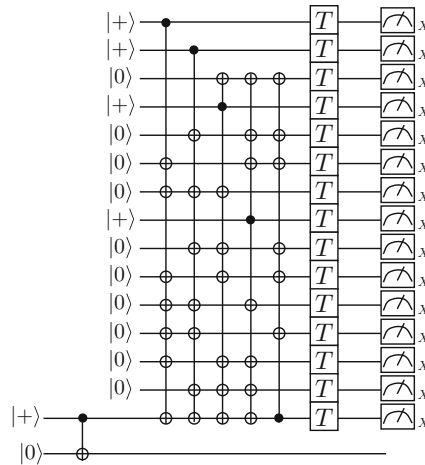
$$\tag{2.90}$$

The Z-basis measurement and CNOT operation are both implemented transversally on a CSS code. Thus, if the preparation of the non-Clifford ancilla state $|T\rangle = (|0_L\rangle + e^{-i\pi/4}|1_L\rangle)/\sqrt{2}$, called a magic state, is done fault-tolerantly, one can ensure fault-tolerance of the logical T gate.

By using an appropriate randomization process, we can prepare a noisy magic state as follows:

$$\rho_T = (1 - p)|T\rangle\langle T| + pZ|T\rangle\langle T|Z. \tag{2.91}$$

Thus, a phase error Z is located on the ideal magic state with probability p . This phase error causes a Z error after the T gate by one-bit teleportation. Because the code space is invariant under the transversal T gate, we can detect such a Z error by the following circuit.



The first part of the above circuit consisting of CNOT operations is an encoding circuit for the quantum Reed-Muller code. The transversal T gate is applied by using one-bit teleportation. The logical $|T\rangle$ state is measured in the X -basis transversally, which detects Z errors on the code state, projecting the code state on the local X -basis. The input state in the second lowest wire is entangled with the ancilla qubit in the lowest wire, where the distilled magic state is teleported.

Let \mathbf{c} be a 15-bit string specifying the location of the Z errors, $E(\mathbf{c}) \equiv \prod_i Z_i^{c_i}$. If $E(\mathbf{c})$ commutes with the X -type stabilizer generators, the state passes through the distillation circuit. To calculate this probability, we define a weight enumerator of a subspace $V \in GF(2^n)$,

$$W_V(x, y) = \sum_{\mathbf{c} \in V} x^{n-\text{wt}(\mathbf{c})} y^{\text{wt}(\mathbf{c})}. \quad (2.92)$$

The probability of passing the distillation circuit is calculated to be

$$p_{\text{pass}} = W_{V_{H_x}^\perp}(1-p, p) = \frac{1}{|V_x|} W_{V_{H_x}}(1, 1-2p) = \frac{1 + 15(1-2p)^8}{16}, \quad (2.93)$$

where the orthogonal subspace $V_{H_x}^\perp$ is equivalent to the kernel of H_x , $\text{Ker}(H_x)$. We also used the MacWilliams identity [91]:

$$W_V(x, y) = \frac{1}{|V|} W_{V^\perp}(x+y, x-y). \quad (2.94)$$

Similarly, the error probability of the output can be calculated to be

$$W_{V_{H_z}}(p, 1-p) = \frac{1}{|V_{H_z}^\perp|} W_{V_{H_z}}(1, 2p-1) \quad (2.95)$$

$$= \frac{1 + 15(2p-1)^8 + 15(2p-1)^7 + (2p-1)^{15}}{32}. \quad (2.96)$$

Accordingly, the error probability, under the condition of passing the distillation circuit, is given by

$$p' = \frac{1 + 15(2p-1)^8 + 15(2p-1)^7 + (2p-1)^{15}}{2[1 + 15(1-2p)^8]} = 35p^3 + O(p^4). \quad (2.97)$$

If $p' > 0.141$, we can reduce the error probability on the magic state via the distillation circuit. After l rounds of distillation, the error probability decreases to $(\sqrt{35}p)^3 / \sqrt{35}$. At each round, we need 15 noisy magic states. Because the probability of successfully passing the distillation circuit converges rapidly to 1, the average number of noisy

magic states consumed after l rounds becomes 15^l . Accordingly, the average number of noisy magic states required to achieve an error probability ε of the magic state scales like

$$[\log(\sqrt{35\varepsilon})/\log(\sqrt{35p})]^{\log(15)/\log(3)} = O(\log^{2.5} \varepsilon). \quad (2.98)$$

In Sect. 2.4, we saw that if an input state is a convex mixture of the Pauli basis states followed by Clifford operations and Pauli basis measurements, the measurement outcomes can be simulated classically in the weak sense. The noisy magic state $\rho_T = (1-p)|T\rangle\langle T| + p|T\rangle\langle T|$ lies on a line in the x - y plane, as shown in Fig. 2.5. If $p > (1-\sqrt{2}/2)/2 = 0.146$, $\rho_{\pi/8}$ lies inside the octahedron, and the Gottesman–Knill theorem is applicable. On the other hand, if $p < 0.141$, magic state distillation allows us to implement universal quantum computation with an arbitrary accuracy as seen above. Unfortunately, magic state distillation based on the Reed-Muller code does not provide a tight distillation threshold against the classically simulatable region. In Ref. [90], a distillation protocol using the 7-qubit code was proposed and achieved a tight threshold $p = (1-\sqrt{2}/2)/2$. In this sense, the classically simulatable and quantum universal regions are divided sharply on the x - y plane.

By combining the magic state distillation and fault-tolerant Clifford operations on the CSS code, we can perform universal quantum computation fault-tolerantly. In order to make the error probability arbitrarily small, we can employ concatenated quantum computation, in which logical qubits of a lower concatenation level are utilized as the physical qubits at a higher level. At the higher level, all operations, including logical qubit preparations and syndrome measurements for QEC, have to be done fault-tolerantly. If the error probability is smaller than a certain constant value, which we call the noise threshold, the logical error probability at the highest concatenation level decreases super-exponentially. On the other hand, the overhead increases exponentially. Thus, we can make the logical error probability small enough to maintain a quantum computation of size N with a polylogarithmic overhead $\text{polylog}(N)$. This implies that we can obtain a quantum benefit even for a quantum algorithm with a quadratic speedup, such as the Grover algorithm [37]. In Appendix A.1, we briefly

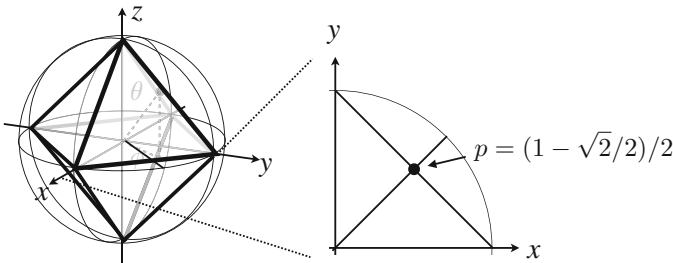


Fig. 2.5 The noisy magic state in the Bloch sphere

review fault-tolerant syndrome measurements, concatenated quantum computation, and the threshold theorem.

While the resource increment for protecting quantum computation scales poly-logarithmically, its constant factor is quite huge. Almost all overheads for the fault-tolerant quantum computation are employed for the magic state distillation [92, 93]. Thus, much effort has been spent recently on developing resource-efficient magic state distillation [88, 94–96].

2.9 Summary and Discussion

In this chapter, we have introduced the stabilizer formalism and have learned how to utilize it to describe the quantum states and the operations of the restricted class efficiently. As examples, we have seen MBQC, quantum error correction codes, and magic state distillation, all of which take important roles in quantum computation.

While MBQC can be understood as sequential quantum teleportation circuits [69], the operator-based explanation, as employed in the original paper [64, 65], would be also useful. Indeed, topologically protected quantum computation was first formalized elegantly in terms of MBQC on a 3D cluster state [4, 5] as we will see in Chap. 5. It was further translated into the circuit-based model in two dimensions (2D) [6, 97], which will be explained in detail in Chap. 4. Specifically, the operator-based formulation is quite useful for describing how correlation is propagated in a topologically protected way.

Recently, quantum stabilizer codes have been employed not only for quantum error correction but also as exactly solvable toy models for topologically ordered condensed matter systems as will be seen in the next chapter. Specifically, the simple but rich structure of the stabilizer formalism allows us to obtain a lot of insights from the toy models, such as scaling laws of entanglement entropy, stability of quantum phases at finite temperature, and statistics of excitations. Besides, a quantum stabilizer code has found its application even to high energy physics recently [98, 99].

The magic state is necessary ingredient for universal quantum computation and hence might be a clue to find the origin of quantum speedup. In Ref. [100], it has been shown that violation of non-contextuality, which is a kind of non-locality of quantum mechanics and prohibits pre-existing hidden valuables for quantum states before measurements, is necessary for distillability of magic states. Moreover, in Ref. [101], a quantum-classical boundary of noisy quantum commuting circuits is derived from the viewpoint of classical (not) simulatability of the output distributions. The boundary, which sharply divides the classical simulatable and not simulatable regions, has its origin in distillability of a magic state.

Chapter 3

Topological Stabilizer Codes

Protecting quantum information from decoherence is of prime importance to realize quantum information processing. Several approaches have been proposed toward reliable quantum information processing, ranging from passive to active protections, such as decoherence-free subspaces [102], dynamical decoupling [103], and quantum error correction [82]. Among these, the most comprehensive approach is fault-tolerant quantum computation based on quantum error correction [104]. Quantum fault-tolerance theory ensures scalable quantum computation with noisy quantum devices as long as the error probability of such devices is smaller than a threshold value (see Appendix A.1). The first threshold values were obtained, at almost the same time (1996), independently by Aharonov et al., who used a polynomial code of distance five [39, 40]; Knill et al., who used the Steane seven-qubit code [86, 87]; and Kitaev, who used toric (surface) codes [105]. All of these studies were based on a concatenated quantum computation and achieved similar threshold values $\sim 10^{-6}$. Since then, fault-tolerant quantum computing has been studied as one of the most important issues in quantum information science. In 2005, Knill proposed a novel scheme based on the C_4/C_6 error-detecting code, namely the Fibonacci scheme [106], and achieved a considerably higher threshold, a few %. Recently, Fujii et al. have further improved this approach by making use of measurement-based quantum computation (MBQC) [4, 5] on logical cluster states, which gives the highest threshold value so far, $\sim 5\%$ [107, 108].

All these approaches rely on the availability of two-qubit gates between arbitrarily separated qubits. However, all physically available interactions are local in space. Moreover, if we consider manufacturing convenience and the selective addressability of individual qubits, 2D nearest-neighbor architectures are necessary. It was thought that, if we restrict the two-qubit gates to nearest-neighbor ones in 2D, then the threshold value would decrease significantly [109]. The situation changed completely by the proposal made by Raussendorf et al., i.e., topologically protected quantum computation [4, 5, 6]. They utilized a surface code, a kind of topological quantum code, originally proposed by Kitaev as a toric code on a torus [110] and investigated

by Dennis et al. in detail [111]. All the stabilizer generators of the surface code are spatially local, as will be seen later, and hence syndrome measurements are done by using nearest-neighbor two-qubit gates. More importantly, Raussendorf et al. developed a novel way to implement a fault-tolerant logical gate operation by braiding defects on the surface, which can be done by using only single-qubit measurements and nearest-neighbor two-qubit gates. Nevertheless, its threshold value is very high $\sim 1\%$ [4, 5, 6, 112]. Based on these analyses, physical implementations and quantum architecture designs have recently been suggested [75, 76, 92, 93, 113, 114, 115, 116, 117], which clarifies the experimental requirements for building a fault-tolerant quantum computer. On the other side, extensive experimental resources have been used to achieve these requirements, and very encouraging results have already been obtained in several experiments in superconducting systems [118, 119, 120, 121].

Another important motivation for studying topological quantum codes are their connection with topologically ordered systems in condensed matter physics [110]. Topological stabilizer codes, whose stabilizer generators are local and translation invariant, provide toy models for topologically ordered systems. Using quantum coding theory with a geometrical constraint such as locality and translation invariance, we can understand the nature of topologically ordered condensed matter systems. One of the most important issues in this direction is to answer whether or not thermally stable topological order exists in three or lower dimensions [122, 123].

When we consider the decoding problems of topological quantum codes, we encounter classical statistical physics [111]. Specifically, the posterior probabilities for the error correction correspond to partition functions of classical spin glass models [111], such as the random-bond Ising model. If the corresponding random-bond Ising model lies inside a ferromagnetic phase, the error correction on a surface code succeeds, and the logical error probability decreases exponentially.

In this chapter, we introduce a representative example of the topological stabilizer codes, the surface codes. Before that, we also briefly review the \mathbf{Z}_2 chain complex, which is a useful tool to define the surface codes. We also apply it to a classical repetition code as an exercise. After introducing the surface codes, we explain how topological error correction is performed. We also mention the relation between topological error correction and the classical spin glass model. Another example of topological stabilizer codes, the topological color codes, is also presented. Finally, we explain the connection between topological stabilizer codes and topologically ordered systems studied in condensed matter physics.

3.1 \mathbf{Z}_2 Chain Complex

Before defining the surface codes, we introduce a mathematical tool, the \mathbf{Z}_2 chain complex, which is useful for describing the surface codes.

Consider a surface $G = (V, E, F)$ that consists of the vertices $V = \{v_k\}$, edges $E = \{e_l\}$, and faces $F = \{f_m\}$. (Here we consider a graph $G = (V, E)$ embedded on a 2D manifold, the *surface*, on which the faces F are defined.) We define a vector space C_0 over \mathbf{Z}_2 , using each vertex $v_k \in V$ as a basis $B(C_0) = \{v_k\}$. A vector in C_0

$$c_0 = \sum_k z_k v_k, \quad (3.1)$$

where $z_k = \{0, 1\}$, is referred to as a 0-chain. The vector space C_0 is an Abelian group under component-wise addition modulo 2. Similarly, we can define the Abelian groups C_1 and C_2 using the edges $E = \{e_l\}$ and faces $F = \{f_m\}$ as base vectors $B(C_1) = \{e_l\}$ and $B(C_2) = \{f_m\}$, respectively. The elements

$$c_1 = \sum_l z_l e_l, \quad (3.2)$$

$$c_2 = \sum_m z_m f_m, \quad (3.3)$$

where $z_l, z_m = \{0, 1\}$, are called the 1-chain and 2-chain, respectively. With a mild abuse of the notation, a set of i -dimensional elements (i.e., vertices, edges, and faces) is also specified by a i -chain $c_i \in C_i$ as a set of elements having $z_j = 1$.

We can define a homomorphism $\partial_i : C_i \rightarrow C_{i-1}$ such that

$$\partial_i \circ \partial_{i-1} = 0. \quad (3.4)$$

Specifically, $\partial_i c_i$ is defined as an $(i-1)$ -chain that is the boundary of c_i . For example, ∂e_l is a 1-chain for which $z_k = 1$ if and only if (iff) v_k is an endpoint of e_l . Thus, the homomorphism ∂_i is called a boundary operator. When there is no risk for confusion, we will denote ∂_i simply by ∂ . A chain c_i is called a cycle, if it is in a kernel of the boundary operator ∂_i , i.e., $\partial c_i = 0$. For example, the boundary $\partial_2 f_m$ of a face f_m is a cycle, because $\partial_2 \circ \partial_1 = 0$. Such a sequence of Abelian groups C_i , connected by the boundary operators ∂_i with $\partial_i \circ \partial_{i-1} = 0$, is called a \mathbf{Z}_2 chain complex. We define a trivial i -cycle c_i if there exists an $(i+1)$ -chain such that $c_i = \partial c_{i+1}$, i.e., $c_i \in \text{Img}(\partial_{i+1})$. By regarding the trivial cycles as generators of the continuous deformation, we may define an equivalent class having the same topology. More precisely, a homology group H_i is defined as a quotient group formed by the cycles and the trivial cycles:

$$H_i = \ker(\partial_i) / \text{Img}(\partial_{i+1}). \quad (3.5)$$

An element of the quotient group, $h \in H_i$, is called a homology class. If two i -chains c_i and c'_i belong to the same homology class, there exists an $(i+1)$ -chain c_{i+1} such that $c_i = c'_i + \partial c_{i+1}$. Two such i -chains c_i and c'_i are said to be homologically equivalent.

We also define the dual surface $\bar{G} = (\bar{V}, \bar{E}, \bar{F})$, where we identify the elements of the dual and original (primal) lattices such that $\bar{V} = F$, $\bar{E} = E$ and $\bar{F} = V$. Specifically, the dual lattice is constructed such that the two vertices $\bar{v}, \bar{v}' \in \bar{V}$ are connected by an edge \bar{e} , if the corresponding two faces f and f' share the same edge e . We can also define a \mathbf{Z}_2 chain complex on \bar{G} using the dual bases $\bar{B}(C_i)$, a dual i -chain $\bar{c}_i \in \bar{C}_i$ and a boundary operator $\bar{\partial}_i : \bar{C}_i \rightarrow \bar{C}_{i-1}$. When there is no risk of confusion, $\bar{\partial}_i$ will be denoted by $\bar{\partial}$.

In the construction of the surface codes, a qubit is defined on each edge $e_l \in E$ of the surface G (or equivalently each dual edge \bar{e}_l of the dual surface \bar{G}). The Pauli product is defined using a 1-chain $c_1 = \sum_l z_l e_l$ (or dual 1-chain \bar{c}_1) such that

$$W(c_1) = \prod_l W_l^{z_l}, \quad (3.6)$$

where $W_l \in \{X_l, Y_l, Z_l\}$ is a Pauli operator acting on the qubit on edge e_l . Specifically, we have

$$W(c_1)W(c'_1) = W(c_1 + c'_1). \quad (3.7)$$

Consider the two operators $X(c_1)$ and $Z(c'_1)$, defined by the two 1-chains c_1 and c'_1 , respectively. The commutability of these two operators is determined by the inner product of the two chains (vectors) $c_1 \cdot c'_1 \equiv \sum_l z_l z'_l$, where the addition is taken modulo 2:

$$c_1 \cdot c'_1 = 0, \quad \text{iff } X(c_1)Z(c'_1) = Z(c'_1)X(c_1) \quad (\text{commute}), \quad (3.8)$$

$$c_1 \cdot c'_1 = 1, \quad \text{iff } X(c_1)Z(c'_1) = -Z(c'_1)X(c_1) \quad (\text{anticommute}). \quad (3.9)$$

Let $M(\partial_i)$ be a matrix representation of ∂_i with respect to the basis vectors $B(C_i)$ and $B(C_{i-1})$. We have

$$(M(\partial_i)c_i) \cdot c_{i-1} = c_i \cdot (M(\partial_i)^T c_{i-1}). \quad (3.10)$$

By identifying the primal and dual bases $B(C_0) = \bar{B}(C_2)$, $B(C_1) = \bar{B}(C_1)$, and $B(C_2) = \bar{B}(C_0)$, the duality relation between primal and dual lattices can be expressed by

$$M(\partial_1) = M(\bar{\partial}_2)^T, \quad M(\partial_2) = M(\bar{\partial}_1)^T, \quad (3.11)$$

where $[\cdot]^T$ indicates the matrix transpose.

Using Eq. (3.4) $\partial_1 \circ \partial_2 = 0$, we obtain $M(\bar{\partial}_2)^T M(\partial_2) = 0$. Thus, for any primal and dual 2-chains, c_2 and \bar{c}_2 , we have

$$\bar{\partial} \bar{c}_2 \cdot \partial c_2 = 0, \quad (3.12)$$

i.e., $X(\partial c_2)$ and $Z(\bar{\partial} \bar{c}_2)$ always commute. This property is useful to construct a stabilizer group, because stabilizer generators have to commute with each other. Using Eqs. (3.10) and (3.11), we also have

$$c_1 \cdot \bar{\partial}_2 \bar{c}_2 = \partial c_1 \cdot \bar{c}_2. \quad (3.13)$$

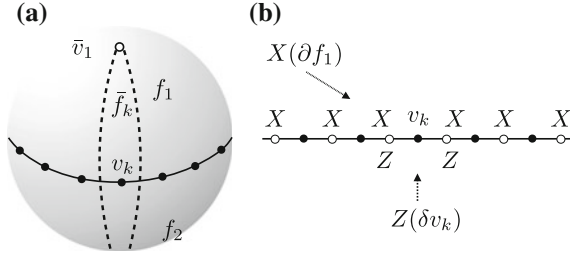


Fig. 3.1 **a** A regular polygon embedded on a sphere. **b** A stabilizer operator defined as $A_k = Z(\partial \bar{f}_k) = \prod_{\delta v_k} Z_k$. The logical X operator is given by $L_X = X(\partial f_1)$. Because primal and dual 1-chains share an even number of qubits, L_X and A_k commute with each other

3.2 A Bit-Flip Code: Exercise

As an exercise, we define a bit-flip code (classical repetition code) using the \mathbf{Z}_2 chain complex. (Readers who are familiar with this topic may skip this section.) Consider a regular polygon $G(V, E, F)$ on a sphere consisting of $n = |E|$ edges and two faces $F = \{f_1, f_2\}$ (corresponding to the top and bottom hemispheres) as shown in Fig. 3.1. The number of qubits is n .

We define a stabilizer generator for each dual face $\bar{f}_k = v_k$ as follows:

$$A_k = Z(\partial \bar{f}_k) = \prod_{e_l \in \delta v_k} Z_l, \quad (3.14)$$

where, for convenience, $\delta v_k \equiv \partial \bar{f}_k$ is defined as a set of edges that are incident to the vertex v_k . (Note that both the dual and primal objects are identified.)

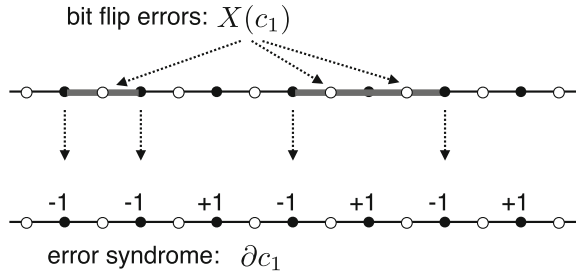
Because $\prod_{v_k \in V} A_k = I$, there are $n - 1$ independent stabilizer generators. The dimension of the stabilizer subspace is two. The code subspace is described by a one-cycle $c_1 = \partial f_1 = \partial f_2$ surrounding the sphere. We define the logical X operator

$$L_X = X(\partial f_1) = X(c_1). \quad (3.15)$$

Note that $L_X = X(\partial f_1)$ and $A_k = Z(\partial \bar{f}_k)$ commute, due to Eq. (3.12). The logical Pauli Z operator, which commutes with $\{A_k\}$ and anticommutes with L_X , is defined as $L_Z = Z_l$. (We may choose any of the Z_l s on the edge e_l , because they act equivalently on the code space.) The code is a bit-flip code, which protects quantum information against bit flip errors.

A string of bit errors $X(c_1)$ is defined using a 1-chain c_1 , which we call an error chain (see Fig. 3.2). The error is detected by measuring the eigenvalues of the stabilizer generators $A_k = Z(\delta v_k)$, i.e., through syndrome measurements. From Eq. (3.13), we have $c_1 \cdot \delta v_k = \partial c_1 \cdot v_k$. Thus, $X(c_1)$ anticommutes with the stabilizer generator A_k if $\partial c_1 \cdot v_k = 1$. Hence, $X(c_1)$ anticommutes with A_k on the vertex v_k that is a boundary $v_k \in \partial c_1$ of the error chain c_1 . The eigenvalue of the stabilizer generator becomes -1 . Error correction is the task of finding a recovery chain c'_1 such that

Fig. 3.2 An X error chain $X(c_1)$ and the corresponding error syndrome ∂c_1 . The eigenvalue of the stabilizer generator becomes -1 on vertices in ∂c_1



$\partial(c_1 + c'_1) = 0$. In the case of the bit-flip code, there are only two possibilities $c_1 + c'_1 = 0$ or $c_1 + c'_1 = \partial f_1$. In the former case, error correction succeeds, while the latter case results in a logical error L_X .

The bit-flip code cannot protect quantum information against phase errors. A natural extension of the classical repetition code to handle both bit and phase errors is the surface codes introduced in the next section.

3.3 Definition of Surface Codes

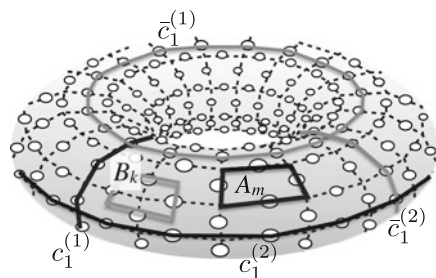
3.3.1 Surface Code on a Torus: Toric Code

Now we are ready to define the surface code. Let us consider a square lattice $G = (V, E, F)$ on a torus with a periodic boundary condition as shown in Fig. 3.3. A dual square lattice $\bar{G} = (\bar{V}, \bar{E}, \bar{F})$ is also defined on the torus. We define stabilizer generators of the Z - and X -types for each face f_m and vertex v_k as follows:

$$A_m = Z(\partial f_m), \quad B_k = X(\delta v_k) = X(\partial \bar{f}_k). \tag{3.16}$$

Because $\partial f_m \cdot \partial \bar{f}_k = 0$ from Eq. (3.12), A_m and B_k commute. The stabilizer generators A_m and B_k are called plaquette and star operators, respectively. By the definition of the stabilizer code, the code state $|\Psi\rangle$ satisfies

Fig. 3.3 A square lattice on a torus with a periodic boundary condition. The Z and X stabilizer generators are defined by being associated with each face and vertex (dual face). The logical Z and X operators are defined by non-trivial cycles



$$A_m|\Psi\rangle = |\Psi\rangle, \quad B_k|\Psi\rangle = |\Psi\rangle, \quad (3.17)$$

for all $f_m \in F$ and $v_k \in V$.

If two 1-chains, c_1 and c'_1 , are homologically equivalent, the actions of $Z(c_1)$ and $Z(c'_1)$ on the code state are the same, because there exists a 2-chain c_2 such that

$$Z(c'_1) = Z(c_1)Z(\partial c_2) = Z(c_1) \left(\prod_{f_m \in c_2} A_m \right). \quad (3.18)$$

We will denote this simply by $Z(c'_1) \sim Z(c_1)$. This is also the case for homologically equivalent dual 1-chains, \bar{c}_1 and \bar{c}'_1 , and the actions of $X(\bar{c}_1)$ and $X(\bar{c}'_1)$ on the code state. Hence, the homology classes, which are equivalent classes over trivial cycles, of the primal and dual 1-chains correspond to the actions of Z - and X -type operators on the code state, because their actions are equal up to stabilizer operators.

Let us define the logical operators $Z(c_1)$ and $X(\bar{c}_1)$. The logical operators have to commute with all stabilizer generators and also be independent of them. The former condition implies that $\partial c_1 = 0$ and $\partial \bar{c}_1 = 0$ from Eq.(3.13). This is because the commutability imposes that $c_1 \cdot \delta v_k = \partial c_1 \cdot v_k = 0$ for all vertices k and $\bar{c}_1 \cdot \delta f_m = \partial \bar{c}_1 \cdot \bar{v}_m = 0$ for all faces (dual vertices) m . The latter condition implies that c_1 and \bar{c}_1 are nontrivial cycles, because for two homologically equivalent cycles c_1 and c'_1 , the actions of $Z(c_1)$ and $Z(c'_1)$ on the code state are the same as seen in Eq. (3.18). We can find two non-trivial cycles, which belong to different homology classes, for each primal and dual 1-chain, as shown in Fig. 3.3. (This is a natural consequence, because the homology group h_1 on the torus is $\mathbf{Z}_2 \times \mathbf{Z}_2$.) Then, we define two pairs of logical Pauli operators

$$\left\{ L_Z^{(1)} = Z(c_1^{(1)}), L_X^{(1)} = X(\bar{c}_1^{(1)}) \right\} \quad \text{and} \quad \left\{ L_Z^{(2)} = Z(c_1^{(2)}), L_X^{(2)} = X(\bar{c}_1^{(2)}) \right\}. \quad (3.19)$$

Note that $L_Z^{(i)}$ and $L_X^{(j)}$ anticommute with each other if $i = j$. Otherwise, they commute:

$$L_Z^{(i)} L_X^{(j)} = (-1)^{\delta_{ij}} L_X^{(j)} L_Z^{(i)}. \quad (3.20)$$

That is, they satisfy a commutation relation equivalent to that of the Pauli operators for two qubits. The logical Pauli basis states are defined as follows:

$$L_Z^{(i)} |\Psi_Z(s_1, s_2)\rangle = (-1)^{s_i} |\Psi_Z(s_1, s_2)\rangle, \quad (3.21)$$

$$L_X^{(i)} |\Psi_X(s_1, s_2)\rangle = (-1)^{s_i} |\Psi_X(s_1, s_2)\rangle. \quad (3.22)$$

The number of stabilizer generators of the $n \times n$ square lattice on the torus is given by

$$|F| + |\bar{F}| - 2 = |F| + |V| - 2 = 2n^2 - 2, \quad (3.23)$$

Table 3.1 The correspondence between \mathbf{Z}_2 chain complex and stabilizer codes

Stabilizer code	Chain complex
Z-type stabilizer generator	Face f_m
X-type stabilizer generator	Vertex v_k (dual face \tilde{f}_k)
Z- and X-type stabilizer operators	Boundaries of 2-chains, $\text{Img}(\partial_2)$ and $\text{Img}(\tilde{\partial}_2)$
Commutability	Cycle conditions: $\partial c_1 = 0$ and $\partial \tilde{c}_1 = 0$
The operators commuting with stabilizer generators	$\ker(\partial_1)$ and $\ker(\tilde{\partial}_1)$
The logical Z and X operators	Nontrivial cycles c_1 and \tilde{c}_1 (homology classes $[c_1] \in \ker(\partial_1)/\text{Img}(\partial_2)$ and $[\tilde{c}_1] \in \ker(\tilde{\partial}_1)/\text{Img}(\tilde{\partial}_2)$)
Z and X errors	1-chains c_1 and \tilde{c}_1
Z and X error syndromes	∂c_1 and $\partial \tilde{c}_1$

where -2 comes from the fact that $\prod_{f_m \in F} A_m = I$ and $\prod_{v_k \in V} B_k = I$, and hence there are two non-independent operators. On the other hand, the number of qubits is given by

$$|E| = 2n^2. \quad (3.24)$$

Thus, we have a $2^{|E|-(|F|+|V|-2)} = 2^2$ -dimensional stabilizer subspace. The above two pairs of logical operators appropriately describe the degrees of freedom in the code space. The code distance, the minimum weight of the logical operators, is determined by the linear length of the square lattice. Thus, the surface code on the torus is an $[[n^2, 2, n]]$ stabilizer code.

In short, an operator that commutes with the stabilizer generators corresponds to a cycle, i.e., $\ker(\partial_1)$. The stabilizer operators correspond to the boundaries of the 2-chains $c_2 \in C_2$. A logical operator, which commutes with and is independent of the stabilizer generators, correspond to a homology class $h_i \in \ker(\partial_1)/\text{Img}(\partial_2)$. We summarize the correspondence between the surface code and the \mathbf{Z}_2 chain complex in Table 3.1.

All properties so far hold for a general tiling $G = (V, E, F)$, and hence we can define a surface code on general, e.g., triangular and hexagonal, lattices [124]. Moreover, the numbers of edges, faces, and vertices are subject to the Euler characteristic formula

$$|F| + |V| - |E| = 2 - 2g, \quad (3.25)$$

where g is the genus, the number of “handles” of the surface. Thus, the dimension of the stabilizer subspace is calculated to be $2^{|E|-(|F|+|V|-2)} = 2^{2g}$. These are the degrees of freedom equivalent to $2g$ qubits in the stabilizer subspace.

Because almost all arguments are made from the operator viewpoint (Heisenberg picture), there is no need to write down $|\Psi_A(s_1, s_2)\rangle$ explicitly. However, for the

readers who worry about such things, we will give an explicit description from the state viewpoint:

$$|\Psi_Z(s_1, s_2)\rangle = \mathcal{N} Z \left(c_1^{(1)} \right)^{s_1} Z \left(c_1^{(2)} \right)^{s_2} \left(\prod_{v_k \in V} \frac{I + B_k}{2} \right) |0\rangle^{\otimes n^2} \quad (3.26)$$

$$|\Psi_X(s_1, s_2)\rangle = \mathcal{N} X \left(\bar{c}_1^{(1)} \right)^{s_1} X \left(\bar{c}_1^{(2)} \right)^{s_2} \left(\prod_{f_m \in F} \frac{I + A_m}{2} \right) |+\rangle^{\otimes n^2} \quad (3.27)$$

where \mathcal{N} is a normalization factor. The code state can be viewed as an equal weight superposition of all cycles belonging to the same homology class.

3.3.2 Planar Surface Code

The periodic boundary condition might be hard to implement in experiments. We can also define a surface code on a planar $n \times (n - 1)$ square lattice with an appropriate boundary condition as shown in Fig. 3.4 [125]. The top and bottom boundaries consist of three-qubit star operators, which we call smooth boundaries, because they are complete plaquette operators. On the other hand, the left and right boundaries consist of three-qubit plaquette operators, which we call rough boundaries. The X operators on a dual 1-chain can terminate at the smooth boundary, while the Z operators on a 1-chain can terminate at the rough boundary. Thus, we should use a relative homology to define logical operators, where two chains c_i and c'_i are said to be (relative) homologically equivalent iff $c'_i = c_i + \partial c_{i+1} + \gamma_i$ with $\gamma_i \in \Gamma_i \subset C_i$. In this case, Γ_1 is chosen, specifically, to be the vector space spanned by the set of 1-chains each of which consists of three edges corresponding to the three-qubit plaquette operator at the top and bottom smooth boundaries. Similarly, we also define $\bar{\Gamma}_1$ at the left and right rough boundaries of the dual lattice. Because $Z(\gamma_1)$ and $X(\bar{\gamma}_1)$ are both stabilizer operators, if the shapes of two logical operators are homologically equivalent, their actions on the code state are the same. The number of edges, faces, and vertices are now $|E| = 2n^2 - 2n + 1$, $|V| = n^2 - n$, and $|F| = n^2 - n$. Thus, the stabilizer subspace is a 2D subspace. We can define the logical operators $L_Z = Z(c_1)$ and $L_X = X(\bar{c}_1)$ using horizontal and vertical 1-chains c_1 and \bar{c}_1 , as shown in Fig. 3.4.

If one chooses all boundaries smooth, the stabilizer state is uniquely defined, and hence there is no logical degree of freedom. By punching a hole, we can introduce a defect on the surface, whereby we can define a nontrivial cycle for a logical Z operator, as shown in Fig. 3.5. The logical X operator can be chosen as a dual 1-chain that connects the defect and the smooth boundary, as shown in Fig. 3.5. As mentioned previously, the properties of the logical operators are the same if the corresponding 1-chains are homologically equivalent.

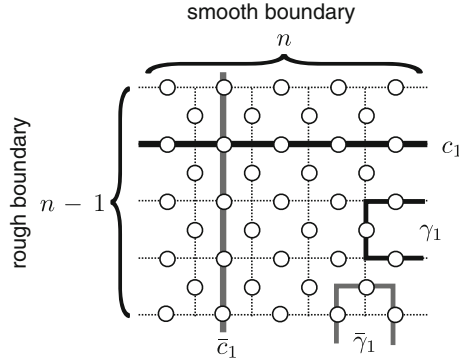
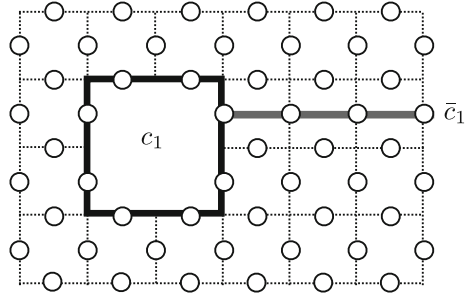


Fig. 3.4 A planar surface code. *Top* and *bottom* are smooth boundaries consisting of three-qubit star operators. *Left* and *right* are rough boundaries consisting of three-qubit plaquette operators. At the boundary, we can define sets of 1-chains $\Gamma_1 = \{\gamma_1\} \in C_1$ and $\bar{\Gamma}_1 = \{\bar{\gamma}_1\} \in \bar{C}_1$ for the primal and dual lattices, respectively, by which the relative homology is defined

Fig. 3.5 A planar surface code with a defect. The stabilizer generators are not defined inside a defect, which provides a nontrivial cycle wrapping around it



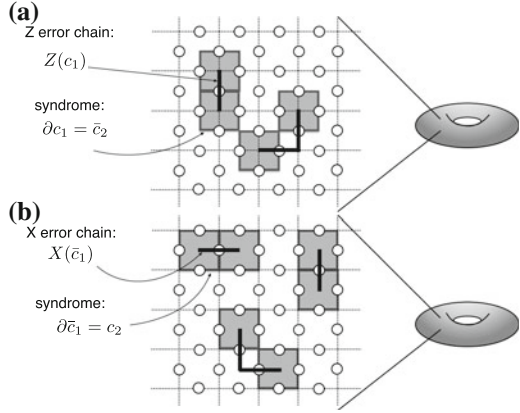
3.4 Topological Quantum Error Correction

Let us return to the surface code on the torus to explain how errors are corrected. There are 2^{2n^2-2} orthogonal subspaces, the so-called syndrome subspaces, each of which is an eigenspace of the stabilizer generators and has the same structure as the code space. These orthogonal subspaces are utilized to identify the location of errors and to infer a recovery operation. Suppose the X and Z errors $X(\bar{c}_1^e)$ and $Z(c_1^e)$, defined by error 1-chains \bar{c}_1^e and c_1^e , respectively, occur on the surface code, as shown in Fig. 3.6. The code state is now mapped into one of the orthogonal subspaces. From Eq. (3.13), $X(\bar{c}_1^e)$ and $Z(c_1^e)$ anticommute with A_m and B_k on the face $f_m \in \partial\bar{c}_1^e$ and vertex $v_k \in \partial c_1^e$. Thus, the orthogonal subspace is specified by error syndromes

$$\partial\bar{c}_1^e \equiv c_2^s, \quad \partial c_1^e \equiv c_0^s. \tag{3.28}$$

More precisely, the eigenvalues with respect to the stabilizer generators A_m and B_k are given by $(-1)^{\bar{z}_m^s}$ and $(-1)^{z_k^s}$, where $c_2^s = \sum_m z_m^s f_m$ and $c_0^s = \sum_k z_k^s v_k$ (see Fig. 3.6), respectively. Error correction is the task of finding recovery 1-chains \bar{c}_1^r and c_1^r such

Fig. 3.6 **a** A Z error chain $Z(c_1)$ is detected at the boundary ∂c_1 . The eigenvalue of the stabilizer generator B_k on the face $f_k \in \bar{c}_2 = \partial c_1$ becomes -1 . **b** An X error chain $X(\bar{c}_1)$ is detected at the boundary $\partial \bar{c}_1$. The eigenvalue of the stabilizer generator A_m on the face $f_m \in c_2 = \partial \bar{c}_1$ becomes -1



that $\partial(\bar{c}_1^e + \bar{c}_1^r) = 0$ and $\partial(c_1^e + c_1^r) = 0$, meaning that the state is returned into the code space by applying $X(\bar{c}_1^r)$ and $Z(c_1^r)$, respectively. Below we will, for simplicity, explain only how to correct the Z errors, but the extension to the X errors is straightforward.

Suppose each Z error occurs with an independent and identical probability p . Conditioned on the error syndrome $c_0^s = \partial c_1^e$, the posterior probability of an error $Z(c_1)$ occurring with $c_1 = \sum_l z_l e_l$ is written as

$$P(c_1 | c_0^s) = \mathcal{N} \prod_l \left(\frac{p}{1-p} \right)^{z_l} \Big|_{\partial c_1 = c_0^s} \quad (3.29)$$

where \mathcal{N} is a normalization factor. One way to find a recovery chain is by maximizing the posterior probability

$$c_1^r \equiv \arg \max_{c_1} P(c_1 | c_0^s) = \arg \min_{c_1} \left(\sum_l z_l \right) \Big|_{\partial c_1 = c_0^s}, \quad (3.30)$$

where $c_1 = \sum_l z_l e_l$. As seen in the l.h.s., this corresponds to minimizing the number of errors $\sum_l z_l^r$ such that $\partial c_1^r = c_0^s$. Hence, this is called minimum distance decoding. If $c_1^r + c_1^e$ is a trivial cycle, the error correction succeeds, because the net action $Z(c_1^r + c_1^e)$ after the recovery operation is identity on the code state as shown in Fig. 3.7. If $c_1^r + c_1^e$ is a nontrivial cycle, the recovery operation results in the logical operation $Z(c_1^r + c_1^e) \sim L_Z$, i.e., a logical error.

Minimum distance decoding is hard in general, because it can be mapped into an integer programming problem, which is NP-hard [126, 127, 128]. However, in the present case, there is a nice geometrical property that makes the decoding problem feasible. The condition $\partial c_1 = c_0^s$ and \min_{c_1} read that it is sufficient to find a 1-chain that connects pairs of two vertices in c_0^s with a minimum Manhattan length. There is a classical polynomial time algorithm to do such a task, the so-called minimum-weight perfect matching (MWPM) algorithm of Edmonds [129, 130, 131]. The algorithm

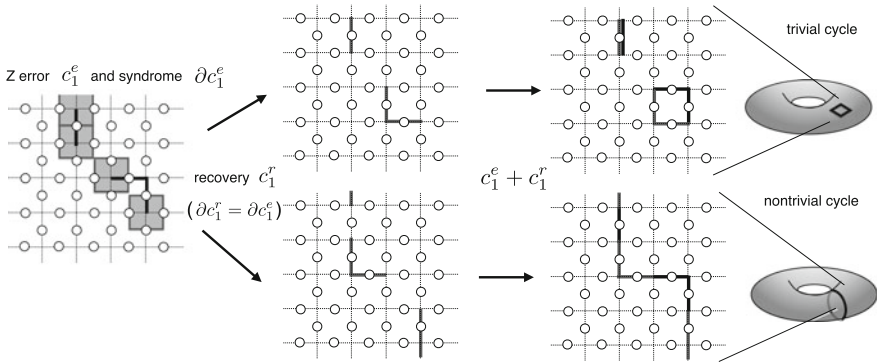


Fig. 3.7 From left to right, error chain $Z(c_1^e)$, recovery chain $Z(c_1^r)$ with $c_1^r = c_0^s$, and net actions $Z(c_1^r + c_1^e)$. If $c_1^r + c_1^e$ is a trivial cycle (top), error correction succeeds. If $c_1^r + c_1^e$ is a non-trivial cycle (bottom), the error correction results in a logical error

scales like $O(n^6)$, with n being the linear length of the lattice and with the fixed error probability p . Typical examples of the error chain c_1^e and the error plus recovery chain $c_1^e + c_1^r$ are shown in Fig. 3.8 for each $p = 0.05$ (b), $p = 0.10$ (c), and $p = 0.15$ (d). Here, we employ an implementation of MWPM, blossom V [132]. The higher the physical error probability is, the longer the error plus recovery chain becomes. For a high physical error probability $p = 0.15$, the error plus recovery chain becomes a large cycle, and unfortunately results in a logical error. Such a logical error probability

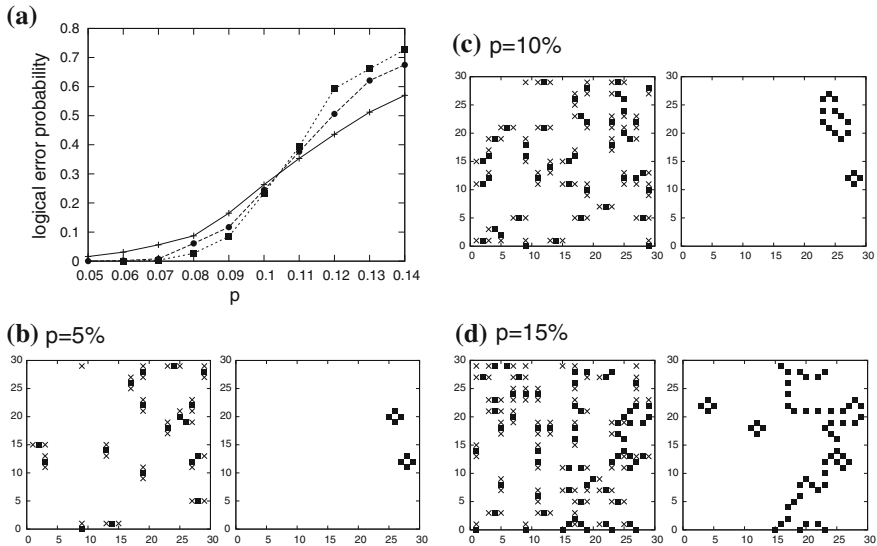


Fig. 3.8 a The logical error probability is plotted as a function of the physical error probability p . Error chains c_1^e and error plus recovery chains $c_1^e + c_1^r$ for $p = 0.05$ (b), $p = 0.10$ (c), and $p = 0.15$ (d)

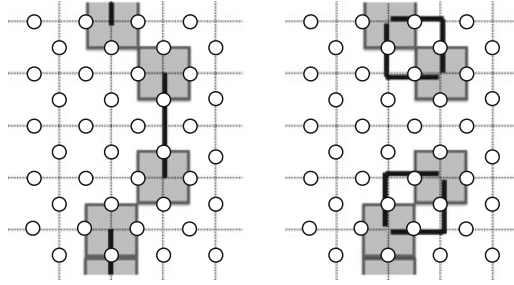


Fig. 3.9 Suppose the *top* and *bottom* boundaries are connected by a periodic boundary condition. The minimum-weight error is shown in the *left panel*. The error shown in the *right panel* is not minimum-weight but has a fourfold degeneracy. The recovery chain should be chosen by comparing the total error probabilities p^3 and $4 \times p^4$

is plotted as a function of the physical error probability p in Fig. 3.8a for each $n = 10$ (solid line), $n = 20$ (dashed line), and $n = 30$ (dotted line). If the error probability is sufficiently smaller than a threshold value, the logical error probability decreases for increasing n . The threshold value for decoding by the MWPM algorithm has been estimated to be 10.3% (MWPM) [111, 133].

The minimum distance decoding with MWPM is not optimal for our purpose, i.e., making the logical error probability as small as possible. An error correction with another recovery chain $c_1^{r'}$, for which $c_1^e + c_1^{r'}$ belongs to the same homology class as $c_1^e + c_1^r$, provides exactly the same result. We may use such a recovery chain $c_1^{r'}$ to correct the error. Thus, we should maximize, not each posterior probability $p(c_1^r | c_0^s)$, but a summation of it over the same homology class. This problem originated from the degeneracy of the surface code, where each syndrome is assigned not uniquely, but for many error instances. A prototypical example of an error syndrome, for which we should consider not only the weight of errors, but also combinatorics (an entropic effect) of the error configurations, is shown in Fig. 3.9.

Denoting the homology class by h_i , the posterior probability for a homology class h_i is given by

$$p_i = \sum_{c_1^r | c_1^e + c_1^r \in h_i} P(c_1^r | c_0^s), \quad (3.31)$$

where c_1^r is a recovery chain satisfying $\partial c_1^r = c_0^s$ and chosen arbitrarily as a reference frame, and the summation is taken over all 1-chains c_1^r such that $c_1^e + c_1^r$ belongs to the homology class h_i .

The posterior probability may be rephrased by using the stabilizer language as follows (see also Appendix B). Let \mathcal{S} and \mathcal{L} be the stabilizer and logical operator groups, respectively. For a given error syndrome c_0^s , we define the recovery operator $Z(c_1^r)$ a priori, such that the erroneous state is returned into the code space. We can decompose an arbitrary error operator $Z(c_1^e)$, providing the syndrome c_0^s , into

$$Z(c_1^e) = Z(c_1^r) GL_i, \quad (3.32)$$

where $G \in \mathcal{G}$ and $L_i \in \mathcal{L}$ are a stabilizer and logical operator, respectively. The posterior probability of the logical operator L_i is calculated by summing over all stabilizer operators $G \in \mathcal{G}$

$$p_i = P(L_i | c_0^s) = \frac{1}{\mathcal{N}} \sum_{G \in \mathcal{G}} P[Z(c_1^r) GL_i]. \quad (3.33)$$

We choose the most likely homology class h_i or, equivalently, the most likely logical operator L_i that maximizes the probability $p_i = P(L_i)$:

$$L_{\bar{i}} \equiv \arg \max_{L_i} P(L_i | c_0^s). \quad (3.34)$$

The error correction is completed by applying $L_{\bar{i}} Z(c_1^r)$, and the logical error probability is given by $1 - p_{\bar{i}}$.

In the next section, we relate the posterior probability summed over the same homology class to a partition function of the random-bond Ising model. The partition function of the Ising model on a planar graph with general coupling strengths (without magnetic fields) can be calculated in polynomial time using the Kasteleyn-Barahona algorithm with the Pfaffian method [130, 134, 135]. Thus, the optimal decoding is also implemented by a polynomial time classical processing, though it takes more overhead than MWPM. The threshold value for optimal decoding is also discussed in the next section.

Efficient decoding has been one of the most important issues for the realization of fault-tolerant quantum computing, because the coherence time of quantum information would be very short; a fast classical processing is essential. Fowler et al. proposed an efficient decoding method based on MWPM [136]. Because a long error chain is exponentially suppressed, we can assign weights between each pairs of vertices having -1 eigenvalues according to their Manhattan length. This allows us to reduce the number of edges from $O(n^4)$ to $O(n^2)$. Moreover, the exponential suppression of longer error chains allows us to search the pairs within a local small region; matching a long-distance pair is exponentially rare. Because the matching process employs almost exclusively local information, this algorithm can be parallelized to an $O(1)$ average time per round.

Another decoding method is using a renormalization technique [137, 138]. As explained in Appendix B, we can efficiently execute an optimal decoding on a concatenated quantum code by using a brief-propagation on a tree factor graph [139]. Although the surface code itself does not have such a hierarchal structure, a renormalization technique is employed to introduce a hierarchal structure on the surface code [137, 138].

Using 12 qubits as a unit cell, we define a couple of level-1 logical qubits, as shown in Fig. 3.10a, which include 2 pairs of logical operators $\mathcal{L}^{(1)}$, 6 stabilizer generators $\mathcal{G}^{(1)}$. We define 6 pure error operators $\mathcal{G}^{(1)}$ each of which anticommutes with a stabilizer generator as shown in Fig. 3.10a (middle). Moreover, we define 4 pairs of anticommuting operators as shown in Fig. 3.10a (bottom), which we call

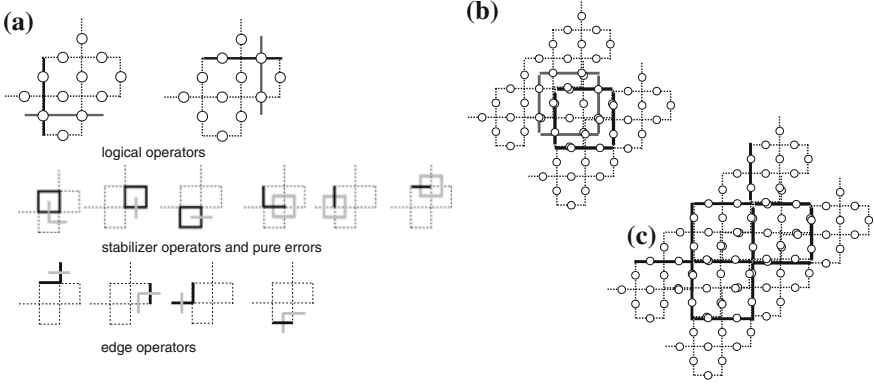


Fig. 3.10 **a** A unit cell of the level-1 logical qubit. Two pairs of logical Pauli operators (*top*). The 6 stabilizer operators and the pure error operators, each of which anticommutes with a stabilizer operator (*middle*). The 4 pairs of edge operators. These 12 pairs of mutually anti-commuting operators generate the Pauli group of the 12 qubits. **b** Using the level-1 logical operators, the level-2 stabilizer generators are defined. **c** Using 12 level-1 logical qubits, a unit cell of the level-2 logical qubit is defined

edge operators $\mathcal{E}^{(1)}$, to generate the Pauli group of the 12 qubits. Any Pauli operator A on the unit cell can be decomposed in terms of these operator,

$$A = L^{(1)}G^{(1)}\bar{G}^{(1)}E^{(1)}, \quad (3.35)$$

where $B^{(1)} \in \mathcal{B}^{(1)}$ for $B = L, G, \bar{G}, E$. The pure error operator $\bar{G}^{(1)}$ is chosen uniquely according to the error syndrome $S^{(1)}$ to return the state into the code space. (If a stabilizer generator G_1 has an eigenvalue -1 , then we employ \bar{G}_1 , which anticommutes with G_1 .) From the error distribution $P(A)$, the posterior probability of the level-1 logical operator is calculated by taking a marginal over $\mathcal{G}^{(1)}$ and $\mathcal{E}^{(1)}$,

$$P(L^{(1)}|S^{(1)}) = \sum_{G^{(1)} \in \mathcal{G}^{(1)}, E^{(1)} \in \mathcal{E}^{(1)}} P(A = L^{(1)}G^{(1)}\bar{G}^{(1)}E^{(1)}|S^{(1)}), \quad (3.36)$$

which are utilized to model the error distribution at the level 2.

Similarly, a level- k unit cell is defined by using 8 level- $(k-1)$ unit cells and the 12 pairs of the level- $(k-1)$ logical operators on them. Similarly to the level-1 case, we define the level- k logical, stabilizer, pure error, edge operators. At the highest level $k = l$, we obtain the logical operators of the surface code. For example, the level-2 stabilizer generators are shown in Fig. 3.10b. A unit cell of the level-2 logical qubit is shown in Fig. 3.10c.

The posterior probability of the level- k logical operator $P(L^{(k)}|S^{(k)})$ is calculated using the posterior probabilities at the level $(k-1)$ by assuming that they are independent for each level- $(k-1)$ unit cell. Under this assumption, we can calculate the posterior probability $P(L^{(l)}|S^{(l)})$ conditioned on all error syndrome $S^{(l)}$ by using the belief propagation [137, 138, 139] (see Appendix B for decoding the concatenated

codes by using the belief propagation). Then the maximization procedure over all logical operators provide us a most likely logical operator,

$$L^* = \arg \max_{L^{(l)}} P(L^{(l)} | S^{(l)}). \quad (3.37)$$

The level- $(k - 1)$ unit cells share physical qubits with each other, and hence the conditional logical error probability is not independent for level- $(k - 1)$ unit cells. If we employ a message passing in order to reweigh the level- $(k - 1)$ logical error probability as the error model for the level- k unit cell, we can further improve the approximation of $P(L^{(l)} | S^{(l)})$. If the decoding process is done in parallel, the belief propagation takes $O(\log_2(n))$ time, similar to the case of the concatenated code as explained in Appendix B. Because the posterior probability is approximated, the decoding based on the renormalization is not optimal, but results in a reasonable threshold value, $\sim 9\%$ for the independent X and Z error and $\sim 15.2\%$ for the depolarizing error (against 10.3 and 15.5% obtained by MWPM, respectively) [137, 138]. The renormalization decoder can also be applied for an arbitrary topological code [140].

3.5 Error Correction and Spin Glass Model

The behavior of the logical error probability in Fig. 3.8 suggests the existence of a critical phenomenon behind the error correction problem. Indeed, there is a beautiful correspondence between quantum error correction on the surface code and a spin glass model, the so-called random-bond Ising model (RBIM) [111]. More precisely, the posterior probability Eq. (3.31) of the logical operator is mapped into a partition function of the RBIM as seen below.

To solve the condition $\partial c_1 = c_0^s$ in Eq. (3.29), we rewrite the recovery chain as $c_1 = c_1^{r_k} + c_1^t$, where $c_1^t \in \text{Img}(\partial_2)$ is a trivial cycle. $c_1^{r_k}$ is further decomposed into $c_1^{r_k} = c_1^e + c_1^{(k)}$, where c_1^e determines the actual location of errors, and $c_1^{(k)}$ is a (nontrivial) cycle belonging to the homology class h_k . Note that this decomposition corresponds to Eq. (3.32). The posterior probability is rewritten as

$$P(c_1 | c_0^s) = \mathcal{N} \prod_l \left(\frac{p}{1-p} \right)^{z_l^t \oplus z_l^{r_k}} \Big|_{c_1^t \in \text{Img}(\partial_2)}, \quad (3.38)$$

where $c_1^\alpha = z_l^\alpha e_l$, with $z_l^\alpha \in \{0, 1\}$. In order to take the condition $c_1^t \in \text{Img}(\partial_2)$ automatically, we introduce a gauge valuable $z_m^s \in \{0, 1\}$ on each dual vertex \tilde{v}_m and a dual 0-chain $\tilde{c}_0^s = \sum_m z_m^s \tilde{v}_m$. Any trivial cycle c_1^t is replaced by the gauge valuables using the following relation

$$z_l^t = \bigoplus_{\tilde{v}_m \in \partial \tilde{e}_l} z_m^s. \quad (3.39)$$

For example, if we choose $\vec{c}_0^g = \vec{v}_m$, we obtain $c_1^t = \partial f_m$. (This corresponds to a multiplication of the face stabilizer generator defined on the face f_m .) There is a one-to-one correspondence between a trivial cycle and a gauge dual 0-chain. Using c_0^g we can formally solve the condition $c_1^t \in \text{Img}(\partial_2)$ in Eq. (3.29),

$$P(c_1|c_0^s) = \mathcal{N} \prod_l \left(\frac{p}{1-p} \right)^{z_l^{r_k} \oplus_{\vec{v}_m \in \partial e_l} z_m^g}. \quad (3.40)$$

The binary valuables $z_i \in \{0, 1\}$ are transformed into spin variables $\sigma_i \in \{+1, -1\}$ by $\sigma_i = (-1)^{z_i}$. Moreover, we define a coupling constant $e^{-J} = \sqrt{p/(1-p)}$. Then Eq. (3.40) is rewritten as

$$P(c_1|c_0^s) = \mathcal{N}' e^{J \sum_l \sigma_l^{r_k} \sigma_{m(l)}^g \sigma_{m'(l)}^g}, \quad (3.41)$$

where \mathcal{N}' is a normalization factor and $m(l)$ and $m'(l)$ are end points of the edge e_l . By changing the notation $l \rightarrow ij$ and $m(l), m'(l) \rightarrow i, j$, the posterior probability is reformulated as a Boltzmann factor of the $\pm J$ RBIM:

$$P(c_1|c_0^s) = \mathcal{N}' e^{\sum_{ij} J_{ij}^{(k)} \sigma_i^g \sigma_j^g}, \quad (3.42)$$

where $J_{ij}^{(k)} = J \sigma_{ij}^{r_k}$. On the edges where the errors are located, anti-ferromagnetic interactions are assigned. The posterior probability of the logical operator Eq. (3.31) is calculated by taking summation over all gauge spin configurations (this corresponds to the summation over all stabilizer operators in Eq. (3.31)):

$$p_k = \sum_{c_1^r | c_1^k + c_1^r \in h_k} P(c_1^r | c_0^s) \quad (3.43)$$

$$= \sum_{c_1^r | c_1^r \in \text{Img}(\partial_2)} P(c_1^r + c_1^k | c_0^s) \quad (3.44)$$

$$= \sum_{\{\sigma_i^g\}} \mathcal{N}' e^{\sum_{ij} J_{ij}^{(k)} \sigma_i^g \sigma_j^g}. \quad (3.45)$$

$$= \mathcal{N}' \mathcal{Z}(\{J_{ij}^{(k)}\}), \quad (3.46)$$

where $\mathcal{Z}(\{J_{ij}^{(k)}\})$ is the partition function of the $\pm J$ RBIM.

Let us consider the performance under this decoding by taking an average of the logarithm of p_k over the error distribution, which corresponds to a sample average with respect to quenched randomness:

$$\bar{p}_k^{\text{ln}} \equiv [\ln(p_k)] = \sum_{c_1^e} P(c_1^e) p_k \quad (3.47)$$

Table 3.2 The correspondence among random-bond Ising model, \mathbf{Z}_2 chain complex and stabilizer codes

RBIM	Stabilizer code	Chain complex
Dual lattice	Z error correction	Primal lattice
Primal lattice	X error correction	Dual lattice
Ising interactions	Qubits	Edge e_l
Gauge spin	Z-type stabilizer generators	Face f_m
Gauge spin	X-type stabilizer generators	Vertex v_k (dual face \bar{f}_k)
Domain wall	Logical X and Z operators	Nontrivial cycles c_1 and \bar{c}_1
Anti-ferromagnetic interactions	X and Z errors	1-chains c_1 and \bar{c}_1
Frustration	X and Z error syndromes	∂c_1 and $\partial \bar{c}_1$

$$= \sum_{\{J_{ij}\}} P(\{J_{ij}^e\}) \ln \mathcal{Z}(\{J_{ij}^{(k)}\}) + \ln \mathcal{N}' \quad (3.48)$$

$$\equiv -F_k + \ln \mathcal{N}', \quad (3.49)$$

where $[\cdot]$ indicates the sample average and \bar{p}_k^{\ln} is the sample average of the logarithm of the logical error probability p_k . The quenched randomness is determined by the error distribution

$$P(J_{ij}^e) = (1 - p')\delta(J_{ij}^e - J) + p'\delta(J_{ij}^e + J), \quad (3.50)$$

where $e^{-J} = \sqrt{p/(1-p)}$. Note that p is a parameter in the posterior probability and could be different from the actual error probability p' . Of course, an optimal decoding, in the sense of Bayesian inference, is achieved by maximizing the posterior probability with a true error probability $p = p'$. $F_0 = -[\ln \mathcal{Z}(\{J_{ij}^{(0)}\})]$ is the free-energy of the RBIM, while $F_k = -[\ln \mathcal{Z}(\{J_{ij}^{(k)}\})]$ ($k = 1, 2, 3$) is the free-energy with respect to the interactions $\{J_{ij}^{(k)}\}$, where a domain wall of anti-ferromagnetic interactions corresponding to a cycle $c_1^{(k)}$ of a homology class h_k is inserted.

From Eqs. (3.46) and (3.49), the relation between quantum error correction and a spin glass model becomes apparent; the posterior probability of the logical operator is proportional to the partition function of RBIM, whose Hamiltonian is given by $H = -\sum_{ij} J_{ij}^{(k)} \sigma_i^g \sigma_j^g$. The location of the Z errors, represented by $J_{ij}^{(k)} = -J$, corresponds to the anti-ferromagnetic interaction due to disorder as shown in Fig. 3.11 (see also Table 3.2). The error syndrome $c_0^s = \partial c_1^e$ corresponds to the distribution of frustrations of the Ising interactions and also the end points (Ising vortex) of the excited domain walls. The probability of the homology class is expressed by the domain-wall free energy: (Fig. 3.12)

$$-(\bar{p}_k^{\ln} - \bar{p}_0^{\ln}) = F_k - F_0. \quad (3.51)$$

If the physical error probability p is smaller than the threshold value, $-\bar{p}_k^{\ln}$ ($k = 1, 2, 3$) diverges in the large n limit. On the other hand, if p is higher than the

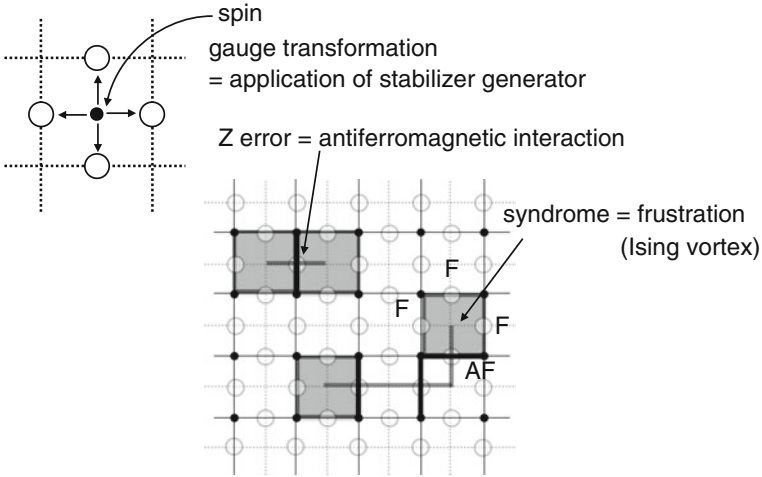


Fig. 3.11 A gauge spin σ_j^g located at the face-center of a plaquette. The distribution of anti-ferromagnetic interactions corresponds to the error chain c_1^e . The error syndrome $c_0^s = \partial c_1^e$ corresponds to the distribution of frustrations. The ground state configuration is determined by a domain-wall consisting of excited domain-wall and anti-ferromagnetic bonds, both of which have frustrations at their end points

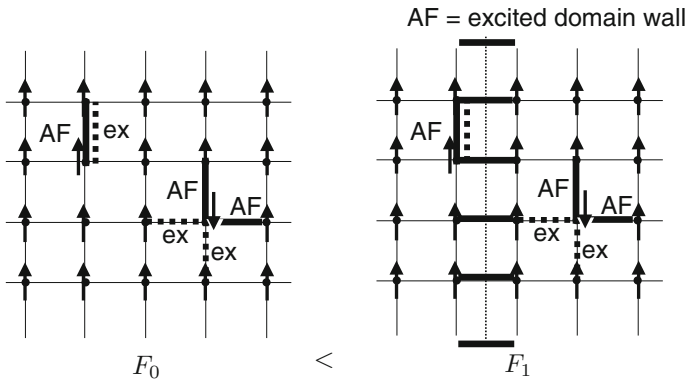
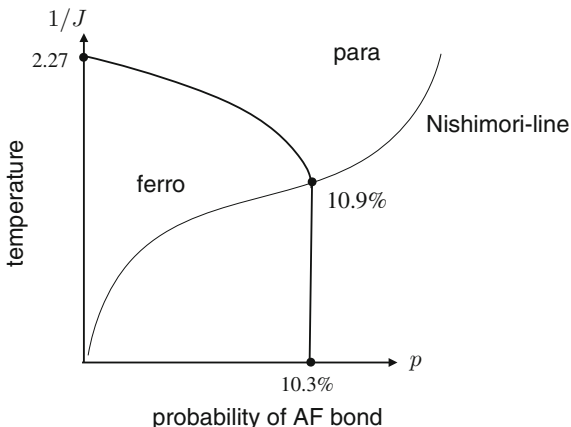


Fig. 3.12 Examples of ground states in a ferromagnetic phase are shown for $\{J_{ij}^{(k)}\}$ with $k = 0, 1$. In the ferromagnetic case, the insertion of the anti-ferromagnetic bonds results in an excited domain-wall and hence increases the free energy (ground state energy at zero temperature)

threshold value, $-\bar{p}_k^{\ln}$ ($k = 0, 1, 2, 3$) converges to $2 \ln 2$ meaning the stored information becomes completely destroyed. This non-analytical behavior also appeared in the r.h.s. of Eq.(3.49), i.e., the free-energy of RBIM. Indeed, the difference between the free energies $\Delta = F_k - F_0$ is an order parameter, the so-called domain-wall free energy [141], for a ferromagnetic ordered phase. The phase diagram of RBIM with respect to $J = (1/2) \ln[p/(1 - p)]$ and p' is shown in Fig. 3.13. As mentioned before, optimal decoding is achieved with $p = p'$, i.e., $e^{-J} = \sqrt{p'/(1 - p')}$, which

Fig. 3.13 A phase diagram of RBIM with respect to the coupling strength $J = (1/2) \ln[p/(1 - p)]$ and the probability of anti-ferromagnetic interaction p' . The probability p in j corresponds to the parameter in the posterior probability. When $p = p'$, called the Nishimori line, an optimal decoding is achieved



is called the Nishimori line [142]. The critical point on the Nishimori line, which is called a multi-critical point, has been numerically calculated to be $10.94 \pm 0.02 \%$ by Honecker et al. and Merz et al. [143, 144, 145]. The optimal threshold value of the surface code is good in the following sense: The existence of CSS codes with asymptotic rate $R \equiv k/n$ is guaranteed if

$$R = 1 - 2H_2(p), \tag{3.52}$$

by the quantum Gilbert-Varshamov bound under independent X and Z errors with probability p [146]. The rate becomes zero with $p = 11.00 \%$. The optimal threshold of the surface code, which consists only of local stabilizer generators, achieves a value very close to this. Indeed, Nishimori conjectured that the multi-critical point of the RBIM is determined by

$$H_2(p) = 1/2, \tag{3.53}$$

arguing from the self-duality in RBIM with a replica method [147, 148, 149] (if the reader is interested this derivation, please see a review in Ref. [150]). Ohzeki has evaluated the multi-critical point precisely to be $p = 0.109187 \%$ by using a real-space renormalization [151], which is in a good agreement with the numerical result $p = 0.10917(3)$ [152, 153].

The minimum distance decoding with MWPM is achieved in the limit $p \rightarrow 0$, which is the low temperature limit $J \rightarrow \infty$ where the entropic effect is suppressed. The threshold of MWPM corresponds to the critical point at zero temperature [130], which has been investigated numerically and determined to be 10.4 ± 0.1 by Kawashima et al. [111, 133, 154].

3.6 Other Topological Codes

The surface codes have been also studied on general lattice tilings, such as, triangle, hexagonal, and random lattices [124, 151, 155, 156, 157, 158]. Suppose a surface code is defined on a lattice $G = (V, E, F)$. As seen in the previous section, the X error correction is mapped into a RBIM on a lattice $G = (V, E)$. On the other hand, the Z error correction is mapped into a RBIM on its dual lattice $\bar{G} = (\bar{V}, \bar{E})$. The mutual duality relation [147] between G and \bar{G} allows us to predict the relation between the optimal threshold values for X and Z errors:

$$H(p_x) + H(p_z) = 1, \quad (3.54)$$

where p_x and p_z are the X and Z error probabilities, respectively. This equality is the same as the quantum Gilbert-Varshamov bound evaluated for the independent X and Z errors with probabilities p_x and p_z , respectively. The precise locations of the optimal thresholds for regular lattices have been investigated by Ohzeki using a real-space renormalization technique [151]. The thresholds have been investigated using the MWPM algorithm by Fujii et al. [124]. The thresholds with MWPM (i.e., the critical points of RBIMs at zero temperature) even approaches Eq. (3.54), as shown in Fig. 3.14. These codes with an asymmetry between the X and Z error tolerances would be useful to correct a biased error [159]. In Refs [155, 156], the asymmetry is continuously controlled by changing a lattice parameter.

A leakage process or qubit loss is an important source of noise. Unlike the X and Z errors discussed so far, the qubit loss is detectable (heralded), and hence we

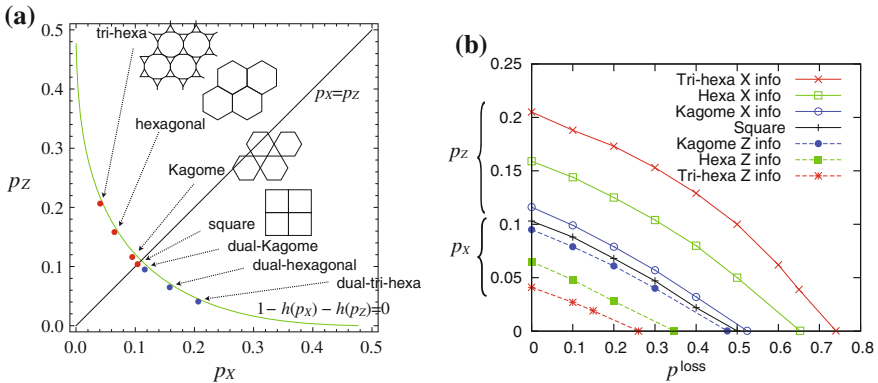


Fig. 3.14 **a** The thresholds (p_x, p_z) of the surface codes on square, Kagome, hexagonal, and triangle-hexagonal lattices, as well as their duals. The curve is the quantum Gilbert-Varshamov bound hitting zero asymptotic rate under a bit and phase flip channel with probabilities p_x and p_z , respectively. **b** The trade-off curves between qubit loss rate p^{loss} and unheralded error rates p_x and p_z are shown for square, Kagome, hexagonal, and triangle-hexagonal lattices. If the loss and error rates are below these curves, the logical information is protected

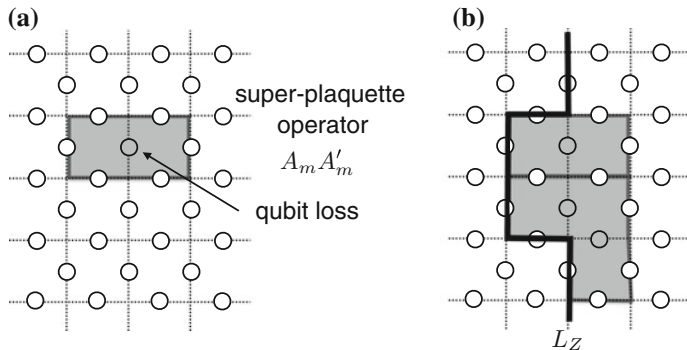


Fig. 3.15 **a** A super-plaquette defined as a product of two plaquette operators. The lost qubit is not contained in the super-plaquette operator. **b** The logical operator is chosen appropriately by avoiding the lost qubits. The lost qubits (bonds) are not percolated, so we can find such a logical operator

can tolerate much more loss rate than for the (undetectable) error rate. Stace et al. proposed to cope with the qubit loss on the surface code [160, 161]. Suppose a qubit is lost on the surface as shown in Fig. 3.15a. The plaquette operators containing the lost qubit are undetermined. However, we can construct a *super-plaquette* multiplying two neighboring plaquette so that the super-plaquette does not have the lost qubit. By using the super-plaquette as a stabilizer generator, we can perform MWPM. If two super-plaquettes are neighbors, the weight between these super-plaquettes has to be modified appropriately in MWPM, because the plaquettes share two physical qubits and the error probability is effectively doubled. The logical operator is defined by avoiding the lost qubits. Unless the lost qubits, which are located on edges, are percolated throughout the lattice, we can find such a logical operator. Thus, the threshold for qubit loss in the large lattice limit without any (undetected) error is determined by the bond percolation threshold. The trade-off curves between qubit loss and (unheralded) error rates with MWPM for various lattice tillings are shown in Fig. 3.14b. The leakage error in a more realistic situation has been investigated and counteracted in the superconducting systems [162, 163].

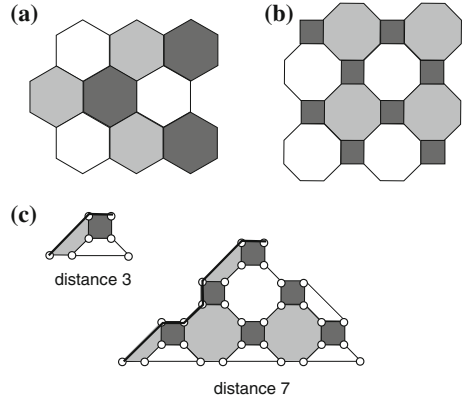
On the other hand, when the error correction problem is mapped into the RBIM, the qubit loss corresponds to a bond-dilution. The duality relation in the presence of the bond-dilution on the square lattice is given by [164]

$$(1 - q)h(p) + q = 1/2. \quad (3.55)$$

From the numerical data in Ref. [124], we could expect a more general equality,

$$(1 - q_x)h(p_x) + (1 - q_z)h(p_z) + q_x + q_z = 1, \quad (3.56)$$

Fig. 3.16 **a, b** Trivalent lattices that can be colored with three distinct colors. **c** Color codes defined on (4.8.8) lattices with a triangle open boundary condition. The logical operators are shown by *solid lines*. Any string at the boundaries can be employed as a logical operator



where q_x and q_z are the probabilities of the heralded X and Z errors, respectively. With $p_x = p_z = 0$, this is reduced to $q_x + q_z = 1$, which corresponds to Kesten's duality relation of the bond percolation thresholds between mutually dual lattices. With $q_x = q_z = 0$, Eq.(3.56) is reduced to Eq.(3.54). With $q_x = q_z = q$ and $p_z = p_x = p$, Eq.(3.56) is reduced to Eq.(3.55).

Another important class of local stabilizer codes is the topological color code proposed by Bombin and Martin-Delgado [165, 166]. The topological color codes are defined on trivalent graphs with faces that can be colored with three colors, such as hexagonal (6.6.6) and (4.8.8) lattices, as shown in Fig. 3.17a and b, respectively. A qubit is defined on each vertex v and stabilizer generators are defined on each face f :

$$B^x = \prod_{v \in f} X_v, \quad B^z = \prod_{v \in f} Z_v. \quad (3.57)$$

Specifically, the color code on a (4.8.8) lattice, shown in Fig. 3.16c, allows all single-qubit Clifford gates transversally. The distance-3 topological color code on the (4.8.8) lattice is equivalent to Steane's 7-qubit code [85]. The extension to a 3D lattice also enables a transversal non-Clifford gate on the code space [166], whose distance-3 version corresponds to the Reed-Muller 15-qubit code [60].

The topological color codes are also described by a \mathbb{Z}_2 chain complex on hyper-graphs [167]. Consider a trivalent graph $G = (V, E)$, on which a topological color code is defined. We define a hyper-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ consisting of the sets of hyper-vertices \mathcal{V} , hyper-edges \mathcal{E} , and hyper-faces \mathcal{F} as follows (see Fig. 3.17): The hyper-vertices \mathcal{V} are vertices \tilde{v} of the dual graph \tilde{G} . A hyper-edge $\tilde{e} \in \mathcal{E}$ is defined as a triplet of (hyper-)vertices $\tilde{v} \in \tilde{V}$ on a face \tilde{f} of the dual graph \tilde{G} . A hyper-face $\tilde{f} \in \mathcal{F}$ is defined as a set of hyper-edges \tilde{e} that are incident to the vertex \tilde{v} of the dual graph \tilde{G} . As defined in Sect. 3.1, we can define both \mathbb{Z}_2 valued vector spaces and Abelian groups $C_{1,2,3}$ with the hyper-graph elements as their bases. Because the hyper-edge and hyper-face are defined as sets of hyper-vertices and hyper-edges, respectively, we can define boundary maps $\partial_i : C_i \rightarrow C_{i-1}$ naturally by such sets. The plaquette and star stabilizer generators of the surface code defined on such a

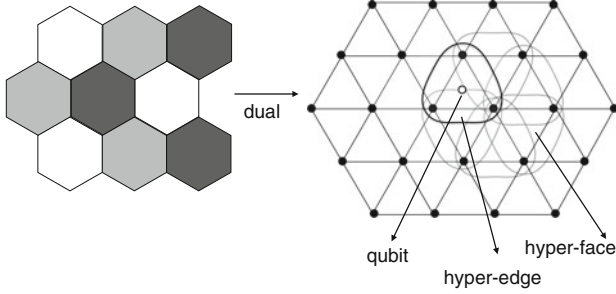


Fig. 3.17 (Left) A hexagonal lattice with three-coloring. (Right) A triangular lattice, the dual of the hexagonal lattice. A hyper-graph is defined on a triangular lattice. The color code on the hexagonal lattice corresponds to a surface code defined on the hyper-graph

hyper-graph are as follows:

$$A_m = \prod_{\tilde{e} \in \partial \tilde{f}} Z_{\tilde{e}}, \quad B_k = \prod_{\tilde{e} \in \delta \tilde{v}} X_{\tilde{e}}. \quad (3.58)$$

This definition is equivalent to the previous definition, Eq.(3.57). In Ref. [167], the authors defined the projections from the \mathbf{Z}_2 chain complex on the hyper-graph into chain complexes on a dual graph \tilde{G} . Specifically, a certain subset of vertices is removed in each projection. This procedure corresponds to the removal of stabilizer generators colored by one of the three colors. This allows us to utilize MWPM to decode the topological color codes efficiently on the projected surface codes [167].

The decoding problem of topological color codes is mapped into random three-body Ising models using the mapping between quantum error correction and spin glass models [168]. This can be understood as follows: for each face center a gauge spin valuable is located associated with each stabilizer generator, and three gauge spin valuables (stabilizer generators) that share the same qubit (vertex) interact with each other. The locations of the optimal thresholds have also been investigated via the spin glass theory and Monte Carlo simulations [169, 170, 171].

3.7 Connection to Topological Order in Condensed Matter Physics

Topological order is an exotic quantum phase of matter, which cannot be characterized by the Landau–Ginzburg theory of symmetry breaking, in which an ordered phase can be characterized by local order parameters. The ground state of a topologically ordered system is degenerate, but its degeneracy cannot be destroyed by any local perturbation. Thus, no local order parameter can succeed to capture the topologically ordered phase. Understanding the nature of topological order is one of the most important goals of modern condensed matter physics. Moreover, the ground

state of a topologically ordered system is, by definition, robust against any local perturbations, and hence it is also useful for storing quantum information. There is a beautiful correspondence between topological quantum codes and topologically ordered systems, which provides a promising way to understand condensed matter physics via quantum information.

Let us first consider the bit-flip code defined in Sect. 3.2. The Hamiltonian, which we call a stabilizer Hamiltonian, is defined as follows:

$$H_{\text{Ising}} = -J \sum_k A_k = -J \sum_{i=1}^{n-1} Z_i Z_{i+1}. \quad (3.59)$$

Here, the summation is taken over all stabilizer generators, but one stabilizer operator, which is not independent, is removed. The Hamiltonian Eq. (3.59) corresponds to the Ising model in 1D with an open boundary condition. By its construction, the Hamiltonian is diagonalizable, and the stabilizer state becomes the ground state. The bit-flip code has a 2D stabilizer subspace spanned by $\{|00\dots 0\rangle, |11\dots 1\rangle\}$. This means that the ground state is degenerate. The bit-flip errors occurring on the code space excite the ground state to an excited state. Thus, the states in the orthogonal subspaces of the bit-flip code correspond to excited states.

To address topological order, let us consider the robustness of the ground state against perturbations from transversal fields $h_x \sum_i X_i$. By using standard perturbation theory [172], we can show that the degeneracy of the ground states is not lifted up to the $(n - 1)$ th order of perturbation. This happens because the code distance of the n -qubit bit flip code against X errors is n , and any X errors of weight up to $n - 1$ map the code state into an orthogonal space. Accordingly, the ground state degeneracy is robust against the transversal fields.

On the other hand, longitudinal fields $h_z \sum_i Z_i$ spoil the ground state degeneracy in the large n limit, even if h_z is small. More precisely, the energy between $|00\dots 0\rangle$ and $|11\dots 1\rangle$ is shifted by nh_z . Thus, a superposition $\alpha|00\dots 0\rangle + \beta|11\dots 1\rangle$ in the ground subspace is easily destroyed by the longitudinal fields. In this sense, the stabilizer Hamiltonian constructed by the bit-flip code is not topologically ordered. However, if any perturbation with respect to the longitudinal fields is prohibited due to some symmetry of nature, the ground state degeneracy is robust under that symmetry. This type of robustness of ground-state degeneracies is called *symmetry protected topological order* [173, 174]. (Note that in this case the ground-state degeneracy is not related to the geometrical property of the underlying manifold in contrast to the genuine topological order in 2D.)

The symmetry prohibiting the longitudinal fields seems to be somewhat artificial. We can, however, impose the symmetry by transforming the Ising Hamiltonian to a free-fermion model using the following Jordan-Wigner transformation [175]:

$$a_{2i-1} = \prod_{k=1}^{i-1} X_k Z_i, \quad (3.60)$$

$$a_{2i} = \prod_{k=1}^{i-1} X_k Y_i. \quad (3.61)$$

The operators, called Majorana fermion operators, are hermitian $a_k = a_k^\dagger$ and satisfy the fermion commutation relation $\{a_k, a_{k'}\} = 2\delta_{k,k'}I$. The Ising stabilizer Hamiltonian is reformulated in terms of a_k :

$$H_{\text{Ising}} = -J \sum_{i=1}^{n-1} (-i) a_{2i} a_{2i+1}. \quad (3.62)$$

The logical operators acting on the degenerated ground states are given by

$$L_Z = a_1 = Z_1, \quad L_X = a_1 a_{2n} = Y_1 \left(\prod_{k=2}^{n-1} X_k \right) Y_n. \quad (3.63)$$

The degree of freedom in the degenerated ground states is called the Majorana zero mode or the unpaired Majorana fermion [176]. If the parity of the number of fermions is preserved, the fermion operators would appear with a quadratic form $a_k a_{k'}$. Under such a symmetry, there is no perturbation that lifts the ground state degeneracy. Thus, the ground state of the unpaired Majorana fermion is symmetry protected.

Next, we will provide a genuine topologically ordered system based on the surface code (Kitaev's toric code). The stabilizer Hamiltonian, the so-called Kitaev's toric code Hamiltonian [105], is given as a summation of all plaquette and star operators:

$$H_{\text{Kitaev}} = -J \sum_m A_m - J \sum_k B_k. \quad (3.64)$$

The ground state has a fourfold degeneracy corresponding to the code space. Errors on the code state correspond to excitations. Specifically, there are two types of excitations, corresponding to the Z error $Z(c_1)$ and X error $X(c_1)$. Excitations appear at the boundaries of the error chains ∂c_1 and $\partial \bar{c}_1$, because the local energy changed from $-J$ to $+J$ there. Such excitations, i.e., at the end points of the error chains, are always created as pairs, can be viewed as a pair creation process on the ground state.

Suppose these two types of the excitations were created on the system and moved as shown in Fig. 3.18 (left top). This process can be described by

$$Z(c_1^{(2)}) X(\bar{c}_1^{(2)}) Z(c_1^{(1)}) X(\bar{c}_1^{(1)}) |\Psi\rangle = -|\Psi\rangle. \quad (3.65)$$

On the other hand, by continuously changing the trajectory of the particles, as shown in Fig. 3.18, this process can also be regarded as a braiding process of an X -type excitation around a Z -type excitation. After the braiding operations, a phase factor is applied to the state as shown in the r.h.s. of Eq. (3.65). Thus, the excitations are neither bosonic nor fermionic, which are invariant under the braiding operation, i.e.,

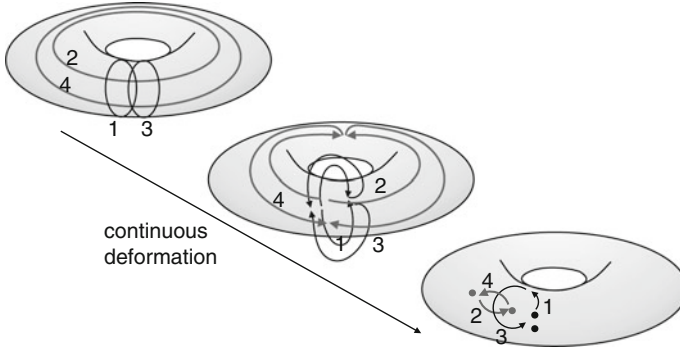


Fig. 3.18 (left top) (1) A pair of X -type excitations are created, moved around the torus, and annihilated. (2) A pair of Z -type excitations are created, moved around the torus, and annihilated. (3) Do the process (1) again. (4) Do the process (2) again. (middle) The creation, movement, and annihilation process is continuously deformed. (right bottom) A braiding operation of an X -type excitation around a Z -type excitation

the swapping operation twice. In this sense, the excitations on the surface code are referred to as anyons. Specifically, since the phase factors is \mathbf{Z}_2 , they are called \mathbf{Z}_2 Abelian anyons. By using the generalized Pauli operators on for a qudit, we can also define \mathbf{Z}_d Kitaev's toric code [110], on which excitations are \mathbf{Z}_d Abelian anyons. The renormalization decoders are also applied for topological quantum error correction on the \mathbf{Z}_d Kitaev's toric codes [177, 178].

More generally, using a finite group G , we can define a quantum state $|g\rangle$ ($g \in G$) in a $|G|$ -dimensional Hilbert space. Then we define four types of operators for each $g \in G$:

$$L_+^g = \sum_{h \in G} |gh\rangle \langle h|, L_-^g = \sum_{h \in G} |hg^{-1}\rangle \langle h|, T_+^h = |h\rangle \langle h|, T_-^h = |h^{-1}\rangle \langle h^{-1}|. \quad (3.66)$$

The non-Abelian Kitaev's toric code model, which is called the quantum double model [110], is defined as

$$H = -J \sum_m A(f_m) - J \sum_k B(v_k), \quad (3.67)$$

in terms of the following plaquette and star operators

$$A(f_m) = \sum_{g_1 g_2 g_3 g_4 = I} T_-^{g_1}(e_{l_1}^m) T_-^{g_2}(e_{l_2}^m) T_+^{g_3}(e_{l_3}^m) T_+^{g_4}(e_{l_4}^m) \quad (3.68)$$

$$B(v_k) = \frac{1}{|G|} \sum_{g \in G} L_+^g(\vec{e}_{l_1}^k) L_+^g(\vec{e}_{l_2}^k) L_-^g(\vec{e}_{l_3}^k) L_-^g(\vec{e}_{l_4}^k) \quad (3.69)$$

where the four edges $e_{1,2,3,4}^m \in \partial f_m$ and $\bar{e}_{1,2,3,4}^k \in \delta v_k = \partial \bar{f}_k$ are labeled clock wise. The quantum double model supports non-Abelian anyonic excitations [38, 110], which allows us to implement universal quantum computation solely by braiding them.

Let us return to the \mathbb{Z}_2 Kitaev’s toric code Hamiltonian. The code distance of the surface code on the $n \times n$ torus is n . Thus, neither X -type (transverse) nor Z -type (longitudinal) fields can lift the ground state degeneracy up to the $(n - 1)$ th order of perturbation. More generally, no local perturbation can lift the ground state degeneracy in the large n limit. Thus, the ground state of the Kitaev’s toric code Hamiltonian is topologically ordered. For any properly defined stabilizer code, we can define a stabilizer Hamiltonian, whose ground state is topologically ordered. However, in condensed matter physics, the local interactions are of central importance. Thus, it is natural to restrict the stabilizer generators to be spatially local, i.e., topological stabilizer codes.

Only the nearest-neighbor two-body interactions are attained in physically natural systems. It has been known that the Kitaev’s toric code Hamiltonian is obtained as an effective low energy model of another model consisting only of two-body nearest-neighbor interactions [179]. Let us consider the following two-body nearest-neighbor model, called Kitaev’s compass model:

$$H_{\text{comp}} = -J_x \sum_{(i,j) \in E_x} X_i X_j - J_y \sum_{(i,j) \in E_y} Y_i Y_j - J_z \sum_{(i,j) \in E_z} Z_i Z_j, \quad (3.70)$$

where E_x, E_y , and E_z are sets of right-up, left-up, and vertical bonds, respectively, of a hexagonal lattice (see Fig. 3.19 (left)). If we take the large J_z limit, each vertical bond favors the two-dimensional subspace spanned by $\{|00\rangle, |11\rangle\}$, because it is stabilized by $Z_i Z_j$. Thus, in the large J_z limit, we can derive an effective low energy Hamiltonian, which commutes with the $Z_i Z_j$ interactions, by using perturbation theory:

$$H_{\text{eff}} = -\frac{J_x^2 J_y^2}{16|J_z|^3} \sum_f \tilde{X}_{e_1^f} \tilde{X}_{e_2^f} \tilde{Y}_{e_3^f} \tilde{Y}_{e_4^f} - J_z \sum_{e \in E_z} \tilde{Z}_e, \quad (3.71)$$

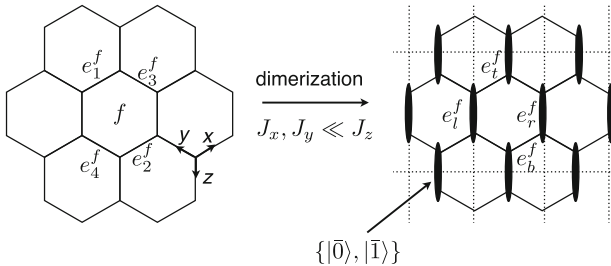


Fig. 3.19 A hexagonal lattice on which Kitaev’s compass model is defined. In the large J_z limit, the two spins on each vertical edge are confined into a 2D subspace forming a dimer. The dimer is located on each edge of a square lattice shown by the *dotted lines*

where $\tilde{A}_e = A_i A_j$ with $e = (i, j)$ and $A = X, Y, Z$, depending on $e \in E_x, E_y, E_z$. The edges $\{e_i^f\}$ are left-up and right-up edges on a hexagonal face f and the summation \sum_f is taken over all faces.

Let us define a qubit $\{|\bar{0}\rangle = |00\rangle, |\bar{1}\rangle = |11\rangle\}$ for each dimerized edge and Pauli operators $\bar{X} = XX$ or $= \bar{Y}Y$ and $\bar{Z} = Z \otimes I$ or $= I \otimes Z$. Now a qubit is assigned to each edge of a square lattice, which is an vertical edge in the hexagonal lattice as shown in Fig. 3.19 (right). Using this definition, the effective Hamiltonian can be reformulated as

$$\bar{H}_{\text{eff}} = -\frac{J_x^2 J_y^2}{16|J_z|^3} \sum_f \bar{X}_{e_l^f} \bar{X}_{e_r^f} \bar{Z}_{e_t^f} \bar{Z}_{e_b^f}, \quad (3.72)$$

where $\{e_l^f, e_r^f, e_t^f, e_b^f\}$ are the left, right, top, and bottom edges, respectively, on a square face f . If we apply the Hadamard transformation on all horizontal edges, we obtain the Kitaev's toric code Hamiltonian. In this way, the stabilizer Hamiltonian can be obtained as a low-energy effective model of a two-body nearest-neighbor system [179]. This shows the validity of employing topological stabilizer codes and quantum coding theory to understand the quantum phase of matter in condensed matter physics. The readers, who are interested in more about the Kitaev's compass and related models, should see the comprehensive review on this fields [180, 181].

3.8 Summary and Discussion

In this chapter, we have introduced the topological stabilizer codes and seen how errors are detected and corrected on them. We have also seen the connection between spin glass models and topological quantum error correction. Since there has been well-established knowledge on the phase diagram of the spin glass models [170, 182, 183], it will help to understand the parameter regions in which topological quantum error correction faithfully works [133, 171, 184] and also help to develop an efficient decoding method [111, 185, 186]. On the experimental side, proof of principle experiments of topologically encoded qubits were conducted with photonic qubits [187] and trapped ion qubits [188].

We have only mentioned topological stabilizer codes on simple 2D manifolds, but as we have seen in Sects. 3.1 and 3.3, we can define the surface codes on general tilings of the 2D manifolds [124] even with negative curvature [189], which allows us to achieve higher encoding rates. We can also define topological stabilizer codes in 3D or higher manifolds, such as the Kitaev's toric codes in arbitrary dimensions [111], the 3D cubic code [190], and the fractal code [191].

A complete classification of the topological stabilizer codes has been obtained in 2D [140, 192]. It has been shown that any topological stabilizer code is equivalent to multiple copies of the toric codes up to local (finite-qubit) unitary operations. Moreover, in arbitrary d dimensions, the color codes have been shown to equal to multiple

copies of d -dimensional toric codes up to local unitary operations and adding and removal of ancilla qubits [193]. The thermal stability of topological order in 2D and 3D Kitaev's toric code Hamiltonians have been analyzed by Nussinov and Oritz [194] in detail, which showed that the lifetime of the logical information is finite even in the thermodynamic (large system size) limit. Moreover, a general condition for existence of the thermal stability has been argued systematically in terms of symmetries of entanglement at both zero and finite temperatures [195, 196]. The thermal stability of the general topological stabilizer Hamiltonians has been investigated via quantum coding theory by Bravyi and Terhal [122] for 2D, and Yoshida for 3D [123]. They showed that topological order in stabilizer Hamiltonian systems does not have thermal stability in 3D or lower dimensions under quite general assumptions such as locality, translation invariance, and scale invariance of the systems.

If there is a thermally stable topological order, we can store quantum information reliably even at a finite temperature without any active error correction, i.e., a self-correcting quantum memory. Of course, if a fault-tolerant quantum computer were realized, we could store quantum information reliably with a repetitively performing quantum error correction, which, however, requires selective addressing of each individual qubits. There are also several intermediate approaches for a reliable quantum storage without selective addressing using global dissipative dynamics [197], an interaction with an engineered environment [198, 199, 200], and decoding by cellular automata with local update rules [201, 202]. However, a genuine topologically ordered self-correcting quantum memory in 3D or lower dimensions seems to be hardly achieved even in the presence of effective long-range interactions [203].

Chapter 4

Topological Quantum Computation with Surface Codes

In this chapter, we explain how to perform topological fault-tolerant quantum computation on the surface code. All operations employed are nearest-neighbor (at most) two-qubit gates and single-qubit measurements on a 2D array of qubits. This property is quite favorable for the fabrication of qubits and their control lines on a chip. Fault-tolerance ensures that no local noise during any sort of operations ever spoil the quantum computation, if the noise strength is smaller than a certain threshold value, as seen below.

We first introduce the defect pair qubit, a logical qubit using a pair of defects on the surface, which allows us to encode many logical qubits on the surface. Then, we explain the elementary operations of defects by local quantum information processing. A braiding operation, based on the elementary operations, of the defects is further utilized to implement the logical CNOT gate between the defect pair qubits. Combining with the topologically protected operations, the magic states are injected and distilled for fault-tolerant universal quantum computation. Based on these understandings, we introduce a topological diagram and the topological calculus on it, as a set of the transformations that preserve the logical actions on the code space. These provide us with a deep understanding of topological quantum computation on the surface. Finally, we return to the microscopic viewpoint to explain how topological quantum error correction is executed and analyzed.

4.1 Defect Pair Qubits

In order to perform quantum computation of N qubits, we have to arrange N logical qubits in the code subspace while keeping their code distance long enough. A way of doing this is to use a surface of higher genus. Specifically, we use a large planar surface and punch holes on it, which we call defects on the surface.

In Sect. 3.3.2, we have seen that we can define a logical qubit by introducing a defect on a planar surface, where the stabilizer generators inside the defect region are removed from the stabilizer group. The logical operators are defined by a nontrivial

cycle surrounding the defect and a chain connecting the defect to the boundary. If we introduced many degrees of freedom, i.e., many defects based on this strategy, the logical operators defined by chains connecting each defect to the boundary would become complicated. To avoid this, we use a pair of defects to define a logical qubit, which we call a *defect pair qubit*.

We define two defect regions $D \in C_2$ and $D' \in C_2$, as shown in Fig. 4.1. We remove all Z -type stabilizer generators inside these regions, i.e., $\{A_m\}_{f_m \in D \cup D'}$. Thus, the operators $Z(\partial D)$ and $Z(\partial D')$ are not stabilizer operators. Instead, we append a Z -type stabilizer operator $Z(\partial D + \partial \bar{D})$ as a stabilizer generator. Moreover, we also append the Pauli X operator on all edge qubits inside (not including the boundary) the regions $D \cup D'$, i.e., $\{X_l\}_{e_l \in (D \cup D') \setminus (\partial D \cup \partial D')}$. By this definition, the star (X -type) stabilizer generators in the defect region are still in the stabilizer group. We choose $Z(\partial D)$ as a logical operator, because it commutes with all stabilizer operators, but does not belong to the stabilizer group. Because $Z(\partial D + \partial \bar{D})$ is a stabilizer operator, $Z(\partial D')$ also acts the same way, i.e., $Z(\partial D) \sim Z(\partial D')$. Moreover, the actions of operators represented by the homologically equivalent cycles are the same. The logical X operator is given by $X(\bar{c}_1)$, with a dual 1-chain \bar{c}_1 connecting two defects, as shown in Fig. 4.1. The actions of the operators represented by any homologically equivalent chains in the sense of relative homology are the same. That is, we may choose any dual 1-chain \bar{c}_1 which connects two defect regions D and D' . The code distance is given by the circumference of the defect or the distance between two defects. Because the plaquette operators on the primal lattice are removed, we call these defects and the logical qubit *primal defects* and *primal defect pair qubit*, respectively.

Similarly, we can also define a logical qubit by removing the star (X -type) stabilizer operators on the dual defect regions \bar{D} and \bar{D}' defined on the dual lattice as shown in Fig. 4.2. We call such defects and the logical qubit *dual defects* and *dual defect pair qubit*, respectively. Hereafter, the planar surface code state, on which the defects are introduced, is referred to as *vacuum*. Below we will explain how defect qubits are created, deformed, and moved in the vacuum.

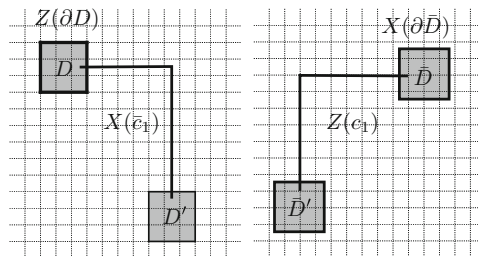


Fig. 4.1 Primal (*left*) and dual (*right*) defect pair qubits. The primal defect pair qubit is defined by removing the plaquette operators inside the defect regions D and D' . Then the cycle ∂D around the defect becomes a nontrivial cycle, by which the logical Z operator $Z(\partial D)$ is defined. The logical X operator $X(\bar{c}_1)$ is defined as a dual 1-chain \bar{c}_1 connecting two defects. Any homologically equivalent logical operators acts the same way on the code space. The dual defect pair is defined similarly, but the basis is changed by the Hadamard transformation

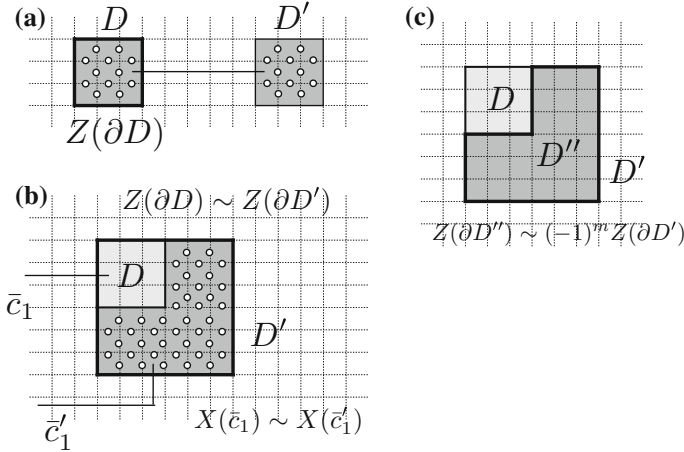


Fig. 4.2 **a** A defect creation. The qubits inside the defect regions (excluding the qubits on the boundary), denoted by *circles*, are measured in the X -basis. **b** The defect region D is expanded into D' by measuring the qubits inside the region D' in the X -basis, where the encoded quantum information is preserved. **c** A defect region D' is contracted into D by measuring the plaquette operators in $D'' = D' \setminus D$. The logical operator Z after the contraction is defined depending on the measurement outcome m

4.2 Defect Creation, Annihilation, and Movement

The defect creation is accomplished by measuring the qubits inside the defect region D , not including the qubits on its boundary, in the X -basis (see Fig. 4.2a). These measurements remove the plaquette operators inside D from the stabilizer group, because these X -basis measurements do not commute with them. On the other hand, the measurements do commute with the star operators, and hence a parity of four measurement outcomes corresponding to a star operator is always even (if there is no measurement error). While the measurement outcomes are random, we can prepare all measured qubits to be in the $|+\rangle$ state by applying Z operators according to the measurement outcomes. (In practice, there is no need to apply the Z operations. It is enough to keep the information of the byproduct Pauli operator dependent on the measurement outcomes.)

As the qubits on the boundary are not measured, the post-measurement state is stabilized by $Z(\partial D)$. In order to create a pair of defects, we do the same thing for another defect D' . Apparently, the resultant state is stabilized by $Z(\partial D + \partial D')$ and satisfies the definition of the defect pair qubit. Moreover, the logical qubit is stabilized by $Z(\partial D)$ and hence a logical Z -basis state is prepared.

The defect region can be expanded by creating a larger defect D' , which includes the original defect region D , i.e., $D \subset D'$, as shown in Fig. 4.2b. We can choose $Z(\partial D')$ as a logical Z operator of the defect pair qubit, so that the information with respect to $Z(\partial D')$ is preserved during this operation. (Recall that logical operators act the same if the corresponding cycles are homologically equivalent. Thus we can

choose a large enough cycle surrounding the defect in advance.) We can expand the defect region step-by-step such that, at each step, a logical X operator is untouched as shown in Fig. 4.2b. (Or equivalently, we can compensate for the difference in the logical X operator according the X -basis measurement outcomes inside the defect region.) Thus, the information with respect to the logical X operator is also preserved. Accordingly, the logical information of the defect pair qubit is stored, but now the defect region D is expanded into $D + D'$.

The defect annihilation is executed by measuring the plaquette operators inside the region D to restore them to the stabilizer group. The surface with a defect D is rewritten by

$$|D\rangle \propto \prod_{e_l \in D} \left(\frac{I + X_l}{2} \right) |v\rangle, \quad (4.1)$$

where $|v\rangle$ indicates the surface code state without defect, i.e., vacuum. Thus, $|D\rangle$ is a superposition of all possible applications of X operators on the vacuum $|v\rangle$. The measurements of the plaquette operators collapse the superposition. By applying a recovery operation inside the region D such that all eigenvalues becomes $+1$, the defect is annihilated. (Note that there is no need to actually apply the recovery operation. It is enough to keep the record of the eigenvalues.) The parity of all measurement outcomes of the plaquette operators inside D corresponds to the eigenvalue of $Z(\partial D)$, because we have

$$Z(\partial D) = \prod_{f_m \in D} Z(\partial f_m) = \prod_{f_m \in D} A_m. \quad (4.2)$$

This indicates that a logical Z -basis measurement can be done by annihilating the defect completely.

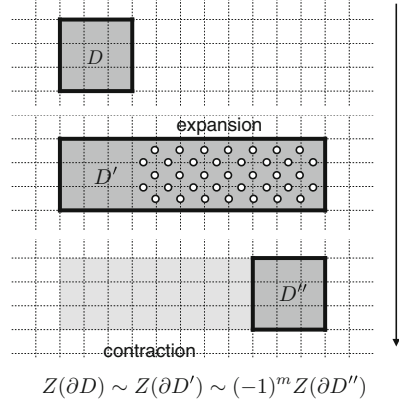
Suppose a defect region D inside the defect region D' (i.e., $D \in D'$) is annihilated by measuring the qubits inside D , as shown in Fig. 4.2c. We can then obtain the eigenvalue $(-1)^m$ ($m = 0, 1$) of $Z(\partial D)$. Let D'' be the complement of D in D' . Because we have

$$Z(\partial D') = Z(\partial D'')Z(\partial D), \quad (4.3)$$

depending on the eigenvalue $(-1)^m$, the operator $(-1)^m Z(\partial D'')$ acts as the logical operator of the defect D'' . Thus, the defect D' , consisting of the defect pair qubit, is contracted into D'' without changing the stored quantum information (up to the logical Pauli X flip).

The defect movement on the surface is implemented by combining the defect expansion and contraction, as shown in Fig. 4.3. At first, we expand a defect D into D' ($D \in D'$) by the previously mentioned procedure. Second, the defect region D is annihilated, and we obtain the eigenvalue $(-1)^m$ of $Z(\partial D)$. As we already pointed out, the stored logical information is unchanged under these procedures up to the logical Pauli X operator depending on the measurement outcome m .

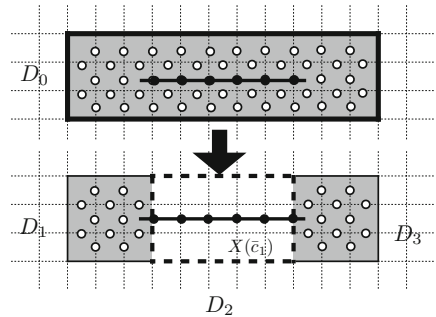
Fig. 4.3 A defect movement by expansion and contraction



The logical X -basis state preparation is executed by combining the defect creation and annihilation, as shown in Fig. 4.4. We first create a defect region $D_0 = D_1 + D_2 + D_3$, which consists of three adjacent defect regions D_1 , D_2 , and D_3 . The two defects D_1 and D_3 are utilized as a defect pair. We can define a logical X operator by employing the qubits inside the defect region D_2 . Because all qubits inside the defect region are in the $|+\rangle$ state, the eigenvalue of the logical X operator at this time is $+1$. Then the defect region D_2 in-between D_1 and D_3 is annihilated by measuring the star operators. These measurements commute with the logical X operator, and hence its eigenvalue is still $+1$ after the annihilation. Now we have a defect pair qubit, which is the eigenstate of the logical X operator. A measurement in the logical X basis can be implemented by doing the X -basis state preparation in an inverse way. More precisely, two defects D_1 and D_3 are connected by making a larger defect $D_0 = D_1 + D_2 + D_3$. We can choose a logical X operator such that all constituent qubits belong to the defect region D_0 . Then, we obtain the eigenvalue of the logical X operator.

The dual defect creation, annihilation, expansion, contraction, and propagation can be done in the same way on the dual lattice with the Hadamard transformation. In Sect. 4.5, these elementary operations of the primal and dual defects are depicted by a topological diagram.

Fig. 4.4 A preparation of the logical X -basis state by creation of a defect region D_0 and annihilation of a defect region D_2 between two defect regions D_1 and D_3 . Note that after the first defect creation, the state is stabilized by the logical X operator $X(\bar{c}_1)$, which is untouched during the following defect annihilation



4.3 Logical Controlled-NOT Gate by Braiding

Using the elementary operations explained in the previous section, we can perform logical gate operations on the defect pair qubits. Indeed, the defects created from the vacuum behave like anyonic particles, i.e., braiding a defect around another defect results in a nontrivial operation. (It is better to say that we change the manifold continuously to deform the ground state, in contrast to the anyons appeared as an excitation on the toric code Hamiltonian.) All operations employed in the elementary operations are single-qubit or local stabilizer measurements, which can be done by nearest-neighbor two-qubit gates and single-qubit measurements. This contrasts with concatenated quantum computation, where the logical operations are implemented as transversal operations and hence, essentially, non-local two-qubit gates are employed.

We first consider the CNOT gate between primal (control) and dual (target) defect pair qubits. Suppose there are primal and dual qubits on the surface code as shown in Fig. 4.5a, where logical operators are specified by the chains $\{\partial D, \bar{c}_1\}$ and $\{\partial \bar{D}, c_1\}$ for the primal and dual qubits, respectively. We can freely move the defect everywhere we want by the defect expansion and contraction. Let us braid the primal defect around the dual defect, as shown in Fig. 4.5a–d. After the braiding operation, the operator $X(\bar{c}'_1)$ in Fig. 4.5d has the same information as the operator $X(\bar{c}_1)$ before the braiding. Using the equivalence relation,

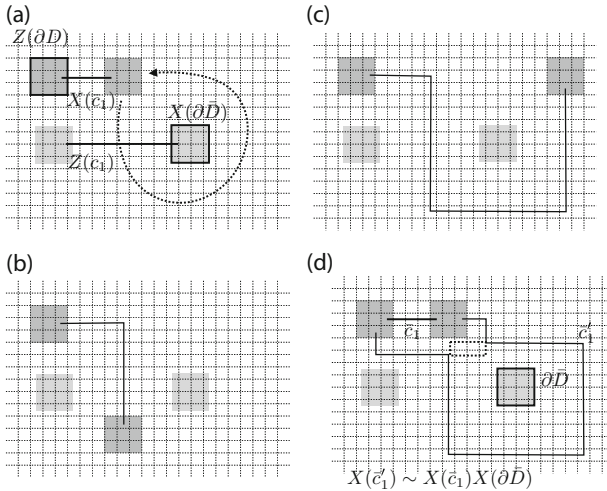


Fig. 4.5 A logical CNOT gate implemented by braiding. **a** A primal defect is braided around a dual defect. **b, c** The logical X operator is deformed continuously during the braiding operations. **d** After the braiding, the logical X operator is represented by the dual chain \bar{c}'_1 winding around the dual defect. By applying star operators inside the rectangle denoted by *dotted lines*, \bar{c}'_1 is decoupled into a chain connecting two primal defects and a chain surrounding the dual defect. The corresponding operators are equivalent to the products of the logical X operators on the primal and dual qubits

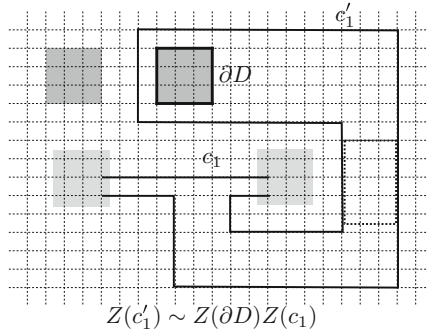


Fig. 4.6 The time evolution of the logical Z operator caused by braiding. The logical Z operator on the dual qubit is transformed into the product of the logical Z operators on the primal and dual qubits

$$X(\bar{c}'_1) \sim X(\bar{c}_1)X(\partial\bar{D}), \quad (4.4)$$

we understand that a correlation is made by the braiding operation between the logical X operators $X(\bar{c}_1)$ and $X(\partial\bar{D})$ of the primal and dual defect pair qubits. A similar observation holds for the logical operator $Z(c_1)$, which is transformed into $Z(c'_1) \sim Z(c_1)Z(\partial D)$ (see Fig. 4.6). On the other hand, $Z(\partial D)$ and $X(\partial\bar{D})$ are invariant under this operation. In short, the braiding operation transforms the logical Pauli operators as follows:

$$Z(\partial D) \rightarrow Z(\partial D), \quad (4.5)$$

$$Z(c_1) \rightarrow Z(c_1)Z(\partial D), \quad (4.6)$$

$$X(\partial\bar{D}) \rightarrow X(\partial\bar{D}), \quad (4.7)$$

$$X(\bar{c}_1) \rightarrow X(\bar{c}_1)X(\partial\bar{D}). \quad (4.8)$$

This transformation is equivalent to that for the CNOT gate, Eqs. (2.14)–(2.17). Thus, braiding the primal defect around the dual defect results in a logical CNOT gate between the primal and dual defect pair qubits.

Unfortunately, in the above CNOT gate, the primal (dual) defect is always a control (target) qubit. Such CNOT gates always commute with each other, which is a natural consequence of the fact that the defect qubits on the surface code are Abelian. In order to realize a genuine CNOT gate between the same type of qubits (and hence noncommuting gates), we utilize a teleportation-based gate [68], as shown in Fig. 4.7. Only CNOT gates between primal (control) and dual (target) qubits are employed. The Pauli basis measurements are also done as mentioned in the previous section. Accordingly, the CNOT gate between primal qubits is realized by braiding the primal defects around *virtual* dual defects, which are created and annihilated as ancillae.

The above braiding operations are implemented with topological quantum error correction at each elementary step, which will be explained later in detail. All operations considered so far can be executed keeping the defect size (circumference) and

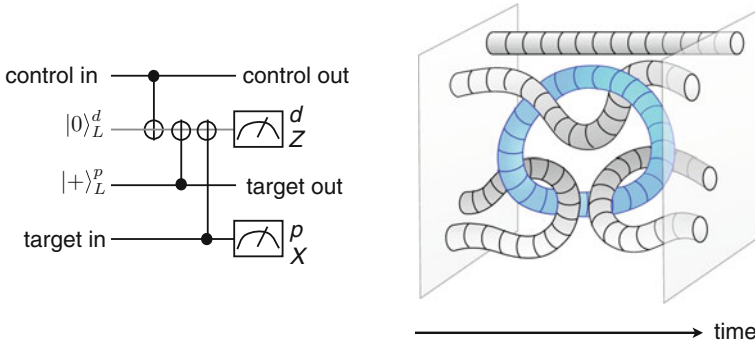


Fig. 4.7 The circuit diagram of the teleportation-based CNOT gate between primal qubits using a dual qubit as ancilla (*left*). The control and target qubits of the CNOT gates are always primal and dual qubits, respectively. The corresponding braiding operation (*right*)

defect distance larger than a length d , which provides a code distance of the logical qubits. Thus, the logical error probability for the logical CNOT gates decreases exponentially by increasing the characteristic length d of the system. In other words, the CNOT gates are topologically protected.

We should mention that the braiding operations explained above are not the only way to perform fault-tolerant operations on the surface code. There are another approaches to perform logical operations fault-tolerantly for the encoded degrees of freedom of the surface code. One is the lattice surgery scheme [204], where the boundary conditions of two planar surface codes are engineered to perform a logical operation. Another is to employ twists, which are topological defects introduced by point defects on lattices [205, 206]. All Clifford gates can be implemented by the twist creation, braiding, and annihilation similarly to the defect pair qubits explained above. All these different approaches can be view as the logical operations by the code deformations [4, 111, 207] and seem to be a unique feature for the topological codes contrasting to the transversal logical gate for the concatenated codes.

4.4 Magic State Injections and Distillation

Unfortunately, the topologically protected CNOT gates do not allow universal quantum computation, because Clifford circuits can efficiently be simulated classically due to the Gottesman–Knill theorem [208]. Here, we explain how to perform single-qubit rotations on the defect pair qubit, while, unfortunately, they are not topologically protected.

Suppose we have a defect pair qubit, whose logical operators are given by $Z(\partial D)$ and $X(c_1)$. We first consider a logical Z rotation $e^{-i(\theta/2)Z(\partial D)}$ on the defect pair qubit. To this end, the defect region D is contracted to a single face f_i by using the annihilation process, as shown in Fig. 4.8a. Now the logical operator $Z(f_i)$ is a four-body operator. (This implies that the code distance becomes 4 at this stage.) A rotation with respect to the logical operator $Z(f_i)$, $e^{-i(\theta/2)Z(f_i)}$, can be implemented

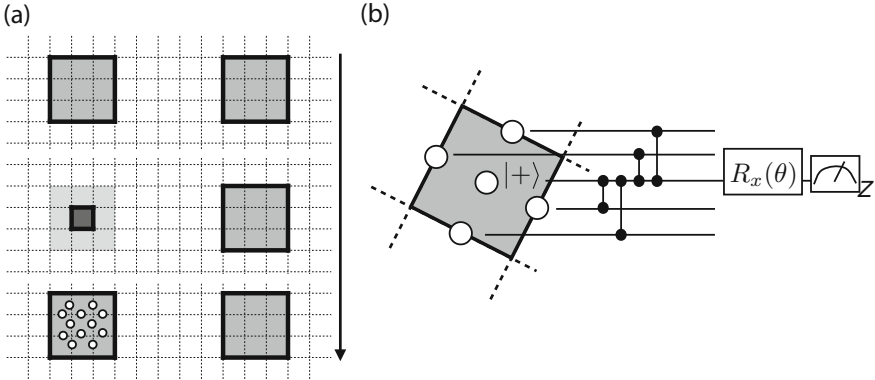
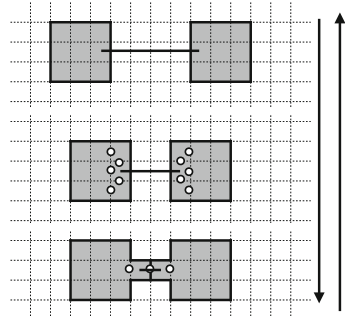


Fig. 4.8 **a** A magic state injection by contracting the defect into an elementary cell. **b** A circuit diagram for the logical Z rotation on the defect of an elementary cell

Fig. 4.9 A magic state injection by connecting two defects. If two defects regions are adjacent, the logical X operator is a physical Pauli X operator between two defects



indirectly by using an ancilla qubit located at the center of the face shown in Fig. 4.8b. First, the four CZ gates are applied between the ancilla and the four edge qubits on the face. Second, after applying a X-rotation $R_x(\theta) = e^{-i(\theta/2)X}$, the ancilla qubit is measured in the Z-basis. According to the measurement outcome $m = 0, 1$, a logical rotation $e^{-i(-1)^m(\theta/2)Z(f_i)}$ is implemented. After the above procedure, the defect, as a single face f_l , is expanded again into the defect region D to restore the code distance of the defect pair qubit.

Next we consider a logical X rotation, $e^{-i(\theta/2)X(c_1)}$. In this case, the two defects are moved and deformed near each other such that $X(c'_1)$ becomes a one-body operator as shown in Fig. 4.9. The rotation $e^{-i(\theta/2)X(c'_1)}$ is now easily implemented by a single-qubit gate. After that, the distance between two defects is restored.

In this way, we can perform an arbitrary single-qubit unitary operation on the defect pair qubit by continuously deforming the defect pair qubit. These deformations and the operations for the logical rotations are depicted in Fig. 4.10. Unfortunately, during the deformation, the code distance inevitably becomes relatively small, so that we can perform logical rotations directly with nearest-neighbor operations. Thus, these processes are not topologically protected.

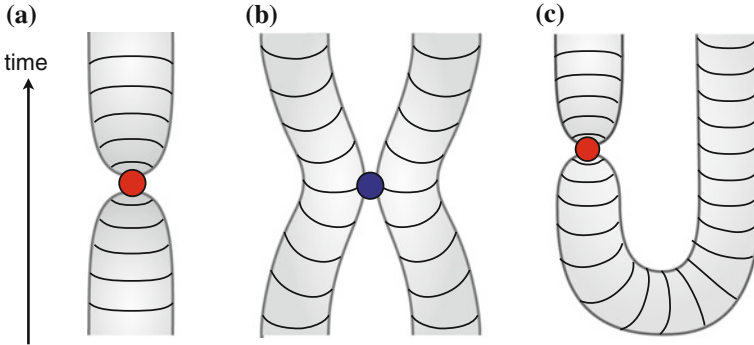
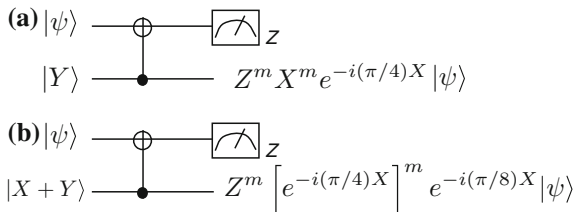


Fig. 4.10 **a** The logical Z rotation. **b** The logical X rotation. **c** The logical $e^{-i(\theta/2)Z}|+\rangle$ state injection

Fig. 4.11 **a, b** Single-qubit rotations by one-bit teleportations [209]



These non-topological operations can be utilized to inject noisy magic states on the surface. Specifically, the logical $e^{-i(\theta/2)Z}|+\rangle$ state is injected as shown in Fig. 4.10c. Then, these noisy magic states are distilled by topologically protected operations to clean magic states, which have a fidelity high enough for reliable quantum computation. Note that we are allowed to use only the CNOT gates, and there is no topologically protected single-qubit Clifford gate. To manage this, we distill two types of magic states. One is the eigenstate of the Pauli- Y operator, which is used to implement the S gate via gate teleportation [68, 209], as shown in Fig. 4.11a. Another is the eigenstate of the $(X + Y)/\sqrt{2}$ operator, which is used to implement the $\pi/8$ gate, a non-Clifford gate necessary for universal quantum computation. The magic state distillation for the Y -basis state is executed solely by the topologically protected CNOT gates using the Steane 7-qubit code, similarly to the method introduced in Sect. 2.8. The topologically protected CNOT gates and the S gates with the distilled Y -basis states are further employed to distill the $(X + Y)/\sqrt{2}$ -states through the Reed–Muller 15-qubit code, as explained in Sect. 2.8.

Using these distilled magic states and the CNOT gates, we can implement the single-qubit gates shown in Fig. 4.11a, b, which together with the CNOT gate constitute a universal set of gates. Accordingly, universal quantum computation is performed reliably on the surface code. Note that all operations employed are single-qubit gates, two-qubit nearest-neighbor gates, and single-qubit measurements on the 2D array of qubits.

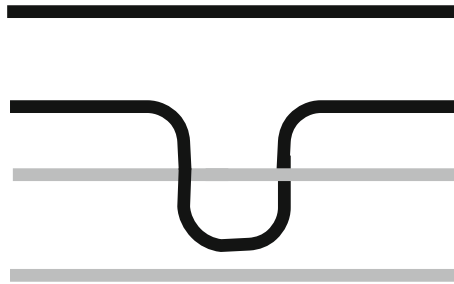
4.5 Topological Calculus

Based on the microscopic understanding of the topological operations, we can introduce a *topological diagram* and a *topological calculus*, which allows us a diagrammatic description of topological quantum computation on the surface codes.

In this diagram, the 3D space-time trajectory of the defects is depicted by projecting it onto a 2D plane like link diagrams mentioned in Sect. 1.5.3. The trajectory of the primal and dual defects as tubes in 3D space-time are denoted by pairs of solid and gray-colored lines:



The braiding operation is denoted by a double crossing as follows:



The logical *X* state preparation and *X*-basis measurement of the primal defect are denoted by closures:



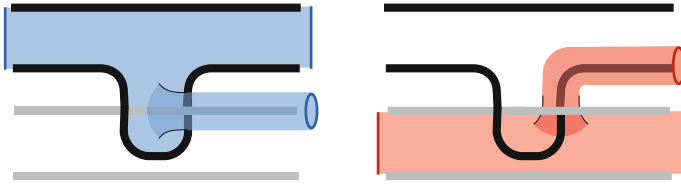
The logical *Z* state preparation and *Z*-basis measurement of the primal defect are denoted by endpoints of the tubes:



The state preparations and measurements for the dual defect pair qubit are depicted in a similar way with gray (curved) lines, except for the basis change by the Hadamard transformation.

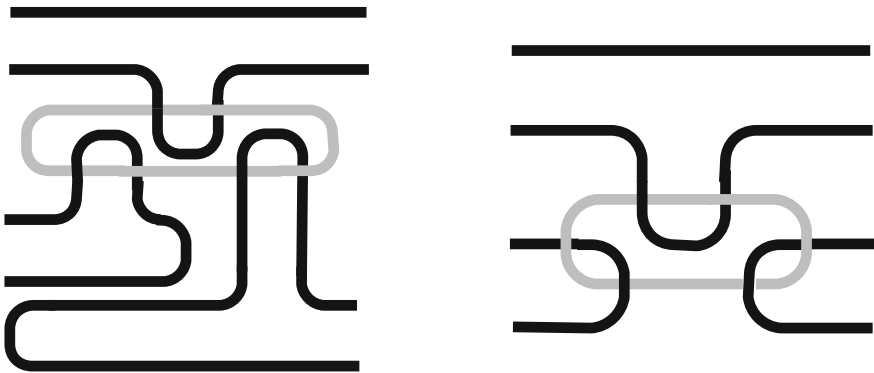
The trajectories of the logical operators are represented by surfaces in 3D space-time like a Seifert surface, which we call *correlation surfaces*. Specifically, the logical *Z* operator for the primal defect corresponds to a surface wrapping around the primal defect. The logical *X* operator for the primal defect corresponds to a surface whose

boundary is the primal defect tubes. (This is also the case for the dual defect, except for the Hadamard transformation.) For example, the time evolution of the logical X and Z operators by the braiding operation can be represented by the following surfaces

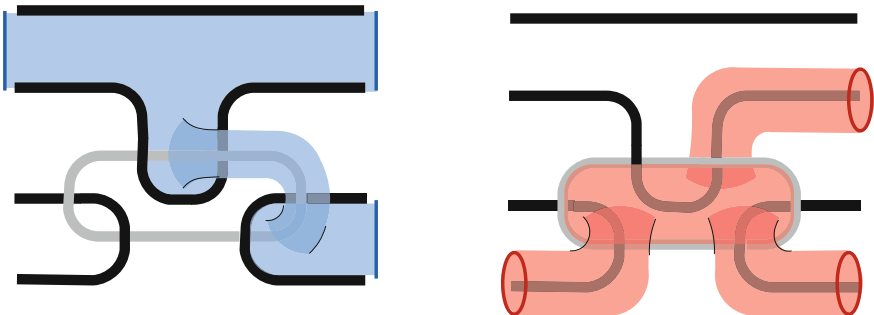


The logical operators before and after the operation correspond to the left and right boundaries of these surfaces, respectively. We can confirm Eqs. (4.5)–(4.8) from the left and right boundaries the correlation surfaces of the above diagrams.

The CNOT gate between the primal defect pair qubits is depicted as follows:

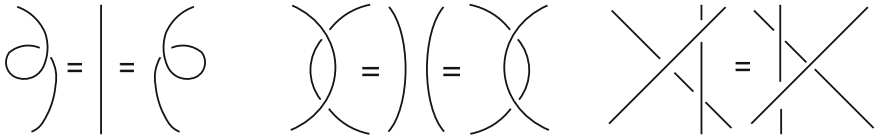


In this diagram, we can easily confirm that the braiding operations transform the logical operators according to the rule for the CNOT gate, i.e., $X \otimes I \rightarrow X \otimes X$ and $I \otimes Z \rightarrow Z \otimes Z$:

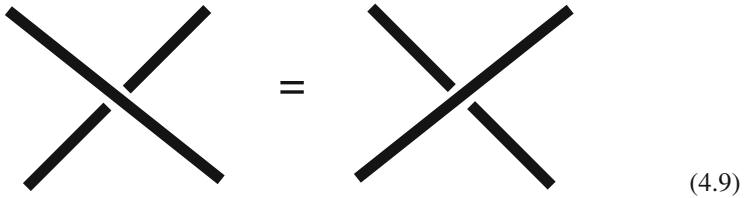


As seen above, the action on the code space is defined by the left and right boundaries of the correlation surfaces. Thus, the logical action of the topological operation is invariant under transformations that do not change the topology of the

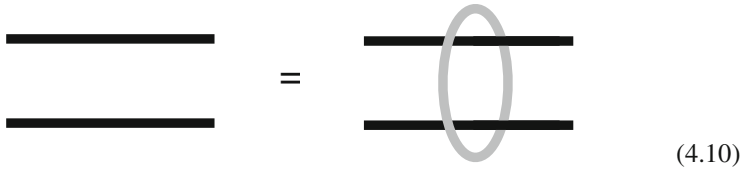
left and right boundaries of the correlation surfaces. Similarly to link diagrams, the boundary topology is invariant under the Reidemeister moves:



In the present case, we have further transformations under which the logical action on the code space is invariant due to the properties of the defect pair qubits. At first, the following two crossings of tubes of the same type are equivalent:



Second, if two defect tubes are a defect pair qubit, a dual loop wrapping around a defect pair can be removed:



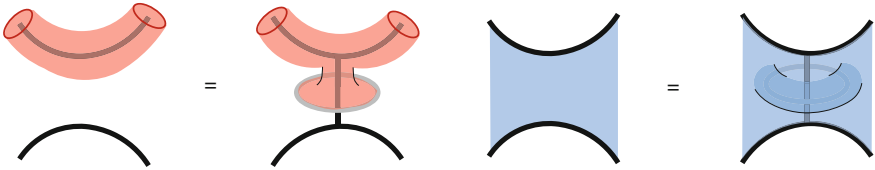
This is because the primal defect pair qubit is stabilized by a Z-type loop operator surrounding the defect pair, and hence the dual loop is nothing but a trivial measurement of the stabilizer operator. Third, because the defect pair qubit is a \mathbf{Z}_2 Abelian anyon, if we braid a primal defect around a dual defect twice, it results in an identity operation:



Fourth, and most importantly, two tubes can be connected or disconnected by the following procedure, which we call a Φ -transformation:

(4.12)

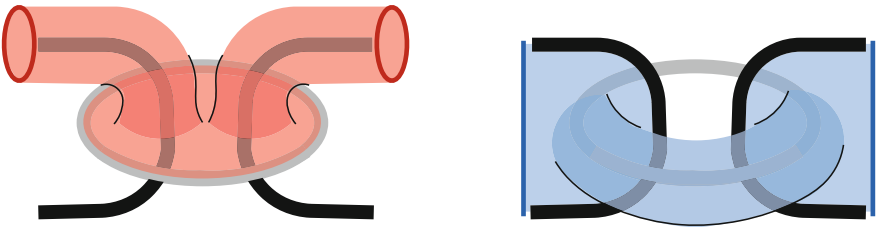
We can easily confirm that the logical operators are invariant under the Φ -transformation:



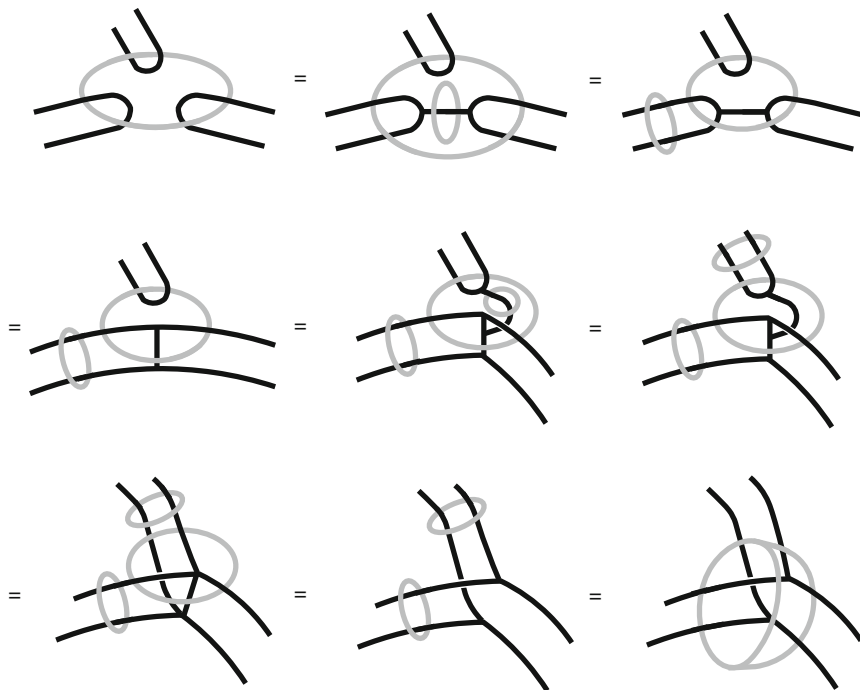
where we note that the dual ring in the middle serves to stop the Z-type correlation surface from propagating toward the bottom and also serves to mediate the X-type correlation surface toward the right. Specifically, if two defect pair qubits are connected by the Φ -transformation between their closures, we can remove the dual ring by using the rule (4.11):

By using the Φ -transformation and rules (4.11) and (4.10), we can transform the defect pair qubit as follows:

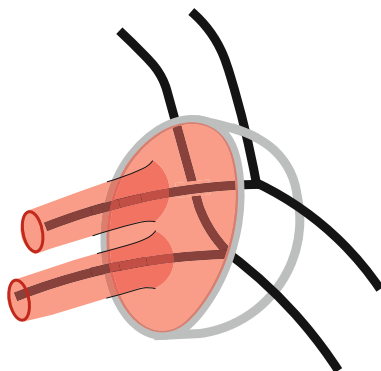
We can also easily confirm that the logical operators are invariant under this transformation as follows:



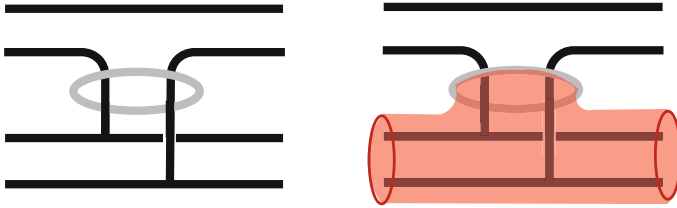
The Φ -transformation is very useful to simplify a complicated topological operation on the surface by joining the tubes as follows:



where we frequently used the Φ -transformation and rule (4.10). Here, the dual triple-ring wrapping around each of the two defect tubes serves to reflect the logical Z operator from the upper to the lower tubes as follows:

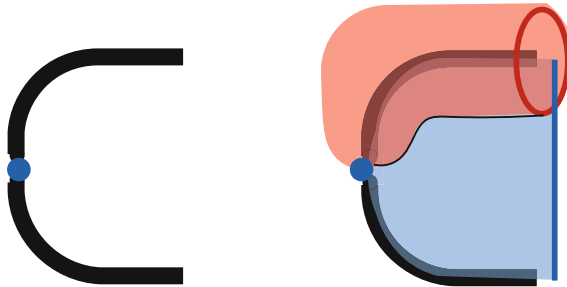


For example, if we apply this transformation to the CNOT gate between the primal defect pair qubits, we obtain a much simpler diagram as follows:

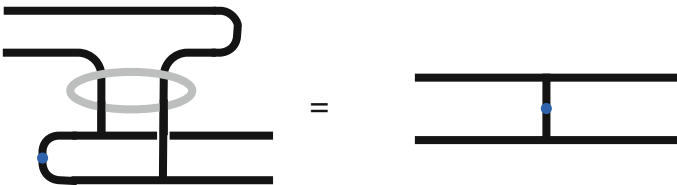


where the dual ring in the middle serves to keep the code space of the defect pair qubit as shown in the left above. It is straightforward to check the transformations of the logical operators. In general, by joining the defect tubes directly, we can transform the logical operators under multiple CNOT gates. However, if two tubes are joined, the definition of the defect pair is broken. In order to keep the defect pair qubit encoding, we need the dual ring wrapping around the two tubes at the joint.

The magic state injection is denoted by the following diagram:



which can be viewed as a superposition of two correlation surfaces for the two anti-commuting logical operators. The non-Clifford gate by one-bit teleportation can be described in the topological diagram as follows:



The right diagram, which is topologically equivalent to the left one, indicates that the non-Clifford gate by one-bit teleportation with the magic state injection is equivalent to simply performing the non-Clifford gate by the method mentioned in the previous section.

The present diagrammatic description of the topological operations and correlation surfaces on them provide us an intuitive understanding of how topological quantum computation on the surface code is performed. These transformation rules will be useful to optimize the complexity (space-time volume required) of the braiding operations.

4.6 Faulty Syndrome Measurements and Noise Thresholds

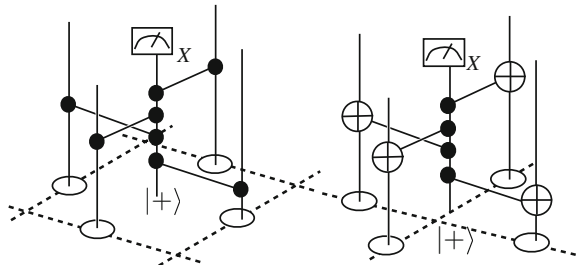
We have seen how universal quantum computation is executed on the surface code. At each step, we have to perform topological quantum error correction to protect the quantum information encoded by the defects. In Sect. 3.4, we analyzed topological quantum error correction on the surface code. However, at that time, we assumed ideal syndrome measurements. However, in a fault-tolerant quantum computation, we have to take into account all reasonable sources of noise, including faulty syndrome measurements, and the quantum computation has to tolerate them as well. Here, we explain how faulty syndrome measurements are handled during topological quantum computation, which completes the big picture of fault-tolerant topological quantum computation.

During the topological operations, the star and plaquette operators are measured in the vacuum region to obtain the error syndrome. These measurements are implemented using ancillae located on each vertex and face center for the star and plaquette operators, respectively (see Fig. 4.12). The ancilla state is prepared to be $|+\rangle$ and the $\Lambda(X)$ or $\Lambda(Z)$ gates are performed from the ancilla qubit as a control to the four qubits on ∂f_l or δv_k . By measuring the ancilla qubit in the X -basis, we obtain the eigenvalue of the star or plaquette operator. If an error is introduced during the measurement, we obtain an incorrect syndrome value, which has to be dealt with in the topological quantum error correction.

Again, we assume that the error is given as a Pauli error for simplicity. We also assume that the X and Z errors, which might be correlated in general, are corrected independently. The former assumption could be justified as follows. Any Kraus operator of the noise map can be decomposed into a superposition of Pauli operators. Such a superposition is collapsed into Pauli errors by the syndrome measurements, which map different Pauli errors (of low weight) into orthogonal subspaces. Note that, in this case, we have to model the error per gate carefully. The latter assumption makes the analysis very simple, but only results in an underestimation of the noise threshold. Below we only consider correction of the Z errors, but it can be applied straightforwardly to the X errors.

Let $c_1^e(t) = \sum_l z_l^e(t) e_l$ be a 1-chain specifying the space-time Z error location, where if $z_l(t) = 1$, a Z error occurs on qubit e_l at time step t . The Z operators on the code state at time t are denoted by $c_1(t) = \sum_l z_l(t) e_l$ (i.e., $Z[c_1(t)]$). The state at

Fig. 4.12 Syndrome measurements for plaquette (left) and star (right) operators, where qubits on a face center and on a vertex are employed as ancillae. The alternative syndrome measurements for the plaquette and star operators are done repeatedly



time t satisfies the following equation of the motion:

$$c_1(t + 1) = c_1(t) + c_1^e(t + 1). \tag{4.13}$$

At each time step t , we measure the star operator $B_k(t)$ and obtain the measurement outcome

$$m_k(t) = \left(\bigoplus_{e_l \in \delta v_k} z_l(t) \right) \oplus z_k^e(t), \tag{4.14}$$

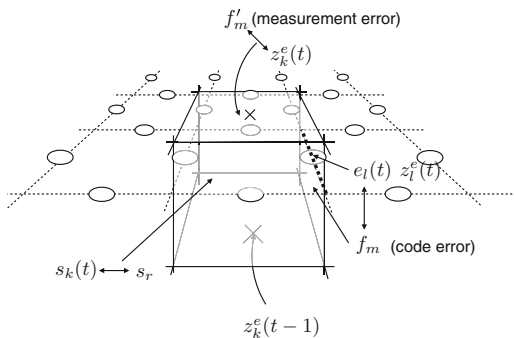
where the bit $z_k^e(t) \in \{0, 1\}$ indicates an error on the measured syndrome, which we call a measurement error. To obtain an equation consisting only of errors, we calculate the parity of the syndrome at time steps t and $t - 1$,

$$s_k(t) = m_k(t) \oplus m_k(t - 1) = \left(\bigoplus_{e_l \in \delta v_k} z_l^e(t) \right) \oplus z_k^e(t) \oplus z_k^e(t - 1). \tag{4.15}$$

Hence, errors on the measured syndrome can be detected by the parity (difference) of the syndrome values at time t and $t - 1$. We redefine $\{s_k(t)\}$ as an error syndrome of the space-time errors including the measurement error.

In the case of the perfect syndrome measurement, the error syndrome is given by the boundary of errors ∂c_1^e , which allows us to use the MWPM algorithm for the error correction. What about the case with the faulty syndrome measurement? Using a chain complex in a 3D manifold, we can again reformulate the error syndrome $\{s_k(t)\}$ as the boundary of a space-time error chain as shown in Fig. 4.13. We consider a chain complex $\{C_0, C_1, C_2, C_3\}$ on a cubic lattice. The basis for the 3-chain is given by a set of cubes $\{q_r\}$. The 3-chain $c_3 = \sum_r z_r q_r \in C_3$ ($z_r \in \{0, 1\}$) is a linear combination over \mathbf{Z}_2 . The dual chain complex $\{\bar{C}_0, \bar{C}_1, \bar{C}_2, \bar{C}_3\}$ is also defined. Specifically, C_i and C_{3-i} are identified via the duality transformation.

Fig. 4.13 A mapping from the chain complex in 2D with time evolution to another chain complex in 3D



Let us explain how to embed the space-time error location to a 1-chain in the chain complex in 3D. The edge $e_l(t)$ and its coefficient $z_l^e(t)$ at time t are mapped into a vertical face f_m and its coefficient z_m^e in the 3D chain complex, respectively. The measurement error $z^e(t)$ is mapped to the coefficient z_m^e of a horizontal face f_m . Then, the space-time error location including the measurement errors is described by a 2-chain or a dual 1-chain in the 3D chain complex:

$$c_2^e = \sum_m z_m^e f_m, \quad \leftrightarrow \quad \bar{c}_1^e = \sum_m z_m^e \bar{e}_m. \quad (4.16)$$

A syndrome $s_k(t)$ is assigned on each cube q_r , or equivalently each dual vertex \bar{v}_r , in the 3D chain complex, and denoted by s_r . Now we realize that

$$s_r = \bigoplus_{f_m \in q_r} z_m^e. \quad (4.17)$$

By using the same argument for the 2D case, $s_r = 1$ iff \bar{v}_r belongs to the boundary $\partial \bar{c}_1^e$ of the dual 1-chain \bar{c}_1^e , specifying the space-time error location. Thus, we can apply the MWPM algorithm on the cubic lattice to perform topological quantum error correction in space-time.

To calculate the noise thresholds, we have to model the noise distribution $P(\bar{c}_1^e)$. First, we consider the simplest case, where the errors are located on each dual edge \bar{e}_l of the dual cubic lattice with probability p . This means that Z errors occur on each qubit independently with probability p at each time step. Moreover, the measured syndrome is flipped with probability p . Such a noise model is called a *phenomenological noise model*. Using the MWPM algorithm, the noise threshold has been estimated to be 2.93 % [133]. The error correction problem under the phenomenological noise model can be mapped into a phase transition of the random-plaquette \mathbf{Z}_2 gauge model by using the same argument as in Sect. 3.5. Specifically, the loop condition for the dual 1-chain (primal 2-chain) in the 3D model can be solved by introducing a gauge spin on each primal edge σ_l and defining the dual trivial 1-cycle $\bar{c}_1 = \sum_m z_m \bar{e}_m \in \text{Img}(\bar{\partial}_2)$ via $(-1)^{z_m} = \prod_{e_l \in \partial f_m} \sigma_l$, where \bar{e}_m and f_m are related through the duality relation. In this way, the variable on the face center is provided as a product of the gauge spins on the boundary of the face, which leads to the random-plaquette \mathbf{Z}_2 gauge model. The ordered Higgs and disordered confinement phases correspond to fault-tolerant and non-fault-tolerant regions, respectively [111]. The threshold value 2.93 % corresponds to the critical point at zero temperature. The optimal threshold is provided on the Nishimori line and has been estimated to be 3.3 % using the Monte Carlo simulation [210].

In fault-tolerant quantum computation, we have to consider any source of noise, including the gate operations, during the syndrome measurement. As a standard way to model a realistic situation, suppose that each elementary gate is followed by a depolarizing channel. This is called a *circuit-based noise model*. More precisely, an ideal single-qubit gate is followed by single-qubit depolarizing noise,

$$(1 - p_1)\rho + \sum_{A \in \{X, Y, Z\}} \frac{p_1}{3} A \rho A. \quad (4.18)$$

An ideal two-qubit gate is followed by two-qubit depolarizing noise,

$$(1 - p_2)\rho + \sum_{A, B \in \{I, X, Y, Z\} \setminus I \otimes I} \frac{p_2}{15} (A \otimes B) \rho (A \otimes B). \quad (4.19)$$

A faulty Pauli-basis state preparation is modeled by the state

$$P_A^+(p_p) = (1 - p_p) \frac{I + A}{2} + p_p \frac{I - A}{2}, \quad (4.20)$$

where $A = X, Y, Z$. A faulty Pauli-basis measurement is modeled by a POVM measurement with POVM elements

$$\{P_A^+(p_m), P_A^-(p_m)\}, \quad (4.21)$$

where

$$P_A^-(p_m) = (1 - p_m) \frac{I - A}{2} + p_m \frac{I + A}{2}. \quad (4.22)$$

Thus, the measurement outcome is flipped with probability p_m .

In Refs. [4–6], the error probabilities were parameterized as $p_1 = p_2 = (3/2)p_p = (3/2)p_m$. The noise threshold with the MWPM algorithm was obtained by numerical simulations to be $p_2 = 0.75\%$. In Ref. [112], the threshold value was further improved to $\sim 1\%$ by assigning a weight for each edge in MWPM appropriately, according to the amount of possible errors. Roughly speaking, the threshold is located where the expectation value of each syndrome becomes 0.7:

$$\langle (-1)^{s_r} \rangle = 0.7. \quad (4.23)$$

In the phenomenological noise model, the expectation value is provided by

$$\langle (-1)^{s_r} \rangle = (1 - 2p)^6, \quad (4.24)$$

which hits 0.7 for $p = 2.89\%$; thus being in good agreement with 2.93%. Moreover, in the case of the circuit-based noise model, p is roughly given by $p_p + p_m + 4p_2$ for the measurement error, which means that one state preparation, one measurement, and four two-qubit gates are employed. The error probability p for the code state is given by $4p_2$, meaning that four two-qubit gates are employed in the syndrome measurements of the star and plaquette operators. This yields

$$\langle (-1)^{s_r} \rangle = (1 - 2p_p - 2p_m - 8p_2)^2 (1 - 8p_2)^4, \quad (4.25)$$

which hits 0.70 for $p_2 = 0.63\%$ with $p_1 = p_2 = (3/2)p_p = (3/2)p_m$. This is again in a good agreement with the numerical result 0.75%. This simple calculation provides rough estimates of the threshold values, but there is no validity. If we need a more accurate threshold value, we should perform a numerical simulation by taking noise propagation and correlation into account [5, 112].

4.7 Summary and Discussion

In this chapter, we have seen how fault-tolerant quantum computation is implemented on the surface code by brading the defects. Specifically, all necessary operations including syndrome measurements are done with nearest-neighbor two-qubit operations and single-qubit operations in 2D, which is favorable for integrating huge number of qubits on a chip. Indeed, topologically protected quantum computation in 2D has been utilized as a platform to design fault-tolerant architectures for quantum computation. One promising approach is on-chip monolithic architectures, such as quantum dots [92, 93], silicon-based nuclear spins [211], and superconducting qubits [116, 117, 119, 121, 212–214], where huge number of qubits are integrated on a single chip, and each individual qubit and the interaction between the qubits are manipulated by multiple lasers or electrodes. Among them, the superconducting qubits system is one of the most promising candidates for implementing topological quantum computation on the surface code, because they can be fabricated on a 2D chip, and all elementary operations have already been demonstrated experimentally [215–218]. The coherence time and gate fidelity of the superconducting systems have improved rapidly. An important breakthrough was made by Martinis's group at University of California Santa Barbara in 2014, where single-qubit gates with a fidelity of 99.92% and two-qubit gates with a fidelity of 99.4% were demonstrated on a 1D array of five superconducting transmon qubits [212]. In addition, repetitive quantum non-demolition measurements were demonstrated on a 1D array of 9 qubits, which improved the fidelity of the stored quantum state by a factor of 8.5, even when using 9 qubits and faulty two-qubit gates and measurements [214]. On the other hand, IBM T. J. Watson Research Center has demonstrated both bit and phase error detections on an integrated superconducting quantum circuits [121, 219]. These are important building blocks of the fault-tolerant quantum computation on the surface code. The readers who are interested in the recent experimental attempts toward fault-tolerant quantum computation should read Refs. [220, 221].

Another approach is the distributed modular architecture, where few-qubit local modules are connected with quantum channels mediating interactions between separate modules [75, 76, 113–115, 222]. The few-qubit quantum module has already been experimentally realized in various physical systems, such as nitrogen-vacancy centers in diamond and trapped ions. Furthermore, entangling operations between separate local modules have been experimentally demonstrated. This experimental and theoretical progress will gradually lead us to large-scale quantum computation.

Chapter 5

Topologically Protected Measurement-Based Quantum Computation

In this chapter, we reformulate topological fault-tolerant quantum computation explained in the previous chapter in terms of measurement-based quantum computation.

5.1 Topological Cluster State in Three Dimensions

Consider a (primal) cubic lattice \mathcal{L} and \mathbf{Z}_2 chain complex on it, $\{C_0, C_1, C_2, C_3\}$, where

$$c_0 = \sum_k z_k v_k \in C_0, \quad c_1 = \sum_l z_l e_l \in C_1, \quad (5.1)$$

$$c_2 = \sum_m z_m f_m \in C_2, \quad c_3 = \sum_n z_n q_n \in C_3, \quad (5.2)$$

with $z_k, z_l, z_m, z_n \in \mathbf{Z}_2$. We also consider a dual cubic lattice $\bar{\mathcal{L}}$ through the relations $v_k \leftrightarrow \bar{q}_k, e_l \leftrightarrow \bar{f}_l, f_m \leftrightarrow \bar{e}_m$, and $q_n \leftrightarrow \bar{v}_n$.

Qubits are defined on the edges and faces of the primal lattice \mathcal{L} (or equivalently primal and dual edges), as shown in Fig. 5.1. We define an operator $A(c_i)$ ($i = 1, 2$) in terms of a 1-chain $c_1 = \sum_j z_j e_j$ or a 2-chain $c_2 = \sum_j z_j f_j$ as

$$A(c_i) = \prod_j A^{z_j}. \quad (5.3)$$

The stabilizer generators of a 3D cluster state for topologically protected MBQC are defined on the primal and dual elementary faces $f_m, \bar{f}_{m'}$ (see Fig. 5.2a):

$$K_{f_m} = X_{f_m} Z(\partial f_m), \quad (5.4)$$

$$K_{\bar{f}_{m'}} = X_{\bar{f}_{m'}} Z(\partial \bar{f}_{m'}). \quad (5.5)$$

Fig. 5.1 A unit cell of the cluster state for topological MBQC. The primal and dual cubes, faces, and edges are also shown

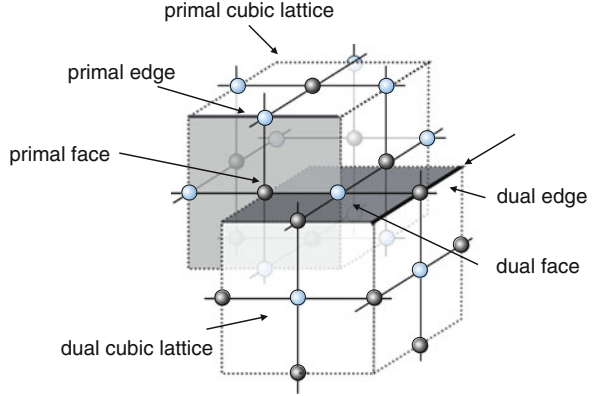
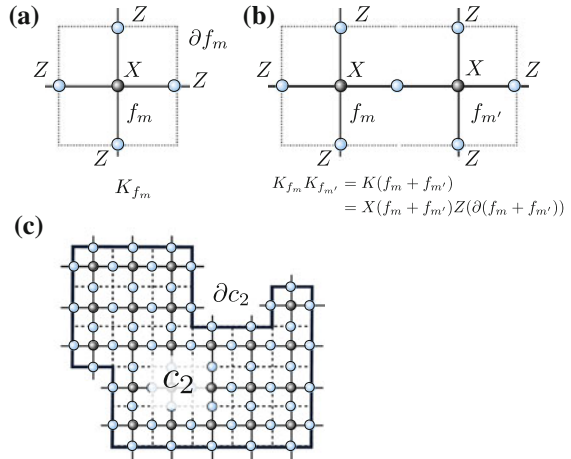


Fig. 5.2 **a** A stabilizer operator defined on a primal face. **b** $K_{f_m} K_{f_{m'}} = X(f_m + f_{m'})Z(\partial(f_m + f_{m'}))$. **c** $K(c_2) \equiv \prod_m K_{f_m}^z = X(c_2)Z(\partial c_2)$



A unit cell of the 3D cluster state is shown in Fig. 5.1. This notion of stabilizer generators of the cluster state is quite useful; it provides a connection between the operators and the chain complex as follows. By multiplying the two stabilizer operators K_{f_m} and $K_{f_{m'}}$, we have

$$K_{f_m} K_{f_{m'}} = X(f_m + f_{m'})Z(\partial(f_m + f_{m'})), \quad (5.6)$$

(see also Fig. 5.2b). By using this property, we can define a stabilizer operator on a 2-chain c_2 ,

$$K(c_2) \equiv \prod_m K_{f_m}^z = X(c_2)Z(\partial c_2), \quad (5.7)$$

(see also Fig. 5.2c). Furthermore, for the two 2-chains, c_2 and c'_2 , we have

$$K(c_2 + c'_2) = K(c_2)K(c'_2). \quad (5.8)$$

Let us see how the 3D cluster state is related to topological quantum computation on the surface code explained in Chap. 4. Recall the circuit diagrams for the syndrome measurements of the plaquette and star operators in Fig. 4.12. The measurement for the plaquette operator is done by applying the CZ gates between the ancilla qubit on the face center and the four qubits on the edges. This operation is the same as generation of the cluster state stabilized by $K(f_m)$ with a horizontal face f_m . The syndrome measurement for the star operator can be done by the CZ gates with the basis change by the Hadamard gates. This corresponds to generation of the cluster state stabilized by $K(\tilde{f}_i)$ with a horizontal dual face \tilde{f}_i . Moreover, the horizontal edge qubits, which constitute the surface code, are connected by applying the CZ gates vertically in order to perform the Hadamard gates for the basis change. In this way, we recover the 3D cluster state stabilized by $K(f_m)$ and $K(\tilde{f}_i)$ for all primal and dual faces f_m and \tilde{f}_i . Two of three dimensions are employed for the spatial degrees of freedom, constituting the surface code. One is for the time evolution of measurement-based quantum computation. The measurements are done along the time-like axis, where even and odd layers, corresponding to the syndrome measurements of the plaquette and star operators respectively, together with constitute an elementary time step of the topologically protected MBQC. Below we will see how the topological operations on the surface code are translated into a measurement pattern of the MBQC on the 3D cluster state.

5.2 Vacuum, Defect, and Singular Qubit Regions

The cubic lattice is divided into three regions: the vacuum \mathcal{V} , defect \mathcal{D} , and singular qubits \mathcal{S} (the detailed definitions are provided later). In the vacuum region, the topological quantum computation is protected through topological quantum error correction. The defect regions are utilized to implement topological quantum computation by braiding defects. We have two types of defects: the primal (D) and dual (\bar{D}) defects. For simplicity, we only consider the primal defect. The extension to the dual case is straightforward by replacing primal by dual in the derivation. The primal defect D is defined as a set of primal cubes. The primal face qubits inside the primal defect (except for those on the boundary ∂D) are measured in the Z -basis to remove the corresponding bonds of the cluster state (or, equivalently, we can prepare the cluster state without those bonds from the beginning). On the boundary ∂D , the primal face qubits are measured in the X -basis. The primal edge qubits belonging to the primal defect (including its boundary) are measured in the X -basis. The measurement pattern for the dual defect is defined similarly.

Unfortunately, only Clifford circuits such as Pauli-basis preparations, measurements, and CNOT gates, are implemented in a topologically protected way. For universal quantum computation, magic states for the non-Clifford gates are injected

on the singular qubits, which are always located in-between two defects. The injections are executed by measuring the singular qubits in the Y - and $(X + Y)/\sqrt{2}$ -bases. These measurements correspond to injections of $(|0\rangle + e^{-i\pi/2}|1\rangle)/\sqrt{2}$ and $(|0\rangle + e^{-i\pi/4}|1\rangle)/\sqrt{2}$ (up to global phases), which are utilized to implement the $S = e^{-i(\pi/4)Z}$ and $T = e^{-i(\pi/8)Z}$ gates via gate teleportation, respectively. The singular qubits are not topologically protected, because two defects are made close to each other resulting in shortening the code distance. However, we can obtain clean magic states with topologically protected Clifford gates through the magic state distillation protocols [60]. In this way, an arbitrary quantum computation is executed fault-tolerantly. Below, we will define these three regions more precisely and see how topological quantum computation is executed in a measurement-based way.

5.3 Elementary Operations in Topological Measurement-Based Quantum Computation

Definition of a Logical Qubit

The logical information is encoded by using a pair of two defects as shown in Fig. 5.3a, b, where the measurements are done from left to right. The logical degree of information at time step t is described a primal 1-chain c_1 surrounding the defect and a dual 1-chain \bar{c}_1 connecting the two defects, as shown in Fig. 5.3a. After measuring qubits up to the $(t - 1)$ th even and odd layers, according to the measurement patterns presented before, the following two operators may become logical operators, which commute with the stabilizer group of the remaining cluster state and are independent of it:

$$L_Z^{(t)} = Z(c_1), \quad L_X^{(t)} = X(\overleftarrow{c}_1)Z(\bar{c}_1), \quad (5.9)$$

where \overleftarrow{c}_1 indicates the dual face qubits on the even layer at time step t that are the left neighbor of \bar{c}_1 , as shown in Fig. 5.3b. These two operators anticommute with each other and represent a logical qubit. (If the cluster state ends at the even layer at the time step t , then the two logical operators are equivalent to the logical operators of the surface code. Because there are the time-like CZ gates for the Hadamard gates, the logical X operator in Eq. (5.9) accompanied by the Z operators.)

Identity Gate

Next, we will see how these logical operators evolve with the measurements. We consider a correlation surface defined by a primal 2-chain c_2 and a dual 2-chain \bar{c}_2 as shown in Fig. 5.4. A stabilizer operator $K(c_2)$ on the correlation surface c_2 surrounding the defect is obtained by multiplying the stabilizer generators K_{f_m} on the primal 2-chain c_2 :

$$K(c_2) \equiv \prod_m K_{f_m}^{\bar{c}_2} = Z(\partial c_2)X(c_2). \quad (5.10)$$

Fig. 5.3 **a** A defect pair logical qubit. **b** The logical operators $L_Z^{(t)}$ and $L_X^{(t)}$ at time step t

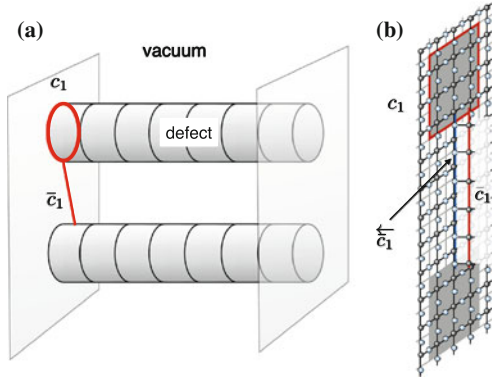
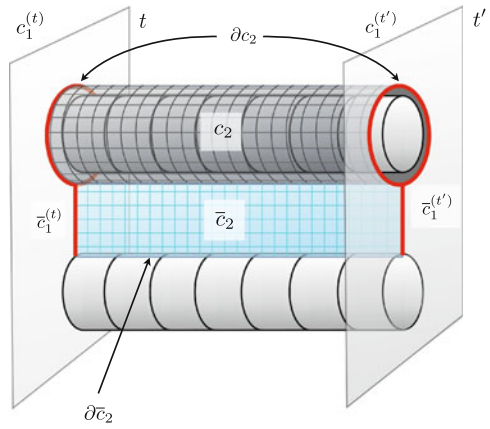


Fig. 5.4 A logical identity gate. The logical operators at time steps t and t' are related by the correlation surfaces $K(c_2)$ and $K(\bar{c}_2)$ via the measurements



Similarly, a stabilizer operator on the dual correlation surface \bar{c}_2 is defined by multiplying $K_{\bar{f}_m}$ on the dual 2-chain \bar{c}_2 :

$$K(\bar{c}_2) \equiv \prod_m K_{\bar{f}_m}^{z_m} = Z(\partial\bar{c}_2)X(\bar{c}_2). \quad (5.11)$$

Suppose measurements are done from the left to the right, except for those qubits on the final even layer. Using the correlation surface, we obtain equivalence relations between the logical operators at time t and t' :

$$L_Z^{(t)} \sim Z(c_1^{(t)})K(c_2) = Z(c_1^{(t')})X(c_2). \quad (5.12)$$

$$L_X^{(t)} \sim X(\bar{c}_1^{(t)})Z(\bar{c}_1^{(t)})K(\bar{c}_2) = X(\bar{c}_1^{(t')})X(\bar{c}_2)Z(\bar{c}_1^{(t')}) \quad (5.13)$$

Here, $A \sim B$ means that A and B are equivalent up to a multiplication of the stabilizer operator of the cluster state, meaning that both A and B act the same on the cluster state. When the qubits on c_2 have been measured in the X -basis, we obtain

$$L_Z^{(t)} \sim L_Z^{(t+1)}, \quad (5.14)$$

$$L_X^{(t)} \sim L_X^{(t+1)}, \quad (5.15)$$

where we assumed that all measurement outcomes are +1 for simplicity. This relation indicates that the logical information at time step t is propagated to time step t' without any operation, i.e., a logical identity operation.

State Preparation and Measurement

Next, we consider how the logical qubit is prepared from the vacuum. To prepare the eigenstate of $L_Z^{(t)}$, we utilize the defect shown in Fig. 5.5a. By considering the correlation surface c_2 , we obtain

$$K(c_2) = X(c_2)L_Z^{(t)}. \quad (5.16)$$

Because $X(c_2)$ commutes with the measurements, the state at time step t is stabilized by $L_Z^{(t)}$, and hence a logical Z-basis state is prepared. Considering another surface c'_2 , shown in Fig. 5.5b, the state at time step t is also stabilized by $Z(\partial c'_2) = L_Z^{(t)}L'_Z{}^{(t)}$. Thus the pair of the defects is appropriately encoded into the code space. Both $L_Z^{(t)}$ and $L'_Z{}^{(t)}$ act equivalently as logical Z operators.

Next, we consider the defect shown in Fig. 5.6. Considering a correlation surface \bar{c}_2 , we obtain

$$K(\bar{c}_2) = Z(\partial \bar{c}_2)X(\bar{c}_2). \quad (5.17)$$

Fig. 5.5 a A logical Z-basis state preparation. b

$$K(c'_2) = X(c'_2)L_Z^{(t)}L'_Z{}^{(t)}$$

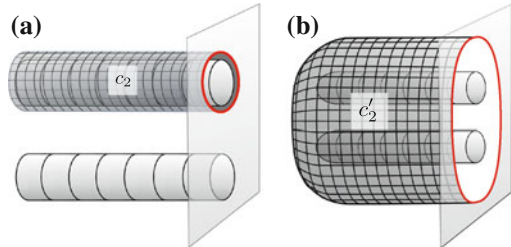
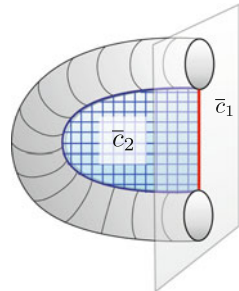


Fig. 5.6 A logical X-basis state preparation



After the measurements, the state at time step t is stabilized by $L_X^{(t)} = X(\overleftarrow{c}_1)Z(c_1)$, where c_1 is a 1-chain on the t th even layer connecting the two defects. Thus, a logical X -basis state is prepared. Again, the logical state is stabilized by $Z(c_1)Z(c'_1)$, with c_1 and c'_1 being a cycle surrounding each defect. Hence, we can choose either $L_Z^{(t)} = Z(c_1)$ or $L_Z^{(t')} = Z(c'_1)$ to serve as the logical operator. The logical measurements of the defect pair qubits can be done with the same defects as the state preparations, but by reversing the time-like direction.

CNOT Gate by Braiding

Let us consider primal defects braiding around a dual defect as shown in Fig. 5.7a. Similarly to the previous case, we calculate the time evolution of logical operators by the measurements. The state at time step t is described by $\{L_Z^{(t)}, L_X^{(t)}\}$ and $\{L_Z'^{(t)}, L_X'^{(t)}\}$ corresponding to $\{c_1^{(t)}, \bar{c}_1^{(t)}\}$ and $\{c_1'^{(t)}, \bar{c}_1'^{(t)}\}$, respectively. We first consider a correlation surface c_2 with respect to $c_1^{(t)}$, as depicted in Fig. 5.7b. Similarly to the identity gate, $L_Z^{(t)}$ is transformed into $L_Z'^{(t)}$. An interesting thing happens when we consider the correlation surface c'_2 with respect to $c_1'^{(t)}$, as shown in Fig. 5.7c. The stabilizer operator on c'_2 is given by

$$\begin{aligned} K(c'_2) &= Z(\partial c'_2)X(c'_2) \\ &= Z(c_1'^{(t)})Z(c_1^{(t)})Z(c_1'^{(t)})Z(c_1^{(t)})X(c'_2), \end{aligned} \tag{5.18}$$

where c_1'' is a cycle in the dual defect of a loop as shown in Fig. 5.7c. Then after the measurements, we obtain an equivalence relation,

$$L_Z^{(t)} \sim L_Z'^{(t)}L_Z'^{(t')}. \tag{5.19}$$

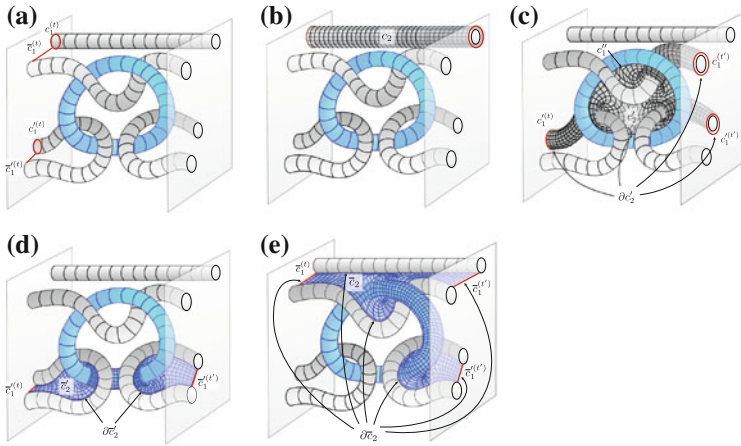


Fig. 5.7 a A diagram for defect braiding for a logical CNOT gate. b–e Time evolutions of the logical operators and the corresponding correlation surfaces

Note that inside the dual defect region, the dual face (primal edge) qubits are measured in the Z -basis, and hence we can obtain the eigenvalue of $Z(c_1')$. After a similar argument using a defect surface \bar{c}_2 and \bar{c}_2' with respect to the dual 1-chain \bar{c}_1 and \bar{c}_1' , shown in Fig. 5.7d and e, respectively, we obtain

$$L_X^{(t)} \sim L_X^{(t)}, \quad (5.20)$$

$$L_X^{(t')} \sim L_X^{(t')} L_X^{(t')}. \quad (5.21)$$

These relations between the logical operators at time steps t and t' are equivalent to those for the CNOT gate. Thus, the defect braiding in Fig. 5.7 results in a logical CNOT gate. Now we realize that the correlation surface introduced in Chap. 4 as a trajectory of the logical operator corresponds to the correlation surface defined by the stabilizer operator of the cluster state.

A Singular Qubit Injection for Magic State Distillation

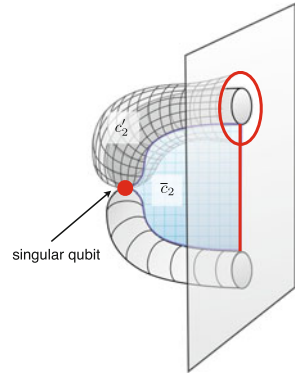
So far, we have shown that the Clifford circuits, Pauli-basis preparation, measurements, and CNOT gate, can all be implemented in a topological way. Unfortunately, these operations are not enough to generate universal quantum computation. To implement universal quantum computation, we inject Y - and $(X + Y)/\sqrt{2}$ -basis states by measuring singular qubits in the Y - and $(X + Y)/\sqrt{2}$ -bases, respectively, as shown in Fig. 5.8. Let us see how this measurement works. Similarly to the previous case, we have two correlation surfaces c_2 and \bar{c}_2 :

$$K(c_2) = Z(\partial c_2)X(c_2) = Z(\partial c_2)X(c_2 \setminus s)X_s, \quad (5.22)$$

$$K(\bar{c}_2) = Z(\partial \bar{c}_2)X(\bar{c}_2) = Z_s Z(\partial \bar{c}_2 \setminus s)X(\bar{c}_2), \quad (5.23)$$

where A_s is a Pauli operator on the singular qubit and $[\cdot] \setminus s$ indicates a chain with a removal of an element corresponding to the singular qubit. Suppose the singular qubit is measured in the Y -basis. After the measurements, the state at time step t is stabilized by

Fig. 5.8 A state injection on the singular qubit



$$K(c_2)K(\bar{c}_2) \simeq L_X^{(t)}L_Z^{(t)} \equiv L_Y^{(t)}. \quad (5.24)$$

Thus, a logical Y -basis state is prepared. When the singular qubit is measured in the $(X + Y)/\sqrt{2}$ -basis, the state at time step t is stabilized by

$$[K(\bar{c}_2) + K(c_2)K(\bar{c}_2)]/\sqrt{2} \simeq (L_X^{(t)} + L_Y^{(t)})/\sqrt{2}, \quad (5.25)$$

which means that a logical $(X + Y)/\sqrt{2}$ -basis state is prepared.

These states are utilized to implement S , T , HSH , and HTH gates using gate teleportation with the CNOT gate. These gates form a universal set of gates.

5.4 Topological Quantum Error Correction in Three Dimensions

Next, we will see how topological quantum error correction is done in 3D. Indeed, in the 3D case, the argument for the noisy syndrome measurements made in Chap. 4 becomes more simple as follows. All measurements in the vacuum region are done in the X -basis. We consider a stabilizer operator on a unit primal cube q ,

$$K(\partial q_n) = \prod_{f_m \in \partial q_n} X_{f_m}, \quad (5.26)$$

where there is no Z operator due to $\partial \circ \partial q_n = 0$. This implies that, if there is no error, the parity of each six X -basis measurement outcomes on the primal cube is always even. The errors are described by using a dual 1-chain $E = Z(\bar{c}_1)$. At a unit cube q_n belonging to $\partial \bar{c}_1$, we have $|q_n \cap \partial \bar{c}_1| = \text{odd}$. (Recall that the primal 3-chain and the dual 0-chain are identified.) From a set of odd parity cubes $\partial \bar{c}_1$, we estimate the actual location of errors $E' = Z(\bar{c}'_1)$ such that $\partial \bar{c}_1 = \partial \bar{c}'_1$. If the total of the actual and estimated error chains $\bar{c}_1 + \bar{c}'_1$ results in a trivial cycle, meaning that there is no defect inside the cycle, it can be contracted and removed by a continuous deformation. If the total of the actual and estimated error chains $\bar{c}_1 + \bar{c}'_1$ results in a nontrivial cycle, meaning a cycle winding around a defect, $EE' = Z(\bar{c}_1 + \bar{c}'_1)$ may result in a logical operator. In such a case, the topological error correction has failed. This property is completely the same as the topological quantum error correction under faulty syndrome measurements argued in Sect. 4.6. If the error probability is smaller than a constant value (the threshold), the failure probability of the topological quantum error correction decreases exponentially in the characteristic size and distance of the defects.

Inside the defect region, the face qubits are measured in the Z -basis. Especially, the Z -basis measurement outcomes near the defect boundary are employed to evaluate the correlation surface. Note that these Z -basis measurements and the removal of the corresponding bonds of the cluster state can instead be done by generating a cluster without connecting the corresponding bonds in advance. In such a case, the errors on the Z -basis measurements do not appear. We can obtain an additional parity K_{f_m}

and $K_{\bar{f}_m'}$ at the primal and dual faces on the boundary of the defects, respectively. If the errors on the face qubits f_m and \bar{f}_m' are suppressed, the errors on the boundary are reduced into errors on a toric code on a 2D surface, ∂D or $\partial \bar{D}$. Again, if the error probability is sufficiently smaller than a constant threshold value, we can correct it faithfully.

The X - and Z -basis state preparations and measurements, and the CNOT gate obtained by braiding, are topologically protected because we can execute these topological operations by keeping the defect size and distance larger than an arbitrarily large constant length. Unfortunately, through the state injection, we shrink the defect size into an elementary unit cell, where the defect size and distance become very small. Thus, the topological protection is broken down around the singular qubit (see Fig. 5.8a). There will also be lower weight errors, which effectively increase the logical error probability on the injected logical states. However, noisy injected states can be purified by using the topologically protected Clifford gates, the so-called magic state distillation. The Y - and $(X + Y)/\sqrt{2}$ -basis states are distilled by using the 7-qubit Steane and 15-qubit Reed-Muller codes, respectively as explained. The distilled states are utilized to implement non-Clifford gates via gate teleportation, as seen before. In this way, universal quantum computation is executed with arbitrary accuracy.

5.5 Applications for Measurement-Based Quantum Computation on Thermal States

Topologically protected MBQC in 3D is useful to study the quantum computational capacity of quantum many-body states at finite temperature. Consider the stabilizer Hamiltonian of the 3D cluster state for topological MBQC [65, 223, 224]:

$$H_{\text{fc}} = -J \left[\sum_f K(f) + \sum_{\bar{f}} K(\bar{f}) \right]. \quad (5.27)$$

The thermal state at temperature $T = 1/(\beta J)$ is given by

$$\rho_{\text{fc}} = e^{-\beta H_{\text{fc}}} / \text{Tr}[e^{-\beta H_{\text{fc}}}], \quad (5.28)$$

Using a unitary operator $U_{CZ} = \prod_{(f_m, \bar{f}_i)} A_{f_m, \bar{f}_i}(Z)$, consisting of CZ gates on all nearest-neighbor two qubits, the thermal state can be mapped into the thermal state of an interaction-free spin model:

$$U_{CZ} \rho_{\text{fc}} U_{CZ}^\dagger = e^{-\beta H_t} / \text{Tr}[e^{-\beta H_t}], \quad (5.29)$$

where

$$H_f \equiv -J \sum_i X_i = U_{CZ} H_{fc} U_{CZ}^\dagger. \quad (5.30)$$

Because H_f is an interaction-free Hamiltonian, the stabilizer Hamiltonian, which we will call the *free cluster Hamiltonian*, does not undergo any thermodynamic phase transition.

The thermal state of the free cluster Hamiltonian is given as a product state of the single-spin density matrix:

$$\rho_f = e^{-\beta H_f} / \text{Tr}[e^{-\beta H_f}] = \prod_i e^{\beta J X_i} / \text{Tr}[e^{\beta J X_i}] \quad (5.31)$$

$$= \prod_i \mathcal{E}_i(p_{\beta J}) (|+\rangle\langle+|)^{\otimes n}, \quad (5.32)$$

where

$$\mathcal{E}_i(p) = (1 - p)\rho + pZ_i\rho Z_i, \quad (5.33)$$

and $p_{\beta J} = e^{-2\beta J} / (1 + e^{-2\beta J})$. Because \mathcal{E}_i and U_{CZ} commute, the thermal state of H_{fc} is rewritten as

$$\begin{aligned} \rho_{fc} &= U_{CZ} \rho_f U_{CZ}^\dagger = \left[\prod_i \mathcal{E}_i(p_{\beta J}) \right] U_{CZ} (|+\rangle\langle+|)^{\otimes n} U_{CZ}^\dagger \\ &= \left[\prod_i \mathcal{E}_i(p_{\beta J}) \right] |\Psi_{3D}\rangle\langle\Psi_{3D}|, \end{aligned} \quad (5.34)$$

where $|\Psi_{3D}\rangle$ is the ground state of H_{fc} , i.e., the 3D cluster state. This means that the thermal state is given as an ideal 3D cluster state, followed by an independent dephasing for each qubit with probability $p_{\beta J} = e^{-2\beta J} / (1 + e^{-2\beta J})$. From the argument made in the previous section, if $p \leq 2.9 - 3.3\%$ and hence $T = 1/(\beta J) \leq 0.57 - 0.59$, we can then perform universal quantum computation reliably on the thermal state at a finite temperature, where the errors originating from the thermal excitations are corrected by the topological quantum error correction. On the other hand, in the high temperature limit $T = 1/(\beta J) \rightarrow \infty$, the thermal state is given by a completely mixed state, and hence MBQC on it can be simulated classically.

A projected-entangled-pair state (PEPS) picture [223–225] allows us to obtain a lower bound for the possibility of a classical simulation. In the PEPS picture, the 3D cluster state is described as follows (see Fig. 5.9). A maximally entangled pair $|\psi_{\text{MES}}\rangle \equiv (|0\rangle|+\rangle + |1\rangle|-\rangle) / \sqrt{2}$ is shared on each bond. On each site consisting of halves of the entangled pair, a projection

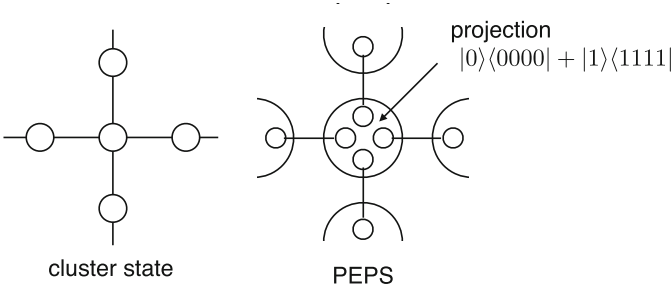


Fig. 5.9 A PEPS picture of the cluster state

$$|0\rangle\langle 00\dots 0| + |1\rangle\langle 11\dots 1| \quad (5.35)$$

is performed with an appropriate normalization. The resultant state is the 3D cluster state. Because the projection and the Z error \mathcal{E}_i^z commute, the effect of the thermal excitations on the shared entangled state can be determined beforehand:

$$\rho_{\text{bond}} = \mathcal{E}_a^z(p)\mathcal{E}_b^z(p)|\psi_{\text{MES}}\rangle\langle\psi_{\text{MES}}| \quad (5.36)$$

If $p \geq (2 - \sqrt{2})/2$, the decohered entangled pair ρ_{bond} becomes a separable state. If two bonds per site are made separable, the 3D cluster state becomes a separable state. A sampling on such a resource state can be simulated efficiently classically [101, 224]. In this case, the Z error probability per site has to be $p_{\beta J} \geq \sqrt{2} - 1$, i.e., $T = 1/(\beta) = 5.77J$. The true critical temperature T_c between the classically simulatable and universal quantum computational phases is located in the range $0.59J < T_c < 5.77J$. Note that this model exhibits a transition of the computational capability, while there is no thermodynamic phase transition in the physical system [223, 224].

In classical information processing, a thermodynamic phase transition, or, more precisely, an ordered phase below a critical temperature is utilized for robust information storage in magnetic storage devices. While there is no such long range order in the previous model, it is natural to ask whether or not a long range ordered phase is useful to enhance the measurement-based quantum computation on many-body thermal states for quantum information processing. To address this issue, Fujii et al. proposed an interacting cluster Hamiltonian [184],

$$H_{\text{ic}} = -J \sum_{\langle f, \bar{f} \rangle} K(f)K(\bar{f}).$$

Because interactions are introduced between the cluster stabilizers, this model is mapped by U_{CZ} into an Ising model on a 3D lattice:

$$H_{\text{Ising}} = U_{CZ}H_{\text{ic}}U_{CZ}^\dagger = -J \sum_{\langle f, \bar{f} \rangle} X_f X_{\bar{f}}.$$

Thus, it undergoes a thermodynamic phase transition at a finite temperature. The degenerate ground states

$$U_{CZ}|+\rangle^{\otimes n} \text{ and } U_{CZ}|-\rangle^{\otimes n} \quad (5.37)$$

are also the 3D cluster states, up to the simultaneous spin flipping due to the global symmetry. Because the eigenvalues of the cluster stabilizer have a long range order (they are likely to be aligned in the same direction) in a ferromagnetic ordered phase, the topologically protected MBQC on the symmetry-breaking thermal state has a special robustness against the thermal excitations. In Ref. [184], topological quantum error correction of this model is mapped to a correlated random plaquette \mathbf{Z}_2 -gauge model in 3D, where the disorder in the signs of the plaquettes has an Ising-type correlation. By using this property and the gauge transformation on the Nishimori line [182], Fujii et al. showed that the critical temperature of this model, and hence the threshold temperature for topological protection, is equal to the critical temperature of the 3D Ising model, which is the unitary equivalent model of the interacting cluster Hamiltonian. This means that the critical temperatures for the topological protection and the thermodynamic phase transition of the underlying physical system coincides exactly. Due to this fact, we can improve the threshold temperature for topological protection by one order of magnitude.

While the above Hamiltonian employs multi-body interactions, the 3D cluster state can be generated from the thermal states of a nearest-neighbor two-body Hamiltonian for spin-3/2 and composite spin-1/2 particles via local filtering operations [226, 227]. Let us consider a system consisting of a spin-3/2 particle located at site \mathbf{r} and a composite particle of two spin-1/2 particles located at the nearest-neighbor site $\mathbf{r} + \mathbf{i}$, with $\mathbf{i} = \mathbf{1}, \mathbf{2}, \mathbf{3}$, as shown in Fig. 5.10a. The Hamiltonian is given by

$$H = \Delta \sum_{\mathbf{r}} \mathbf{S}_{\mathbf{r}} \cdot (\mathbf{I}_{\mathbf{r}+\mathbf{1}} + \mathbf{I}_{\mathbf{r}+\mathbf{2}} + \mathbf{I}_{\mathbf{r}+\mathbf{3}})$$

where $\mathbf{S}_{\mathbf{r}} \equiv (S_{\mathbf{r}}^x, S_{\mathbf{r}}^y, S_{\mathbf{r}}^z)$ is the spin-3/2 operator of the center particle at the position \mathbf{r} and $\mathbf{I}_{\mathbf{r}+\mathbf{a}} = \mathbf{A}_{\mathbf{r}+\mathbf{a}}$ or $\mathbf{B}_{\mathbf{r}+\mathbf{a}}$ depending on the interaction types (line or dash). Here, $\mathbf{A}_{\mathbf{r}+\mathbf{a}} \equiv (A_{\mathbf{r}+\mathbf{a}}^x, A_{\mathbf{r}+\mathbf{a}}^y, A_{\mathbf{r}+\mathbf{a}}^z)$ and $\mathbf{B}_{\mathbf{r}+\mathbf{a}} \equiv (B_{\mathbf{r}+\mathbf{a}}^x, B_{\mathbf{r}+\mathbf{a}}^y, B_{\mathbf{r}+\mathbf{a}}^z)$ are two independent spin-1/2 operators on the composite particle at the position $\mathbf{r} + \mathbf{a}$ ($\mathbf{a} = \mathbf{1}, \mathbf{2}, \mathbf{3}$). The above Hamiltonian H can be reformulated as

$$H = \sum_{\mathbf{r}} H_{\mathbf{r}} = \Delta/2 \sum_{\mathbf{r}} (\mathbf{T}_{\mathbf{r}}^2 - \mathbf{S}_{\mathbf{r}}^2 - \mathbf{I}_{\mathbf{r}}^2)$$

where $\mathbf{I}_{\mathbf{r}} \equiv \mathbf{I}_{\mathbf{r}+\mathbf{1}} + \mathbf{I}_{\mathbf{r}+\mathbf{2}} + \mathbf{I}_{\mathbf{r}+\mathbf{3}}$ and $\mathbf{T}_{\mathbf{r}} \equiv \mathbf{S}_{\mathbf{r}} + \mathbf{I}_{\mathbf{r}}$. The ground state $|G\rangle = \bigotimes_{\mathbf{r}} |g_{\mathbf{r}}\rangle$ is given by $T_{\mathbf{r}} = 0$, $S_{\mathbf{r}} = 3/2$, and $I_{\mathbf{r}} = 3/2$, where $L_{\mathbf{r}}(L_{\mathbf{r}} + 1)$ ($L = T, S, I$) is the eigenvalue of the operator $\mathbf{L}_{\mathbf{r}}^2$. Each center particle in the ground state $|G\rangle$ is filtered by using the POVM measurement:

$$\{F^{\alpha} = (S_{\mathbf{r}}^{\alpha 2} - 1/4)/\sqrt{6}\} \quad (\alpha = x, y, z). \quad (5.38)$$

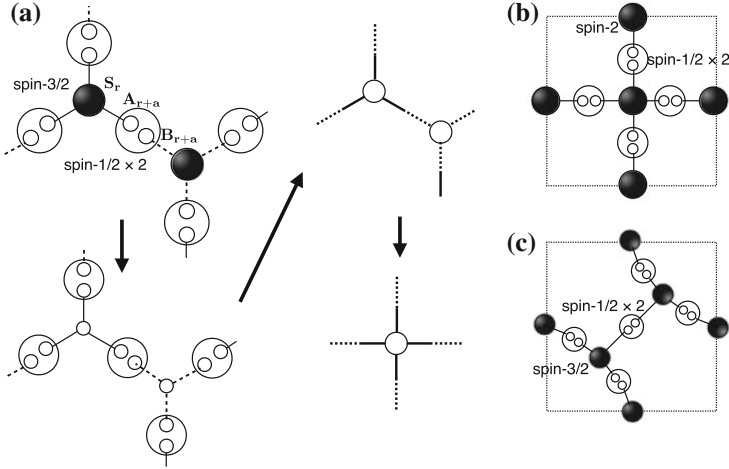


Fig. 5.10 **a** A system consisting of spin-3/2 particles and composite particles of two spin-1/2. After the filtering operation on the ground state, we obtain cluster states. **b** A system consisting of spin-2 particles and composite particles of two spin-1/2 for the 3D cluster state. **c** A system consisting of spin-3/2 particles and composite particles of two spin-1/2 for the 3D cluster state

If the measurement outcome is $\alpha = z$, we obtain a four-qubit GHZ (Greenberger-Horne-Zeilinger) state [228] as the post-POVM measurement state:

$$|\text{GHZ}_r^4\rangle \equiv \frac{1}{\sqrt{2}}(|\tilde{0}\rangle + + +) + |\tilde{1}\rangle - - -),$$

where $|\tilde{1}\rangle$ and $|\tilde{0}\rangle$ are eigenstates of S^z with eigenvalues $+3/2$ and $-3/2$, respectively, and $|\pm\rangle$ are the eigenstates of A^z or B^z with eigenvalues ± 1 , respectively. Even if we obtain other outcomes, we can transform the post-POVM measurement state to $|\text{GHZ}_r^4\rangle$ by local operations. The four-qubit GHZ state is subsequently used to construct the 2D honeycomb cluster state, which is a universal resource for MBQC, by measuring the operators $A^z \otimes B^x$ and $A^x \otimes B^z$ on the bond particle as shown in Fig. 5.10a.

In the case of finite temperature, we have the thermal state $\bigotimes_r \rho_r$ with $\rho_r \equiv e^{-\beta H_r} / \mathcal{Z}$ instead of the ground state, where \mathcal{Z} indicates the partition function and $\beta = T^{-1}$ for a temperature T . Then, the GHZ state becomes a noisy, say thermal, GHZ state, $\sigma_r \equiv F^\alpha \rho_r F^{\alpha \dagger} / \text{Tr}[F^\alpha \rho_r F^{\alpha \dagger}]$. In the low temperature case, the thermal GHZ state is calculated, in the leading order, to be $\mathcal{E}_4(|\text{GHZ}_r^4\rangle\langle\text{GHZ}_r^4|)$ with

$$\begin{aligned} \mathcal{E}_4 = & (1 - q_1 - 3q_2 - 3q_3)[I] + q_1[Z_r] \\ & + q_2 \sum_{\mathbf{a}=1,2,3} [Z_{r+\mathbf{a}}] + q_3 \sum_{\mathbf{a}=1,2,3} [Z_r Z_{r+\mathbf{a}}], \end{aligned} \quad (5.39)$$

where q_1 , q_2 , and q_3 are the error probabilities as functions of the temperature T , the Pauli Z operator $Z_{\mathbf{b}}$ on the qubit at the position \mathbf{b} , and $[C]\rho \equiv C\rho C^\dagger$, respectively. The probability of other errors such as $Z_r Z_{r+a} Z_{r+a'}$ is several orders of magnitude smaller than $q_{1,2,3}$.

To obtain the 3D cluster state for topological MBQC, as done in Ref. [226], the five-qubit GHZ state $|\text{GHZ}_r^5\rangle$ is generated in a similar way by using spin-2 particles and composite particles of spin-1/2, as shown in Fig. 5.10b. Instead of the spin-2 particles, spin-3/2 particles were employed in Ref. [227] to obtain the 3D cluster state shown in Fig. 5.10c. After the filtering operation and local operations, the two four-qubit GHZ states are connected to obtain the five-qubit GHZ state for building the 3D cluster state.

By using the threshold for topologically protected MBQC, we can calculate the threshold temperatures $T = 0.21\Delta$ and $T = 0.18\Delta$ for the cases of spin-2 and spin-3/2 center particles, respectively [226, 227]. Accordingly, we can perform fault-tolerant universal measurement-based quantum computation even on the thermal states of local two-body Hamiltonians at finite temperature.

5.6 Summary and Discussion

In this chapter, we have reformulated topological quantum computation with the surface code in measurement-based quantum computation. In this formulation, the space-time trajectory of the errors with imperfect syndrome measurements discussed in Sect. 4.6 becomes much clearer because they are already mapped into a spatial degree of the 3D cluster state. For certain physical systems, the measurement-based model is more feasible than the circuit model. For example, even if the entangling gates are non-deterministic, we can generate the resource state efficiently [71–76], and universal quantum computation can be executed scalably in the measurement-based model. There have been several proposals of optically connected distributed systems for large scale fault-tolerant quantum computation with the surface code [113, 115, 229, 230].

Appendix A

Fault-Tolerant Quantum Computation

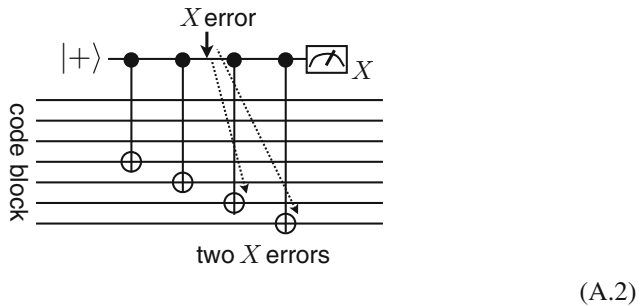
A.1 Fault-Tolerant Syndrome Measurements

In the case of the stabilizer code, the syndrome measurements are done by simply measuring the stabilizer operators. Several QEC gadgets have been proposed to implement the stabilizer measurement fault-tolerantly [104, 106, 231].

DiVincenzo-Shor's gadget—The first QEC gadget was proposed by David DiVincenzo and Peter Shor, who used cat states as ancillae for the syndrome measurement [104]. It is based on an indirect measurement of the observable A with the eigenvalues ± 1 :

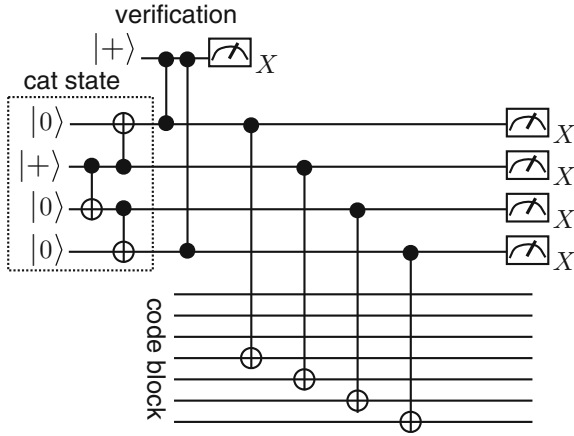
$$\begin{array}{c}
 |+\rangle \text{---} \bullet \text{---} \boxed{X} \\
 | \psi \rangle \text{---} \boxed{A} \text{---} \frac{I \pm A}{2} | \psi \rangle
 \end{array}
 \tag{A.1}$$

For example, the stabilizer S_1 of the seven-qubit code can be measured as the transversal X measurements of the corresponding physical qubits in the code block:



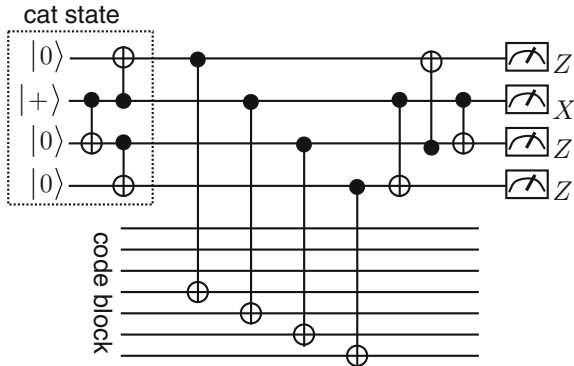
Unfortunately, this measurement is not fault tolerant, because the errors in the CNOT gates [$A = X$ in Eq. (A.1)] are spread by the following CNOT gates, as shown

in the above circuit. To make it fault-tolerant, a cat state $|\text{cat}\rangle = (|000\dots 0\rangle + |111\dots 1\rangle)/\sqrt{2}$ is used as an ancilla for the measurement as follows:



(A.3)

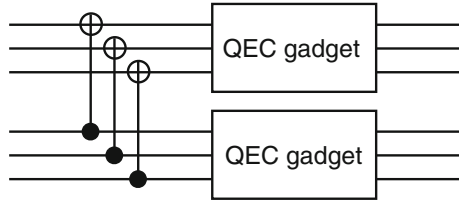
where the cat state is verified before connecting with the code state. Because the qubits in the code block interact with different ancilla qubits, this measurement does not spread the errors in the CNOT gates. Similarly, other stabilizers S_2, \dots, S_6 are measured fault-tolerantly to obtain the error syndrome. Instead of the verification, one can perform a suitable recovery operation by postprocessing the ancilla state after its interaction with the code state [232]:



(A.4)

The DiVincenzo-Shor’s QEC gadget and its improved version both require a lot of physical gate operations, which results in deterioration of the performance.

Steane’s gadget—Subsequently, a relatively simple QEC gadget was proposed by Andrew Steane [231], where encoded ancilla states are used to extract the syndrome with transversal operations. In particular, for the case of the CSS code, the logical



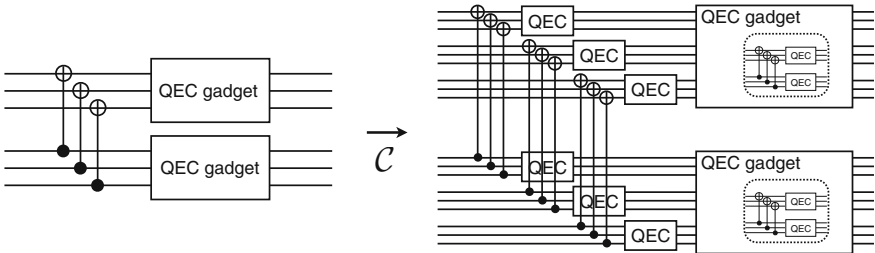
(A.7)

where the code block is depicted as though it is a three-qubit code. A QEC gadget is attached to each logical output of the transversal CNOT gate. Because a single error never will be propagated as multiple errors in a fault-tolerant gate, a logical error is caused by two (or more) simultaneous physical errors. Denoting the number of such faulty pairs of error locations as C , the logical error probability is given by Cp^2 , with p being the physical error probability. If the physical error probability is sufficiently small, $p < 1/C$, one can improve the accuracy of the gate and achieve a fault-tolerant computation.

A.3 Concatenated Quantum Computation

For a reliable computation of a large size, the logical error probability should be reduced arbitrarily. This is done by a concatenated fault-tolerant computation [39, 40, 86, 87]. In the concatenated computation, each physical gate operation is repeatedly replaced by a logical gate operation followed by QEC gadgets.

Suppose A is a quantum computation, which consists of some physical gate operations. Then the first level concatenated computation is defined by $\mathcal{C}(A)$, where the operation \mathcal{C} indicates replacing each physical gate with a logical one, followed by QEC gadgets. For example, $\mathcal{C}(\text{CNOT})$ is described in the diagram (A.7). The second level concatenated CNOT gate $\mathcal{C} \circ \mathcal{C}(\text{CNOT})$ is also described as follows:



(A.8)

where each physical gate in the QEC gadgets is replaced by the logical one, followed by the QEC gadgets. By repeating this procedure, the l th level concatenated computation of A is given by $\mathcal{C}^l(A)$. Specifically, for a physical gate operation G (e.g., Hadamard, CNOT gate, etc.), $\mathcal{C}^l(G)$ is called the level- l G gate. The l th level concatenated code state is called the level- l qubit, denoted by $|0^{(l)}\rangle, |+\rangle^{(l)}$.

As mentioned previously, the logical error probability $p_g^{(1)}$ of the level-1 gate is given by

$$p^{(1)} = C(p^{(0)})^2, \quad (\text{A.9})$$

where $p^{(0)} = p$ and C denotes the number of all faulty pairs of error locations. The constant C differs between the logical gates. It is, however, sufficient to choose the maximum value. Due to the self-similarity of the concatenation, the logical error probability of the level-2 gate is given in terms of $p^{(1)}$ by

$$p^{(2)} = C(p^{(1)})^2. \quad (\text{A.10})$$

Similarly, the logical error probability $p^{(l)}$ of the level- l gate is obtained recursively as

$$\begin{aligned} p^{(l)} &= C(p^{(l-1)})^2 \\ &= (Cp^{(0)})^{2^l} / C. \end{aligned} \quad (\text{A.11})$$

We conclude that if $p^{(0)} < p_{\text{th}} \equiv 1/C$, the logical error probability can be reduced super-exponentially with the concatenation level l . This is the so-called *threshold condition*.

On the other hand, the resources usage, $R^{(l)}$, consumed for the level- l gate is estimated roughly as

$$R^{(l)} = N^l, \quad (\text{A.12})$$

where N indicates the total number of physical gates in the level-1 gate. Suppose that the size of the computation is $10^{n-1} = M$. Then an accuracy of $p^{(l)} < 10^{-n}$ is required for each logical gate at the highest level. The total resources to perform a reliable quantum computation of size M amount to

$$R_{\text{tot}} = N^{\bar{l}} M = \left(\frac{n}{\log_{10}(Cp^{(0)})^{-1}} \right)^{\log_2 N} M = \text{poly}(\log(M))M, \quad (\text{A.13})$$

where $\bar{l} \simeq \log_2 [n / \log_{10}(Cp^{(0)})]$ is the number of levels necessary to achieve the required accuracy. This result clearly shows that if the physical error probability $p^{(0)}$ is smaller than p_{th} , one can execute quantum computation to an arbitrary accuracy with only polylogarithmic overhead. This is the celebrated *threshold theorem* and the critical value p_{th} is called the *noise threshold* [9, 39, 86, 87, 105]. The noise thresholds have been calculated to be about 10^{-4} – 10^{-2} for several fault-tolerant schemes under varying degrees of assumption and rigor [39–43, 86, 87, 106, 238–241].

Appendix B

Decoding Stabilizer Codes

Consider an n -qubit stabilizer code, whose stabilizer group is given by $\mathcal{G} = \{G_i\}$. The group consisting of the logical operators is denoted by $\mathcal{L} = \{L_i\}$. Suppose a Pauli product $E \in \{I, X, Y, Z\}^{\otimes n}$ acts as an error on the stabilizer code state. The error syndrome, a set of eigenvalues of the stabilizer generators, of an error E is denoted by $S = \mathcal{S}(E)$. For each error syndrome S , we define a pure error operator $R(S) \in \{I, X, Y, Z\}^{\otimes n}$ such that $S = \mathcal{S}[R(S)]$. The pure error operator $R(S)$ is chosen arbitrarily as long as $S = \mathcal{S}[R(S)]$. The error E is decomposed uniquely into logical, stabilizer, and pure error operators:

$$E = L_i G_j R[\mathcal{S}(E)]. \tag{B.1}$$

We define a decoding map \mathcal{D} , which computes the logical operator L_i from the error E , i.e., $L_i = \mathcal{D}(E)$.

The decoding problem consists of finding an optimal logical operator L_i that maximizes the posterior probability $P(L|S)$:

$$\arg \max_{L_i \in \mathcal{L}} P(L_i|S). \tag{B.2}$$

The posterior probability is given by

$$P(L_i|S) = \frac{1}{\mathcal{N}} \sum_E P(E) \delta[L_i = \mathcal{D}(E)] \delta[S = \mathcal{S}(E)] \tag{B.3}$$

$$= \frac{1}{\mathcal{N}} \sum_{G_j \in \mathcal{G}} P[L_i G_j R(S)]. \tag{B.4}$$

where $1/\mathcal{N}$ is a normalization factor, $\delta(\dots)$ is an indicator function, and $P(E)$ is the probability of the error E . In general, computing $P(L|S)$ is hard because the summation over all stabilizer operators, \sum_{G_j} , takes an exponential time. However,

if the code has a good structure, such as a concatenated code, we can manage it efficiently.

A concatenated code is defined recursively using the logical qubits at the lower level as physical qubits at the higher level. Using the logical Pauli operators $L_i^{(k)} \in \mathcal{L}^{(k)}$ at the k th level ($k = 1, 2, \dots$), we define a stabilizer group $\mathcal{G}^{(k+1)} = \{G_j^{(k+1)}\}$ and logical operators $\mathcal{L}^{(k+1)} = \{L_i^{(k+1)}\}$ at the $(k+1)$ th level. The whole stabilizer group of the concatenated code is given by the union of all stabilizer groups $\cup_{k=1}^l \mathcal{G}^{(k)}$, where we consider the l th concatenated code. A logical operator of the l th concatenated code is given by a logical operator at the highest level, $L_i^{(l)} \in \mathcal{L}^{(l)}$. At each level, we define a level- k error $E^{(k)}$. In the case of $k = 0$, the level-0 error $E^{(0)} \in \{I, X, Y, Z\}^{\otimes n}$ is a physical error. At a higher level, the level- k error $E^{(k)}$ is a level- k logical operator, which will be defined later. At each level, we define an error syndrome $S^{(k)} = \mathcal{S}^{(k)}(E^{(k-1)})$, i.e., a set of eigenvalues of the level- k stabilizer operators $\{G_j^{(k)}\}$ under the level- $(k-1)$ error $E^{(k-1)}$. The level- k pure error operator $R^{(k)}(S^{(k)})$ is defined arbitrarily such that $\mathcal{S}^{(k)}[R^{(k)}(S^{(k)})] = S^{(k)}$.

The level- k error $E^{(k)}$ is now defined recursively as follows. Any physical error $E^{(0)}$ can be decomposed into the stabilizer, logical, and pure error operators of level-1:

$$E^{(0)} = G_j^{(1)} L_i^{(1)} R^{(1)}(S^{(1)}). \quad (\text{B.5})$$

The obtained level-1 logical operator $L_i^{(1)} = E^{(0)} G_j^{(1)} R^{(1)}(S^{(1)})$ is further regarded as the level-1 error $E^{(1)} \equiv L_i^{(1)}$. Recursively, the level- k error $E^{(k)}$ is decomposed into the stabilizer, logical, and pure error operators of level- $(k+1)$:

$$E^{(k)} = G_j^{(k+1)} L_i^{(k+1)} R^{(k+1)}(S^{(k+1)}). \quad (\text{B.6})$$

Then, we define the level- $(k+1)$ error $E^{(k+1)} = L_i^{(k+1)} = E^{(k)} G_j^{(k+1)} R^{(k+1)}(S^{(k+1)})$. Let $\mathcal{D}^{(k)}$ be such a level- k decoding map $L_i^{(k)} = \mathcal{D}^{(k)}(E^{(k-1)})$, which computes the level- k logical operator from the level- $(k-1)$ error $E^{(k-1)}$. At the highest level, we have a decomposition

$$E^{(0)} = L_i^{(l)} \prod_{k=1}^l G_{j_k}^{(k)} R^{(k)}[\mathcal{S}^{(k)}(E^{(k-1)})], \quad (\text{B.7})$$

where $E^{(k)}$ is defined by Eq. (B.6).

The decoding is performed by maximizing the posterior probability of the logical operator $L_i^{(l)}$, conditioned on the union of the error syndrome $\bar{S}^{(l)} = \cup_{k=1}^l S^{(k)}$ up to the l th concatenation level,

$$P\left(L_i^{(l)}|\bar{S}^{(l)}\right) = \frac{1}{\mathcal{N}} \sum_{G_{i_l}^{(l)}, G_{i_{l-1}}^{(l-1)}, \dots, G_{i_1}^{(1)}} P(E^{(0)}), \quad (\text{B.8})$$

where $E^{(0)}$ is given by Eq. (B.7). Using the hierarchal structure, Eq. (B.8), can be rewritten as

$$\begin{aligned} & P\left(L_i^{(l)}|\bar{S}^{(l)}\right) \quad (\text{B.9}) \\ &= \sum_{L_{i_{l-1}}^{(l-1)} \in \mathcal{L}^{(l-1)}} P\left(L_i^{(l)}|\bar{S}^{(l)}, L_{i_{l-1}}^{(l-1)}\right) P\left(L_{i_{l-1}}^{(l-1)}|\bar{S}^{(l)}\right) \\ &= \sum_{L_{i_{l-1}}^{(l-1)} \in \mathcal{L}^{(l-1)}} \delta\left[L_{i_l}^{(l)} = \mathcal{D}^{(l)}\left(L_{i_{l-1}}^{(l-1)}\right)\right] \frac{P\left(L_{i_{l-1}}^{(l-1)}, \bar{S}^{(l)}\right)}{P\left(\bar{S}^{(l)}\right)} \\ &= \sum_{L_{i_{l-1}}^{(l-1)} \in \mathcal{L}^{(l-1)}} \delta\left[L_{i_l}^{(l)} = \mathcal{D}^{(l)}\left(L_{i_{l-1}}^{(l-1)}\right)\right] \delta\left[S^{(l)} = \mathcal{S}^{(l)}\left(L_{i_{l-1}}^{(l-1)}\right)\right] \frac{P\left(L_{i_{l-1}}^{(l-1)}, \bar{S}^{(l-1)}\right)}{P\left(\bar{S}^{(l)}\right)} \\ &= \sum_{L_{i_{l-1}}^{(l-1)} \in \mathcal{L}^{(l-1)}} \frac{\delta\left[L_{i_l}^{(l)} = \mathcal{D}^{(l)}\left(L_{i_{l-1}}^{(l-1)}\right)\right] \delta\left[S^{(l)} = \mathcal{S}^{(l)}\left(L_{i_{l-1}}^{(l-1)}\right)\right]}{P\left(\bar{S}^{(l)}|\bar{S}^{(l-1)}\right)} P\left(L_{i_{l-1}}^{(l-1)}|\bar{S}^{(l-1)}\right) \end{aligned} \quad (\text{B.10})$$

If the physical errors occur independently for each qubit, we can factorize the posterior probability of the $(l-1)$ th level into posterior probabilities for level- $(l-1)$ code blocks

$$P\left(L_{i_{l-1}}^{(l-1)}|\bar{S}^{(l-1)}\right) = \prod_j P\left(L_{i_{l-1}}^{(l-1,j)}|\bar{S}^{(l-1,j)}\right), \quad (\text{B.11})$$

where $L_{i_{l-1}}^{(l-1,j)}$ is a level- $(l-1)$ logical operator acting on the j th code block, and $\bar{S}^{(l-1,j)}$ is a level- $(l-1)$ syndrome with respect to the stabilizer operator on the j th code block. By repeating this procedure, we can rewrite the posterior probability Eq. (B.8) as a summation over logical operators defined at each concatenation level $L_{i_k}^{(k,j)} \in \mathcal{L}^{(k,j)}$, where we have defined the group of the level- k logical operators on the j th code block, $\mathcal{L}^{(k,j)}$.

The summation can be reformulated as a marginalization problem on a factor graph, which is a bipartite graph consisting of two types of nodes, circles, and boxes [139]. The variable x_c and a function $f_b(\partial b)$ are assigned on each circle c and box b . Here, ∂b is a set of circles neighboring b . Specifically, in the present case, the factor graph is a tree graph. The marginal distribution on a tree factor graph

can be computed efficiently by using the brief-propagation method as follows. From circles to boxes, we pass a message,

$$\mu_{c \rightarrow b}(x_c) = \prod_{b' \in \delta c \setminus b} v_{b' \rightarrow c}(x_c). \quad (\text{B.12})$$

Then from boxes to circles, we pass another message

$$v_{b \rightarrow c}(x_c) = \sum_{\delta b \setminus x_c} \prod_b f_b(\delta b) \prod_{c' \in \delta b \setminus c} \mu_{c' \rightarrow b}(x'_c). \quad (\text{B.13})$$

By repeating these procedures alternatively, we can obtain $P(L_i^{(l)} | \bar{S}^{(l)})$ as a message $v_{b \rightarrow c}(x_c)$, where the $\mu_{c \rightarrow b}(x_c)$ and $v_{b \rightarrow c}(x_c)$ are updated from the bottom leaf nodes to the top root node, while using marginal distributions. By replacing $\mu_{c \rightarrow b}(x_c)$ and $f_b(\delta b)$ with $P(L_{i_{k-1}}^{(k-1,j)} | \bar{S}^{(k-1,j)})$ and $\delta[L_{i_k}^{(k)} = \mathcal{D}^{(k)}(L_{i_{k-1}}^{(k-1)})] \delta[S^{(k)} = \mathcal{S}^{(k)}(L_{i_{k-1}}^{(k-1)})]$, we obtain the posterior probability $P(L_i^{(l)} | \bar{S}^{(l)})$ as $v_{b \rightarrow c}(x_c)$ at the top root node. In this way, decoding by maximizing the posterior probability can be executed efficiently for the concatenated stabilizer codes [139].

References

1. Feynman, R.P.: *Int. J. Theor. Phys.* **21**, 467 (1982)
2. Deutsch, D.: *Proc. R. Soc. Lond. A. Math. Phys. Sci.* **400**, 97 (1985)
3. Lloyd, S.: *Nature* **400**, 720 (1999)
4. Raussendorf, R., Harrington, J., Goyal, K.: *Ann. Phys.* **321**, 2242 (2006)
5. Raussendorf, R., Harrington, J., Goyal, K.: *New J. Phys.* **9**, 199 (2007)
6. Raussendorf, R., Harrington, J.: *Phys. Rev. Lett.* **98**, 190504 (2007)
7. Dawson, C.M., Nielsen, M.A.: *Quantum Inf. Comput.* **6**, 081 (2005)
8. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D.P., Margolus, N., Shor, P., Sleator, T., Smolin, J.A., Weinfurter, H.: *Phys. Rev. A* **52**, 3457 (1995)
9. Nielsen, M.A., Chuang, I.L.: *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge (2000)
10. Reck, M., Zeilinger, A., Bernstein, H.J., Bertani, P.: *Phys. Rev. Lett.* **73**, 58 (1994)
11. DiVincenzo, D.P.: *Phys. Rev. A* **51**, 1015 (1995)
12. Bernstein, E., Vazirani, U.: In: *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pp. 11–20. ACM (1993)
13. Shor, P.W.: In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science. FOCS'94*, pp. 124–134 (1994)
14. Shor, P.W.: *SIAM J. Comput.* **26**, 1484 (1997)
15. Aharonov, D., Jones, V., Landau, Z.: *Algorithmica* **55**, 395 (2009)
16. Aharonov, D., Arad, I.: arXiv preprint [arXiv:quant-ph/0605181](https://arxiv.org/abs/quant-ph/0605181) (2006)
17. Kitaev, A.Y.: arXiv preprint [arXiv:quant-ph/9511026](https://arxiv.org/abs/quant-ph/9511026) (1995)
18. Aharonov, D., Arad, I., Eban, E., Landau, Z.: arXiv preprint [arXiv:quant-ph/0702008](https://arxiv.org/abs/quant-ph/0702008) (2007)
19. De las Cuevas, G., Dür, W., Van den Nest, M., Martin-Delgado, M.: *New J. Phys.* **13**, 093021 (2011)
20. Iblisdir, S., Cirio, M., Boada, O., Brennen, G.: *Ann. Phys.* **340**, 205 (2014)
21. Matsuo, A., Fujii, K., Imoto, N.: *Phys. Rev. A* **90**, 022304 (2014)
22. Knill, E., Laflamme, R.: *Phys. Rev. Lett.* **81**, 5672 (1998)
23. Shor, P.W., Jordan, S.P.: *Quantum Inf. Comput.* **8**, 681 (2008)
24. Jordan, S.P., Wocjan, P.: *Quantum Inf. Comput.* **9**, 264 (2009)
25. Poulin, D., Blume-Kohout, R., Laflamme, R., Ollivier, H.: *Phys. Rev. Lett.* **92**, 177906 (2004)
26. Morimae, T., Fujii, K., Fitzsimons, J.F.: *Phys. Rev. Lett.* **112**, 130502 (2014)
27. Morimae, T., Nishimura, H., Fujii, K., Tamate, S.: arXiv preprint [arXiv:1409.6777](https://arxiv.org/abs/1409.6777) (2014)
28. Kitaev, A.Y.: arXiv preprint [arXiv:quant-ph/9511026](https://arxiv.org/abs/quant-ph/9511026) (1995)
29. Freedman, M., Kitaev, A., Larsen, M., Wang, Z.: *Bull. Am. Math. Soc.* **40**, 31 (2003)
30. Witten, E.: *Commun. Math. Phys.* **121**, 351 (1989)
31. Lickorish, W.R.: *An Introduction to Knot Theory*, vol. 175. Springer Science & Business Media, New York (1997)

32. Temperley, H.N., Lieb, E.H.: Proc. R. Soc. Lond. A. Math. Phys. Sci. **322**, 251 (1971)
33. Wocjan, P., Yard, J.: arXiv preprint [arXiv:quant-ph/0603069](https://arxiv.org/abs/0603069) (2006)
34. Petruccione, F., Breuer, H.P.: The Theory of Open Quantum Systems. Oxford University Press, Oxford (2002)
35. Briegel, H.J., Englert, B.G.: Phys. Rev. A **47**, 3311 (1993)
36. Dür, W., Hein, M., Cirac, J., Briegel, H.J.: Phys. Rev. A **72**, 052326 (2005)
37. Grover, L.K.: In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, pp. 212–219. ACM (1996)
38. Pachos, J.K.: Introduction to Topological Quantum Computation. Cambridge University Press, Cambridge (2012)
39. Aharonov, D., Ben-Or, M.: In: Proceedings of the ACM Symposium on Theory of Computing, pp. 176–188. ACM (1997)
40. Aharonov, D., Ben-Or, M.: SIAM J. Comput. **38**, 1207 (2008)
41. Aliferis, P., Gottesman, D., Preskill, J.: Quantum Inf. Comput. **6**, 97 (2006)
42. Aliferis, P., Gottesman, D., Preskill, J.: Quantum Inf. Comput. **8**, 181 (2008)
43. Aliferis, P., Preskill, J.: Phys. Rev. A **79**, 012332 (2009)
44. Terhal, B.M., Burkard, G.: Phys. Rev. A **71**, 012336 (2005)
45. Aharonov, D., Kitaev, A., Preskill, J.: Phys. Rev. Lett. **96**, 050504 (2006)
46. Zurek, W.H.: Phys. Rev. Lett. **53**, 391 (1984)
47. Vaidman, L., Goldenberg, L., Wiesner, S.: Phys. Rev. A **54**, R1745 (1996)
48. Duan, L.M., Guo, G.C.: Phys. Rev. A **57**, 2399 (1998)
49. Viola, L., Lloyd, S.: Phys. Rev. A **58**, 2733 (1998)
50. Viola, L., Knill, E., Lloyd, S.: Phys. Rev. Lett. **82**, 2417 (1999)
51. Facchi, P., Tasaki, S., Pascazio, S., Nakazato, H., Tokuse, A., Lidar, D.: Phys. Rev. A **71**, 022302 (2005)
52. Zanardi, P., Rasetti, M.: Phys. Rev. Lett. **79**, 3306 (1997)
53. Duan, L.M., Guo, G.C.: Phys. Rev. Lett. **79**, 1953 (1997)
54. Lidar, D.A., Chuang, I.L., Whaley, K.B.: Phys. Rev. Lett. **81**, 2594 (1998)
55. Einstein, A., Podolsky, B., Rosen, N.: Phys. Rev. **47**, 777 (1935)
56. Gottesman, D.: Stabilizer codes and quantum error correction. Ph.D. thesis, California Institute of Technology (1997)
57. Gottesman, D.: arXiv preprint [arXiv:quant-ph/9807006](https://arxiv.org/abs/9807006) (1998)
58. Bremner, M.J., Jozsa, R., Shepherd, D.J.: Proc. R. Soc. A: Math. Phys. Eng. Sci. **467**, 459 (2011)
59. Terhal, B., DiVincenzo, D.: Quantum Inf. Comput. **4**, 134 (2004)
60. Bravyi, S., Kitaev, A.: Phys. Rev. A **71**, 022316 (2005)
61. Hein, M., Dür, W., Eisert, J., Raussendorf, R., Van der Nest, M., Briegel, H.: Int. Sch. Phys. Enrico Fermi **162** (2006)
62. Briegel, H.J., Raussendorf, R.: Phys. Rev. Lett. **86**, 910 (2001)
63. Van den Nest, M., Dehaene, J., De Moor, B.: Phys. Rev. A **69**, 022316 (2004)
64. Raussendorf, R., Briegel, H.J.: Phys. Rev. Lett. **86**, 5188 (2001)
65. Raussendorf, R., Browne, D.E., Briegel, H.J.: Phys. Rev. A **68**, 022312 (2003)
66. Raussendorf, R.: Measurement-based quantum computation with cluster states. Ph.D. thesis, Ludwig-Maximilians Universität München (2003)
67. Bennett, C.H., Brassard, G., Crépeau, C., Jozsa, R., Peres, A., Wootters, W.K.: Phys. Rev. Lett. **70**, 1895 (1993)
68. Gottesman, D., Chuang, I.L.: Nature **402**, 390 (1999)
69. Nielsen, M.A.: Rep. Math. Phys. **57**, 147 (2006)
70. Yoran, N., Reznik, B.: Phys. Rev. Lett. **91**, 037903 (2003)
71. Nielsen, M.A.: Phys. Rev. Lett. **93**, 040503 (2004)
72. Browne, D.E., Rudolph, T.: Phys. Rev. Lett. **95**, 010501 (2005)
73. Duan, L.M., Raussendorf, R.: Phys. Rev. Lett. **95**, 080503 (2005)
74. Matsuzaki, Y., Benjamin, S.C., Fitzsimons, J.: Phys. Rev. Lett. **104**, 050501 (2010)
75. Li, Y., Barrett, S.D., Stace, T.M., Benjamin, S.C.: Phys. Rev. Lett. **105**, 250502 (2010)

76. Fujii, K., Tokunaga, Y.: *Phys. Rev. Lett.* **105**, 250503 (2010)
77. Broadbent, A., Fitzsimons, J., Kashefi, E.: In: 50th Annual IEEE Symposium on Foundations of Computer Science. FOCS'09, pp. 517–526. IEEE (2009)
78. Childs, A.M.: *Quantum Inf. Comput.* **5**, 456 (2005)
79. Arrighi, P., Salvail, L.: *Int. J. Quantum Inf.* **4**, 883 (2006)
80. Morimae, T., Fujii, K.: *Phys. Rev. A* **87**, 050301 (2013)
81. Morimae, T., Fujii, K.: *Nat. Commun.* **3**, 1036 (2012)
82. Shor, P.W.: *Phys. Rev. A* **52**, R2493 (1995)
83. Laflamme, R., Miquel, C., Paz, J.P., Zurek, W.H.: *Phys. Rev. Lett.* **77**, 198 (1996)
84. Bennett, C.H., DiVincenzo, D.P., Smolin, J.A., Wootters, W.K.: *Phys. Rev. A* **54**, 3824 (1996)
85. Steane, A.M.: *Phys. Rev. Lett.* **77**, 793 (1996)
86. Knill, E., Laflamme, R., Zurek, W.H.: *Proc. R. Soc. Lond. Ser. A: Math. Phys. Eng. Sci.* **454**, 365 (1998)
87. Knill, E., Laflamme, R., Zurek, W.H.: *Science* **279**, 342 (1998)
88. Meier, A.M., Eastin, B., Knill, E.: arXiv preprint [arXiv:1204.4221](https://arxiv.org/abs/1204.4221) (2012)
89. Jones, C.: *Phys. Rev. A* **87**, 042305 (2013)
90. Reichardt, B.W.: *Quantum Inf. Process.* **4**, 251 (2005)
91. MacWilliams, F.J., Sloane, N.J.A.: *The Theory of Error Correcting Codes*, vol. 16. Elsevier, Amsterdam (1977)
92. Van Meter, R., Ladd, T.D., Fowler, A.G., Yamamoto, Y.: *Int. J. Quantum Inf.* **8**, 295 (2010)
93. Jones, N.C., Van Meter, R., Fowler, A.G., McMahon, P.L., Kim, J., Ladd, T.D., Yamamoto, Y.: *Phys. Rev. X* **2**, 031007 (2012)
94. Bravyi, S., Haah, J.: *Phys. Rev. A* **86**, 052329 (2012)
95. Eastin, B.: *Phys. Rev. A* **87**, 032321 (2013)
96. Jones, C.: *Phys. Rev. A* **87**, 022328 (2013)
97. Fowler, A.G., Stephens, A.M., Groszkowski, P.: *Phys. Rev. A* **80**, 052312 (2009)
98. Almheiri, A., Dong, X., Harlow, D.: *J. High Energy Phys.* **2015**, 1 (2014)
99. Pastawski, F., Yoshida, B., Harlow, D., Preskill, J.: arXiv preprint [arXiv:1503.06237](https://arxiv.org/abs/1503.06237) (2015)
100. Howard, M., Wallman, J., Veitch, V., Emerson, J.: *Nature* **510**, 351 (2014)
101. Fujii, K., Tamate, S.: arXiv preprint [arXiv:1406.6932](https://arxiv.org/abs/1406.6932) (2014)
102. Lidar, D.A., Chuang, I.L., Whaley, K.B.: *Phys. Rev. Lett.* **81**, 2594 (1998)
103. Viola, L., Knill, E., Lloyd, S.: *Phys. Rev. Lett.* **82**, 2417 (1999)
104. DiVincenzo, D.P., Shor, P.W.: *Phys. Rev. Lett.* **77**, 3260 (1996)
105. Kitaev, A.Y.: *Russ. Math. Surv.* **52**, 1191 (1997)
106. Knill, E.: *Nature* **434**, 39 (2005)
107. Fujii, K., Yamamoto, K.: *Phys. Rev. A* **81**, 042324 (2010)
108. Fujii, K., Yamamoto, K.: *Phys. Rev. A* **82**, 060301 (2010)
109. Svore, K.M., Terhal, B.M., DiVincenzo, D.P.: *Quantum Inf. Comput.* **7**, 297 (2007)
110. Kitaev, A.Y.: *Ann. Phys.* **303**, 2 (2003)
111. Dennis, E., Kitaev, A., Landahl, A., Preskill, J.: *J. Math. Phys.* **43**, 4452 (2002)
112. Wang, D.S., Fowler, A.G., Hollenberg, L.C.: *Phys. Rev. A* **83**, 020302 (2011)
113. Fujii, K., Yamamoto, T., Koashi, M., Imoto, N.: arXiv preprint [arXiv:1202.6588](https://arxiv.org/abs/1202.6588) (2012)
114. Li, Y., Benjamin, S.C.: *New J. Phys.* **14**, 093008 (2012)
115. Nickerson, N.H., Li, Y., Benjamin, S.C.: *Nat. Commun.* **4**, 1756 (2013)
116. Ghosh, J., Fowler, A.G., Geller, M.R.: *Phys. Rev. A* **86**, 062318 (2012)
117. Fowler, A.G., Mariantoni, M., Martinis, J.M., Cleland, A.N.: *Phys. Rev. A* **86**, 032324 (2012)
118. Chow, J.M., Gambetta, J.M., Córcoles, A., Merkel, S.T., Smolin, J.A., Rigetti, C., Poletto, S., Keefe, G.A., Rothwell, M.B., Rozen, J., et al.: *Phys. Rev. Lett.* **109**, 060501 (2012)
119. Chow, J.M., Gambetta, J.M., Magesan, E., Srinivasan, S.J., Cross, A.W., Abraham, D.W., Masluk, N.A., Johnson, B., Ryan, C.A., Steffen, M.: *Nat. Commun.* **5**, 4015 (2014)
120. Barends, R., Kelly, J., Megrant, A., Veitia, A., Sank, D., Jeffrey, E., White, T., Mutus, J., Fowler, A., Campbell, B., et al.: *Nature* **508**, 500 (2014)
121. Córcoles, A., Magesan, E., Srinivasan, S.J., Cross, A.W., Steffen, M., Gambetta, J.M., Chow, J.M.: arXiv preprint [arXiv:1410.6419](https://arxiv.org/abs/1410.6419) (2014)

122. Bravyi, S., Terhal, B.: *New J. Phys.* **11**, 043029 (2009)
123. Yoshida, B.: *Ann. Phys.* **326**, 2566 (2011)
124. Fujii, K., Tokunaga, Y.: *Phys. Rev. A* **86**, 020303 (2012)
125. Bravyi, S.B., Kitaev, A.Y.: arXiv preprint [arXiv:quant-ph/9811052](https://arxiv.org/abs/1908.09462) (1998)
126. Hsieh, M.H., Le Gall, F.: *Phys. Rev. A* **83**, 052331 (2011)
127. Kuo, K.Y., Lu, C.C.: In: *International Symposium on Information Theory and its Applications (ISITA)*, pp. 208–211. IEEE (2012)
128. Iyer, P., Poulin, D.: arXiv preprint [arXiv:1310.3235](https://arxiv.org/abs/1310.3235) (2013)
129. Edmonds, J.: *Can. J. Math.* **17**, 449 (1965)
130. Barahona, F.: *J. Phys. A: Math. Gen.* **15**, 3241 (1982)
131. Cook, W., Rohe, A.: *INFORMS J. Comput.* **11**, 138 (1999)
132. Kolmogorov, V.: *Math. Program. Comput.* **1**, 43 (2009)
133. Wang, C., Harrington, J., Preskill, J.: *Ann. Phys.* **303**, 31 (2003)
134. Kasteleyn, P.W.: *Physica* **27**, 1209 (1961)
135. Bravyi, S., Raussendorf, R.: *Phys. Rev. A* **76**, 022304 (2007)
136. Fowler, A.G., Whiteside, A.C., Hollenberg, L.C.: *Phys. Rev. Lett.* **108**, 180501 (2012)
137. Duclos-Cianci, G., Poulin, D.: *Phys. Rev. Lett.* **104**, 050504 (2010)
138. Duclos-Cianci, G., Poulin, D.: In: *Information Theory Workshop (ITW)*, IEEE. pp. 1–5. IEEE (2010)
139. Poulin, D.: *Phys. Rev. A* **74**, 052333 (2006)
140. Bombin, H., Duclos-Cianci, G., Poulin, D.: *New J. Phys.* **14**, 073048 (2012)
141. Hukushima, K.: *Phys. Rev. E* **60**, 3606 (1999)
142. Nishimori, H.: *Prog. Theor. Phys.* **66**, 1169 (1981)
143. Reis, F.A., De Queiroz, S., Dos Santos, R.R.: *Phys. Rev. B* **60**, 6740 (1999)
144. Honecker, A., Picco, M., Pujol, P.: *Phys. Rev. Lett.* **87**, 047201 (2001)
145. Merz, F., Chalker, J.: *Phys. Rev. B* **65**, 054425 (2002)
146. Calderbank, A.R., Shor, P.W.: *Phys. Rev. A* **54**, 1098 (1996)
147. Nishimori, H., Nemoto, K.: *J. Phys. Soc. Jpn.* **71**, 1198 (2002)
148. Nishimori, H.: *J. Phys. C: Solid State Phys.* **12**, L905 (1979)
149. Kramers, H.A., Wannier, G.H.: *Phys. Rev.* **60**, 252 (1941)
150. Ohzeki, M.: In: Nakahara, M., Tanaka, S. (eds.) *Lectures on Quantum Computing, Thermodynamics and Statistical Physics*, vol. 8. World Scientific (2013)
151. Ohzeki, M.: *Phys. Rev. E* **79**, 021129 (2009)
152. Hasenbusch, M., Toldin, F.P., Pelissetto, A., Vicari, E.: *Phys. Rev. E* **77**, 051115 (2008)
153. Toldin, F.P., Pelissetto, A., Vicari, E.: *J. Stat. Phys.* **135**, 1039 (2009)
154. Kawashima, N., Rieger, H.: *Europhys. Lett.* **39**, 85 (1997)
155. Röthlisberger, B., Wootton, J.R., Heath, R.M., Pachos, J.K., Loss, D.: *Phys. Rev. A* **85**, 022313 (2012)
156. Ohzeki, M., Fujii, K.: *Phys. Rev. E* **86**, 051121 (2012)
157. Al-Shimary, A., Wootton, J.R., Pachos, J.K.: *New J. Phys.* **15**, 025027 (2013)
158. Kay, A.: *Phys. Rev. A* **89**, 032328 (2014)
159. Alieris, P., Brito, F., DiVincenzo, D.P., Preskill, J., Steffen, M., Terhal, B.M.: *New J. Phys.* **11**, 013061 (2009)
160. Stace, T.M., Barrett, S.D., Doherty, A.C.: *Phys. Rev. Lett.* **102**, 200501 (2009)
161. Stace, T.M., Barrett, S.D.: *Phys. Rev. A* **81**, 022317 (2010)
162. Ghosh, J., Fowler, A.G., Martinis, J.M., Geller, M.R.: *Phys. Rev. A* **88**, 062329 (2013)
163. Ghosh, J., Fowler, A.G.: *Phys. Rev. A* **91**, 020302 (2015)
164. Ohzeki, M.: *Phys. Rev. A* **85**, 060301 (2012)
165. Bombin, H., Martin-Delgado, M.A.: *Phys. Rev. Lett.* **97**, 180501 (2006)
166. Bombin, H., Martin-Delgado, M.: *Phys. Rev. Lett.* **98**, 160502 (2007)
167. Delfosse, N.: *Phys. Rev. A* **89**, 012317 (2014)
168. Bombin, H., Martin-Delgado, M.: *Phys. Rev. A* **77**, 042322 (2008)
169. Ohzeki, M.: *Phys. Rev. E* **80**, 011141 (2009)

170. Bombin, H., Andrist, R.S., Ohzeki, M., Katzgraber, H.G., Martin-Delgado, M.: *Phys. Rev. X* **2**, 021004 (2012)
171. Andrist, R.S., Katzgraber, H.G., Bombin, H., Martin-Delgado, M.: *New J. Phys.* **13**, 083006 (2011)
172. Kato, T.: *Perturbation Theory for Linear Operators*, vol. 132. Springer, New York (1995)
173. Gu, Z.C., Wen, X.G.: *Phys. Rev. B* **80**, 155131 (2009)
174. Pollmann, F., Turner, A.M., Berg, E., Oshikawa, M.: *Phys. Rev. B* **81**, 064439 (2010)
175. Jordan, P., Wigner, E.P.: *Z. Phys.* **47**, 631 (1928)
176. Kitaev, A.Y.: *Physics-Usppekhi* **44**, 131 (2001)
177. Duclos-Cianci, G., Poulin, D.: *Phys. Rev. A* **87**, 062338 (2013)
178. Duclos-Cianci, G., Poulin, D.: *Quantum Inf. Comput.* **14**, 721 (2014)
179. Kitaev, A.: *Ann. Phys.* **321**, 2 (2006)
180. Nussinov, Z., van den Brink, J.: *Rev. Mod. Phys.* **87**, 1 (2015)
181. Nussinov, Z., Brink, J.v.d.: arXiv preprint [arXiv:1303.5922](https://arxiv.org/abs/1303.5922) (2013)
182. Nishimori, H.: *Statistical Physics of Spin Glasses and Information Processing*, vol. 187. Oxford University Press, Oxford (2001)
183. Ohzeki, M., Jacobsen, J.L.: *J. Phys. A: Math. Theor.* **48**, 095001 (2015)
184. Fujii, K., Nakata, Y., Ohzeki, M., Murao, M.: *Phys. Rev. Lett.* **110**, 120502 (2013). doi:[10.1103/PhysRevLett.110.120502](https://doi.org/10.1103/PhysRevLett.110.120502)
185. Wootton, J.R., Loss, D.: *Phys. Rev. Lett.* **109**, 160503 (2012)
186. Hutter, A., Wootton, J.R., Loss, D.: *Phys. Rev. A* **89**, 022326 (2014)
187. Yao, X.C., Wang, T.X., Chen, H.Z., Gao, W.B., Fowler, A.G., Raussendorf, R., Chen, Z.B., Liu, N.L., Lu, C.Y., Deng, Y.J., et al.: *Nature* **482**, 489 (2012)
188. Nigg, D., Müller, M., Martinez, E., Schindler, P., Hennrich, M., Monz, T., Martin-Delgado, M., Blatt, R.: *Science* **345**, 302 (2014)
189. Breuckmann, N.P., Terhal, B.M.: arXiv preprint [arXiv:1506.04029](https://arxiv.org/abs/1506.04029) (2015)
190. Haah, J.: *Phys. Rev. A* **83**, 042330 (2011)
191. Yoshida, B.: *Phys. Rev. B* **88**, 125122 (2013)
192. Yoshida, B.: *Ann. Phys.* **326**, 15 (2011)
193. Kubica, A., Yoshida, B., Pastawski, F.: arXiv preprint [arXiv:1503.02065](https://arxiv.org/abs/1503.02065) (2015)
194. Nussinov, Z., Ortiz, G.: *Phys. Rev. B* **77**, 064302 (2008)
195. Nussinov, Z., Ortiz, G.: *Ann. Phys.* **324**, 977 (2009)
196. Nussinov, Z., Ortiz, G.: *Proc. Natl. Acad. Sci.* **106**, 16944 (2009)
197. Fujii, K., Negoro, M., Imoto, N., Kitagawa, M.: arXiv preprint [arXiv:1401.6350](https://arxiv.org/abs/1401.6350) (2014)
198. Hamma, A., Castelnovo, C., Chamon, C.: *Phys. Rev. B* **79**, 245122 (2009)
199. Pedrocchi, F.L., Hutter, A., Wootton, J.R., Loss, D.: *Phys. Rev. A* **88**, 062313 (2013)
200. Kapit, E., Chalker, J.T., Simon, S.H.: arXiv preprint [arXiv:1408.0959](https://arxiv.org/abs/1408.0959) (2014)
201. Herold, M., Campbell, E., Eisert, J., Kastoryano, M.: arXiv preprint [arXiv:1406.2338](https://arxiv.org/abs/1406.2338) (2014)
202. Harrington, J.W.: *Analysis of quantum error-correcting codes: symplectic lattice codes and toric codes*. Ph.D. thesis, California Institute of Technology (2004)
203. Landon-Cardinal, O., Yoshida, B., Poulin, D., Preskill, J.: *Phys. Rev. A* **91**, 032303 (2015). doi:[10.1103/PhysRevA.91.032303](https://doi.org/10.1103/PhysRevA.91.032303)
204. Horsman, C., Fowler, A.G., Devitt, S., Van Meter, R.: *New J. Phys.* **14**, 123011 (2012)
205. Bombin, H.: *Phys. Rev. Lett.* **105**, 030403 (2010)
206. Bombin, H.: *New J. Phys.* **13**, 043005 (2011)
207. Bombin, H., Martin-Delgado, M.: *J. Phys. A: Math. Theor.* **42**, 095302 (2009)
208. Gottesman, D.: arXiv preprint [arXiv:quant-ph/9807006](https://arxiv.org/abs/quant-ph/9807006) (1998)
209. Zhou, X., Leung, D.W., Chuang, I.L.: *Phy. Rev. A* **62**, 052316 (2000)
210. Ohno, T., Arakawa, G., Ichinose, I., Matsui, T.: *Nucl. Phys. B* **697**, 462 (2004)
211. O’Gorman, J., Nickerson, N.H., Ross, P., Morton, J.J., Benjamin, S.C.: arXiv preprint [arXiv:1406.5149](https://arxiv.org/abs/1406.5149) (2014)
212. Barends, R., Kelly, J., Megrant, A., Veitia, A., Sank, D., Jeffrey, E., White, T., Mutus, J., Fowler, A., Campbell, B., et al.: *Nature* **508**, 500 (2014)

213. Ristè, D., Poletto, S., Huang, M.Z., Bruno, A., Vesterinen, V., Saira, O.P., DiCarlo, L.: *Nat. Commun.* **6** (2015)
214. Kelly, J., Barends, R., Fowler, A.G., Megrant, A., Jeffrey, E., White, T.C., et al.: arXiv preprint [arXiv:1408.0959](https://arxiv.org/abs/1408.0959) (2014)
215. Nakamura, Y., Pashkin, Y.A., Tsai, J.: *Nature* **398**, 786 (1999)
216. Yamamoto, T., Pashkin, Y.A., Astafiev, O., Nakamura, Y., Tsai, J.S.: *Nature* **425**, 941 (2003)
217. Chiorescu, I., Nakamura, Y., Harmans, C.M., Mooij, J.: *Science* **299**, 1869 (2003)
218. Wallraff, A., Schuster, D.I., Blais, A., Frunzio, L., Huang, R.S., Majer, J., Kumar, S., Girvin, S.M., Schoelkopf, R.J.: *Nature* **431**, 162 (2004)
219. Córcoles, A., Magesan, E., Srinivasan, S.J., Cross, A.W., Steffen, M., Gambetta, J.M., Chow, J.M.: *Nat. Commun.* **6** (2015)
220. Gibney, E.: *Nature* **516**, 24 (2014)
221. Benjamin, S., Kelly, J.: *Nat. Mater.* **14**, 561 (2015)
222. Monroe, C., Raussendorf, R., Ruthven, A., Brown, K., Maunz, P., Duan, L.M., Kim, J.: *Phys. Rev. A* **89**, 022317 (2014)
223. Raussendorf, R., Bravyi, S., Harrington, J.: *Phys. Rev. A* **71**, 062313 (2005)
224. Barrett, S.D., Bartlett, S.D., Doherty, A.C., Jennings, D., Rudolph, T.: *Phys. Rev. A* **80**, 062328 (2009)
225. Verstraete, F., Cirac, J.I.: *Phys. Rev. A* **70**, 060302 (2004)
226. Li, Y., Browne, D.E., Kwek, L.C., Raussendorf, R., Wei, T.C.: *Phys. Rev. Lett.* **107**, 060501 (2011)
227. Fujii, K., Morimae, T.: *Phys. Rev. A* **85**, 010304 (2012)
228. Greenberger, D.M., Horne, M.A., Zeilinger, A.: *Bells Theorem, Quantum Theory and Conceptions of the Universe*, pp. 69–72. Springer, Berlin (1989)
229. Devitt, S.J., Fowler, A.G., Stephens, A.M., Greentree, A.D., Hollenberg, L.C., Munro, W.J., Nemoto, K.: *New J. Phys.* **11**, 083032 (2009)
230. Nemoto, K., Trupke, M., Devitt, S.J., Stephens, A.M., Scharfenberger, B., Buczak, K., Nöbauer, T., Everitt, M.S., Schmiedmayer, J., Munro, W.J.: *Phys. Rev. X* **4**, 031022 (2014)
231. Steane, A.M.: *Phys. Rev. Lett.* **78**, 2252 (1997)
232. DiVincenzo, D.P., Aliferis, P.: *Phys. Rev. Lett.* **98**, 020501 (2007)
233. Plenio, M., Vedral, V., Knight, P.: *Phys. Rev. A* **55**, 4593 (1997)
234. Dür, W., Aschauer, H., Briegel, H.J.: *Phys. Rev. Lett.* **91**, 107903 (2003)
235. Aschauer, H., Dür, W., Briegel, H.J.: *Phys. Rev. A* **71**, 012319 (2005)
236. Dawson, C.M., Haselgrove, H.L., Nielsen, M.A.: *Phys. Rev. Lett.* **96**, 020501 (2006)
237. Dawson, C.M., Haselgrove, H.L., Nielsen, M.A.: *Phys. Rev. A* **73**, 052306 (2006)
238. Steane, A.M.: *Phys. Rev. A* **68**, 042322 (2003)
239. Aliferis, P., Cross, A.W.: *Phys. Rev. Lett.* **98**, 220502 (2007)
240. Steane, A.M.: *Nature* **399**, 124 (1999)
241. Cross, A.W., DiVincenzo, D.P., Terhal, B.M.: *Quantum Inf. Comput.* **9**, 0541 (2009)

Index

A

Abelian anyon, [82](#)
Aharonov-Jones-Landau algorithm, [16](#)

B

Bloch sphere, [2](#)
Braid group, [17](#)
Byproduct operator, [36](#), [37](#)

C

Calderbank-Shor-Steane code (CSS code),
[48](#)

\mathbb{Z}_2 chain complex, [57](#)
Chernoff-Hoeffding bound, [13](#)
Classical linear code, [47](#)
Clifford gate, [3](#)
Clifford operation, [26](#)
Cluster states, [32](#)
Code distance, [46](#)
Completely-positive-trace-preserving
(CPTP), [21](#)
Correlation surface, [96](#)

D

Defect creation, annihilation, and movement,
[88](#)
Defect pair qubit, [87](#)
Deterministic quantum computation with
one clean qubit (DQC1), [14](#)

F

Feedforward, [38](#)
Five-qubit code, [46](#)

G

Gate teleportation, [37](#), [38](#)
Gottesman-Knill theorem, [30](#)
Graph state, [32](#)

H

Hadamard test, [13](#)
Homology class, [58](#)

J

Jones polynomial, [16](#)
Jordan-Wigner transformation, [80](#)

K

Kauffman bracket, [17](#)
Kitaev's compass model, [83](#)
Kitaev's phase estimation algorithm, [15](#)
Kitaev's toric code Hamiltonian, [81](#)
Kraus operators, [22](#)

L

Logical operator, [25](#), [39](#), [45](#)
Logical qubit, [46](#)

M

Magic state, [49](#)
Majorana fermion, [81](#)
Measurement-based quantum computation,
[35](#)
Minimum-weight perfect matching
(MWPM), [66](#)
Mutual duality relation, [76](#)

N

Nishimori line, 75

O

One-bit teleportation, 36

P

Parity check matrix, 47
 Path-model representation, 19
 Pauli-basis measurement, 29
 Pauli group, 24
 Pauli operator, 2
 Pauli product, 24
 Plaquette and star operators, 61
 Positive-operator-valued measure measurement (POVM measurement), 3
 Projected-entangled-pair state (PEPS), 117

Q

Quantum error correction (QEC), 43
 Quantum Fourier transform, 15
 Quantum Gilbert-Varshamov bound, 75
 Qubit, 1
 9-qubit Shor code, 44
 7-qubit Steane code, 48
 Quotient group, 58

R

Random-bond Ising model (RBIM), 71

Reed-Muller code, 51

Reidemeister move, 98

S

Shor's factorization algorithm, 14
 Stabilizer formalism, 24
 Stabilizer generator, 25
 Stabilizer group, 25
 Stabilizer Hamiltonian, 80
 Stabilizer QEC code, 45
 Stabilizer state, 25
 Stabilizer subspace, 25
 Symmetry protected topological order, 80

T

Temperley-Lieb (TL) algebra, 18
 Three-qubit bit flip code, 43
 Three-qubit phase flip code, 44
 Topological calculus, 96
 Topological color code, 78
 Topological diagram, 96
 Topological order, topological ordered system, 79
 Topological stabilizer code, 57
 Φ -transformation, 98
 Transversality, 45
 Trivial cycle, 58

W

Weight, 46