UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Luciano Alves Vieira

Processos de Construção e de Decodificação de Códigos Quânticos Tóricos

Campinas

2022

Luciano Alves Vieira

Processos de Construção e de Decodificação de Códigos Quânticos Tóricos

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica, na Área de Telecomunicações e Telemática.

Orientador: Prof. Dr. Reginaldo Palazzo Júnior

Coorientadora: Profa. Dra. Clarice Dias de Albuquerque

Este trabalho corresponde à versão final da dissertação defendida pelo aluno Luciano Alves Vieira, orientada pelo Prof. Dr. Reginaldo Palazzo Júnior e coorientada pela Profa. Dra. Clarice Dias de Albuquerque.

Campinas

2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

Vieira, Luciano Alves, 1996-
V673p Processos de construção e decodificação de códigos quânticos tóricos / Luciano Alves Vieira. – Campinas, SP : [s.n.], 2022.

Orientador: Reginaldo Palazzo Júnior.
Coorientador: Clarice Dias de Albuquerque.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Computadores quânticos. 2. Correção quântica de erros. 3. Informação quântica. 4. Topologia. I. Palazzo Júnior, Reginaldo, 1951-. II. Albuquerque, Clarice Dias de. III. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Construction and decoding processes of toric quantum codes
Palavras-chave em inglês:
Quantum computers
Quantum error correction
Quantum information
Topology
Área de concentração: Telecomunicações e Telemática
Titulação: Mestre em Engenharia Elétrica
Banca examinadora:
Clarice Dias de Albuquerque [Coorientador]
Max Henrique Machado Costa
Giuliano Gadioli La Guardia
Data de defesa: 28-04-2022
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-8699-5875
- Currículo Lattes do autor: http://lattes.cnpq.br/1705059711508690

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidato(a): Luciano Alves Vieira RA: 264365
Data de defesa: 28 de abril de 2022
Título da Dissertação: "Processos de Construção e de Decodificação de Códigos Quânticos Tóricos"

Profa. Dra. Clarice Dias de Albuquerque (Presidente)
Prof. Dr. Max Henrique Machado Costa
Prof. Dr. Giuliano Gadioli La Guardia

A Ata de Defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

Aos meus pais e à minha irmã.

Agradecimentos

A Deus, por todas as oportunidades que me proporcionou e por colocar no meu caminho todas as pessoas que tornaram tudo até agora possível.

Ao meu orientador, Reginaldo Palazzo Júnior, um exemplo por sua educação, conhecimento e humildade. Agradeço a oportunidade de poder ser seu orientando durante estes anos, por ter acreditado em mim e sempre me incentivado.

À minha coorientadora, Clarice Dias de Albuquerque, uma pessoa altamente humilde e generosa, por ter me apresentado o mundo dos "códigos", por todas as discussões que tanto contribuíram para este trabalho e, acima de tudo, por sempre me incentivar e fazer o possível para ajudar.

Aos meus pais, Raimundo e Meyre Vieira, pelo amor, apoio e cuidado que sempre tiveram comigo. Nunca irei esquecer de todas as renúncias que fizeram pelos filhos e como sempre colocaram nossa educação em primeiro lugar.

À minha irmã, Leylianne Alves, pelo incentivo e ajuda em todos os momentos. A sua coragem em desbravar e enfrentar novos desafios sempre foi um exemplo.

À Larissa Lima, por me fazer rir nos momentos de angústia e continuar trabalhando nos de desânimo. Por estar ao meu lado e me ajudar a lidar com todas as mudanças e desafios.

À professora Maria Silvana Alcântara Costa, minha primeira orientadora, durante a graduação na UFCA, que me ajudou a acreditar que seria possível mudar de área e hoje estar aqui.

À Larissa Guimarães, por sempre saber o quê, quando e como falar o que preciso ouvir e com quem sei que sempre poderei contar. Uma amizade que aguentou a distância, o tempo e os caminhos distintos tomados.

A Felipe Duarte, pelos mais de 10 anos de amizade e por todas as conversas, momentos de descontração e diversão.

À Julia Dias, pela amizade que tornou mais fácil a mudança de cidade, e de vida, que veio com o mestrado. Guardo as boas lembranças de todos os cafezinhos e lanches.

Aos membros da banca por aceitarem o convite, pela atenção dedicada a este trabalho e pelas sugestões e considerações.

A todos que, direta ou indiretamente, contribuíram para a conclusão deste trabalho.

O presente trabalho foi realizado com apoio do CNPq, Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil

“Há muitos grãos de incenso sobre
o mesmo altar: um caiu mais cedo,
outro mais tarde. Não faz diferença”

Marco Aurélio

Resumo

Neste trabalho, apresentamos um estudo sobre construção e decodificação do código tórico, primeiro código quântico topológico proposto. Inicialmente, compilamos os conceitos preliminares necessários à compreensão do tema. Para a apresentação do código tórico, estudamos sua estrutura e identificamos seus parâmetros por meio de duas abordagens, a primeira fundamentada principalmente na teoria de homologia e co-homologia, e a segunda visando entender o comportamento dos anyons dentro do sistema. Por fim, o problema da decodificação é introduzido para dois modelos de erros: um que considera a síndrome fidedigna e outro que prevê a existência de erros nas medições dos geradores do código. Analisamos as particularidades que surgem na decodificação com cada um dos casos. Ao longo do trabalho, buscamos utilizar exemplos que reforcem os conceitos apresentados.

Palavras-chaves: código tórico; computação quântica; códigos quânticos corretores de erros; códigos quânticos topológicos.

Abstract

In this work, we present a study on the construction and decoding of the toric code, the first proposed topological quantum code. Initially, we compiled the preliminary concepts necessary to understand the topic. For the presentation of the toric code, we studied its functioning and identified its parameters through two approaches, the first based mainly on the theory of homology and co-homology, and the second aiming to understand the behavior of anyons within the system. Finally, the decoding problem is introduced for two error models, one that considers the syndrome to be reliable and the other that predicts the existence of errors in the measurements of the code generators. We analyze the particularities that arise in the decoding with each case. Throughout the work, we seek to use examples that reinforce the concepts presented.

Keywords: toric code; quantum computing; quantum error correction codes; topological quantum codes.

Lista de ilustrações

Lista de tabelas

Lista de acrônimos

CQCE	códigos quânticos corretores de erros
CSS	código quântico Calderbank-Shor-Steane
CE	caminhada evasiva
PA	polígono auto-evitável

Lista de símbolos

$\mathbb{C}^n$	espaço vetorial complexo de dimensão n
$\mathcal{H}^n$	espaço de Hilbert de dimensão n
$\|v\rangle$	vetor (ket)
$\langle v\|$	vetor dual (bra)
$(\cdot,\cdot)$ e $\langle\cdot\|\cdot\rangle$	produto interno
$\|\|\|v\rangle\|\|$	norma do estado $\|v\rangle$
$A^\dagger$	operador adjunto ou conjugado hermitiano do operador A
$\otimes$	produto tensorial
δ_{ij}	delta de Kronecker
$[\cdot,\cdot]$	comutador
$\{\cdot,\cdot\}$	anticomutador
ρ	operador de densidade
$\mathcal{G}_n$	grupo de Pauli de ordem n
$S \in \mathcal{G}_n$	estabilizador
$\mathcal{C}_S$	código estabilizador associado a S
$f(E_a)$	síndrome do erro E_a
$C(S)$	centralizador de S
$N(S)$	normalizador de S
$\mathbf{C}_p$	conjunto das p-cadeias
∂_p	operador bordo de $\mathbf{C}_p$
$\mathbf{Z}_p$	núcleo do homomorfismo ∂_p
$\mathbf{B}_p$	imagem do homomorfismo ∂_{p+1}
$\mathbf{H}_p$	grupo de homologia de dimensão p
$\mathbf{C}^p$	conjunto das p-cocadeias
δ_p	operador cobordo de $\mathbf{C}^p$
$\mathbf{Z}^p$	núcleo do homomorfismo ∂_p
$\mathbf{B}^p$	imagem do homomorfismo δ_{p-1}
$\mathbf{H}^p$	grupo de cohomologia de dimensão p
$\{p,q\}$	tesselação regular com q polígonos de p lados se encontrando em cada vértice
g	gênero de uma superfície
χ	característica de Euler
X_c	portas X atuando sobre os qubits da cadeia $c \in C_1$
Z_c	portas Z atuando sobre os qubits da cadeia $c \in C_1$
X_{c^*}	portas X atuando sobre os qubits da cocadeia $c^* \in C^1$

Z_{c^*}	portas Z atuando sobre os qubits da cocadeia $c^* \in C^1$
$A(v)$	operador vértice
$B(f)$	operador face
$\|\xi\rangle$	estado fundamental do código tórico
e, m e ϵ	tipos de *anyons* do código tórico
$\bar{X}$	operador lógico X
$\bar{Z}$	operador lógico Z
L	número de arestas em um lado de um reticulado quadrado do toro
n	número de qubits físicos de um código quântico
k	número de qubits lógicos de um código quântico
d	distância de um código quântico
p	taxa de erros por qubit
q	taxa de erros na medição da síndrome
p_c	limite de precisão
Sdr	cadeia da síndrome

Sumário

1 Introdução

Os computadores estão tão inseridos na sociedade que é difícil imaginar nossas vidas sem eles. A capacidade atual de processamento dessas ferramentas permite que sejam realizadas milhões de operações em curtos espaços de tempo e com bastante precisão. Com os recursos de software e de hardware atuais, eles superaram os humanos em muitas tarefas. Mas nem sempre isso foi encarado com tanta naturalidade.

Em 1997, Kasparov, campeão mundial de xadrez da época, perdeu um match de xadrez contra o Deep Blue [1], um supercomputador construído pela IBM especificamente pare esse fim. Hoje, não são necessários supercomputadores: os comuns são capazes de derrotar os maiores jogadores do mundo sem muita dificuldade. Mesmo tarefas em que se imaginava ser necessário um fator humano para realização, hoje podem ser realizadas por computadores, tais como a criação de pinturas, sinfonias ou textos. Isso tudo sem considerar os atuais supercomputadores, capazes de trabalhar com bancos de dados com terabytes de tamanho e realizar previsões climáticas sofisticadas, análises e simulações do funcionamento do corpo humano, ou mesmo investigar o passado e o futuro do universo. Contadas essas façanhas, parece impossível que haja tarefas além da capacidade dessas máquinas. Mas há.

Um problema particularmente difícil para computadores clássicos é a simulação de sistemas quânticos com várias partículas, pois, em geral, os recursos computacionais, tempo e memória, crescem exponencialmente com o tamanho do sistema. Devido a essa dificuldade, em 1982 Feynman [2] apresentou a primeira proposta de um computador quântico, ou seja, um computador baseado nas propriedades da mecânica quântica. A possibilidade de construção de uma máquina de Turing quântica foi sugerida por Deutsch em 1985 [3], sendo dele a atual concepção de computador quântico.

Desde então, muitos pesquisadores têm se debruçado sobre o tema da computação quântica e sobre os possíveis ganhos dessa nova tecnologia, de forma que hoje já são conhecidos vários algoritmos quânticos que desempenham funções de forma superior aos clássicos. Os dois mais famosos são o algoritmo para fatoração de Shor [4] e o algoritmo para busca em bancos de dados desordenados de Grover [5], (para um vislumbre da quantidade de algoritmos quânticos propostos atualmente, [6] pode ser consultado). Dessa forma, espera-se que o computador quântico surja como uma tecnologia disruptiva, capaz de superar os computadores clássicos em determinadas atividades e funções.

Nos últimos anos, temos presenciado uma verdadeira corrida em busca da construção de computadores quânticos funcionais, capazes de lidar com uma quantidade de informação suficiente para serem utilizados em problemas práticos. Dentre os “competidores”

estão gigantes da tecnologia, como Google [7], IBM [8], Intel [9], Microsoft [10], e até mesmo empresas de ramos distintos, como o Grupo Alibaba [11]. Como amostra dessa competição, em 2019 a Google proclamou ter alcançado a supremacia quântica[1] com o uso de seu processador de 53 qubits, o Sycamore. Segundo Arute *et al.* [12], o Sycamore foi capaz de, em 200 segundos, obter um resultado que um supercomputador clássico atual levaria 10 mil anos.

O resultado, entretanto, foi controverso. No mesmo ano, Pednault *et al.*, pesquisadores da IBM, questionaram o resultado em um preprint [13], afirmando que um computador clássico poderia resolver o problema, na verdade, em 2.5 dias, utilizando outra técnica. Mesmo assim, o resultado obtido por Arute *et al.* é, sem dúvida, um marco significativo e justifica a atenção que vem sendo dada a esse novo tipo de tecnologia.

Há projeções otimistas que apontam a construção desses dispositivos nos próximos anos, como as realizadas pela IBM [14]. Após construir um processador com 100 qubits [15], a companhia espera que em 2023 consiga ter em mãos um computador quântico com mais de 1000 qubits, para então alcançar o objetivo de ter um processador com mais de 1 milhão de qubits a partir de 2024.

Apesar da possibilidade fascinante de se utilizar as propriedades da mecânica quântica para conseguir realizar o processamento da informação de forma mais rápida, os computadores quânticos são muito frágeis [16], e é por isso mesmo que dispositivos com poucos qubits têm sido comemorados. O próprio ambiente em que eles estão inseridos é uma possível fonte de erros, em função do fenômeno quântico da decoerência. Por isso, os pesquisadores da computação quântica têm atuado em duas principais missões. A primeira é a busca por novos algoritmos. A segunda, conseguir realizar uma computação quântica que seja tolerante a falhas [16].

Os estudos de Códigos Quânticos Corretores de Erros (CQCE) e de computação quântica tolerante a falhas buscam desenvolver ferramentas que possibilitem a manipulação e o armazenamento da informação quântica de forma confiável, o que permitiria a computação. Tal como os códigos corretores de erros clássicos, os códigos quânticos buscam proteger a informação contra o efeito de erros. No entanto, existem algumas diferenças importantes entre a informação quântica e a informação clássica que precisam ser levadas em conta para que a correção de erros seja possível: 1) é impossível clonar informação quântica; 2) os erros são contínuos; e 3) as observações geralmente destroem a informação na mecânica quântica [17]. Ainda assim, é possível contornar essas diferenças e realizar a correção de erros em sistemas quânticos.

O primeiro código quântico corretor de erros foi proposto por Shor em 1995

[1] A supremacia quântica é alcançada quando uma tarefa computacional, que não pode ser realizada por qualquer algoritmo clássico conhecido em um supercomputador clássico existente e em um período de tempo razoável, é realizada em um computador quântico existente.

[18], sendo um análogo ao código de repetição clássico que ficou conhecido na literatura como *código de Shor*. Ele codifica um qubit de informação em nove qubits e é capaz de corrigir um erro arbitrário que ocorra em um qubit. Paralelamente a Shor, Steane também apresentou um código quântico, que codifica um qubit em sete e possui distância mínima três [19].

Mais tarde, esses códigos foram aprimorados independentemente por Calderbank e Shor [20] e Steane [21], originando a classe *códigos CSS*. Embora essa primeira construção dos códigos CSS tenha sido feita para alfabetos binários, Klappenecker *et al.* [22] generalizaram tais códigos de forma natural, mas não trivial, para alfabetos q-ários, em que q é potência de um número primo. O código de Steane, um código CSS construído a partir do código de Hamming $(7, 4, 3)$, codifica um qubit em sete e é capaz de corrigir erros em um único qubit.

Em 1996, Gottesman [23] propôs uma das classes de códigos quânticos mais abrangentes, conhecida por *códigos estabilizadores*. Tal classe de códigos engloba, por exemplo, os CSS, e tem sua base fundamentada na teoria de grupos. Um código dessa classe é um subespaço fixado por um subgrupo abeliano do grupo de Pauli. Entre as vantagens do formalismo estabilizador estão a possibilidade de uma representação mais compacta e a maior facilidade na identificação dos erros corrigíveis pelo código.

Kitaev apresentou, em 1997, o *código tórico* [24], que veio a ser o primeiro código de uma nova subclasse dos códigos estabilizadores, os *códigos quânticos topológicos*. Com o código tórico, Kitaev encontrou uma forma de realizar computação quântica tolerante a falhas, de modo que a informação é protegida fisicamente durante o processo de computação.

O objetivo principal deste trabalho é o estudo do processo de codificação e dos princípios da decodificação dos códigos quânticos tóricos. A escolha desse código é justificada pois, como primeira proposta dentro da classe de códigos topológicos, a sua compreensão é fundamental para o entendimento das demais classes de códigos e, consequentemente, para a construção de novas. Além disso, o código tórico tem contado com o interesse da Google, que recentemente conseguiu preparar um estado fundamental do hamiltoniano daquele código utilizando seu processador quântico Sycamore [25].

A dissertação está organizada como segue.

No Capítulo 2, serão apresentados os conceitos fundamentais ao entendimento do código tórico, o que passa por tópicos básicos das teorias da mecânica quântica e da informação quântica e pelas teorias matemáticas de reticulados, tesselações e superfícies.

Já o Capítulo 3 conecta os assuntos expostos no capítulo anterior para definir formalmente o código tórico. A fim de entender o seu funcionamento, recorremos a dois pontos de vista, um mais matemático, fundamentado principalmente na teoria de homologia,

e outro mais físico, que considera a existência e ação dos *anyons* no sistema. Deixamos, ainda, um gancho para o processo de decodificação, abordado no próximo capítulo.

O Capítulo 4 introduz a decodificação em códigos tóricos. Para isso, introduzimos primeiramente o problema da decodificação e os elementos que a compõem. Serão dados destaque às particularidades do processo de decodificação que surgem quando é considerada a existência de erros nas medições da síndrome. Além disso, ao final deste capítulo, e do anterior, apresentamos a construção de exemplos que permitem ilustrar o que foi apontado na teoria e facilitar o entendimento para o leitor.

Por fim, o Capítulo 5 reúne as considerações finais sobre o tema após toda a reflexão teórica apresentada nesta dissertação.

2 Conceitos preliminares

Neste capítulo serão apresentados alguns conceitos preliminares para o entendimento do código tórico. Em primeiro lugar, serão introduzidas a unidade básica da informação quântica, o qubit, bem como duas propriedades da mecânica quântica fundamentais ao desempenho dos computadores quânticos: a superposição e o emaranhamento. Em seguida, será realizada uma breve revisão de alguns tópicos da álgebra linear sob o aspecto da notação de Dirac, padrão da mecânica quântica e introduzidos os códigos quânticos corretores de erros, com ênfase nos códigos estabilizadores. Por fim, são introduzidos alguns conceitos referentes a teoria de homologia, reticulados, tesselações e superfícies.

Os tópicos aqui apresentados possuem uma extensa teoria desenvolvida; aqui destacamos objetivamente a parte de interesse para a pesquisa de cada um desses tópicos. Pretende-se, assim, permitir ao leitor um ponto de partida para a compreensão dos Capítulos 3 e 4, que tratarão, especificamente, do código tórico.

Enfatizamos que os resultados e definições apresentados nas Seções 2.1 a 2.4 podem ser consultados em [17], enquanto as Seções 2.5 e 2.6 são um apanhado dos resultados de [17, 26] e [23]. Por fim, a Seções 2.7 e 2.8 são baseadas fundamentalmente nas referências [27, 28, 29].

2.1 O bit quântico e as propriedades quânticas de superposição e emaranhamento

Para a computação clássica, a unidade fundamental da informação é o bit, capaz de assumir dois estados distintos, 0 ou 1. Na computação quântica existe uma unidade análoga, o *bit quântico* ou *qubit*. Os qubits são objetos físicos como, por exemplo, os elétrons, em que um spin-up representa o estado $|1\rangle$ e um spin-down representa o estado $|0\rangle$. Porém, tratar os qubits como objetos matemáticos com certas propriedades específicas possui a vantagem de se poder estudar a teoria da computação e da informação quântica sem a preocupação acerca de qual sistema quântico específico está sendo utilizado [17].

A diferença entre bits e qubits é que os últimos também podem assumir estados diferentes de $|0\rangle$ e $|1\rangle$. Mais especificamente, um qubit pode ser qualquer combinação linear dos estados $|0\rangle$ e $|1\rangle$.

A notação "$|\cdot\rangle$" é conhecida como *ket* e faz parte da *notação de Dirac*, padrão da mecânica quântica. O ket é usado para representar estados quânticos.

Definição 1. *Um estado de um qubit pode ser visto como um vetor unitário (vetor de estado) em um espaço vetorial complexo* $\mathbb{C}^2$.

Antes de prosseguir, deixamos a seguir algumas definições que ajudam a descrever com mais detalhes o espaço $\mathbb{C}^2$.

Definição 2. *Um espaço vetorial com produto interno* V *é completo quando toda sequência de Cauchy em* V *é convergente, sendo a convergência em relação ao produto interno de* V.

Definição 3. *Uma sequência* x_n *em um espaço vetorial com produto interno* V *é uma sequência de Cauchy se, para todo* $\varepsilon > 0$ *dado, existem* $n_0 \in \mathbb{N}$ *tal que* $m, n > n_0 \Rightarrow d(x_m, x_n) < \varepsilon$, *em que* $d(\cdot, \cdot)$ *é uma função* $V \times V \to \mathbb{R}$ *denominada distância, satisfazendo algumas condições que podem ser consultadas em [30].*

Dados $v, w \in V$, podemos, por exemplo, tomar $d(v, w) = |v - w|$, em que $|\cdot|$ é a norma induzida pelo produto interno $(\cdot, \cdot)$ de V, de forma que, para $v \in V$, $|v| = \sqrt{(v, v)}$. O leitor que desejar se aprofundar nessas e em outras definições correlatas, encontrará em [30] uma referência adequada.

Por fim, definimos um *espaço de Hilbert*.

Definição 4. *Dizemos que um espaço vetorial com produto interno* V *é um espaço de Hilbert se* V *é um espaço vetorial com produto interno, completo em relação a esse produto interno.*

Para espaços vetoriais complexos de dimensões finitas, um *espaço de Hilbert* é equivalente a um espaço vetorial com produto interno [17]. Logo, o espaço vetorial complexo $\mathbb{C}^2$ e o espaço de Hilbert $\mathcal{H}^2$ são equivalentes.

Assim, um estado arbitrário de um qubit pode ser expresso por

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

em que α e $\beta \in \mathbb{C}$ são as *amplitudes* associadas aos estados $|0\rangle$ e $|1\rangle$, respectivamente. As amplitudes α e β devem satisfazer à condição

$$|\alpha|^2 + |\beta|^2 = 1. \tag{2.2}$$

A Equação (2.2) surge porque o quadrado da amplitude associada a um estado é igual à probabilidade de se encontrar o qubit no respectivo estado quando for feita uma medição do mesmo, e, como sabemos, a soma de todas as probabilidades de um determinado universo de eventos deve ser igual a 1. Assim, quando o qubit no estado arbitrário da Equação (2.1) é medido, o estado $|0\rangle$ é obtido com probabilidade $|\alpha|^2$ e o estado $|1\rangle$ é obtido

com probabilidade $|\beta|^2$. Quando é feita uma observação (ou medição) de um estado em superposição, ele colapsará para um dos estados possíveis com a probabilidade associada.

Dessa forma, enquanto um bit clássico é similar a uma moeda honesta, pois só consegue assumir dois estados (0 ou 1), um qubit consegue existir em um *continuum* de estados entre $|0\rangle$ e $|1\rangle$, as chamadas superposições. Um exemplo de um possível estado em superposição de um qubit é o estado $|+\rangle$:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

Com isso, a realização de uma medição sobre $|+\rangle$ resultaria no estado $|0\rangle$ com probabilidade $\left|1/\sqrt{2}\right|^2 = 1/2$ e no estado $|1\rangle$ com a mesma probabilidade. É importante ressaltar que a medição do qubit colapsa a superposição de estados, de forma que, após a medição, o estado do qubit passa a ser o resultado apresentado pela própria medição. Assim, para determinarmos as amplitudes α e β do qubit da Equação (2.1), seria necessário preparar infinitos qubits de forma idêntica e medi-los. Apesar disso, a propriedade de superposição pode ser usada para obter ganho computacional. A título de exemplo, o algoritmo de Grover [5] faz uso da propriedade de superposição para conseguir realizar uma busca em um banco de dados desordenado de forma mais eficiente que os algoritmos clássicos.

Quando um sistema possui n qubits, um estado geral $|\psi\rangle$ é uma superposição dos 2^n estados $|00\ldots0\rangle, |00\ldots1\rangle, \ldots |11\ldots1\rangle$. Isso ficará mais claro ao discutirmos os postulados da mecânica quântica, na Seção 2.3. De toda forma, podemos associar a sequência dentro dos kets à representação binária dos números $0, 1, \ldots, 2^n - 1$. Assim, o estado arbitrário $|\psi\rangle$ pode ser representado por

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle ,$$

sujeito à condição

$$\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1.$$

Outra propriedade quântica importante é o *emaranhamento*. Partículas emaranhadas apresentam forte correlação entre suas propriedades. Tal correlação faz com que as características de um estado emaranhado só possam ser bem definidas como um todo, ou seja, suas características não são armazenadas localmente em cada partícula. Dessa forma, uma medição realizada sobre uma dessas partículas pode afetar todas as demais que formam o estado. A criptografia quântica faz uso dessa propriedade para conseguir transmitir mensagens com segurança absoluta.

2.2 Tópicos de álgebra linear

Nesta subseção, revisaremos alguns conceitos básicos da álgebra linear sob a ótica da mecânica quântica, fazendo uso da notação de Dirac. A princípio, como um qubit pode ser interpretado como um vetor em $\mathbb{C}^2$, então podemos fazer uso de uma representação matricial. Assim, o estado arbitrário de um qubit $|\psi\rangle$, da Equação (2.1), na representação matricial se torna

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Portanto, a representação matricial dos estados $|0\rangle$ e $|1\rangle$ é

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{e} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Essa representação é conveniente para o entendimento da manipulação e da evolução dos estados quânticos. Contudo, com o aumento do número de qubits, ela se torna impraticável.

Os estados $|0\rangle$ e $|1\rangle$ formam uma base ortonormal para o espaço vetorial $\mathbb{C}^2$, conhecida como *base computacional*. Outra base bastante utilizada é a *base conjugada*, formada pelos estados

$$|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}; \quad |-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}.$$

Definiremos agora os *operadores lineares*, responsáveis pela modificação dos estados quânticos e por realizarem a computação.

Definição 5. *Uma transformação linear entre os espaços V e W é qualquer aplicação $A : V \to W$ que seja linear em sua ação:*

$$A\left(\sum_i a_i |v_i\rangle\right) = \sum_i a_i \left(A |v_i\rangle\right).$$

Usualmente, utiliza-se $A|v\rangle$ para denotar $A(|v\rangle)$. Quando uma transformação linear A é uma transformação de V em V dizemos que A é um operador linear. Para um espaço vetorial V, o operador identidade I_V satisfaz $I_V |v\rangle = |v\rangle$, para todo $|v\rangle \in V$. Quando não houver necessidade de distinção, o operador identidade será escrito apenas como I.

Consideremos os espaços vetoriais V, W e X e os operadores lineares $A : V \to W$ e $B : W \to U$. A *composição de B com A* é um operador $BA : V \to U$, dado por $(BA)\,|v\rangle = B(A\,|v\rangle) = BA\,|v\rangle$.

Da mesma forma como podemos representar os estados por meio de *vetores coluna*, os operadores lineares também possuem uma representação matricial. Consideremos o operador linear $A : V \to W$ e os conjuntos $\{|v_1\rangle, |v_2\rangle, \ldots, |v_m\rangle\}$ e $\{|w_1\rangle, |w_2\rangle, \ldots, |w_n\rangle\}$ sejam bases para os espaços V e W, respectivamente. Então, para cada $j \in \{1, 2, \ldots, m\}$ existem números complexos $a_{1j}, a_{2j}, \ldots, a_{nj}$ tais que $A\,|v_j\rangle = \sum_i a_{ij}\,|w_i\rangle$. A matriz correspondente ao operador A é aquela cujas entradas são os números complexos a_{ij}.

Entre os operadores de um qubit se destacam as *matrizes de Pauli*:

$$I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X \equiv \sigma_X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Y \equiv \sigma_Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad \text{e} \quad Z \equiv \sigma_Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Vejamos o efeito de cada uma dessas matrizes sobre um qubit em um estado arbitrário $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, recorrendo para isso, à representação matricial de $|\psi\rangle$.

A aplicação de I sobre o estado de um qubit não o altera,

$$I\,|\psi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = |\psi\rangle\,.$$

Por sua vez, o operador X tem como efeito a troca entre as amplitudes de $|0\rangle$ e $|1\rangle$:

$$X\,|\psi\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \beta \\ \alpha \end{bmatrix} = \beta\,|0\rangle + \alpha\,|1\rangle\,.$$

Ou seja, X age de modo similar à porta NOT da computação clássica.

Para o operador Y, temos

$$Y\,|\psi\rangle = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} -\beta i \\ \alpha i \end{bmatrix} = -\beta i\,|0\rangle + \alpha i\,|1\rangle\,.$$

Por fim, o operador Z atua da seguinte forma:

$$Z\,|\psi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ -\beta \end{bmatrix} = \alpha\,|0\rangle - \beta\,|1\rangle\,.$$

Sendo assim, o operador Z insere um fator de fase entre os estados $|0\rangle$ e $|1\rangle$, multiplicando por -1 a amplitude de $|1\rangle$.

As matrizes de Pauli possuem algumas propriedades algébricas notáveis. A primeira delas é que o quadrado de cada um desses operadores é igual ao operador identidade: $I^2 = X^2 = Y^2 = Z^2 = I$. A segunda é que, com exceção de I, esses operadores anticomutam entre si. Por exemplo,

$$XZ = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = - \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = - \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = -ZX.$$

Uma das razões que levam à avaliação das matrizes de Pauli como significativamente importantes é que as mesmas formam uma base para o espaço vetorial das operações sobre um qubit [31]. Além disso, veremos à frente que, a partir das matrizes de Pauli, é possível construir o chamado grupo de Pauli, que se destaca, principalmente, para a classe de códigos estabilizadores.

A porta de Hadamard H, apresentada na Equação (2.3), é outra porta quântica de um qubit com boa aplicabilidade.

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.3}$$

Sua ação promove a mudança da base computacional $\{|0\rangle, |1\rangle\}$ para a base conjugada $\{|+\rangle, |-\rangle\}$:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = |+\rangle,$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = |-\rangle,$$

$$H|+\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle,$$

$$H|-\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle.$$

O uso da porta H permite a criação de superposições e de estados emaranhados, como os *estados de Bell*[1].

O *dual* de um vetor $|v\rangle \in \mathbb{C}^n$, denotado por $\langle v|$, é o conjugado transposto de $|v\rangle$. A operação de conjugação transposta é representada por †, de forma que

$$\langle v| = (|v\rangle^*)^T = (|v\rangle)^\dagger,$$

em que $|v\rangle^*$ representa a operação de conjugação complexa dos elementos do vetor coluna $|v\rangle$, enquanto $|v\rangle^T$ representa a transposição do vetor coluna $|v\rangle$. A notação "$\langle\cdot|$" é chamada *bra*.

Apresentaremos a seguir um produto interno para um espaço vetorial complexo com base no vetor dual $\langle v|$.

Lembramos que para uma função do tipo $(\cdot,\cdot) : V \times V \to \mathbb{C}$ ser um produto interno, ela deve satisfazer as seguintes condições:

1. $(\cdot,\cdot)$ é linear em relação ao segundo argumento:

$$(|v\rangle, \sum_i \lambda_i |w_i\rangle) = \sum_i \lambda_i (|v\rangle, |w_i\rangle).$$

2. $(|v\rangle, |w\rangle) = (|w\rangle, |v\rangle)^*$

3. $(|v\rangle, |v\rangle) \geq 0$, com a igualdade valendo se, e somente se, $|v\rangle = 0$.

Assim, a função $(|v\rangle, |w\rangle) = (|v\rangle)^\dagger |w\rangle$ é um produto interno para o espaço vetorial complexo $\mathbb{C}^n$.

Na notação de Dirac, o produto interno $(|v\rangle, |w\rangle)$ se torna $\langle v|w\rangle$.

Seguem algumas definições simples:

Definição 6. *Dois estados $|v\rangle$ e $|w\rangle$ são ortogonais se $\langle v|w\rangle = \langle w|v\rangle = 0$.*

Definição 7. *A norma de um estado $|v\rangle$ é definida por $|||v\rangle|| = \sqrt{\langle v|v\rangle}$.*

Definição 8. *Um vetor é dito unitário se $|||v\rangle|| = 1$.*

Definição 9. *Um conjunto de vetores $|i\rangle$ é dito ortonormal se cada vetor for unitário e vetores distintos do conjunto forem dois a dois ortogonais, ou seja, $\langle i|j\rangle = 0$, se $i \neq j$.*

[1] Os estados de Bell ou pares EPR (iniciais de Einstein, Podolsky e Rose) são: $|\phi_{00}\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, |\phi_{01}\rangle = \frac{|01\rangle + |10\rangle}{\sqrt{2}}, |\phi_{10}\rangle = \frac{|00\rangle - |11\rangle}{\sqrt{2}}$ e $|\phi_{11}\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$.

Outra notação para representar que um conjunto de vetores $|i\rangle$ é ortonormal é $\langle i|j\rangle = \delta_{ij}$, em que δ_{ij} é o delta de Kronecker, que assume valor 1 quando i e j são iguais e 0 caso contrário.

O produto interno também possui uma representação matricial. Seja V um espaço vetorial, $|v\rangle, |w\rangle \in V$ tais que $|v\rangle = \sum_i v_i |i\rangle$ e $|w\rangle = w_i \sum_i |i\rangle$, em que os vetores $|i\rangle$ são uma base ortonormal para V. A representação matricial para o produto interno é, então,

$$\langle v|w\rangle = \begin{bmatrix} v_1^* & \dots & v_n^* \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix},$$

onde v_i^* representa o complexo conjugado de $v_i \in \mathbb{C}$.

O uso do produto interno permite ainda uma nova representação para os operadores lineares, chamada *representação de produto externo*.

Definição 10. *Sejam $|v\rangle$ e $|w\rangle$ vetores dos espaços vetoriais V e W, respectivamente, com ambos os espaços com produto interno. Define-se o produto externo $|w\rangle\langle v|$ como a transformação linear de V para W, para o qual $(|w\rangle\langle v|)|v'\rangle = |w\rangle\langle v|v'\rangle = \langle v|v'\rangle|w\rangle$.*

Outros conceitos essenciais são os de *autovetores* e *autovalores*.

Definição 11. *Um autovetor de um operador linear A é um vetor não-nulo $|v\rangle$, tal que $A|v\rangle = \lambda|v\rangle$, em que $\lambda \in \mathbb{C}$ é o autovalor associado ao autovetor $|v\rangle$.*

Todo operador linear A possui pelo menos um autovalor e um correspondente autovetor. O *autoespaço* correspondente a um autovalor λ é o conjunto dos vetores que possui autovalor λ, acrescido do vetor nulo. Um autoespaço é um subespaço do espaço vetorial sobre o qual o operador linear age.

Definição 12. *Uma representação diagonal de um operador A definido no espaço vetorial V é uma representação da forma $A = \sum_i \lambda_i |i\rangle\langle i|$, em que os vetores $|i\rangle$ formam um conjunto de autovetores ortogonais com autovalores associados λ_i. Se um operador possui uma representação diagonal, este é dito diagonalizável.*

Veremos que um tipo específico de operador linear com aplicação recorrente na computação quântica e na teoria de códigos quânticos é o *operador adjunto* ou *conjugado hermitiano*, definido a seguir.

Definição 13. *Seja A um operador linear em um espaço de Hilbert V. Definimos o operador adjunto ou conjugado hermitiano do operador A como o único operador linear $A^\dagger \in V$, tal que para todos os vetores $|v\rangle, |w\rangle \in V$, satisfaça*

$$(|v\rangle, A|w\rangle) = (A^\dagger|v\rangle, |w\rangle).$$

Segue da definição que $(AB)^\dagger = B^\dagger A^\dagger$. Por convenção, o adjunto de um vetor $|v\rangle$ é o seu dual, $|v\rangle^\dagger = \langle v|$. Logo, $(A|v\rangle)^\dagger = \langle v|A^\dagger$. Além disso, $(A^\dagger)^\dagger = A$.

Na representação matricial, a ação da conjugação hermitiana sobre um operador A é levar a matriz de A na sua complexa conjugada, $A^\dagger = (A^*)^T$. Em outras palavras, para a obtenção de $A^\dagger$ deve-se fazer a transposta de A e substituir seus elementos pelos respectivos complexos conjugados.

Definição 14. *Se um operador A é tal que seu adjunto é ele mesmo, $A^\dagger = A$, dizemos que ele é Hermitiano ou auto-adjunto.*

Dentre os operadores Hermitianos, destacamos a classe dos *projetores*.

Definição 15. *Seja W um subespaço vetorial de dimensão k do espaço vetorial V de dimensão d. Usando o processo de Gram-Schimidt é possível construir uma base ortonormal $\{|1\rangle, |2\rangle, \ldots, |d\rangle\}$ para o espaço V, de forma que $\{|1\rangle, |2\rangle, \ldots, |k\rangle\}$ é uma base ortonormal para o subespaço W. Definimos o projetor sobre W pelo operador*

$$P \equiv \sum_{i=1}^{k} |i\rangle\langle i|.$$

Da definição de operador adjunto segue que $|v\rangle\langle v|$ é Hermitiano para qualquer vetor $|v\rangle$, e, portanto, P é Hermitiano.

Definição 16. *Um operador A é dito normal se $AA^\dagger = A^\dagger A$.*

Logo, todo operador Hermitiano é normal.

Definição 17. *Se um operador A satisfaz $AA^\dagger = A^\dagger A = I$ (ou seja, $A^{-1} = A^\dagger$) ele é chamado unitário.*

Os operadores unitários têm a propriedade de preservar o produto interno, ou seja, $(U|v\rangle, U|w\rangle) = \langle v|U^\dagger U|w\rangle = \langle v|w\rangle$.

Para entendermos o funcionamento de sistemas com mais de uma partícula, é necessária a introdução do conceito de *produto tensorial*. Considere dois espaços de Hilbert, V e W, com dimensões m e n, respectivamente. O produto tensorial $V \otimes W$ é um espaço vetorial de dimensão mn. Os elementos desse espaço são combinações lineares dos produtos tensoriais $|v\rangle \otimes |w\rangle$, em que $|v\rangle \in V$ e $|w\rangle \in W$.

Se $|i\rangle$ for uma base ortonormal para V e $|j\rangle$ for uma base ortonormal para o espaço W, então o produto tensorial dos vetores dessas bases é uma base para $V \otimes W$.

Outras notações que podem ser utilizadas para o produto tensorial $|v\rangle \otimes |w\rangle$ são $|v\rangle |w\rangle$ e $|vw\rangle$.

Considere os espaços vetoriais V e W, em que $|v\rangle, |v_1\rangle, |v_2\rangle \in V$ e $|w\rangle, |w_1\rangle, |w_2\rangle \in W$ são todos estados arbitrários e $z \in \mathbb{C}$. O produto tensorial é caracterizado pelas seguintes propriedades:

- $z(|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle)$;
- $(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle |w\rangle + |v_2\rangle |w\rangle$;
- $|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle |w_1\rangle + |v\rangle |w_2\rangle$.

Se A e B são operadores lineares em V e W, respectivamente, então $A \otimes B$ é o operador linear em $V \otimes W$ definido por

$$(A \otimes B)(|v\rangle \otimes |w\rangle) \equiv A|v\rangle \otimes B|w\rangle, \tag{2.4}$$

em que $|v\rangle$ é um vetor de V e $|w\rangle$ é um vetor de W.

Percebamos que, se A e B são operadores lineares em V e C e D são operadores lineares em W, então, segue, diretamente da Equação 2.4, que $(A \otimes C)(B \otimes D) = AB \otimes CD$. Veja:

$$\begin{aligned}(A \otimes C)(B \otimes D)(|v\rangle \otimes |w\rangle) &= (A \otimes C)(B|v\rangle \otimes D|w\rangle) \\ &= AB|v\rangle \otimes CD|w\rangle \\ &= (AB \otimes CD)(|v\rangle \otimes |w\rangle).\end{aligned}$$

A definição de $A \otimes B$ é estendida aos elementos de $V \otimes W$, por linearidade:

$$(A \otimes B)(\sum_i a_i |v_i\rangle \otimes |w_i\rangle) = \sum_i a_i A|v_i\rangle \otimes B|w_i\rangle.$$

Os produtos internos dos espaços V e W podem ser utilizados para definir um produto interno em $V \otimes W$:

$$\left(\sum_i a_i |v_i\rangle \otimes |w_i\rangle, \sum_j b_j \left|v'_j\right\rangle \otimes \left|w'_j\right\rangle\right) \equiv \sum_{ij} a_i^* b_j \left\langle v_i \middle| v'_j \right\rangle \left\langle w_i \middle| w'_j \right\rangle.$$

A ação do produto tensorial na representação matricial é conhecida como *produto de Kronecker*. Suponha que A é uma matriz $m \times n$ e B é uma matriz $p \times q$. Então, a representação matricial de $A \otimes B$ é uma matriz $mp \times nq$, definida por

$$A \otimes B \equiv \begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1n}B \\ A_{21}B & A_{22}B & \dots & A_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1}B & A_{m2}B & \dots & A_{mn}B \end{bmatrix},$$

em que A_{ij} representa o elemento de da linha i e coluna j de A, e $A_{ij}B$ é uma submatriz de dimensões $p \times q$ que resulta do produto usual complexo de A_{ij} por cada elemento de B.

A notação $|\psi\rangle^{\otimes n}$ representa o produto tensorial do estado $|\psi\rangle$ por ele mesmo n vezes. A notação $A^{\otimes n}$ possui significado semelhante, só que relacionado ao operador linear A.

Definimos, agora, duas operações entre operadores lineares: o *comutador* e o *anticomutador*.

Definição 18. *Dados dois operadores lineares A e B, o comutador entre eles é dado por*

$$[A, B] = AB - BA.$$

Se $[A, B] = 0$, dizemos que A e B comutam.

Definição 19. *Dados dois operadores lineares A e B, o anticomutador entre eles é dado por*

$$\{A, B\} = AB + BA.$$

Se $\{A, B\} = 0$, dizemos que A e B anticomutam. Como dito, os operadores de Pauli diferentes de I anticomutam entre si, de forma que $\{X, Y\} = \{X, Z\} = \{Y, Z\} = 0$.

Existe uma conexão entre o comutador e a possibilidade de diagonalização simultânea de operadores Hermitianos, apresentada no Teorema 1:

Teorema 1. *(Teorema da diagonalização simultânea [17]) Sejam A e B operadores Hermitianos. $[A, B] = 0$ se, e somente se, existir uma base ortonormal tal que A e B sejam diagonalizáveis em relação a essa base. Nesse caso, diz-se que A e B são simultaneamente diagonalizáveis.*

2.3 Postulados da mecânica quântica

A mecânica quântica é uma estrutura matemática acionada para o desenvolvimento de uma teoria física. No entanto, por si só, a mecânica quântica não estabelece quais leis um sistema físico deve obedecer: esta fornece o aparato matemático e conceitual

para o desenvolvimento de tais leis. Os postulados da mecânica quântica providenciam a conexão necessária entre o mundo físico e o formalismo da mecânica quântica [17]. A seguir, apresentamos os quatro postulados.

Postulado 1. *A qualquer sistema físico isolado existe associado um espaço vetorial complexo com produto interno (ou seja, um espaço de Hilbert), conhecido como espaço de estados do sistema. O sistema é completamente descrito pelo seu vetor de estado, um vetor unitário no espaço de estados.*

Postulado 2. *A evolução de um sistema quântico fechado é descrita por uma transformação unitária. Ou seja, o estado $|\psi\rangle$ de um sistema em um tempo t_1 está relacionado ao estado $|\psi'\rangle$ do sistema em t_2 por um operador unitário U que depende somente de t_1 e t_2:*

$$|\psi'\rangle = U\,|\psi\rangle\,.$$

Postulado 3. *As medições quânticas são descritas por determinados operadores de medição M_m. Esses operadores atuam sobre o espaço de estados do sistema. O índice m se refere aos possíveis resultados da medição. Se o estado de um sistema quântico for $|\psi\rangle$, imediatamente antes da medição, a probabilidade de um resultado m ocorrer é dada por*

$$p(m) = \langle\psi|\, M_m^\dagger M_m\,|\psi\rangle\,,$$

e o estado do sistema após a medição será

$$\frac{M_m\,|\psi\rangle}{\sqrt{\langle\psi|\, M_m^\dagger M_m\,|\psi\rangle}}.$$

Os operadores de medição satisfazem a relação de completude:

$$\sum_m M_m^\dagger M_m = I.$$

Postulado 4. *O espaço de estados de um sistema físico composto é o produto tensorial dos espaços de estados dos sistemas individuais.*

O Postulado 1 da mecânica quântica descreve que o local no qual o sistema quântico funciona é um espaço com produto interno, ou seja, um espaço de Hilbert. O Postulado 2 caracteriza a evolução temporal dos estados quânticos isolados ou fechados no tempo, relacionando os estados de um sistema em dois instantes de tempos distintos por meio de um operador linear unitário U.

Um sistema quântico fechado é aquele que não interage com nenhum outro sistema. De fato, qualquer sistema, com exceção do universo como um todo, interage com outros sistemas, porém, existem sistemas que podem ser descritos, com uma boa aproximação, como sendo fechados e por evoluções unitárias [17].

O Postulado 3 apresenta os operadores de medição, que atuam no sistema quando ele é observado, por exemplo, para a extração de informações. Um caso importante acontece quando os operadores de medição são projetores, P_m. Nesse caso, a probabilidade de se medir o estado $|\psi\rangle$ e obter o resultado m é dada por $p(m) = \langle\psi| P_m |\psi\rangle$, enquanto o estado após a medição é $\dfrac{P_m |\psi\rangle}{\sqrt{p(m)}}$. Por fim, o Postulado 4 explica como espaços de estados de sistemas quânticos diferentes se combinam para formar sistemas compostos por meio do produto tensorial.

2.4 Operadores de densidade

Existe ainda mais uma forma de descrever sistemas quânticos: por meio do *operador densidade* ou *matriz densidade*.

Definição 20. *Dados um conjunto de estados $\{|\psi_1\rangle, |\psi_2\rangle, \ldots |\psi_n\rangle\}$ e um sistema quântico que esteja em um desses estados $|\psi_i\rangle$ com probabilidade p_i, chamamos o conjunto $\{p_i, |\psi_i\rangle\}$ de um ensemble de estados puros. O operador densidade do sistema é definido por*

$$\rho = \sum_{i=1}^{n} p_i |\psi_i\rangle \langle\psi_i| .$$

Os quatro postulados da mecânica quântica podem ser reformulados em termos dos operadores de densidade. Entretanto, apresentaremos apenas a evolução de um sistema quântico sob essa descrição, posto que o utilizaremos quando formos descrever a incidência de erros em canais quânticos. Os demais postulados reformulados podem ser consultados em [17].

Postulado 2*. *Dado um sistema descrito pelo operador de densidade ρ, a evolução desse sistema por meio de um operador unitário U é descrita como*

$$\rho = \sum_{i=1}^{n} p_i |\psi_i\rangle \langle\psi_i| \longrightarrow \sum p_i U |\psi_i\rangle \langle\psi_i| U^\dagger = U\rho U^\dagger.$$

2.5 Códigos quânticos corretores de erros

Os erros em qubits ocorrem devido a imprecisões no controle dos qubits ou a interações deles com o ambiente. No segundo caso, as interações podem destruir estados em superposição, o que causa a perda de informação ou decoerência.

Uma das formas de proteger a informação quântica contra erros é o uso de códigos quânticos corretores de erros (CQCE). Assim como os códigos corretores de erros

clássicos, tais códigos codificam a informação de forma a torná-la resistente ao ruído, decodificando-a quando for necessário recuperar o estado original.

Um CQCE que codifica k qubits lógicos em n qubits físicos possui uma base com 2^k palavras-código, que correspondem à base dos estados originais. Uma combinação linear das palavras-código da base é também uma palavra-código. O espaço código $\mathcal{C}$, ou o espaço das palavras-código válidas, é um espaço de Hilbert, ou, mais precisamente, um subespaço de dimensão 2^k de um espaço de Hilbert de dimensão 2^n.

Se um código corrige os erros E e F, então ele consegue corrigir erros da forma $aE + bF$, combinações lineares desses erros. Portanto, só é necessário verificar se o código consegue corrigir uma base de erros. Uma base conveniente é o conjunto dos operadores formados pelos tensoriais de I, X, Y e Z [32]. De fato, os tensoriais de X (e I) e Z (e I) já são suficientes para gerar todos os erros.

O produto tensorial de ordem n das matrizes de Pauli, conjuntamente com os fatores multiplicativos ± 1 e $\pm i$, sob operação de multiplicação matricial, formam o *grupo de Pauli de ordem n*, denotado por $\mathcal{G}_n$. Como exemplo, o grupo $\mathcal{G}_1$ é da forma

$$\mathcal{G}_1 = \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}.$$

Antes de falar das propriedades de $\mathcal{G}_n$, vamos definir formalmente o que é um grupo.

Definição 21. *Seja G um conjunto não vazio no qual está definida uma operação binária $*$ entre elementos de G, denotada por*

$$\begin{aligned} * : G \times G &\to G \\ (x, y) &\to x * y. \end{aligned}$$

Dizemos que o par $G, *$ é um grupo se forem válidas as seguintes propriedades:

1. *$a * (b * c) = (a * b) * c$, $\forall\ a, b, c \in G$ (associatividade);*
2. *$\exists\ e \in G$ tal que $a * e = e * a$, $\forall\ a \in G$ (elemento neutro);*
3. *$\forall\ a \in G, \exists\ b \in G$ tal que $a * b = b * a = e$ (existência de inverso).*

O grupo de Pauli $\mathcal{G}_n$, denominado grupo de erros, possui algumas características de destaque:

1. todo elemento de $\mathcal{G}_n$ é unitário, ou seja, $\forall M \in \mathcal{G}_n, M^{-1} = M^\dagger$;

2. para todo elemento $M \in \mathcal{G}_n$ temos que $M^2 = \pm I$. Se $M^2 = I$, então M é Hermitiano ($M = M^\dagger$). Caso $M^2 = -I$, então M é anti-Hermitiano ($M = -M^\dagger$);

3. dados M e $N \in \mathcal{G}_n$, M e N comutam ($MN = NM$) ou anticomutam ($MN = -NM$).

Dado um elemento de $\mathcal{G}_n$, o seu *peso* é o número de fatores no tensor que diferem de I.

Para que um código consiga corrigir dois erros E_a e E_b, é necessário que sempre consigamos distinguir a ação de E_a sobre uma palavra-código da base, $|\psi_i\rangle$, da ação de E_b sobre outra palavra-código da base, $|\psi_j\rangle$. Só é possível garantir isso se $E_a |\psi_i\rangle$ e $E_b |\psi_j\rangle$ forem ortogonais, ou seja,

$$\langle\psi_i| E_a^\dagger E_b |\psi_j\rangle = 0, \tag{2.5}$$

para todo $|\psi_i\rangle, |\psi_j\rangle \in T$, com $i \neq j$. Caso a ortogonalidade não seja satisfeita, há a possibilidade de se confundir os erros, o que levaria à aplicação da operação de correção incorreta, e, consequentemente, à perda de informação. Normalmente se considera I no conjunto dos possíveis erros, posto que visamos diferenciar quando ocorreu um erro em um qubit ou nenhum erro aconteceu.

É necessário garantir também que, quando realizarmos medições para descobrir um possível erro, nada seja revelado sobre a palavra-código em si. Caso contrário, causaríamos uma pertubação nas superposições dos estados da base e, com isso, apesar de conseguirmos corrigi-los, não seríamos capazes de corrigir um estado arbitrário.

Sabemos sobre o erro através da medição $\langle\psi_i| E_a^\dagger E_b |\psi_i\rangle$. Para que ela não revele informação sobre o estado $|\psi_i\rangle$, é necessário que o resultado seja o mesmo para todas as palavras-código da base

$$\langle\psi_i| E_a^\dagger E_b |\psi_i\rangle = \langle\psi_j| E_a^\dagger E_b |\psi_j\rangle . \tag{2.6}$$

Combinando as condições das Equações (2.5) e (2.6), chegamos a uma única equação:

$$\langle\psi_i| E_a^\dagger E_b |\psi_j\rangle = C_{ab}\delta_{ij}, \tag{2.7}$$

em que $|\psi_i\rangle$ e $|\psi_j\rangle$ percorrem todos os elementos da base do código, E_a e E_b percorrem todos os possíveis erros, C_{ab} é independente de i e j e δ_{ij} é o *delta de Kronecker*, que possui valor 1 quando $i = j$ e valor 0 quando $i \neq j$. A condição da Equação (2.7) foi descoberta por Bennett *et al* [33] e por Knill e Laflamme [34]. Para que os erros $\{E_a\}$ sejam corrigidos pelo código $\mathcal{C}$, a condição (2.7) não é apenas necessária, mas também suficiente [32].

Na Equação (2.7), se $E_a, E_b \in \mathcal{G}_n$, então $E = E_a^\dagger E_b \in \mathcal{G}_n$. O peso do menor operador $E \in \mathcal{G}_n$ para o qual a Equação (2.7) não é satisfeita é a distância do código. Para que um código $\mathcal{C}$ consiga corrigir até t erros, é necessário que sua distância seja ao menos $d = 2t + 1$. Para que um código consiga identificar s erros, é necessário que ele possua

uma distância de no mínimo $d = s + 1$ [32]. Um código $[[n, k, d]]$ é um código que codifica k qubits lógicos em n qubits físicos e possui distância d.

A seguir, apresentamos os códigos de três qubits para erros *bit-flip* e *phase-shift* e o resultado da concatenação dos dois códigos, o código de Shor. Também será introduzida a classe de códigos CSS.

2.5.1 Código de três qubits para erros *bit-flip*

Suponha que um canal utilizado para o envio de qubits os mantém inalterados com probabilidade $1 - p$ e os inverte com probabilidade p. Esse canal é conhecido como *canal bit-flip* e faz com que o estado $|\psi\rangle$ seja levado no estado $X|\psi\rangle$ com probabilidade p, em que X é a matriz de Pauli apresentada anteriormente, também chamada de *operador bit-flip*.

Naturalmente, podemos pensar na utilização de um código de repetição quântico para lidar com esse canal. Entretanto, o *teorema da não-clonagem* torna impossível a implementação de tal código. Tal teorema afirma a impossibilidade de se fazer cópias de um estado quântico arbitrário desconhecido. Vejamos o enunciado desse teorema e uma prova.

Teorema 2. *(Teorema da não-clonagem) Não existe operador unitário U que evolua o estado $|\psi\rangle|\phi\rangle$ para o estado $|\psi\rangle|\psi\rangle$, para $|\psi\rangle$ arbitrário.*

Demonstração. Suponha que exista operador U capaz de realizar essa cópia. Então, para os estados arbitrários $|\psi\rangle$ e $|\rho\rangle$, vale que

$$U|\psi\rangle|\phi\rangle = |\psi\rangle|\psi\rangle \quad \text{e} \quad U|\rho\rangle|\phi\rangle = |\rho\rangle|\rho\rangle .$$

Segue da linearidade que

$$U(a|\psi\rangle + b|\rho\rangle)|\phi\rangle = a\,U|\psi\rangle|\phi\rangle + b\,U|\rho\rangle|\phi\rangle = a|\psi\rangle|\psi\rangle + b|\rho\rangle|\rho\rangle . \tag{2.8}$$

Por outro lado, como U clona estados arbitrários, então vale que

$$\begin{aligned} U(a|\psi\rangle + b|\rho\rangle)|\phi\rangle =& (a|\psi\rangle + b|\rho\rangle)(a|\psi\rangle + b|\rho\rangle) \\ =& a^2|\psi\rangle|\psi\rangle + b^2|\rho\rangle|\rho\rangle + ab|\psi\rangle|\psi\rangle + ab|\psi\rangle|\psi\rangle . \end{aligned} \tag{2.9}$$

Porém, as Equações (2.8) e (2.9) são diferentes, a não ser que a ou b sejam iguais a 0. □

Assim, em geral, dada uma base ortonormal podemos copiar os estados da base, mas não é possível copiar corretamente as superposições desses estados.

Apresentamos o código *bit-flip*, usado para proteger qubits contra os efeitos do ruído desse canal. Considere um qubit em um estado arbitrário $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ e que esse estado seja codificado com o uso de três qubits em $\alpha|000\rangle + \beta|111\rangle$. Essa codificação pode ser expressa como

$$|0\rangle \rightarrow |0_L\rangle \equiv |000\rangle \quad \text{e} \quad |1\rangle \rightarrow |1_L\rangle \equiv |111\rangle\,,$$

onde as notações $|0_L\rangle$ e $|1_L\rangle$ indicam os *estados lógicos* $|0\rangle$ e $|1\rangle$, respectivamente.

Suponha que o estado inicial $\alpha|0\rangle + \beta|1\rangle$ seja perfeitamente codificado como $\alpha|000\rangle + \beta|111\rangle$. Envia-se, então, cada um dos três qubits por meio de uma cópia do canal *bit-flip*. Considere, agora, que uma inversão ocorra em, no máximo, um qubit. O procedimento de correção de erros para a recuperação do estado original, $\alpha|000\rangle + \beta|111\rangle$, é dividido em duas etapas.

Primeiramente, com o intuito de detectar a existência de erros, é realizada uma medição sobre o estado quântico. Chamamos esse resultado de *síndrome do erro*. Para o canal *bit-flip* existem quatro possíveis síndromes de erro, correspondentes aos seguintes operadores de projeção:

$$\begin{aligned}
P_0 &= |000\rangle\langle 000| + |111\rangle\langle 111| && \text{sem erros}\\
P_1 &= |100\rangle\langle 100| + |011\rangle\langle 011| && \text{inversão do primeiro qubit}\\
P_2 &= |010\rangle\langle 010| + |101\rangle\langle 101| && \text{inversão do segundo qubit}\\
P_3 &= |001\rangle\langle 001| + |110\rangle\langle 110| && \text{inversão do terceiro qubit.}
\end{aligned}$$

O valor da síndrome corresponde ao subíndice dos projetores.

Para entender o funcionamento dos operadores P_i, suponha que durante o envio do estado $|\psi\rangle = \alpha|000\rangle + \beta|111\rangle$ ocorra um erro no primeiro qubit. Dessa forma, o estado quântico recebido após a transmissão será $|\psi'\rangle = \alpha|100\rangle + \beta|011\rangle$.

Decorrente da medição sobre os projetores P_1, P_2, P_3 e P_4, temos que

$$\begin{aligned}
\langle\psi'|P_1|\psi'\rangle &= |\beta|^2 + |\alpha|^2 = 1\\
\langle\psi'|P_i|\psi'\rangle &= 0, \text{ para } i = 0, 2, 3.
\end{aligned}$$

Consequentemente, a síndrome é 1. É importante observar que a medição da síndrome contém informação apenas sobre o erro ocorrido. Ela não traz informação sobre as amplitudes α e β. Portanto, a medição da síndrome não destrói a informação associada ao estado quântico. Se houver a inversão de apenas um dos qubits, é possível identificá-lo com o código *bit-flip*.

Identificado o erro, a segunda etapa de correção é a utilização do valor da síndrome para determinar o procedimento a ser adotado para recuperação do estado inicial. Se, como no exemplo anterior, a síndrome de erro for 1, ou seja, ocorreu um erro no

primeiro qubit, para a recuperação do estado inicial é necessária a aplicação do operador X sobre o primeiro qubit. A Tabela 2.1 apresenta as quatro possíveis síndromes de erro e os procedimentos correspondentes que devem ser utilizados para correção.

Tabela 2.1 – Síndrome de erro e procedimento para correção de erro.

síndrome	procedimento
0	não faça nada
1	inverta o primeiro qubit
2	inverta o segundo qubit
3	inverta o terceiro qubit

Fonte: elaborado pelo autor.

Esse procedimento funciona perfeitamente se a inversão de bit ocorrer em no máximo um dos três qubits. A probabilidade de que isso aconteça é de $(1-p)^3+3p(1-p)^2 = 1-3p^2+2p^3$. Se $p < 1/2$, a codificação e a decodificação aumentam a confiabilidade no armazenamento do estado quântico.

Existe ainda um modo alternativo para identificação do erro *bit-flip* no qual, em vez de serem medidos os projetores P_0, P_1, P_2 e P_3, executam-se as medições, $Z_1Z_2 = Z \otimes Z \otimes I$ e $Z_2Z_3 = I \otimes Z \otimes Z$. Cada um desses observáveis possui autovalores ± 1 e, por isso, cada medição fornece um único bit de informação, totalizando dois bits de informação. A primeira medição, Z_1Z_2, compara o valor do primeiro e do segundo qubits. Para verificar isso, perceba que

$$Z_1Z_2 = (|00\rangle\langle 00| + |11\rangle\langle 11|) \otimes I - (|01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I,$$

o que corresponde a uma medição projetiva com os projetores $(|00\rangle\langle 00| + |11\rangle\langle 11|) \otimes I$ e $(|01\rangle\langle 01| + |10\rangle\langle 10|) \otimes I$. Dessa forma, medir Z_1Z_2 é equivalente a comparar o valor dos dois primeiros qubits, retornando 1, se eles forem iguais, ou -1, se forem diferentes. Da mesma forma, o operador Z_2Z_3 faz a comparação entre o segundo e o terceiro qubits. A partir do resultado das duas medições é possível verificar se ocorreu a inversão de algum qubit, e, caso positivo, em qual. Essas medições também não fornecem informações sobre α e β do estado quântico codificado e, portanto, não destroem a superposição de estados quânticos que se deseja preservar. A Tabela 2.2 apresenta as possíveis síndromes para esses operadores de medição e a identificação do qubit em que ocorreu o erro.

O código *bit-flip* é um código $[[3, 1, 3]]$, ou seja, codifica um qubit em três qubits e possui distância 3.

Tabela 2.2 – Síndrome de erro para os operadores Z_1Z_2 e Z_2Z_3 do código *bit-flip*

Autovalores de (Z_1Z_2, Z_2Z_3)	qubit com erro
(+1,+1)	nenhum
(+1,-1)	terceiro qubit
(-1,+1)	primeiro qubit
(-1,-1)	segundo qubit

Fonte: elaborado pelo autor.

2.5.2 Código de três qubits para inversão de fase

Passemos à discussão sobre o canal ruidoso conhecido como *phase-shift*. Ele canal tem como característica preservar o qubit com probabilidade $1-p$ e trocar a fase entre os estados $|0\rangle$ e $|1\rangle$ com probabilidade p. Ou seja, o estado $|\psi\rangle$ é levado ao estado $Z|\psi\rangle$ com probabilidade p, sendo Z uma das matrizes de Pauli. Se o estado inicial é $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, então a atuação de Z resultará em $Z|\psi\rangle = \alpha|0\rangle - \beta|1\rangle$.

Não há um equivalente clássico para este canal porque classicamente não existe essa característica. No entanto, é possível transformá-lo em um canal de *bit-flip*. Para isso, utiliza-se a base conjugada $\{|+\rangle, |-\rangle\}$, pois a ação de Z leva $|+\rangle$ em $|-\rangle$, e vice-versa.

Para o código *phase-shift*, os qubits lógicos na base conjugada são escritos como:

$$|0_L\rangle \equiv |+++\rangle \quad \text{e} \quad |1_L\rangle \equiv |---\rangle .$$

Considerando a base conjugada, codificação, detecção de erro e recuperação do estado inicial ocorrem de modo similar ao caso do bit-flip. A mudança de base é obtida com a aplicação da porta de Hadamard sobre a base computacional, pois $H|0\rangle = |+\rangle$ e $H|1\rangle = |-\rangle$.

Na codificação, cada qubit de informação é codificado em três qubits e, na sequência, a porta de Hadamard é aplicada sobre cada um dos qubits. Com isso, os estados lógicos são:

$$|0\rangle \to |0_L\rangle \equiv \frac{1}{2\sqrt{2}}(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle)$$
$$|1\rangle \to |1_L\rangle \equiv \frac{1}{2\sqrt{2}}(|000\rangle - |001\rangle - |010\rangle + |011\rangle - |100\rangle + |101\rangle + |110\rangle - |111\rangle)$$

Para a detecção de um erro são realizadas as mesmas medições projetivas do canal *bit-flip*, mas conjugadas com as portas de Hadamard: $P_i \to P_i' \equiv H^{\otimes 3} P_i H^{\otimes 3}$, em que $i = 0, 1, 2, 3$.

A correção de erros no canal *phase-shift* utiliza os operadores empregados no canal *bit-flip*, conjugados pela matriz de Hadamard. Assim, caso seja identificada a inversão de fase do i-ésimo qubit, aplica-se o operador $Z = HXH$ sobre ele, para fazer a correção.

De forma equivalente ao código *bit-flip*, os observáveis $X_1X_2 = H^{\otimes 3}Z_1Z_2H^{\otimes 3}$ e $X_2X_3 = H^{\otimes 3}Z_2Z_3H^{\otimes 3}$ podem ser utilizados para medir as síndromes, comparando os sinais do primeiro e segundo qubits e do segundo e terceiro qubits, respectivamente. A Tabela 2.3 apresenta as possíveis síndromes para esses operadores de medição e a identificação do qubit em que ocorreu o erro *phase-shift*.

Tabela 2.3 – Síndrome de erro para os operadores X_1X_2 e X_2X_3 do código *phase-shift*

Autovalores de (X_1X_2, X_2X_3)	qubit com erro
(+1,+1)	nenhum
(+1,-1)	terceiro qubit
(-1,+1)	primeiro qubit
(-1,-1)	segundo qubit

Fonte: elaborado pelo autor.

Os parâmetros do código *phase-shift* são os mesmos do código *bit-flip*, ou seja, é um código $[[3, 1, 3]]$.

2.5.3 Código de Shor

Agora, veremos um CQCE capaz de proteger um estado de um qubit contra os efeitos de um erro arbitrário, o *código de Shor*. Esse código é obtido de uma combinação dos códigos de três qubits para erros *bit-flip* e de *phase-shift*, por meio de uma técnica chamada *concatenação*. Tal técnica é uma maneira eficiente de se obter novos códigos a partir de códigos conhecidos.

Primeiro o estado do qubit é codificado utilizando o código *phase-shift*, ou seja, $|0\rangle \rightarrow |+++\rangle, |1\rangle \rightarrow |---\rangle$. Em seguida, cada um dos qubits resultantes da primeira codificação é codificado utilizando o código *bit-flip*, de forma que $|+\rangle$ é codificado em $(|000\rangle + |111\rangle)/\sqrt{2}$ e $|-\rangle$ em $(|000\rangle - |111\rangle)/\sqrt{2}$. O resultado final é um código de nove qubits, no qual os estados lógicos são:

$$|0\rangle \rightarrow |0_L\rangle \equiv \frac{(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)}{2\sqrt{2}}$$
$$|1\rangle \rightarrow |1_L\rangle \equiv \frac{(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)}{2\sqrt{2}}$$

O código de Shor é capaz de corrigir erros *bit-flip* e *phase-shift* em qualquer qubit. Mais do que isso, tal código é capaz de corrigir os dois tipos de erros simultaneamente, desde que atuem em um mesmo qubit.

A identificação de erros do tipo *bit-flip* é realizada por meio de observáveis que comparam os valores dos qubits dentro de cada bloco de três qubits. Os observáveis que verificam o primeiro bloco de qubits são Z_1Z_2 e Z_2Z_3. Comparando suas medições, é possível determinar se há algum qubit diferente no bloco e, nesse caso, corrigi-lo. Da mesma

forma as medições de $\{Z_4Z_5, Z_5Z_6\}$ e $\{Z_7Z_8, Z_8Z_9\}$ podem ser utilizadas para verificar os qubits do segundo e do terceiro blocos contra erros de inversão de bit, respectivamente. Como em geral não se sabe em qual bloco ocorreu o possível erro, é necessária a utilização dos seis observáveis. A correção do erro é realizada pela aplicação do operador X sobre o qubit em que é identificado o erro.

Para a detecção de um erro do tipo *phase-shift*, os observáveis $X_1X_2X_3X_4X_5X_6$ e $X_4X_5X_6X_7X_8X_9$ são os utilizados. O primeiro observável compara os sinais do primeiro e do segundo blocos de qubits, enquanto o segundo compara os sinais do segundo e do terceiro blocos. Se um erro de inversão de fase é identificado no primeiro bloco de qubits, a correção é realizada pela aplicação do operador $Z_1Z_2Z_3$. Caso o erro tenha acontecido no segundo ou no terceiro bloco, deve-se utilizar, respectivamente, os operadores $Z_4Z_5Z_6$ e $Z_7Z_8Z_9$.

O código de Shor consegue proteger estados quânticos contra a ação de erros totalmente arbitrários, contanto que apenas um qubit tenha sido afetado. Para isso, não é necessário o uso de operações adicionais, realizando-se apenas o procedimento descrito acima. Assim, corrigindo apenas um subconjunto discreto de erros, outros possíveis erros serão automaticamente corrigidos. O código de Shor é um código $[[9, 1, 3]]$.

2.5.4 Códigos CSS

Os códigos de Calderbank-Shor-Steane (CSS), são uma ampla classe de códigos quânticos construídos a partir de códigos lineares clássicos, C_1 e C_2, tais que $C_2 \subseteq C_1$ e C_1 e $C_2^{\perp}$ corrigem ambos t erros. Suponha que C_1 é um código (n, k_1) e C_2 é (n, k_2). Considere $x \in C_1$, ou seja, x é uma palavra-código em C_1. O estado quântico $|x + C_2\rangle$ é dado por:

$$|x + C_2\rangle \equiv \frac{1}{\sqrt{|C_2|}} \sum_{y \in C_2} |x + y\rangle \,,$$

em que a soma $x + y$ é realizada módulo 2. Seja x' um elemento de C_1 tal que $x - x' \in C_2$. Mostra-se que $|x + C_2\rangle = |x' + C_2\rangle$ e, portanto, o estado $|x + C_2\rangle$ depende somente da classe lateral C_1/C_2 à qual x pertence. Temos também que se x e x' pertencem a diferentes classes laterais de C_2, então para nenhum $y, y' \in C_2$ ocorre $x + y = x' + y'$ e, portanto, $|x + C_2\rangle$ e $|x' + C_2\rangle$ são estados ortogonais.

O CQCE $CSS(C_1, C_2)$ binário é definido como o espaço vetorial gerado pelos estados $|x + C_2\rangle$, para todo $x \in C_1$. O número de classes laterais de C_2 em C_1 é $|C_1|/|C_2| = 2^{k_1 - k_2}$, portanto, o número de qubits codificados é $k_1 - k_2$. Assim, um código $CSS(C_1, C_2)$ é um CQCE $[[n, k_1 - k_2]]$. Relembraremos mais uma vez que essa construção dos códigos CSS foi generalizada por Klappenecker *et al.* [22] para alfabetos q-ários, em que q é potência de um número primo.

2.6 Códigos estabilizadores

Definição 22. *Dado um subgrupo abeliano $S \in \mathcal{G}_n$ que não contenha $-I$, é possível associar um código $\mathcal{C}_S$ ao autoespaço simultâneo com autovalor 1 de todos os elementos de S. O subgrupo S é denotado por estabilizador e o código $\mathcal{C}_S$ é o código estabilizador associado a S. Matematicamente, o código $\mathcal{C}_S$ pode ser expresso por:*

$$\mathcal{C}_S = \{|\psi\rangle \, ; M \, |\psi\rangle = |\psi\rangle \, , \forall M \in S\}.$$

A denominação de S como estabilizador vem do fato desse grupo preservar todas as palavras-código [26].

Vamos entender porque é necessário que S seja abeliano e $-I \notin S$. Suponha que $-I \in S$, então, dado $|\psi\rangle \in \mathcal{C}_S$, temos que $-I \, |\psi\rangle = |\psi\rangle$, ou seja, $|\psi\rangle = 0$. Já se S não fosse abeliano, então existem $M, N \in S$, tais que $MN = -NM$, e dado $|\psi\rangle \in \mathcal{C}_S$, teríamos que $-|\psi\rangle = -NM \, |\psi\rangle = MN \, |\psi\rangle = |\psi\rangle$, ou seja, $|\psi\rangle = 0$.

Como vimos na Seção 2.5, o quadrado de todo elemento de $\mathcal{G}_n$ é igual a $\pm I$. Devido ao fato de $-I \notin S$, então temos que $\forall M \in S$ vale que $M^2 = I$, caso contrário, $-I \in S$. Ou seja, todos os elementos do estabilizador são hermitianos.

Definição 23. *Diz-se que um conjunto $\{M_1, M_2, \ldots M_l\}$ de elementos independentes de S gera S se todo elemento de S pode ser escrito como um produto de $M_1, M_2, \ldots, M_l$.*

Fazer uso do conjunto de geradores é uma forma mais eficiente e compacta de trabalhar com o estabilizador S. Por exemplo, para verificar se um estado $|\psi\rangle$ pertence ao espaço do código $\mathcal{C}_S$, basta avaliar se esse estado é fixado por todos os geradores de S. Em outras palavras, não há a necessidade de verificar se todo elemento do estabilizador fixa $|\psi\rangle$. Os geradores funcionam, então, de forma análoga às matrizes verificação de paridade dos códigos lineares clássicos, sendo as medições desses operadores utilizadas para a identificação de erros.

A seguir, mostraremos quantos qubits um código estabilizador pode codificar a partir do número de geradores do estabilizador. Contudo, primeiro precisamos provar um resultado auxiliar.

Afirmação 1. *Seja M uma matriz $n \times n$. Se λ^2 é um autovalor da matriz M^2, então λ ou - λ é um autovalor de M.*

Demonstração. Como λ^2 é um autovalor de M^2, então vale que $\det(M^2 - \lambda^2 I) = 0$.

É possível escrever $M^2 - \lambda^2 I$ da seguinte forma: $M^2 - \lambda^2 I = (M + \lambda I)(M - \lambda I)$. Segue daí que:

$$\begin{aligned} 0 &= \det\Big(M^2 - \lambda^2 I\Big) \\ &= \det((M+\lambda I)(M-\lambda I)) \\ &= \det(M+\lambda I)\det(M-\lambda I). \end{aligned}$$

Portanto, ou $\det(M+\lambda I)=0$ ou $\det(M-\lambda I)=0$, o que implica que, ou λ ou $-\lambda$ é um autovalor de M.

□

Afirmação 2. *Se S tem $n-k$ geradores, então o espaço do código $\mathcal{C}_S$ tem dimensão 2^k, ou seja, $\mathcal{C}_S$ codifica k qubits.*

Demonstração. Suponha que o estabilizador S possua um conjunto de $n-k$ geradores, $S_1, S_2, \dots S_{n-k}$. Tomemos o primeiro gerador, S_1. Como $S_1^2 = I$, então, da Afirmação 1, temos que S_1 possui autovalores ± 1.

Além disso, o traço de S_1 é zero. Vejamos o porquê. Vale que $\mathrm{Tr}[A\otimes B] = \mathrm{Tr}[A]\,\mathrm{Tr}[B]$ e, com exceção de I, todas as matrizes de Pauli (X, Y e Z) possuem traço 0. Assim, todo $M \in \mathcal{G}_n \neq \pm I$ possui ao menos um fator no tensor diferente de I e, portanto, $\mathrm{Tr}[M]=0$. Como I não pode ser um dos geradores de S (pois como $S_i^2 = I$, então o menor conjunto de elementos que gera S não seria $S_1, S_2, \dots S_{n-k}$), então $\mathrm{Tr}[S_1]=0$. Assim, S_1 possui 2^{n-1} autovalores 1 e 2^{n-1} autovalores -1. Com isso, $S_1|\psi\rangle = |\psi\rangle$ divide o espaço de Hilbert de dimensão n pela metade.

Vamos analisar dentro do autoespaço relacionado a $S_1|\psi\rangle = |\psi\rangle$, qual a dimensão do subespaço que surge quando se é imposto que $S_2|\psi\rangle = |\psi\rangle$. Para isso, é preciso recorrer ao projetor $\frac{1}{2}(I+S_1)$, responsável por projetar no espaço do autovalor $+1$ de S_1. Perceba que $\mathrm{Tr}\Big[\frac{1}{2}(I+S_1)S_2\Big] = 0$, pois $\mathrm{Tr}\Big[\frac{1}{2}(I+S_1)S_2\Big] = \frac{1}{2}(\mathrm{Tr}[S_2] + \mathrm{Tr}[S_1S_2])$ e $S_2, S_1S_2 \in \mathcal{G}_n - \{-I\}$. Assim, do espaço de dimensão 2^{n-1} que satisfaz $S_1|\psi\rangle = |\psi\rangle$, um subespaço de dimensão 2^{n-2} satisfaz $S_2|\psi\rangle = |\psi\rangle$. Seguindo da mesma forma para os demais geradores, concluímos que S_i divide o espaço fixado por $S_1, S_2, \dots S_{i-1}$ pela metade. Assim, para um conjunto de $n-k$ geradores, o autoespaço simultâneo com autovalor $+1$ tem dimensão $2^{n-(n-k)} = 2^k$.

□

2.6.1 Correção e síndrome de erros em códigos estabilizadores

Seja $\varepsilon = \{E_a\} \subset \mathcal{G}_n$ um conjunto de erros. Um operador de erro $E_a \in \varepsilon$ satisfaz uma das seguintes situações:

(i) $\{E_a, M\} = 0$ para algum gerador $M \in S$;

(ii) $[E_a, M] = 0$ para todos os geradores M de S e $E_a \in S$;

(iii) $[E_a, M] = 0$ para todos os geradores M de S, mas $E_a \notin S$.

Analisemos as três situações.

(i) Se o erro E_a anticomutar com algum gerador M do estabilizador, então, para todo estado $|\psi\rangle \in \mathcal{C}_S$, temos

$$ME_a |\psi\rangle = -E_a M |\psi\rangle = -E_a |\psi\rangle$$

e, portanto, $E_a |\psi\rangle$ é um autovetor de M, associado ao autovalor -1 e não pertence ao espaço do código $\mathcal{C}_S$. Como consequência, conseguimos identificar o erro E_a por meio de uma medição de M.

(ii) Para o segundo caso, como E_a comuta com todos os geradores de S, temos que para todo gerador M:

$$ME_a |\psi\rangle = E_a M |\psi\rangle = E_a |\psi\rangle . \tag{2.10}$$

Ou seja, $E_a |\psi\rangle$ pertence ao espaço do código. Como $E_a \in S$, então a ação do erro sobre a palavra-código é trivial ($E_a |\psi\rangle = |\psi\rangle$) e, portanto, não há perda de informação.

(iii) No último caso, a Equação (2.10) também é válida. Porém, como $E_a \notin S$, então alguma palavra-código não é fixada por E_a. Dessa forma, o efeito de E_a sobre o código $\mathcal{C}_S$ é um rearranjo dos elementos, o que causa perda de informação quântica.

O uso dos geradores permite, ainda, definir a *síndrome de erro para um código estabilizador*, a qual será dada por uma sequência binária de $(n-k)$ bits. Sejam M_i um gerador de S e E_a um erro, definimos $f_{M_i} : \mathcal{G}_n \to \mathbb{Z}_2$ por

$$f_{M_i}(E_a) = \begin{cases} 0, \text{ se } [M_i, E_a] = 0 \\ 1, \text{ se } \{M_i, E_a\} = 0 \end{cases} .$$

A síndrome do erro E_a é dada, então, por $f(E_a) = (f_{M_1}(E_a), f_{M_2}(E_a), \ldots, f_{M_{n-k}}(E_a))$. No caso de o código ser não-degenerado, cada erro possuirá uma síndrome diferente e, portanto, a medição dos $(n-k)$ geradores permitirá diagnosticar o erro completamente. Porém, se o código é degenerado, existem erros que possuem a mesma síndrome e, assim, a medição dos geradores identificará um conjunto de erros.

Para realizar a correção, o primeiro passo é medir os autovalores dos geradores do estabilizador: para cada gerador M_i, o autovalor associado será igual a $(-1)^{f_{M_i}(E_a)}$, o

que permite identificar a síndrome. Com a síndrome caracterizada, procuramos o erro, ou conjunto de erros, que a produz e realizamos a correção.

2.6.2 Distância de um código estabilizador

Para definirmos a distância de um código estabilizador iremos primeiro introduzir dois conceitos da Teoria de Grupos: *centralizador* e *normalizador*. O conjunto dos elementos de $\mathcal{G}_n$ que comutam com todos os elementos de S é o centralizador $C(S)$ de S em $\mathcal{G}_n$,

$$C(S) = \{P \in \mathcal{G}_n \mid NP = PN, \forall\, N \in S\}.$$

O normalizador $N(S)$ de S em $\mathcal{G}_n$ é o conjunto de elementos de $\mathcal{G}_n$ que fixam S sob a operação de conjugação,

$$N(S) = \{P \in \mathcal{G}_n \mid PNP^\dagger \in S, \forall\, N \in S\}.$$

Devido ao fato de que $-I \notin S$ temos que $C(S) = N(S)$. Observe que $S \subseteq N(S)$, pois S é abeliano. Considere, agora, um erro $E \in \mathcal{G}_n$ tal que $E \in N(S) - S$, ou seja, E comuta com todos os elementos do estabilizador, mas não pertence a ele. Como analisamos na Subseção 2.6.1, esse tipo de erro causa um rearranjo das palavras-código e, portanto, não é detectável. Como os elementos de $N(S)$ movem as palavas do código $\mathcal{C}_S$ dentro do próprio espaço de $\mathcal{C}_S$, tais elementos podem ser interpretados como operações codificadas.

A distância de um código estabilizador $\mathcal{C}_S$ será d, se esse for o peso do operador de menor peso que pertence a $N(S) - S$. Em outras palavras, se um código estabilizador $\mathcal{C}_S$ tem distância d, então todo $E \in \mathcal{G}_n$ com peso menor que d ou pertence ao estabilizador ou anticomuta com algum elemento dele. Se não houver elementos no estabilizador com peso menor que d, então o código é não-degenerado.

2.6.3 O código de Shor sob o formalismo estabilizador

Como exemplo de código estabilizador, revisitemos o código de Shor.

O código de nove qubits de Shor possui oito geradores no estabilizador: Z_1Z_2, Z_2Z_3, Z_4Z_5, Z_5Z_6, Z_7Z_8, Z_8Z_9, $X_1X_2X_3X_4X_5X_6$ e $X_4X_5X_6X_7X_8X_9$. É possível verificar que todo operador de peso igual a 1 do grupo de Pauli, anticomuta com algum gerador. Além disso, todos os operadores de peso 2 desse grupo anticomutam com algum gerador ou pertencem ao estabilizador do código. O mesmo não pode ser verificado para os operadores de peso 3. Por exemplo, $X_1X_2X_3$ comuta com todos os geradores, contudo, não fixa as duas palavras-código, pois $X_1X_2X_3\,|0_L\rangle = |0_L\rangle$, mas $X_1X_2X_3\,|1_L\rangle = -\,|1_L\rangle$. Portanto, a distância do código é três, confirmando que o código de Shor é $[[9, 1, 3]]$.

Os operadores X e Z codificados, denotados por $\bar{X}$ e $\bar{Z}$, respectivamente, podem ser tomados como

$$\bar{X} = Z_1Z_2Z_3Z_4Z_5Z_6Z_7Z_8Z_9,$$
$$\bar{Z} = X_1X_2X_3X_4X_5X_6X_7X_8X_9,$$

pois $\bar{X}\,|0_L\rangle = |1_L\rangle$, $\bar{X}\,|1_L\rangle = |0_L\rangle$, $\bar{Z}\,|0_L\rangle = |0_L\rangle$ e $\bar{Z}\,|1_L\rangle = -\,|1_L\rangle$.

2.7 Conceitos básicos sobre reticulados e tesselações

Para entender a construção do código tórico também é necessário discutir conceitos de reticulados e tesselações. Nesta subseção, introduziremos os dois temas.

2.7.1 Reticulados

Definição 24. *Dado um conjunto de vetores linearmente independentes, $\{v_1, v_2, \ldots, v_m\}$, em $\mathbb{R}^n$ (tal que $m \leq n$), o conjunto de pontos*

$$\Lambda = \{x = \sum_{i=1}^{m} \lambda_i v_i \mid \lambda_i \in \mathbb{Z}\}$$

é denominado reticulado de posto m e $\{v_1, v_2, \ldots, v_m\}$ é denominado base do reticulado.

Assim, um reticulado do $\mathbb{R}^n$ é um conjunto infinito de pontos dispostos regularmente nesse espaço.

Seja $\{v_1, v_2, \ldots, v_m\}$ uma base para Λ, tal que $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$, para $i = 1, 2, \ldots, m$. A matriz

$$M = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{bmatrix},$$

é chamada de *matriz geradora* do reticulado Λ. A matriz $G = MM^T$ é denominada *matriz de Gram* para o reticulado.

Dessa forma, o conjunto de pontos do reticulado Λ pode ser representado por

$$\Lambda = \{v \mid v = \lambda M, \lambda \in \mathbb{Z}\}.$$

Dois reticulados são ditos equivalentes se um puder ser obtido a partir do outro por meio de uma rotação, reflexão ou multiplicação por escalar. Isso significa que, dadas duas matrizes geradoras M e M', elas definem reticulados equivalentes, se, e somente se, [27]

$$M' = c\,UMB,$$

em que c é uma constante não nula, U é uma matriz com entradas inteiras e determinante ± 1, ou seja, uma matriz unimodular com entradas inteiras, e B é uma matriz real ortogonal (com $BB^T = I$).

Definimos, ainda, o *reticulado dual* de Λ como

$$\Lambda^* = \{x \in \mathbb{R}^n \mid x \cdot u \in \mathbb{Z}, \forall u \in \Lambda\},$$

em que $x \cdot u$ é o produto interno canônico (produto escalar) em R^n entre x e u.

2.7.2 Tesselações do plano euclidiano

Definição 25. *Um recobrimento total do plano (euclidiano ou hiperbólico) com o uso de um conjunto de polígonos, de forma que não haja sobreposições entre eles, é denominado tesselação. Em particular, se os polígonos são regulares e congruentes, tal tesselação é chamada de tesselação regular.*

Em uma tesselação regular, o número de polígonos que se encontra em cada vértice é sempre o mesmo. Denotamos por $\{p, q\}$ a tesselação regular com q polígonos de p lados se encontrando em cada vértice.

Analisemos o caso do plano euclidiano. A soma dos ângulos internos de um polígono de p lados é igual a $(p-2)\pi$, e, portanto, o ângulo interno de cada vértice do polígono é $(p-2)\pi/p$. Além disso, a soma dos ângulos internos que se encontram em um vértice deve ser 2π. Como são q polígonos que se encontram em cada vértice, então,

$$2\pi = q\frac{(p-2)\pi}{p},$$

ou, equivalentemente,

$$(p-2)(q-2) = 4.$$

Existem apenas três soluções para essa equação, dado que p e q devem ser inteiros não negativos. Logo, as três tesselações regulares do plano euclidiano são

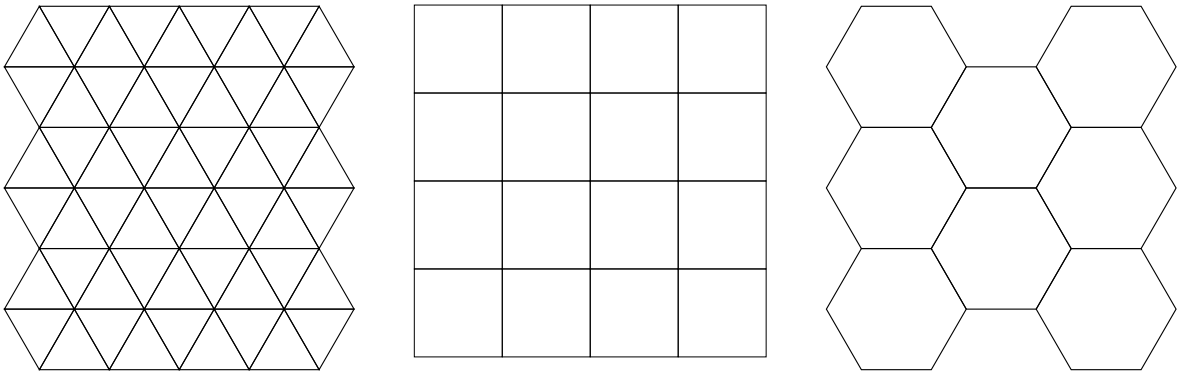
$$\{3,6\}, \{4,4\} \text{ e } \{6,3\}.$$

A Figura 2.1 apresenta essas três possíveis tesselações do plano euclidiano.

A *tesselação dual* de uma tesselação regular é obtida tomando o centro de cada polígono como um vértice e unindo os centros de polígonos adjacentes, aqueles que compartilham arestas. As tesselações triangular ($\{3,6\}$) e hexagonal ($\{6,3\}$) são a dual uma da outra. A tesselação dual da tesselação quadrada ($\{4,4\}$) é também uma tesselação quadrada, ou seja, ela é *autodual* ($p = q$).

Figura 2.1 – Tesselações regulares do plano euclidiano

(a) Tesselação {3,6} (b) Tesselação {4,4} (c) Tesselação {6,3}



Fonte: elaborado pelo autor.

2.8 Superfícies

Por fim, finalizamos esse capítulo introduzindo uma estrutura topológica de destaque para os códigos topológicos: as superfícies.

Definição 26. *Uma superfície é uma variedade bidimensional conexa.*

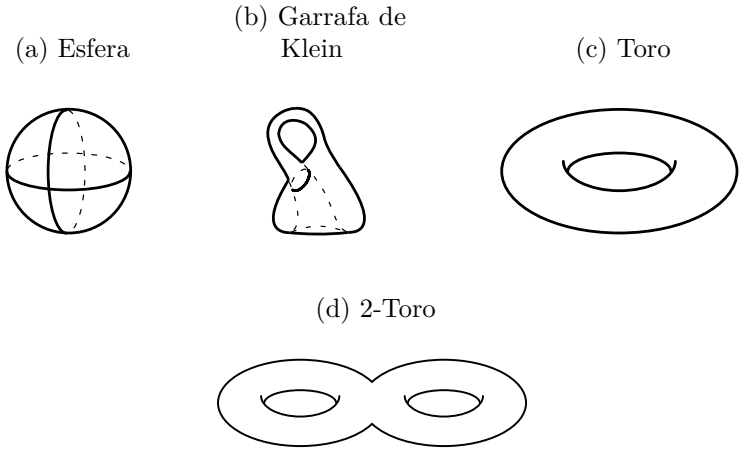
Uma variedade bidimensional é definida como a seguir.

Definição 27. *[35] Uma variedade bidimensional é um espaço de Hausdorff X com uma base contável tal que cada ponto $x \in X$ possui uma vizinhança que é homeomorfa*[2] *a um subconjunto aberto de $\mathbb{R}^2$.*

Isso significa que, localmente, uma superfície se parece com um plano euclidiano O termo conexa indica que é possível se deslocar entre dois pontos da superfície sem ser preciso realizar "saltos" durante o trajeto. A Figura 2.2 apresenta alguns exemplos de superfícies.

[2] Um homeomorfismo é uma bijeção contínua cuja inversa também é uma função contínua.

Figura 2.2 – Exemplos de superfícies



Fonte: elaborado pelo autor.

Restringiremos nossa atenção às superfícies compactas orientáveis, pois o toro, superfície utilizada na construção da classe de códigos à qual dá nome, pertence a essa classe. Podemos entender por compactas as superfícies limitadas, uma vez que podem estar contidas em um espaço de dimensão finita e não possuem um bordo de dimensão 1. Já uma superfície orientável é definida a seguir.

Definição 28. *[36](p. 122) Uma superfície regular S é orientável se for possível cobri-la com uma família de vizinhanças coordenadas, de tal modo que se um ponto $p \in S$ pertence a duas vizinhanças dessa família, então a mudança de coordenadas tem Jacobiano em p. A escolha de uma tal família é chamada uma orientação de S, e S, neste caso, diz-se orientável.*

As Figuras 2.2(a), 2.2(c) e 2.2(d), a esfera, o toro e o 2-toro, respectivamente, são exemplos de superfícies compactas orientáveis. A garrafa de Klein, Figura 2.2(b), é uma superfície compacta, mas não orientável. Como exemplo de superfície não limitada podemos pensar no plano infinito.

A Topologia classifica as superfícies compactas orientáveis por meio de seu *gênero*, g, que pode ser entendido como o número de "furos", ou "buracos", que essa superfície possui. Por exemplo, uma esfera é uma superfície que possui $g = 0$, um toro é uma superfície com $g = 1$ e um 2-toro é uma superfície com $g = 2$. De modo geral, um n-toro é uma superfície com $g = n$.

Dadas duas superfícies, dizemos que são homeomorfas se uma delas puder ser continuamente deformada na outra. Com isso, queremos dizer que podemos dobrar, alongar e comprimir, mas não podemos cortar ou colar pontos da superfície. Por exemplo, podemos deformar continuamente uma caneca até que ela se torne um toro, ou seja, essas superfícies são homeomorfas. Entretanto, é impossível transformar uma esfera em um toro

sem a realização de um corte dizemos, então, que a esfera e o toro são superfícies não homeomorfas.

Dado uma tesselação de uma superfície compacta orientável, existe uma relação entre os números de vértices V, arestas E e faces F e o gênero g da superfície, a *característica de Euler* χ. Definimos χ por

$$\chi = V - E + F. \tag{2.11}$$

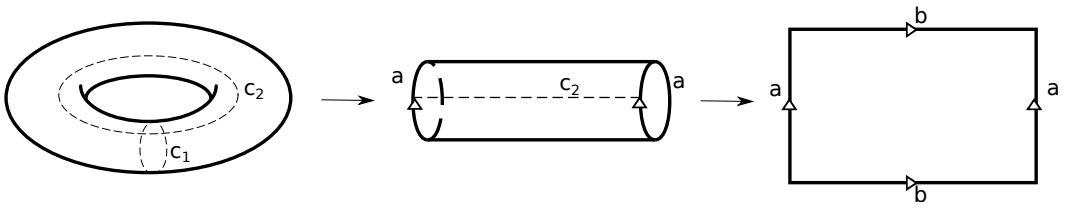
Porém, é possível expressar χ também em função do gênero da superfície,

$$\chi = 2(1 - g). \tag{2.12}$$

A característica de Euler é um *invariante topológico*, uma estrutura matemática que depende apenas da topologia do objeto sobre o qual é construído. Se os invariantes topológicos de dois objetos diferem, é possível afirmar que tais objetos não são homeomorfos.

É possível representar uma superfície com o uso do *modelo planar*, que associa um polígono plano a uma superfície por intermédio do emparelhamento de suas arestas de acordo com certas condições. A Figura 2.3 mostra como chegar ao modelo planar do toro a partir do seu modelo espacial. Isso é obtido por meio de operações de "corte" da superfície, representadas na imagem pelas curvas c_1 e c_2. Inversamente, é possível sair do modelo planar e chegar na superfície correspondente, agora utilizando operações de "colagem", em que unem-se arestas com a mesma identificação, a ou b na figura, de acordo com as orientações indicadas pelas setas presentes nas arestas.

Figura 2.3 – Modelo planar do toro



Fonte: elaborado pelo autor.

3 O código tórico

Neste capítulo, tratamos do principal objeto de estudo desta dissertação: o código tórico. Apresentado por Kitaev [24] em 1997, ele abriu caminho para uma nova classe, os denominados códigos quânticos topológicos.

Devido a sua formulação simples, mas extremamente elegante, esse código permite ser entendido por mais de um prisma. Uma das possibilidades, é compreendê-lo sob a ótica dos *anyons*, quasipartículas sub-atômicas com propriedades bem particulares que surgem no reticulado do código a partir do acontecimento de erros. Os estados em que serão codificadas as informações, a identificação da existência de erros e o retorno de estados para o espaço do código são exemplos do que pode ser compreendido por meio das interações entre essas partículas.

Outra forma de entender o funcionamento do código tórico é por intermédio da principal estrutura matemática associada ao mesmo, a topologia. Definido sobre uma tesselação do toro, nesse código os erros podem ser relacionados a conjuntos de arestas da tesselação e, com o uso da teoria de homologia, bordos e ciclos nortearão o funcionamento dele. Além disso, por se tratar de um código estabilizador, todas as características e propriedades dessa classe de códigos valem para tal código.

Nas próximas páginas, definiremos o código tórico, tanto do ponto de vista da topologia quanto dos *anyons*. No geral, usaremos uma tesselação quadrada do toro, a mais usual e que permite uma codificação simétrica para os erros *bit-flip* e *phase-shift*, mas também faremos nota do funcionamento do código em uma tesselação triangular/hexagonal. Apresentaremos, ainda, o código planar, cuja estrutura é similar ao código tórico, mas com características espaciais de construção diferentes. Por fim, um exemplo de código tórico quadrado com 50 qubits será exposto, explicitando os elementos e estruturas apontados ao longo do capítulo.

3.1 Códigos estabilizadores locais

Entre as propriedades potenciais que os códigos quânticos podem possuir, a *localidade* merece destaque, pois facilita a implementação física potencial dos sistemas [37]. De forma simplificada, um código estabilizador é dito local quando cada gerador do seu estabilizador atua em apenas um pequeno conjunto de qubits próximos. A localidade normalmente implicará que os qubits físicos do código estejam localizados sobre um reticulado e que as interações ocorram apenas entre os qubits próximos [38]. Enquanto os códigos topológicos são naturalmente locais, devido à sua construção, os códigos

estabilizadores são, em sua maioria, não locais. Se o suporte de cada gerador do estabilizador, ou seja, os qubits sobre os quais cada estabilizador atua, é limitado a n qubits, então dizemos que o código é n-local.

Definição 29. *[38] Uma família de códigos estabilizadores é local se for possível escolher os geradores do seu estabilizador de forma que:*

1. *O número de qubits no suporte de cada gerador seja limitado;*
2. *o número de elementos do estabilizador que suporta cada qubit seja limitado;*
3. *a família contenha códigos de distâncias arbitrariamente grandes.*

De forma natural, é possível definir geometricamente o conceito de localidade em códigos.

Definição 30. *[38] Uma família de códigos quânticos será local em D dimensões se:*

1. *Os qubits estiverem localizados em uma array de dimensão D;*
2. *os qubits fixados por cada estabilizador estiverem contidos em um hipercubo de tamanho limitado;*
3. *a família contiver códigos de distâncias arbitrariamente grandes.*

Em que *arrays* são arranjos de elementos de forma sistemática em D dimensões. Por exemplo, em dimensão 1 essas estruturas são equivalentes a vetores, enquanto em duas, são equivalentes a matrizes. Uma família de códigos que seja local no sentido geométrico, também o será no sentido geral [38]. Como código topológico, o código tórico é local. Em sua definição, os qubits no suporte de cada gerador são explicitados espacialmente. Como consequência, a extração de sua síndrome também será local. Veremos essas condições em detalhes nas duas próximas seções.

3.2 O código tórico sob uma abordagem topológica

A fim de seguir com a abordagem topológica, é preciso buscar o ponto de interseção entre os temas de reticulados, tesselações, superfícies e homologia. Como resultado, temos a homologia de curvas, que será apresentada na próxima subseção, após uma introdução à homologia.

3.2.1 Homologia de curvas

O entendimento da codificação de qubits com o uso dos *códigos tóricos* passa por conceitos introdutórios da *Teoria de Homologia e Co-homologia*, tais como: p-cadeias, p-ciclos e bordos, além de seus equivalentes co-homólogos. Esta subseção tem o objetivo de introduzir esses conceitos. Sugerimos ao leitor a consulta de [39] para se aprofundar no assunto.

Um módulo é uma estrutura algébrica análoga a um espaço vetorial [40], porém a multiplicação por um escalar não acontece sobre um corpo, mas sim sobre um anel.

Definição 31. *Um grupo abeliano aditivo M dotado de uma multiplicação por escalar*

$$\begin{aligned} A \times M &\longrightarrow M \\ (a, m) &\longmapsto a \cdot m \end{aligned}$$

é dito um A-módulo se satisfizer os axiomas abaixo, $\forall a_1, a_2 \in A$ e $\forall m_1, m_2 \in M$:

- $1 \cdot m_1 = m_1$;
- $(a_1 a_2) \cdot m_1 = a_1(a_2 \cdot m_1)$;
- $(a_1 + a_2) \cdot m_1 = a_1 \cdot m_1 + a_2 \cdot m_1$;
- $a_1 \cdot (m_1 + m_2) = a_1 \cdot m_1 + a_1 \cdot m_2$.

Dados $a \in A$ e $m \in M$, também utilizamos a notação am para representar o elemento $a \cdot m \in M$.

Definição 32. *Seja A um anel comutativo com unidade. Um complexo de cadeias com coeficientes em A é uma sequência $\mathcal{C} = (\mathbf{C}_p, \partial_p)$, com p inteiro não negativo. Os elementos $\mathbf{C}_p$ são A-módulos e $\partial_p : \mathbf{C}_p \to \mathbf{C}_{p-1}$ são homomorfismos tais que $\partial_p \circ \partial_{p+1} = 0$. Escreve-se:*

$$\mathcal{C} : \cdots \to \mathbf{C}_{p+1} \xrightarrow{\partial_{p+1}} \mathbf{C}_p \xrightarrow{\partial_p} \mathbf{C}_{p-1} \xrightarrow{\partial_{p-1}} \cdots \to \mathbf{C}_1 \xrightarrow{\partial_1} \mathbf{C}_0 \xrightarrow{\partial_0} 0. \tag{3.1}$$

Os elementos de $\mathbf{C}_p$ são denominados *p-cadeias* ou *cadeias de dimensão p*. Se $x \in \mathbf{C}_p$ é tal que $\partial_p x = 0$, então dizemos que x é um *p-ciclo*, ou apenas que é um *ciclo*. Denotamos por $\mathbf{Z}_p$ o conjunto de todos os ciclos de $\mathbf{C}_p$. De fato, $\mathbf{Z}_p$ é o núcleo do homomorfismo $\partial_p : \mathbf{C}_p \to \mathbf{C}_{p-1}$.

Dados $y \in \mathbf{C}_p$ e $x \in \mathbf{C}_{p+1}$, se $y = \partial_{p+1} x$, então dizemos que a p-cadeia y é o *bordo* da $(p+1)$-cadeia x. Denotamos por $\mathbf{B}_p$ o conjunto de todas as p-cadeias que são bordo de alguma $(p+1)$-cadeia. Logo, $\mathbf{B}_p$ é a imagem do homomorfismo $\partial_{p+1} : \mathbf{C}_{p+1} \to \mathbf{C}_p$.

Quando não há risco de confusão, pode-se escrever ∂ em vez de ∂_p. Com isso, a relação $\partial_p \partial_{p-1} = 0$ é reescrita como $\partial\partial x = 0$, para toda cadeia $x \in \mathbf{C}_p$.

A relação $\partial_p \circ \partial_{p+1} = 0$ implica que todo bordo é também um ciclo, e, portanto, $\mathbf{B}_p \subset \mathbf{Z}_p$. O A-módulo quociente $\mathbf{H}_p = \mathbf{H}_p(\mathcal{C}) = \mathbf{Z}_p/\mathbf{B}_p$ é chamado de *grupo de homologia de dimensão p* do complexo $\mathcal{C}$, cujos elementos são as *classes de homologia*, definidas por

$$[z] = z + \mathbf{B}_p = \{z + \partial x; x \in \mathbf{C}_{p+1}\},$$

onde $z \in \mathbf{Z}_p$.

Dois ciclos p-dimensionais, z e z', possuem a mesma classe de homologia, $[z] = [z']$, se, e somente se, $z - z' = \partial x$, para algum $x \in \mathbf{C}_{p+1}$. Nesse caso, dizemos que z e z' são *ciclos homólogos*.

Definição 33. *Um complexo de cocadeias é uma sequência $\mathcal{C} = (\mathbf{C}^p, \delta_p)$, com p inteiro não negativo. Os elementos $\mathbf{C}^p$ são A-módulos e $\delta_p : \mathbf{C}^p \to \mathbf{C}^{p+1}$ são homomorfismos tais que $\delta_{p+1} \circ \delta_p = 0$. Escreve-se:*

$$\mathcal{C} : \mathbf{C}^0 \xrightarrow{\delta_0} \mathbf{C}^1 \xrightarrow{\delta_1} \cdots \xrightarrow{\delta_{p-2}} \mathbf{C}^{p-1} \xrightarrow{\delta_{p-1}} \mathbf{C}^p \xrightarrow{\delta_p} \mathbf{C}^{p+1} \xrightarrow{\delta_{p+1}} \cdots .$$

Os elementos de $\mathbf{C}^p$ são chamados *cocadeias* de dimensão p, ou apenas *p-cocadeias*. Se $\delta_p x = 0$, então dizemos que x é um *p-cociclo*. O conjunto de todos os p-coclicos é denotado por $\mathbf{Z}^p$ e é o núcleo do homomorfismo $\delta^p : \mathbf{C}^p \to \mathbf{C}^{p+1}$. A imagem do operador $\delta_{p-1} : \mathbf{C}^{p-1} \to \mathbf{C}^p$ é o conjunto $\mathbf{B}^p$. A relação $\delta_p \delta_{p-1} = 0$ implica que $\mathbf{B}^p \subset \mathbf{Z}^p$. Assim como acontece com o operador ∂, quando não há risco de confusão, podemos escrever δ no lugar de δ_p.

O A-módulo quociente $\mathbf{H}^p(\mathcal{C}) = \mathbf{Z}^p/\mathbf{B}^p$ é chamado de *grupo de cohomologia de dimensão p* do complexo $\mathcal{C}$. Os elementos de $\mathbf{H}^p(\mathcal{C})$, as *classes de cohomologia*, são definidas por

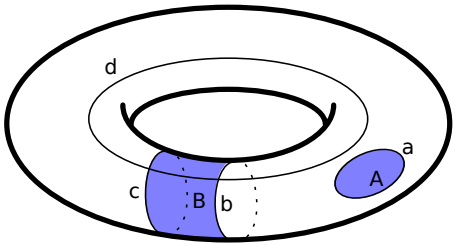
$$[z] = z + \mathbf{B}^p = \{z + \delta x; x \in \mathbf{C}^{p-1}\},$$

onde z é um cociclo de $\mathbf{C}^p$. Dados dois cociclos $z, z' \in \mathbf{C}^p$, com $[z] = [z']$, se, e somente se, $z - z' = \delta x$, para algum $x \in \mathbf{C}^{p-1}$. Nesse caso, dizemos que z e z' são *ciclos cohomólogos*.

Considere o toro exibido na Figura 3.1 e as curvas a, b, c e d construídas sobre ele. A curva a é, individualmente, o bordo da região A e, por isso, é possível deformá-la continuamente até que encolha a um único ponto. Em outras palavras, diz-se que a curva a é *homologicamente trivial*. Por outro lado, as curvas b, c e d não conseguem sozinhas delimitar o bordo de nenhuma região. Diz-se que elas são *homologicamente não triviais*. Por outro lado, se as curvas b e c forem consideradas de forma conjunta, vê-se que elas

delimitam a região B. É dito, então, que b e c são *homologicamente equivalentes*. Entretanto, não há região do toro em que o bordo seja formado apenas pelas curvas b e d, por isso essas curvas não são homologicamente equivalentes. Por meio dessa análise simplificada, acabamos de dividir as curvas sobre o toro em diferentes classes de homologia.

Figura 3.1 – Classes homológicas das curvas sobre um toro



Fonte: adaptado de Bombin [38]

A representação de reticulados e tesselações sobre uma superfície torna-se mais fácil se for utilizado o modelo planar para sua representação. Devido ao modelo planar do toro ser equivalente a um retângulo, ele possui as mesmas três tesselações regulares que o plano euclidiano, ou seja, $\{3,6\}$, $\{4,4\}$ e $\{6,3\}$. A Figura 3.2 exemplifica as tesselações de um toro no modelo planar.

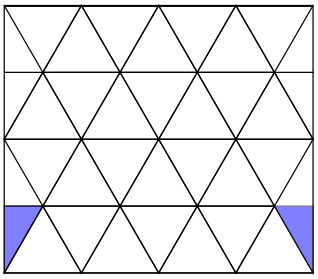
Na figura, não explicitamos as arestas do modelo planar que devem ser coladas, nem a direção dessas colagens, porém, implicitamente, essa colagem segue a estrutura apresentada na Figura 2.3. Com isso, um polígono da tesselação que for cortado por uma borda do modelo planar continua na aresta oposta. Por exemplo, nas Figuras 3.2(a) e 3.2(c), os polígonos cortados destacados em azul representam um só objeto. Além disso, haverá periodicidade entre as arestas e vértices sobre os bordos do modelo planar. Assim, por exemplo, na Figura 3.2(b), as duas arestas destacadas em azul representam um mesmo elemento.

A partir daqui, utilizaremos o reticulado quadrado, ou seja, um reticulado equivalente à tesselação quadrada ($\{4,4\}$), para apresentar o código tórico. Existem algumas justificativas para essa escolha: 1) esse foi o reticulado utilizado por Kitaev quando apresentou o código tórico; 2) o código tórico sob o reticulado quadrado gera um código simétrico em relação a suas distâncias para erros do tipo X e Z; e 3) fixar o tipo de reticulado, ao invés de utilizar uma abordagem generalizada, permite continuar com o uso de representações visuais, o que facilitará o entendimento.

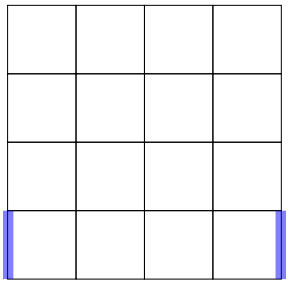
Agora, considere um reticulado quadrado do toro com V vértices, E arestas e F faces. Denomina-se cada vértice, aresta e face por 0-célula, 1-célula e 2-célula, respectivamente. É possível atribuir uma estrutura de grupo abeliano aos conjuntos $\mathbf{C}_i$ das i-células,

Figura 3.2 – Tesselações regulares do toro

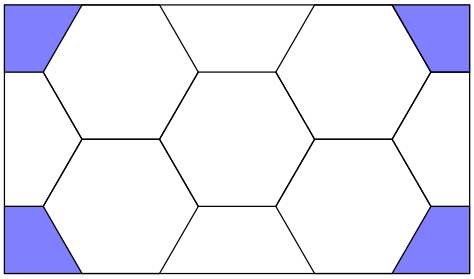
(a) Tesselação {3,6} do toro



(b) Tesselação {4,4} do toro



(c) Tesselação {6,3} do toro



Fonte: elaborado pelo autor.

com $i = 0, 1, 2$. Por exemplo, dado o conjunto $\mathbf{C}_1$ das 1-células, inicialmente rotulamos as arestas como $\{e_i\}_1^E$, ou seja, cada uma das E arestas é enumerada e, com isso, a i-ésima aresta recebe o rótulo e_i. Com isso, é possível representar qualquer subconjunto de arestas E' como uma soma formal de elementos de e_i,

$$c = \sum_{i=1}^{E} c_i e_i, \quad c_i = \begin{cases} 0 \text{ se } e_i \notin E' \\ 1 \text{ se } e_i \in E' \end{cases} . \tag{3.2}$$

A soma c é dita uma 1-cadeia. Definindo-se que $e_i + e_i = 0$, para $i = 1, 2, \ldots, E$, então a operação $+$ é uma união disjunta e o resultado da soma de duas 1-cadeias também será uma 1-cadeia. Desse modo, uma estrutura de grupo abeliano foi atribuída ao conjunto $\mathbf{C}_1$ das 1-células, onde o elemento neutro desse grupo corresponderá ao conjunto vazio, a ausência de todos os elementos e_i. Naturalmente, é possível representar uma 1-cadeia c por um vetor binário de dimensão E, cuja i-ésima coordenada seja igual a 1, se a aresta pertencer a c, ou igual a 0 no caso contrário. Segue daí que $\mathbf{C}_1 \equiv \mathbb{Z}_2^E$. Da mesma forma, é possível atribuir estruturas de grupos abelianos aos conjuntos $\mathbf{C}_0$ e a $\mathbf{C}_2$, sendo $\mathbf{C}_0 \equiv \mathbb{Z}_2^V$ e $\mathbf{C}_2 \equiv \mathbb{Z}_2^F$, cujos elementos são chamados, respectivamente, 0-cadeias e 2-cadeias.

Pode-se encarar $\mathbf{C}_2, \mathbf{C}_1$ e $\mathbf{C}_0$ como os últimos elementos de um complexo de cadeias da forma da Equação (3.1). Para isso, basta definir uma família de homomorfismos entre os $\mathbb{Z}_2$-módulos. Os homomorfismos ∂, denotados por operadores bordo, são do tipo

$$\partial_2 : \mathbf{C}_2 \to \mathbf{C}_1, \qquad \partial_1 : \mathbf{C}_1 \to \mathbf{C}_0.$$

Geometricamente, os operadores ∂ levam os objetos em seus bordos, ou seja, em suas fronteiras.

Assim, para uma face f do reticulado quadrado, delimitada pelas arestas $\{e'_1, e'_2, e'_3, e'_4\}$, vale que

$$\partial_2 f = e'_1 + e'_2 + e'_3 + e'_4.$$

No caso de um conjunto de faces $F' = \{f'_1, f'_2, \ldots, f'_l\}$, a região F' pode ser expressa pela 2-cadeia $r = f'_1 + f'_2 + \cdots + f'_l$, enquanto o bordo de r deve satisfazer

$$\partial_2 r = \partial_2 f'_1 + \partial_2 f'_2 + \cdots + \partial_2 f'_l. \tag{3.3}$$

Como foi definido que $e_i + e_i = 0$, então as arestas que forem compartilhadas por duas faces em F' se anulam, de forma que $\partial_2 r = e''_1 + e''_2 + \cdots e''_m$, onde $\{e''_1, e''_2, \ldots e''_m\}$ são as arestas que limitam a região F'. Para uma aresta e, vale que

$$\partial_1 e = v_1 + v_2,$$

em que v_1 e v_2 são os vértices das suas extremidades. Para um conjunto de arestas, um vértice fará parte do seu bordo se, e somente se, um número ímpar de arestas do conjunto partir dele. Em geral, quando não houver necessidade de especificar, os dois operadores bordo serão referidos apenas por ∂, sem subíndice. A Figura 3.3(a) exemplifica a atuação do operador bordo sobre elementos simples, enquanto a 3.3(b) mostra tal atuação sobre conjuntos de elementos.

Como visto, dois subconjuntos importantes do conjunto das p-cadeias $\mathbf{C}_p$ são $\mathbf{Z}_p$ e $\mathbf{B}_p$. Em particular, os subconjuntos de 1-cadeias $\mathbf{Z}_1, \mathbf{B}_1 \subset \mathbf{C}_1$ são de interesse para o código tórico. Relembrando, o conjunto $\mathbf{Z}_1$ é o grupo das 1-cadeias z que não possuem bordo, ou seja $\partial_1 z = 0$, e os elementos de $\mathbf{Z}_1$ são chamados de ciclos. O grupo $\mathbf{B}_1$ é a imagem do operador ∂_2, ou seja, é o conjunto de todas as 1-cadeias b que são o bordo de alguma 2-cadeia, explicitamente, $b = \partial_2 c$, para algum $c \in \mathbf{C}_2$. Sabe-se que $\mathbf{B}_1 \subset \mathbf{Z}_1$, ou seja, todos os bordos são ciclos, pois $\partial_1 \partial_2 = 0$.

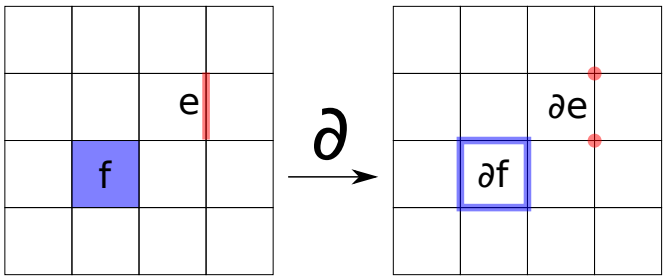
Define-se o grupo de homologia de dimensão 1 de uma superfície $\mathbf{H}_1$ como o quociente

$$\mathbf{H}_1 = \mathbf{Z}_1 / \mathbf{B}_1.$$

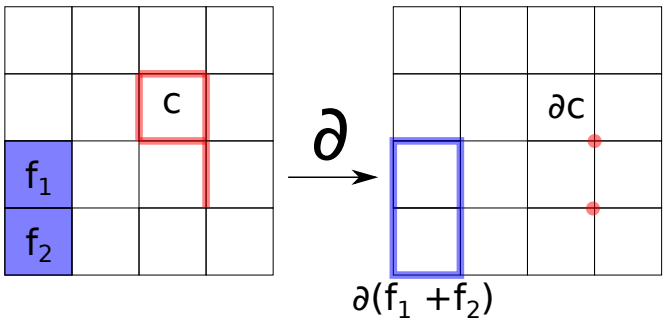
Os elementos $\mathbf{H}_1$ são as classes laterais da forma $\overline{z} := \{z + b | b \in \mathbf{B}_1\}$. A adição em $\mathbf{H}_1$ é inerente a $\mathbf{Z}_1$, $\overline{z} + \overline{z'} = \overline{z + z'}$, e o elemento neutro é a classe $\overline{0} = \mathbf{B}_1$. O grupo $\mathbf{H}_1$ depende apenas da topologia da superfície (relacionado à quantidade de classes de ciclos homologicamente não triviais), de forma que [38]

Figura 3.3 – O operador bordo ∂

(a) Bordo de elementos simples



(b) Bordo de conjunto de elementos



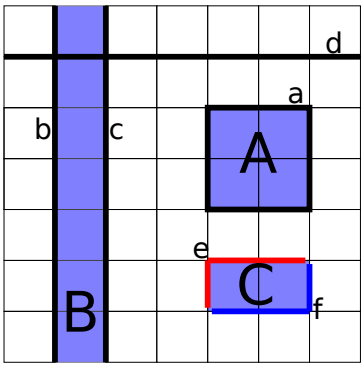
Fonte: elaborado pelo autor.

$$\mathbf{H}_1 \equiv \mathbb{Z}_2^{2g}.$$

Assim, para uma esfera, temos que $g = 0$, o que implica em $\mathbf{H}_1 \equiv \mathbb{Z}_2^0$ e, por isso, só há uma classe lateral, $\overline{0} = \mathbf{B}_1$. Consequentemente, todos os ciclos são triviais em uma esfera, ou seja, todo ciclo é fronteira de alguma região e, portanto, $\mathbf{B}_1 = Z_1$. Para o toro, $g = 1$, $\mathbf{H}_1 = \mathbb{Z}_2 \times \mathbb{Z}_2$. Logo, o grupo de homologia do toro possui 2 geradores e 4 elementos. Por exemplo, considerando a tesselação quadrada do toro e as curvas destacadas na Figura 3.4, temos que $\overline{a} = \overline{0}, \overline{b} = \overline{c} \neq \overline{0}$. Tomando $\overline{b}$ e $\overline{d}$ como geradores, os elementos de $\mathbf{H}_1$ são $\overline{0}, \overline{b}, \overline{d}$ e $\overline{b+d}$.

Também é possível particionar o grupo das 1-cadeias $\mathbf{C}_1$ em classes homológicas. O quociente $\mathbf{C}_1/\mathbf{B}_1$ possui elementos da forma $\overline{c} = \{c + b | b \in \mathbf{B}_1\}$, em que $c \in \mathbf{C}_1$. Dessa forma, duas 1-cadeias são homologicamente equivalentes se juntas elas formarem o bordo de uma região. Na Figura 3.4, a curva b é equivalente à c, pois formam juntas o bordo da região B, enquanto as curvas e e f também são equivalentes, pois formam o bordo da região C (as curvas e e f foram destacadas em cores diferentes a fim de facilitar a visualização, já que têm suas extremidades nos mesmos vértices). As curvas a, b, c e d são ciclos. Como a curva a é individualmente a fronteira da região A, dizemos que a é um ciclo trivial. Porém, como b não é individualmente a fronteira de nenhuma região, então dizemos que b é um ciclo não trivial. O mesmo vale para c e d.

Figura 3.4 – Classes homológicas de um toro com reticulado quadrado



Fonte: adaptado de Bonbim [38].

3.2.2 Definições iniciais

Nos códigos tóricos, dado um reticulado da superfície, um qubit é associado a cada uma das arestas desse reticulado. Com isso, cada elemento da base computacional pode ser interpretado como uma 1-cadeia $c \in \mathbf{C}_1$:

$$|c\rangle := \bigotimes_i |c_i\rangle\,, \qquad c \in \mathbf{C}_1,$$

onde o índice i percorre todos as arestas do reticulado, e os coeficientes c_i são os mesmos da Equação (3.2).

Os produtos tensoriais de operadores X e Z também podem ser rotulados com 1-cadeias,

$$X_c := \bigotimes_i X_i^{c_i}, \quad Z_c := \bigotimes_i Z_i^{c_i}, \quad c \in \mathbf{C}_1, \tag{3.4}$$

onde o índice i indica a aresta e_i em que a porta X, ou Z, atua. Vale que $X_i^1 = X_i, Z_i^1 = Z_i$ e $X_i^0 = Z_i^0 = I_i$. Perceba que, para essa notação, dados $c, c' \in \mathbf{C}_1$ vale para o produto de operadores que

$$X_c \cdot X_{c'} = X_{c+c'}, \quad Z_c \cdot Z_{c'} = Z_{c+c'}. \tag{3.5}$$

3.2.3 O reticulado dual

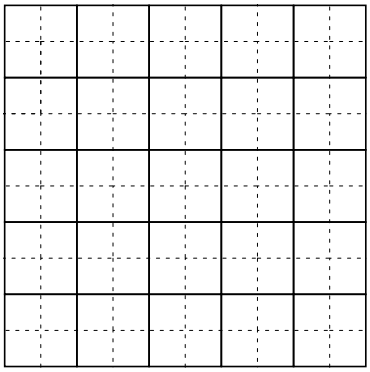
Dada uma tesselação quadrada do toro, considere a sua tesselação dual e o reticulado associado a ela. Apesar de não ser a definição de reticulado dual apresentada na Seção 2.7, faremos referência a ele como um reticulado dual. Assim, em alguns momentos do texto, os dois termos, reticulado e tesselação (dual), serão utilizados com o mesmo significado. Como visto na Seção 2.7.2, na construção da tesselação dual:

- os vértices da tesselação primal são levados em faces da tesselação dual;

- as faces da tesselação primal, em vértices da tesselação dual;
- as arestas da tesselação primal, em arestas duais.

A tesselação quadrada é autodual, ou seja, a sua tesselação dual é também uma tesselação quadrada. A Figura 3.5 ilustra uma tesselação quadrada do toro (em linha contínua) e sua tesselação dual relacionada (em linha tracejada).

Figura 3.5 – Tesselação quadrada do toro e respectiva tesselação dual



Fonte: elaborado pelo autor.

Será utilizado o símbolo $*$ para denotar os elementos duais. Assim, f^*, e^* e v^* representam, respectivamente, o vértice dual, a aresta dual e a face dual associados aos elementos f, e e v do reticulado primal. O elemento dual equivalente às cadeias é as cocadeias, ou cadeias duais. Denotam-se os conjuntos das 0-cadeias duais, das 1-cadeias duais e das 2-cadeias duais por $\mathbf{C}^0, \mathbf{C}^1$ e $\mathbf{C}^2$, respectivamente. Em alguns momentos, quando não houver necessidade de especificação, chamaremos tanto as 1-cadeias quanto as 1-cadeias duais apenas de cadeias, na tentativa de tornar o texto menos cansativo.

É necessária certa atenção com a notação dual, uma vez que há uma inversão nos índices utilizados no reticulado primal. Por exemplo, o conjunto $\mathbf{C}^0$ corresponde às faces duais, sendo que no reticulado primal $\mathbf{C}_0$ corresponde aos vértices.

Assim como no reticulado primal foram definidos os operadores bordo, no reticulado dual se definem os operadores bordo duais δ,

$$\delta_0 : \mathbf{C}^0 \to \mathbf{C}^1, \qquad \delta_1 : \mathbf{C}^1 \to \mathbf{C}^2,$$

atuando em cadeias duais. Os operadores δ têm ação equivalente aos operadores ∂, levando cada cocadeia em seu cobordo. Os operadores bordo dual δ produzem os grupos de ciclos duais $\mathbf{Z}^1$ e de bordos duais $\mathbf{B}^1$, além do grupo de homologia dual $\mathbf{H}^1 = \mathbf{Z}^1/\mathbf{B}^1$.

Os operadores bordo e bordo dual podem ser relacionados explicitamente pelas seguintes relações

$$v \in \partial_1 e \iff e^* \in \delta_0 v^*, \qquad e \in \partial_2 f \iff f^* \in \delta_1 e^*, \tag{3.6}$$

onde $\partial_1 e$ e $\partial_2 f$ são interpretados como conjuntos de vértices e arestas, respectivamente, e $\delta_1 e^*$ e $\delta_0 v^*$ como conjuntos de vértices duais e arestas duais, nessa ordem.

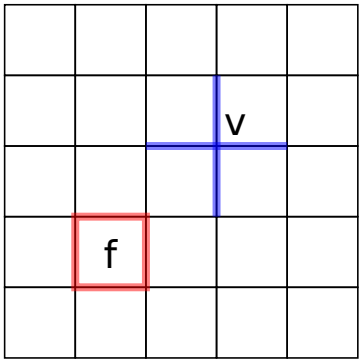
3.2.4 O grupo estabilizador do código tórico e seus geradores

Definição 34. *Considere um vértice v e uma face f de um reticulado quadrado de um toro. Definimos os operadores vértice e face (também conhecido como plaquete), respectivamente, por*

$$A(v) := \prod_{e|v\in\partial_1 e} X_e \qquad e \qquad B(f) := \prod_{e\in\partial_2 f} Z_e. \tag{3.7}$$

Em outros termos, $A(v)$ é o produto tensorial de operadores X atuando sobre as arestas que possuem v como vértice comum e operadores identidade agindo nas demais arestas. Já o operador $B(f)$ é o produto tensorial de operadores Z atuando sobre as arestas que são o bordo de f e operadores identidade nas demais arestas. Segundo as definições da Seção 3.1, o código tórico quadrado é 4-local. A Figura 3.6 representa operadores $A(v)$'s e $B(f)$'s para um reticulado quadrado do toro. Nessa figura representamos qubits (arestas) em que atuam as portas X e Z. Ao longo do texto, buscaremos manter esse esquema de cores.

Figura 3.6 – Operadores vértice e face

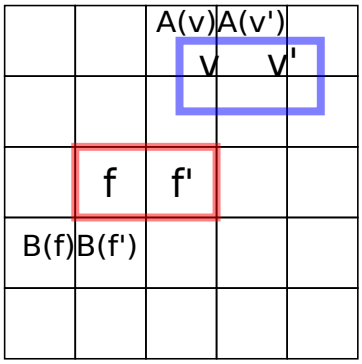


Fonte: elaborado pelo autor.

Um fato interessante é que um operador face no reticulado primal corresponde a um operador vértice (de portas Z) no reticulado dual. Da mesma forma, operadores vértice no reticulado primal correspondem a operadores face (de portas X) no reticulado dual.

Como $Z^2 = I$, então o produto de dois operadores face adjacentes resultará em uma cadeia de operadores Z englobando as duas faces, pois as operações Z sobre à qual pertencem as duas faces se cancelam. Assim, todo $1-$ciclo de operadores Z pode ser decomposto em operadores face. O mesmo vale para $1-$cociclos de operações X e operadores vértice, basta observar o reticulado dual. Por exemplo, considere as faces vizinhas f e f' na Figura 3.7. O ciclo de rotações Z que engloba as duas faces pode ser escrito como $B(f)B(f')$. Por sua vez, o produto de operadores vértice $A(v)A(v')$ corresponde a um ciclo no reticulado dual que engloba as faces duais v^* e v'^*.

Figura 3.7 – Ciclos $A(v)A(v')$ e $B(f)B(f')$



Fonte: adaptado de Pachos [16].

Como $A(v)^2 = B(f)^2 = I$, então os autovalores associados a esses operadores são ± 1, consequência direta da Afirmação 1, presente no Capítulo 2.

Utilizando a notação da Equações (3.4) e (3.6), esses operadores podem ser expressos como

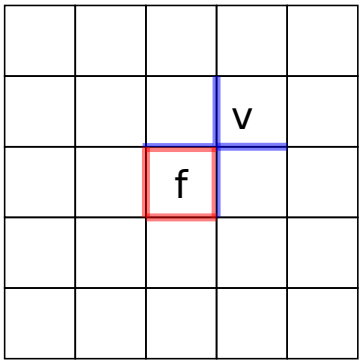
$$A(v) = X_{e|v\in\partial e} \qquad \text{e} \qquad B(f) = Z_{\partial f}. \tag{3.8}$$

Afirmação 3. *Todos os operadores face e vértice comutam.*

Demonstração. Naturalmente, os operadores $A(v)$ comutam uns com os outros, já que são o produto tensorial de operadores X's e I's. Da mesma forma, temos que os operadores $B(f)$ comutam entre si. Além disso, os operadores vértice e face também comutam entre si, pois, apesar dos operadores X e Z anticomutarem quando são aplicados sobre um mesmo qubit, temos que um operador $A(v)$ e um operador $B(f)$, ou não compartilham aresta nenhuma, como na Figura 3.6, ou compartilham duas arestas ($v \in \partial e$, para algum $e \in \partial f$), como na Figura 3.8, e, portanto, nos dois casos eles comutam. □

Tem-se, então, que os operadores vértice e face geram um subgrupo abeliano de $\mathcal{G}_n$. Assim, podemos tomar os operadores vértice e face como geradores do código tórico e, portanto, será também um código estabilizador.

Figura 3.8 – Operadores vértice e face que compartilham arestas



Fonte: elaborado pelo autor.

Perceba, entretanto, que os operadores vértice e face não são todos independentes, pois

$$\prod_f A(v) = I; \tag{3.9}$$

$$\prod_v B(f) = I. \tag{3.10}$$

Essas igualdades independem do reticulado utilizado, pois se originam da definição dos operadores face e vértice. Existem, então, $V + F - 2$ geradores independentes, uma vez que existam V operadores vértice, F operadores face e duas condições, Equações (3.9) e (3.10). Consideraremos os seguintes fatos:

- o número de qubits físicos n é igual ao número de arestas do reticulado E;
- um reticulado qualquer do toro deve satisfazer à característica de Euler, Equações (2.11) e (2.12);
- como código estabilizador, o código tórico deve satisfazer a Afirmação 2, ou seja, se o estabilizador possui $n - k$ geradores, então o espaço do código tem dimensão 2^k.

Assim, conclui-se que o número de qubits codificados por um código tórico, independentemente do tipo e do tamanho do reticulado, é igual a

$$\begin{aligned} k =& E - (V + F - 2) \\ =& 2 - (V - E + F) \\ =& 2 - 2(1 - g) \\ =& 2 - 2(1 - 1) \\ =& 2. \end{aligned} \tag{3.11}$$

3.2.5 O hamiltoniano de um código tórico

O hamiltoniano H de um sistema quântico é um operador que corresponde à energia total do sistema. Para o código tórico, o hamiltoniano associado corresponde ao somatório dos operadores face e vértice:

$$H = -\sum_v A(v) - \sum_f B(f).$$

Um dos estados fundamentais desse hamiltoniano, que corresponde aos estados de menor energia do sistema, é dado por

$$\begin{aligned} |\xi\rangle =& \prod_v \frac{1}{\sqrt{2}}(I + A(v)) \prod_f \frac{1}{\sqrt{2}}(I + B(f)) \,|00\ldots0\rangle \\ =& \frac{1}{\sqrt{2}} \prod_v (I + A(v)) \,|00\ldots0\rangle \,. \end{aligned} \tag{3.12}$$

A segunda igualdade vale porque $Z\,|0\rangle = |0\rangle$. De fato, $|\xi\rangle$ é um autovetor de todos os operadores $A(v)$ e $B(f)$, associado ao autovalor 1, portanto é uma palavra-código. Para verificar que $|\xi\rangle$ é uma palavra-código, considere quaisquer geradores $A(v')$ e $B(f')$. Vale, então, que

$$\begin{aligned} A(v')\,|\xi\rangle &= A(v')\frac{1}{\sqrt{2}} \prod_v (I + A(v)) \,|00\ldots0\rangle \\ &= A(v')(I + A(v'))\frac{1}{\sqrt{2}} \prod_{v\neq v'} (I + A(v)) \,|00\ldots0\rangle \\ &= (A(v') + I)\frac{1}{\sqrt{2}} \prod_{v\neq v'} (I + A(v)) \,|00\ldots0\rangle \\ &= \frac{1}{\sqrt{2}} \prod_v (I + A(v)) \,|00\ldots0\rangle = |\xi\rangle \end{aligned}$$

e

$$\begin{aligned} B(f')\,|\xi\rangle &= B(f')\frac{1}{\sqrt{2}} \prod_v (I + A(v)) \,|00\ldots0\rangle \\ &= \frac{1}{\sqrt{2}} \prod_v (I + A(v))B(f') \,|00\ldots0\rangle \\ &= \frac{1}{\sqrt{2}} \prod_v (I + A(v)) \,|00\ldots0\rangle = |\xi\rangle \,. \end{aligned}$$

Logo, $|\xi\rangle$ é fixado pelos operadores face e vértice.

O resultado do produtório $\prod_v (I + A(v))$ é igual à soma de todos os possíveis subconjuntos de operadores vértice $A(v)$. Como os operadores vértice são equivalentes

a operadores face duais, então é possível encarar esse estado fundamental como uma superposição de todos os possíveis cociclos triviais do reticulado, ou seja, dos elementos de $\mathbf{B}^1$.

3.2.6 Os elementos do grupo de Pauli

Agora, considere um produto tensorial de operadores Z's (e I's). Com o uso da notação da Equação (3.4), esse produto tensorial é escrito como Z_c, com $c \in \mathbf{C}_1$. Todo operador face $B(f)$ comuta com Z_c, já que ambos são produtos tensoriais de Z's. Por outro lado, Z_c só comuta com um operador vértice $A(v)$ se o número de arestas compartilhadas por Z_c e $A(v)$ for par. Ora, então, para que Z_c comute com todos os geradores do código tórico, é necessário que c não possua bordo. Em outras palavras, que seja um ciclo, $c \in \mathbf{Z}_1$.

De modo similar, considere um produto tensorial de operadores X's (e I's), utilizando uma notação equivalente à da Equação (3.4), mas no reticulado dual. Escreve-se esse operador como X_{c^*}, com $c^* \in \mathbf{C}^1$. Aqui, as cadeias c e c^* dos operadores Z_c e X_{c^*} representam elementos genéricos de $\mathbf{C}_1$ e $\mathbf{C}^1$, respectivamente, sem qualquer relação direta entre c e c^*. Todo operador $A(v)$ comuta com X_{c^*}. Além disso, um operador face $B(f)$ comuta com X_{c^*} se, e somente se, o número de arestas compartilhadas pelos dois for um número par. Portanto, um operador X_{c^*} comuta com todos os geradores do código tórico apenas se c^* for um ciclo dual, $c^* \in \mathbf{Z}^1$. A fim de visualizar a afirmação anterior, lembre que um operador face do reticulado primal atua como um operador vértice no reticulado dual.

Com base no que foi discutido, e tendo em mente que os tensoriais de X's (e I's) e Z's (e I's) geram os elementos do grupo de Pauli, a menos de um fator de fase global, pode-se escrever qualquer operador de Pauli A da forma

$$A = i^\alpha Z_c X_{c^*}, \quad (c, c^*) \in \mathbf{C}_1 \times \mathbf{C}^1, \alpha \in \mathbb{Z}_4. \tag{3.13}$$

Dados um operador Z_c e um operador X_{c^*}, eles anticomutam se a 1-cadeia c e a 1-cocadeia c^* se cruzarem um número ímpar de vezes, pois cada interseção representa um qubit em que os dois operadores atuam.

Qualquer operador de Pauli no código tórico pode ser relacionado a um par de uma cadeia e uma cocadeia, como na Equação (3.13). Há relação entre $N(S)/S$ e $\mathbf{Z}_1/\mathbf{B}_1$. Perceba que, dados $c \in \mathbf{C}_1$, $c* \in \mathbf{C}^1$, sendo v um vértice e f uma face do reticulado, então

$$[Z_c, A(v)] = 0 \iff v \notin \partial_1 c, \qquad [X_{c^*}, B(f)] = 0 \iff f^* \notin \delta_1 c^*, \tag{3.14}$$

pois, em situação diferente, o número de qubits compartilhados por Z_c e $A(v)$ (ou por X_{c^*} e $B(f)$) seria ímpar e eles anticomutariam. Considere um operador A na forma da

Equação (3.13) e as relações apresentadas na Equação (3.14), segue que $A \in N(S)$ se, e somente se, c e c^* possuírem ambas bordo nulo, $(c, c^*) \in \mathbf{Z}_1 \times \mathbf{Z}^1$.

Um elemento do estabilizador é o produto de um subconjunto dos geradores e pode ser expresso como $B = \prod_i A(v_i) \prod_j B(f_j)$, para algum subconjunto de faces $\{f_j\}$ e algum subconjunto de vértices $\{v_i\}$. Podemos reescrever esse operador B em função dos bordo dos conjuntos $\{f_j\}$ e $\{v_i\}$, veja:

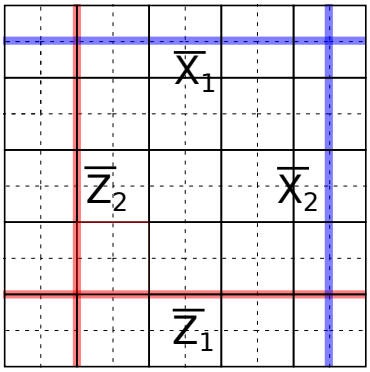
$$\begin{aligned} B &= \prod_i A(v_i) \prod_j B(f_j) \\ &= \left(\prod_i X_{e|v_i \in \partial e}\right)\left(\prod_j Z_{\partial f_j}\right) \\ &= \left(\prod_i X_{\delta v_i^*}\right)\left(\prod_j Z_{\partial f_j}\right) \\ &= \left(X_{\sum_i \delta v_i^*}\right)\left(Z_{\sum_j \partial f_j}\right) \\ &= \left(X_{\delta \sum_i v_i^*}\right)\left(Z_{\partial \sum_j f_j}\right) \\ &= X_{\delta c_0^*} Z_{\partial c_2}, \end{aligned} \tag{3.15}$$

onde $c_0^* = \sum_i v_j^*$ é uma 0-cocadeia e $c_2 = \sum_i f_j$ é uma 2-cadeia. Portanto, δc_0^* é o cobordo de um conjunto de faces duais e ∂c_2 é o bordo de um conjunto de faces. Na segunda igualdade, foi usada a notação da Equação (3.8). A passagem para a terceira igualdade usa a relação entre os operadores ∂ e δ, apresentada na Equação (3.6). Já nas quarta e quinta igualdades foram utilizadas as propriedades das Equações (3.5) e (3.3), respectivamente.

Assim, comparando as Equações (3.13) e (3.15), percebemos que um operador A pertence ao estabilizador se $\alpha = 0$ e $(c, c^*) \in \mathbf{B}_1 \times \mathbf{B}^1$. Resumidamente, ao normalizador $N(S)$ estão associados os ciclos e cociclos, enquanto que aos elementos do estabilizador S estão associados os bordos e cobordos.

Da teoria de códigos estabilizadores sabemos que os elementos de $N(S) - S$ equivalem às operações lógicas sobre a informação codificada. No código tórico, os operadores lógicos de Pauli, diferentes de I, tomam a forma de ciclos e cociclos não triviais, pois, apesar de comutarem com todos os geradores, os ciclos e cociclos não triviais não são o bordo de quaisquer conjunto de faces e cofaces e, por isso, não são produtos dos geradores. Na Figura 3.9 é apresentada uma possível escolha de operadores lógicos $\overline{Z}$, desenhados sobre o reticulado primal, e $\overline{X}$, desenhados sobre o reticulado dual. De fato, como comentado, qualquer ciclo não trivial corresponde a uma operação lógica. O efeito de cada operador lógico sobre a informação codificada dependerá da classe $N(S)/S$ à qual pertence o ciclo, ou cociclo, da operação.

Figura 3.9 – Operadores lógicos



Fonte: elaborado pelo autor.

3.2.7 Síndrome dos erros

Considere um erro, ou seja, um operador de Pauli E. Desconsiderando a fase na Equação (3.13), pode-se escrever $E = Z_c X_{c^*}$. Como o código tórico é um código estabilizador, a identificação de erros é realizada por meio da medição dos geradores de seu estabilizador. Na Seção 2.6.1 foi visto que a síndrome de um erro para um código estabilizador identifica aqueles geradores para os quais o autovalor medido é -1. Relembrando, um elemento gerador M apresentará autovalor -1 sempre que ele anticomutar com o erro E que ocorreu sobre o código.

Baseando-se na Equação (3.14), um operador $A(v)$ anticomutará com um operador Z_c apenas se o número de qubits compartilhados pelos dois for ímpar. Ora, essa é a condição para que v pertença ao bordo de c. Um argumento similar pode ser utilizado para os operadores $B(f)$ e X_{c^*}. Logo, os geradores com autovalor -1 indicam os vértices e as faces que formam o bordo de c e c^*, respectivamente.

Perceba que a síndrome relacionada a E é compartilhada por qualquer erro $E' = Z_d X_{d^*}$, com $(\partial d, \delta d^*) = (\partial c, \delta c^*)$. Assim, a síndrome $(\partial c, \delta c^*)$ não identifica unicamente um erro, mas sim um conjunto de erros, ou seja, o código tórico é degenerado. Para corrigir o estado e retorná-lo ao espaço do código é preciso escolher um operador E' que possua a mesma síndrome e o aplicar sobre o código. No Capítulo 4 discutiremos com mais profundidade o processo de correção e decodificação, apresentando possíveis modelos de erros, o processo para escolha do operador de correção e os efeitos que a escolha incorreta do operador acarreta.

3.3 Um segundo ponto de vista: *anyons*

Nesta seção, o código tórico será analisado como o modelo *anyônico* que é. Muitos dos elementos e estruturas abordados aqui foram vistos na seção anterior sob o ponto de vista homológico. Seguiremos utilizando alguns desses elementos, tais como

cadeias, porém o nosso objetivo agora será entender o código a partir de um ponto de vista físico, sob a ótica dos *anyons*.

Antes de partirmos diretamente ao modelo, apresentaremos os *quantum double models*. Os estados codificados desses modelos são também os estados fundamentais do seu hamiltoniano, fazendo com que os erros sejam penalizados de forma energética. Isso corresponde a estados excitados. Ou seja, a informação é protegida de forma energética. A citação a esses modelos decorre do fato de que o código tórico é o *quantum double model* mais simples.

3.3.1 *Quantum double models*

Os *quantum double models* estão relacionados a reticulados particulares de sistemas topológicos. Esses modelos são caracterizados pela existência de um grupo finito G que atua sobre estados de *spin* localizados sobre as arestas do reticulado. Como consequência, os estados fundamentais dos hamiltonianos desses modelos se comportam como códigos corretores de erros [16].

As operações realizadas sobre os *spins* levam ao surgimento de partículas conhecidas como *anyons* nos vértices e nas faces do reticulado. As propriedades de fusão e trançamento desses *anyons* dependem diretamente do grupo G que, quando for um grupo abeliano, fará com que os *anyons* sejam denominados abelianos. Por outro lado, o uso de grupos não abelianos implica em *anyons* não abelianos.

O exemplo mais simples de um *quantum double model* é o código tórico, que é baseado no grupo abeliano Z_2, composto dos elementos $\{0, 1\}$, e da operação $*$ para qual vale que $0 * 0 = 0$, $0 * 1 = 1 * 0 = 1$ e $1 * 1 = 0$. Sendo assim, o código tórico é um modelo abeliano. Nele, o grupo Z_2 age sobre partículas *spin*-1/2 alocadas nas arestas do reticulado. Um exemplo de partícula *spin*-1/2 é o elétron.

Devido à sua simplicidade, o código tórico é um dos modelos topológicos mais estudados, servindo ainda como modelo de teste para as propriedades dos sistemas topológicos [16].

3.3.2 Os *anyons*

A estatística é uma propriedade da Mecânica Quântica que descreve a mudança na função de onda de um sistema quando duas partículas idênticas são trocadas. Em espaços tridimensionais, se duas partículas idênticas forem trocadas de lugar, elas podem apresentar dois comportamentos distintos: bósons ou férmions. No caso dos bósons, a troca das partículas é representada por um operador identidade que mantém a função de onda inalterada, não havendo mudança na amplitude do sistema. Como exemplo de bósons há os fótons. Por outro lado, a permutação de férmions resulta em uma fase -1 sobre a

função de onda. Como exemplo de férmions existem os elétrons. Uma consequência direta dessa propriedade é que os bósons podem ocupar o mesmo estado, enquanto os férmions só podem ser agrupados juntos com cada partícula ocupando um estado diferente [16].

Em dimensão 2, há uma variedade de tipos de estatística de partículas possível. Partículas indistinguíveis em duas dimensões, que não são bósons nem férmions, são chamadas *anyons* [26]. Apesar de parecerem uma construção teórica, os *anyons* podem ser estudados em laboratório. É possível a sintetização de um gás bidimensional de elétrons, prendendo-os em uma camada fina entre duas placas de semicondutores, de modo que, em baixas energias, o movimento do elétron na direção ortogonal à camada seja impedido. Em um campo magnético suficientemente forte e em uma temperatura suficientemente baixa, e se os elétrons no material são suficientemente móveis, o gás de elétron bidimensional atinge um estado fundamental profundamente emaranhado que é separado de todos os estados excitados por um *gap* de energia diferente de zero [26].

A primeira discussão apresentada sobre a realização de computação quântica tolerante a falhas a partir do uso dessas partículas foi proposta por Kitaev em 1997 [24], sendo este o código tórico. Usando *anyons*, é possível alcançar tal estado no nível físico, com uma resistência intrínseca à decoerência e a outros erros [26].

3.3.3 Definições iniciais

Mais uma vez, consideremos um reticulado quadrado do toro com qubits posicionados em suas arestas. Consideremos, também, os operadores face e vértice, já definidos na Equação (3.7), e o estado fundamental $|\xi\rangle$, apresentado na Equação (3.12) e que corresponde, por definição, a um vácuo anyônico, ou seja, a ausência de *anyons*.

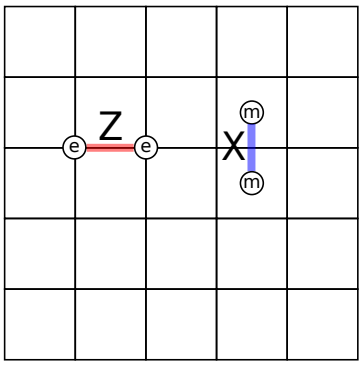
É possível excitar pares de *anyons* no reticulado do código tórico por meio do uso de alguns operadores de Pauli. Por exemplo, a aplicação de um operador Z sobre um qubit do reticulado cria um par de quasipartículas localizadas nos dois vértices das extremidades da aresta em que se encontra o qubit e que correspondem ao autovalor -1 nos operadores $A(v)$ dos dois vértices dessa aresta. São os chamados *anyons* do tipo e. Já um par de *anyons* do tipo m surge nas faces vizinhas a um qubit que é rotacionado por X. A Figura 3.10 exemplifica o processo de criação de *anyons* tipo e e m, indicando a porta utilizada para cada um dos tipos de *anyon*. Denota-se o estado resultante do surgimento dos *anyons* tipos e e m, respectivamente, por

$$|e, e\rangle = Z\,|\xi\rangle \qquad \text{e} \qquad |m, m\rangle = X\,|\xi\rangle\,,$$

em que a ausência de subíndice nos operadores X e Z indica que essas portas atuam sobre um qubit arbitrário. Por sua vez, a combinação dos *anyons* e e m gera um novo tipo de *anyon*, ϵ, com

$$|\epsilon, \epsilon\rangle = ZX\,|\xi\rangle\,. \tag{3.16}$$

Figura 3.10 – Criação de pares *anyons* e e m



Fonte: adaptado de Pachos [16].

3.3.4 Fusão e aniquilação de *anyons*

A presença das quasipartículas e, m e ϵ é detectada por meio da medição dos autovalores dos operadores $A(v)$ e $B(f)$ correspondentes. O autovalor $+1$ corresponde ao vácuo anyônico, enquanto -1 detecta a presença das quasipartículas. Quando um mesmo tipo de rotação é aplicado sobre spins, de forma que o resultado é a criação de um par de *anyons* de um mesmo tipo sobrepostos sobre um vértice v ou uma face f, então o resultado da medição do autovalor de $A(v)$, ou $B(f)$, será $+1$. Mais precisamente, o resultado da fusão de dois *anyons* do mesmo tipo é o vácuo, a aniquilação dos *anyons*.

Com base na fusão de *anyons* é possível fazer uso dos operadores de Pauli para mover *anyons* sobre o reticulado. Indica-se a posição de uma sequência de rotações do tipo Z por meio de uma 1-cadeia (relembrando, uma 1-cadeia é um subconjunto das arestas do reticulado): os *anyons* e estarão sempre em suas extremidades. O mesmo vale para as rotações X, 1-cocadeias e *anyons* tipo m.

Um número par de rotações Z aplicadas sobre os spins das arestas de um determinado vértice v fazem com que $A(v)$ possua autovalor $+1$, enquanto um número ímpar levará a um autovalor -1. Da mesma forma, um número par de rotações X sobre os spins de uma mesma face f leva a um autovalor $+1$ do operador $B(f)$ e um número ímpar ao autovalor -1. Junto com a lei de composição para a criação de quasipartículas ϵ, Equação (3.16), essas propriedades geram as seguintes leis de fusão,

$$
\begin{aligned}
e \times e &= I; \\
m \times m &= I; \\
\epsilon \times \epsilon &= I; \\
e \times m &= \epsilon; \\
\epsilon \times e &= m; \\
\epsilon \times m &= e,
\end{aligned}
$$

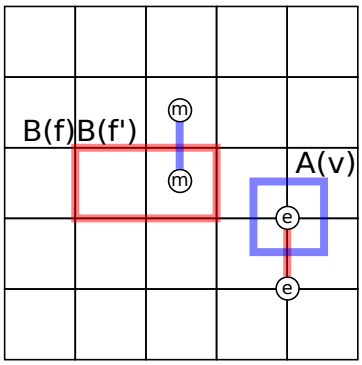
que descrevem a combinação de *anyons*, onde I representa o vácuo anyônico. Como dito, se dois *anyons* são criados no mesmo vértice ou na mesma face, eles se aniquilam. A operação de aniquilação permite a união entre 1-cadeias (1-cocadeias), de rotação Z (X), o que leva à construção de 1-cadeias (1-cocadeias) maiores, sempre com um par de *anyons* em suas extremidades.

3.3.5 Identificando a existência de *anyons*

Se uma 1-cadeia de rotações Z forma um ciclo, então os *anyons* em suas pontas se aniquilam, removendo qualquer excitação anyônica. Os menores ciclos de rotações Z são os operadores face $B(f)$. Por sua vez, os cociclos de rotações X também correspondem a estados sem excitação anyônica e os menores desses cociclos são os operadores $A(v)$.

O autovalor de um ciclo de rotações Z consegue detectar se há um número par ou ímpar de *anyons* do tipo m internos a ele. Se há um número par, o autovalor é 1, se for ímpar, é -1. O mesmo vale para *anyons* do tipo e e cociclos de rotações X. Veja a Figura 3.11. O operador $B(f)B(f')$ consegue identificar a presença de um *anyon* m que ele engloba, assim como o operador $A(v)$ identifica a existência de um *anyon* e.

Figura 3.11 – Detecção de *anyons* com o uso de conjuntos de operadores $A(v)$ e $B(f)$



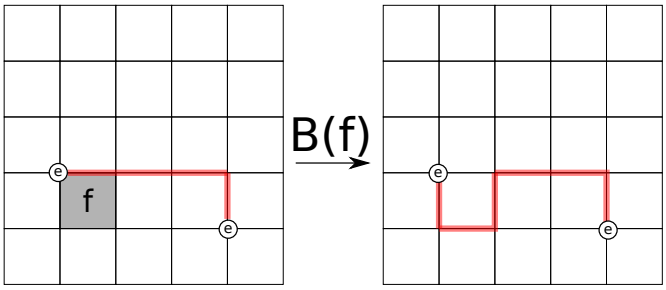
Fonte: adaptado de Pachos [16].

Como já citado na Seção 3.2.5, o fato de os operadores $A(v)$ serem os menores cociclos de rotações X permite enxergar o estado $|\xi\rangle$ da Equação (3.12), como uma

superposição de todas as possíveis combinações de cociclos elementares $A(v)$. A aplicação de qualquer cociclo contrátil, ou seja, um cociclo homologicamente trivial, sobre $|\xi\rangle$ resulta no próprio estado $|\xi\rangle$, mediante o rearranjo de seus elementos.

Considere dois *anyons* do tipo e no fim de uma 1-cadeia de operadores Z. O estado é invariante no que diz respeito a deformações no formato da cadeia, desde que os pontos finais sejam mantidos fixos. Para comprovar isso, perceba que o estado dos dois *anyons* e não se altera se for aplicado qualquer estabilizador $B(f)$. Se alguma das arestas que pertence ao operador $B(f)$ também pertencer à 1-cadeia, então esta será removida da 1-cadeia e um novo traçado será apresentado, com a adição das outras arestas de $B(f)$. Porém, as duas 1-cadeias, inicial e deformada, dão origem aos mesmos estados anyônicos. A Figura 3.12 mostra a operação de deformação de uma 1-cadeia de rotações Z. O mesmo vale para *anyons* do tipo m: a aplicação de operadores $A(v)$ deforma a 1-cocadeia associada aos *anyons*, mas o estado se mantém invariante.

Figura 3.12 – Efeito de um operador $B(f)$ sobre uma cadeia de anyons e



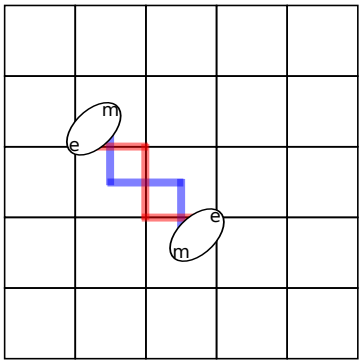
Fonte: elaborado pelo autor.

3.3.6 Estatística anyônica

O comportamento da estatística dos três tipos de *anyons*, e, m e ϵ, será analisado agora. Considere inicialmente dois *anyons* e. Podemos trocar as suas posições com a utilização de um ciclo de rotações Z que passe por ambas. Como as faces em que esse ciclo atua não possuem nenhum *anyon* do tipo m, então ele retornará à identidade. Logo, o estado após a troca dos *anyons* será igual ao estado inicial, ou seja, *anyons* tipo e possuem estatística mútua bosônica. Com um argumento equivalente concluímos que os *anyons* do tipo m também são bosônicos. Porém, o mesmo não vale para *anyons* ϵ. É possível demonstrar que a estatística deles é ferminiônica. Considere um par de partículas ϵ, que são formados a partir de *anyons* tipo e e m e ocupam uma face e um vértice vizinhos, como na Figura 3.13.

Na troca dos dois *anyons* ϵ, deseja-se não rotacioná-los, pois isso pode causar fatores de fase extras, devido ao spin que a partícula ϵ pode ter [16]. A Figura 3.14 demonstra como pode ser feita a troca dos três tipos de *anyons*. As linhas contínuas coloridas representam as operações que levaram à criação dos *anyons*, enquanto as linhas

Figura 3.13 – Anyons do tipo ϵ

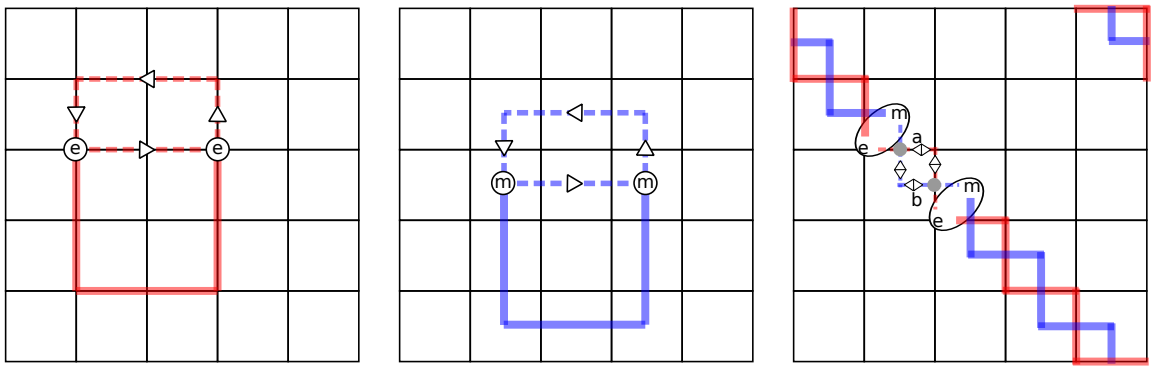


Fonte: adaptado de Pachos [16].

pontilhadas representam as operações realizadas para a troca dos *anyons*, com as setas indicando a direção em que eles são transportados.

Figura 3.14 – Troca de *anyons*

(a) Troca de *anyons* tipo e (b) Troca de *anyons* tipo m (c) Troca de *anyons* tipo ϵ



Fonte: elaborado pelo autor.

Para os *anyons* ϵ da Figura 3.14(c), uma possível troca seria realizada por

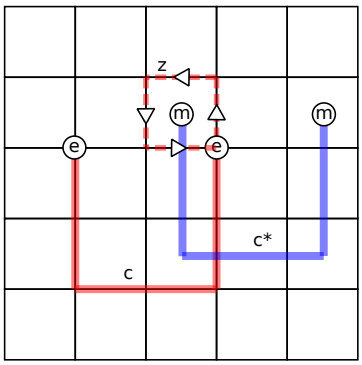
$$(X_a Z_a Z_b X_b)(X_b Z_b X_a Z_a) = -I,$$

onde os subíndices a e b indicam em qual dos dois qubits a porta está sendo aplicada. Os dois qubits estão representados na imagem por círculos cinzas, com o rótulo ao lado. Na equação, cada parêntese corresponde ao conjunto de rotações responsável por mover uma das partículas de sua posição inicial para a final. Como o resultado final de todos os operadores é uma fase global -1, identificamos então que os *anyons* ϵ possuem estatística ferminiônica.

Uma vez que os *anyons* e e m são distinguíveis, não é possível trocá-los diretamente, mas é possível trançá-los, o que corresponde a fazer duas trocas. É possível atribuir a estatística da troca como a raiz quadrada do resultado da evolução. Considere um estado inicial $|\psi\rangle = Z_c X_{c^*} |\xi\rangle$. Durante o trançamento, o *anyon* e é movido ao redor

do *anyon* m por meio de um caminho gerado por sucessivas rotações do tipo Z, o que dá origem a um ciclo $z \in \mathbf{Z}_1$, como na Figura 3.15.

Figura 3.15 – Trança de *anyons* e e m



Fonte: elaborado pelo autor.

Como o operador Z_z cruzará o operador X_{c^*} um número ímpar de vezes, então vale que

$$Z_z X_{c^*} = -X_c Z_z.$$

Portanto, se for aplicado Z_z sobre o estado $|\psi\rangle$, então

$$\begin{aligned} Z_z |\psi\rangle &= Z_z Z_c X_{c^*} |\xi\rangle \\ &= Z_c Z_z X_{c^*} |\xi\rangle \\ &= -Z_c X_{c^*} Z_z |\xi\rangle \\ &= -Z_c X_{c^*} |\xi\rangle \\ &= -|\psi\rangle , \end{aligned}$$

pois, como z é um ciclo, o efeito de Z_z sobre o estado base $|\epsilon\rangle$ é trivial. Esse resultado é independente do ciclo Z_z, desde que ele circule o *anyon* m apenas uma vez. O fator de fase -1 revela uma estatística não trivial entre os *anyons* e e m. A mesma fase é obtida se trançarmos *anyons* e ou m com ϵ.

As partículas e e m são bósons e possuem spin trivial. Como não possuem um senso de orientação, elas não podem ser rotacionadas. Já as partículas ϵ possuem spin $1/2$ e são férmions. De fato, uma rotação anti-horária de 2π de um *anyon* ϵ corresponde a mover o *anyon* e ao redor do *anyon* m. O ciclo associado ao movimento de e é um operador $B(f)$, o qual terá em seu centro uma *anyon* m, e, portanto, levará a um fator de fase -1, comprovando o seu spin $1/2$ [16].

3.3.7 Codificação da informação em um código tórico

Por mais que a topologia da superfície não influencie a estatística dos *anyons* gerados por excitações no código tórico, ela tem papel fundamental para a codificação da informação quântica. A utilização de superfícies de gênero não trivial dá origem a múltiplos estados fundamentais. Por sua vez. a transformação de um estado fundamental em outro envolve a criação de um par de *anyons*, a movimentação deles ao longo de ciclos não triviais da superfície e a sua posterior aniquilação. O estado final da operação não possui excitações anyônicas e, portanto, é um estado fundamental. Além disso, ele é diferente do estado inicial, já que o ciclo utilizado é não contrátil.

Dado o primeiro estado fundamental do toro, $|\psi^1\rangle = |\xi\rangle$, como na Equação (3.12), com o uso dos operadores $X_{z_1^*}$ e $X_{z_2^*}$ e com $z_1^*, z_2^* \in \mathbf{Z}^1$ cociclos não triviais não equivalentes, é possível criar outros estados fundamentais,

$$\left|\psi^1\right\rangle = |\xi\rangle\,, \left|\psi^2\right\rangle = X_{z_1^*}\,|\xi\rangle\,, \left|\psi^3\right\rangle = X_{z_2^*}\,|\xi\rangle \text{ e } \left|\psi^4\right\rangle = X_{z_1^*}X_{z_2^*}\,|\xi\rangle\,. \tag{3.17}$$

Tais estados são invariantes sob deformações contínuas dos ciclos anyônico, de forma que apenas eles podem ser criados nessas circunstâncias. Os quatro estados citados são linearmente independentes, já que correspondem à superposição de ciclos que diferem em relação ao seu enrolamento em torno do toro. De forma equivalente, podemos criar estados fundamentais empregando os anyons e ou combinações dos anyons e e m. Porém, eles vão possuir uma relação de dependência linear com os estados da Equação (3.17) [16].

Surge, então, um espaço de Hilbert de dimensão quatro, que conseguirá codificar dois qubits. De forma mais geral, se um código como o tórico for definido sobre uma superfície de gênero g, ele conseguirá codificar $2g$ qubits. Isso, como esperado, condiz com o resultado encontrado por meio do número de geradores do estabilizador na Equação (3.11).

3.4 Parâmetros de um código tórico quadrado

Partindo das características do código tórico apresentadas nas Seções 3.2 e 3.3, agora serão resumidos os parâmetros deste código corretor de erros.

Considere um código tórico $\mathcal{C}$ definido sobre um reticulado quadrado com $L \times L$ arestas, ou seja, há L^2 quadrados no reticulado, cada quadrado possui quatro arestas e cada aresta pertence a dois quadrados simultaneamente, logo, há um total de $2L^2$ arestas no reticulado. Como cada aresta está associada a um qubit, o número de qubits físicos do código tórico quadrado é $n = 2L^2$. Os quatro estados fundamentais do hamiltoniano do

sistema geram o subespaço do código $\mathcal{C}$, no qual a informação será codificada. Isso permite que $k = 2$ qubits sejam codificados.

Os erros ocorrem quando há rotações indesejadas dos spins, que causam a criação de *anyons*. Os ciclos não contráteis de rotações correspondem às operações codificadas. Por ser um código estabilizador, sabe-se que a distância do código é igual ao peso da menor dessas operações lógicas, o que corresponde a um ciclo com comprimento igual ao lado do reticulado, ou seja, $d = L$. Conclui-se, então, que o código tórico quadrado é um código $[[2L^2, 2, L]]$.

A detecção de erros é realizada por meio da medição dos operadores vértice e face e a correção de erros corresponde à fusão dos *anyons* identificados, ambas com o objetivo de retornar o sistema ao estado fundamental em que se encontrava inicialmente. É importante que durante a tentativa de correção não sejam criados ciclos não contráteis, o que causaria a aplicação de uma operação lógica não desejada sobre a informação.

A geração de um erro no código é penalizada com um *gap* de energia, pois os estados que o formam correspondem a estados sem excitação anyônica. Isso acaba fazendo com que a probabilidade de existência de grandes cadeias de rotações seja exponencialmente suprimida, tornando os códigos tóricos de grandes tamanhos não favoráveis à codificação. Apesar da geração de erros ser penalizada pelos *gaps* de energia, ou seja, eles são tratados em um nível físico, o código tórico não é adequado para a realização de cálculos na computação quântica porque suas portas lógicas produzem apenas fatores de fase, ou seja, não produz porta lógica que faça emaranhamento, e, por isso, não são universais [16]. Entretanto, esses sistemas são altamente favoráveis à utilização como memórias quânticas.

Cabe aqui uma explicação sobre o que é uma computação quântica universal. Define-se que um conjunto de portas quânticas é universal para a computação quântica se qualquer operação unitária puder ser aproximada com precisão arbitrária por um circuito quântico construído utilizando apenas essas portas. As portas $CNOT, H$ e T são um exemplo de conjunto de portas quânticas universais [31]. Já conhecemos a porta de Hadamard H. A $CNOT$, ou NÃO-CONTROLADA, possui um qubit de controle e um qubit alvo; sempre que o qubit de controle for $|1\rangle$, o qubit alvo terá seu estado trocado de $|0\rangle$ para $|1\rangle$, e vice-versa. A Equação (3.18) apresenta a representação matricial desse operador e a Equação (3.19) o seu efeito sobre um estado $(a\,|0\rangle + b\,|1\rangle)\,|0\rangle = a\,|0\rangle\,|0\rangle + b\,|1\rangle\,|0\rangle$.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.18}$$

$$CNOT(a\,|0\rangle\,|0\rangle + b\,|1\rangle\,|0\rangle) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ 0 \\ b \\ 0 \end{bmatrix} = \begin{bmatrix} a \\ 0 \\ 0 \\ b \end{bmatrix} = a\,|0\rangle\,|0\rangle = b\,|1\rangle\,|1\rangle\,. \tag{3.19}$$

Por fim, a porta T promove uma rotação de fase $\pi/8$ e é representada pela Equação (3.20).

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}. \tag{3.20}$$

O efeito da porta T sobre um estado arbitrário $|\psi\rangle = a\,|0\rangle + b\,|1\rangle$ é

$$T\,|\psi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} a \\ e^{\frac{i\pi}{4}} b \end{bmatrix}.$$

3.5 Exemplo: o código tórico [[50, 2, 5]]

Para consolidar o que foi exposto até aqui, apresentaremos como exemplo o código tórico $[[50, 2, 5]]$. Esse código é construído sobre um reticulado quadrado de lado $L = 5$ e, portanto, possui $n = 2L^2 = 50$ qubits físicos.

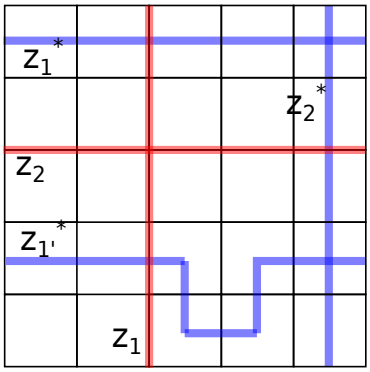
Como $k = 2$ qubits lógicos são codificados, então uma base para o espaço de Hilbert de origem possui $2^k = 2^2$ estados. Uma possível escolha para tal base é o conjunto $\{|00\rangle\,, |01\rangle\,, |10\rangle\,, |11\rangle\}$. Considerando os estados fundamentais da Equação (3.17), uma possível escolha de estados lógicos é dada por

$$\begin{aligned} |00_L\rangle &\to |\xi\rangle\,; \\ |01_L\rangle &\to X_{z_1^*}\,|\xi\rangle\,; \\ |10_L\rangle &\to X_{z_2^*}\,|\xi\rangle\,; \\ |11_L\rangle &\to X_{z_1^*} X_{z_2^*}\,|\xi\rangle\,, \end{aligned}$$

onde $z_1^*, z_2^* \in \mathbf{Z}^1$ são dois cociclos não triviais de classes homológicas diferentes. Por exemplo, na Figura 3.16 é possível escolher as ciclos z_1^* e z_2^*, ou $z_{1'}^*$ e z_2^* para fazer essa codificação, mas não z_1^* e $z_{1'}^*$, pois os dois últimos pertencem à mesma classe homológica.

Os operadores $X_{z_1^*}$ e $X_{z_2^*}$ correspondem justamente aos operadores lógicos codificados $\bar{X}$. Aqui, por escolha, denotaremos $X_{z_1^*} = \bar{X}_1$ e $X_{z_2^*} = \bar{X}_2$. Aplicar $X_{z_1^*}$ sobre

Figura 3.16 – Exemplo de operadores lógicos no código tórico $[[50, 2, 5]]$



Fonte: elaborado pelo autor.

uma palavra-código é equivalente a aplicar uma porta X sobre um dos qubits lógicos originais, e $X_{z_1^*}$ tem o mesmo efeito, só que sobre o outro qubit lógico. Por exemplo,

$$X_{z_1^*} |00_L\rangle = |01_L\rangle \text{ e } X_{z_2^*} |11_L\rangle = |01_L\rangle .$$

Por sua vez, os ciclos z_1 e z_2 da Figura 3.16 podem ser tomados para construir os operadores Z_{z_1} e Z_{z_2}, que correspondem, respectivamente, aos operadores lógicos $\bar{Z}_1$ e $\bar{Z}_2$. Os índices das portas lógicas codificadas indicam que esses operadores atuam nos mesmos qubits lógicos que $\bar{X}_1$ e $\bar{X}_2$, nessa ordem. Veja, como Z_{z_1} e $X_{z_1^*}$ se interceptam em apenas um qubit, então vale que $Z_{z_1} X_{z_1^*} = -X_{z_1^*} Z_{z_1}$, e por isso vale, por exemplo, que

$$Z_{z_1} |01_L\rangle = Z_{z_1} X_{z_1^*} |\xi\rangle = -X_{z_1^*} Z_{z_1} |\xi\rangle = -X_{z_1^*} |\xi\rangle = - |01_L\rangle .$$

Outras relações entre $\bar{X}_1, \bar{X}_2, \bar{Z}_1$ e $\bar{Z}_2$ e os estados lógicos $\{|00_L\rangle , |01_L\rangle , |10_L\rangle , |11_L\rangle\}$ podem ser obtidas sem esforço.

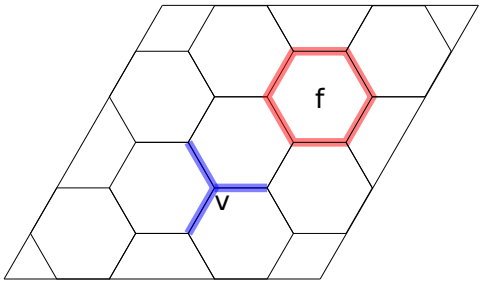
Por fim, a distância do código é $d = 5$, que é o comprimento, em número de arestas, dos menores ciclos não triviais, como z_1, z_2, z_1^* e z_2^* da Figura 3.16. Um exemplo com as análises da síndrome e da identificação de erros para esse mesmo código será apresentado no Capítulo 4.

3.6 O código tórico hexagonal

Como já vimos, o toro pode ser tesselado pela tesselação $\{6, 3\}$ e sua dual $\{3, 6\}$, e portanto, é possível definir um código tórico sobre tais tesselaçõe. Comentaremos sobre o código tórico hexagonal. Neste caso, o reticulado sobre o qual estão os qubits é hexagonal, o que trará certas particularidades em relação ao código tórico quadrado.

Considere a tesselação hexagonal do toro da Figura 3.17. Aqui, o modelo planar do toro foi tomado de forma que suas bordas coincidam em angulação com as arestas dos hexágonos que compõem a tesselação.

Figura 3.17 – Código tórico hexagonal e seus operadores vértice e face



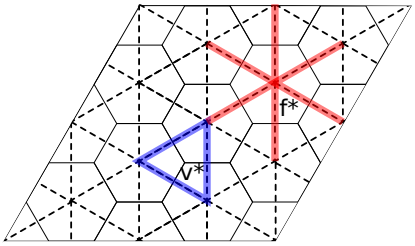
Fonte: elaborado pelo autor.

O número de qubits lógicos continua sendo $k = 2$, já que esse parâmetro não depende do reticulado, mas sim do gênero da superfície. Por sua vez, o número de qubits físicos n pode ser calculado contando o número de hexágonos na tesselação n_{hex} [1]. A partir daí, cada hexágono possui seis arestas, e cada aresta pertence a dois hexágonos, portanto,

$$n = \frac{6n_{hex}}{2} = 3n_{hex}.$$

Há diferenças nos geradores do estabilizador. Eles continuam sendo operadores vértice e face, mas o número de qubits sobre os quais cada um atua agora é diferente, a saber, três e seis, respectivamente. A Figura 3.17 também exemplifica esses operadores. Quando um operador vértice é encarado no reticulado dual, ele novamente se torna um operador face. No entanto, devido à tesselação dual de uma hexagonal ser uma tesselação triangular, ela atua sobre as faces de um triângulo. A Figura 3.18 traz a representação dos operadores da Figura 3.17 no reticulado dual.

Figura 3.18 – Código tórico hexagonal e seus operadores vértice e face no reticulado dual

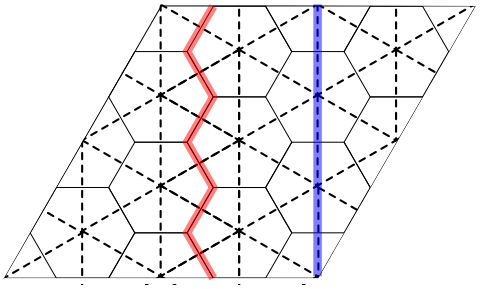


Fonte: elaborado pelo autor.

[1] n_{hex} é a razão entre a área do paralelogramo do modelo planar e a área do hexágono da tesselação, desde que esta razão seja inteira. O mesmo vale para o dual.

Outra diferença em relação ao código tórico quadrado é quanto à distância. Novamente, devido ao reticulado nesse caso não ser autodual, os ciclos não triviais nos reticulados primal e dual possuem comprimentos distintos. Como consequência, o código possuirá uma distância para erros X e outra para erros Z, ou seja, o código tem diferentes capacidades de proteção, de acordo com o tipo de erro. A Figura 3.19 exemplifica um ciclo e um cociclo não triviais de comprimento mínimo.

Figura 3.19 – Código tórico hexagonal e seus ciclos não triviais



Fonte: elaborado pelo autor.

A síndrome de erro, por sua vez, segue igual. A identificação do bordo das cadeias de erros é feita por intermédio de geradores cujas medições retornarem autovalores -1.

Esta breve discussão sobre o código hexagonal buscou demonstrar as características que o diferenciam do quadrado. As particularidades destacadas tornam o código hexagonal mais complexo de se trabalhar do que o código tórico original, e por isso este é menos estudado. Entretanto, em condições em que os erros X e Z não aconteçam de forma equiprovável, o uso dessa variedade de código tórico pode ser útil.

3.7 O código planar

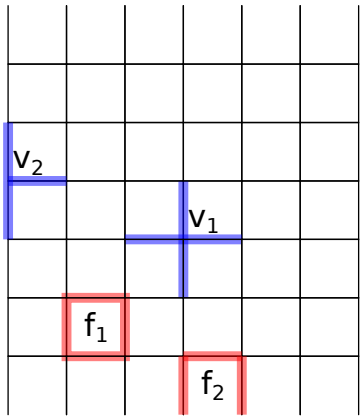
Apesar de o código tórico ser um modelo extremamente elegante e simples de correção de erros, na prática, arranjar os qubits em uma topologia de toro pode ser inconveniente, principalmente se houver interesse em que os qubits de diferentes toros interajam [41]. Como alternativa, o código planar foi proposto por Bravyi e Kitaev [42] e Freedman e Meyer [43]. Nele, os operadores de verificação continuam sendo locais, como no código tórico, mas os qubits são arranjados em "folhas planas".

Em um código planar, os qubits também são identificados com as arestas de um reticulado quadrado $L \times L$, mas, diferentemente do código tórico, esse reticulado possui bordos, de forma que não há continuação entre as arestas em extremidades opostas do reticulado como no código tórico. Para entender a afirmação anterior, perceba que ao caminhar em uma mesma direção sobre as arestas do reticulado de um código tórico quadrado retornaremos à aresta inicial, enquanto que ao encontrarmos um dos bordos

durante uma caminhada sobre um código planar não poderemos mais seguir caminho nessa direção. O código planar é um código estabilizador e também conta com geradores vértice e face. Há, porém, a existência de dois tipos de bordo no reticulado, áspero e liso, que levam à deformações nos operadores face e vértice em suas proximidades.

A Figura 3.20 representa o reticulado de um código planar. Os bordos ásperos são o superior e o inferior, enquanto os lisos são os à direita e à esquerda. Os operadores vértice e face no interior do reticulado são iguais aos do código tórico, como o operador vértice v_1 e o operador face f_1 da Figura 3.20. Já os operadores vértice v_2 e face f_2, localizados, respectivamente, nos bordos liso e áspero, atuam em apenas três arestas cada.

Figura 3.20 – Operadores vértice e face do código planar



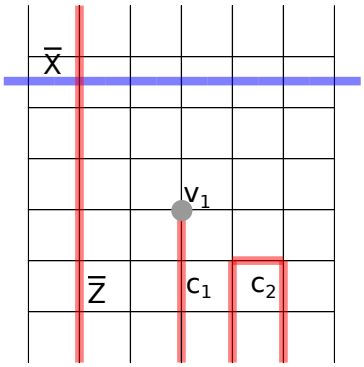
Fonte: elaborado pelo autor.

Assim como ocorre no código tórico, para que um conjunto de operadores Z's comute com os elementos do estabilizador, é necessário que para cada vértice do reticulado um número par de Z's atue sobre ele. Além dos ciclos triviais, se as arestas sobre as quais as portas Z atuam formarem uma cadeia contínua que começa e termina em uma borda áspera, então este operador também comutará com o estabilizador. Diz-se, neste caso, que a 1-cadeia é um ciclo relativo aos bordos ásperos. Similarmente, o conjunto de operadores X's que comutam com o estabilizador compreende, além dos cociclos triviais, os cociclos relativos aos bordos lisos.

Os ciclos relativos a arestas ásperas podem ser de dois tipos. Se uma cadeia parte e chega em uma mesma borda áspera, então é possível que o seu interior seja ladrilhado pelas faces da tesselação (incluindo as do bordo de origem e chegada). Com isso, o produto de Z's associado à 1-cadeia pode ser escrito como o produto dos operadores face e, por isso, pertence ao estabilizador. Diz-se que esse 1-ciclo relativo é a borda relativa de uma 2-cadeia. Porém, um ciclo relativo que vai de uma borda áspera a outra não é o bordo relativo de nenhuma 2-cadeia. De fato, ele é um elemento de uma classe de homologia relativa não trivial, que comuta com todos os elementos do estabilizador, mas não pertence a ele. Podemos, então, tomar um desses ciclos como o operador lógico $\bar{Z}$ atuando sobre

um qubit lógico. De forma semelhante, os ciclos associados às bordas suaves também irão possuir duas variedades, sendo que um produto de X's associado ao ciclo que se estende de uma borda suave a outra pode ser tomado como a operação lógica $\bar{X}$. A Figura 3.21 exemplifica os dois tipos de ciclos relativos. A cadeia c_2 parte e chega ao mesmo bordo rugoso, enquanto as cadeias $\bar{X}$ e $\bar{Z}$ podem ser utilizadas como operadores lógicos, pois são ciclos relativos que partem de um bordo e chegam ao oposto.

Figura 3.21 – Ciclos relativos e operadores lógicos



Fonte: elaborado pelo autor.

Como no reticulado o menor caminho de uma borda áspera à outra, e de uma borda suave à outra, contêm ambos L arestas, então o código possui distância $d = L$. O reticulado possui $L^2 + (L-1)^2$ arestas, $L(L-1)$ vértices e $L(L-1)$ faces, veja, por exemplo, a Figura 3.21. Devido à existência dos bordos, todos os operadores face e vértice são independentes, consequentemente, da Afirmação 2, identificamos que o número de qubits codificados será: $k = L^2 + (L-1)^2 - 2L(L-1) = 1$. A introdução dos bordos no reticulado faz com que o código planar codifique apenas um qubit lógico, sendo, portanto, um código $[[L^2 + (L-1)^2, 1, L]]$.

Enquanto no código tórico os bordos dos erros sempre aparecem em pares, nos planares podem surgir erros com bordos individuais. É o caso, por exemplo, de uma cadeia de erros Z's que começa no interior da tesselação e se estende até uma borda áspera, ou erros X's que surgem no interior do reticulado e vão até uma borda suave. Na Figura 3.21 a cadeia de erros Z que atua sobre c_1 anticomuta apenas com o operador $A(v_1)$. Perceba que cadeias de erros do tipo Z que atingem as bordas suaves continuam apresentando dois bordos. O mesmo vale para erros do tipo X's que atingem bordas ásperas.

Se erros do tipo *phase-shift* forem mais comuns que erros *bit-flip*, ou vice-versa, então a informação quântica poderá ser armazenada em códigos planos assimétricos, nos quais a distância de uma borda áspera a outra seja maior que aquela entre bordas suaves. Entretanto, os códigos assimétricos são menos convenientes para o processamento da informação codificada [41].

O processo de decodificação para códigos planos é conceitualmente idêntico

ao de códigos tóricos, de forma que muitos decodificadores podem ser adaptados para os mesmos. Assim, os resultados que serão expostos no capítulo a seguir podem ser adaptados para essa classe específica de códigos.

3.8 Códigos de superfície

Os códigos tóricos podem ser generalizados, de forma que qualquer tesselação de uma superfície pode ser associada a um código quântico. Esta classe é chamada *códigos de superfície*. Para uma superfície orientável de gênero g, é possível codificar $2g$ qubits, um resultado direto da Equação (3.11). A distância de um código de superfície será sempre o comprimento do menor ciclo não trivial, quer na tesselação primal ou dual.

Os códigos planares também podem ser generalizados. Considere um total de e bordas ásperas distintas separadas por e bordas suaves distintas, então, será possível codificar $e - 1$ qubits neste código [41]. A distância entre os códigos, por sua vez, será dada pelo comprimento do menor caminho indo de uma borda áspera à outra ou de uma borda suave à outra.

Entretanto, quando passamos a lidar com superfícies de gênero $g \geq 2$, a geometria associada deixa de ser a euclidiana e passa ser a hiperbólica. Em [44] é apresentado uma construção de códigos quânticos topológicos em superfícies compactas de gênero $g \geq 2$.

4 Decodificação em códigos tóricos

Como visto no capítulo anterior, a síndrome do código tórico é responsável pela identificação de erros e retorna seus bordos, ou seja, os vértices e as faces nos quais estão localizados *anyons*. De forma simplificada, no processo de decodificação os anyons são aniquilados e, ao final, o sistema estará em um vácuo anyônico.

No entanto, uma vez que o código tórico é degenerado, a síndrome não identifica univocamente o erro que a gerou, mas sim todo um conjunto de possíveis erros. Além disso, se durante o processo de correção forem criados ciclos não triviais, então operações lógicas serão aplicadas sobre a informação codificada, comprometendo-a. Por isso, diz-se que o processo de decodificação foi eficiente apenas se o estado obtido após o processo for igual ao estado inicial, ou seja, antes da ocorrência dos erros.

Neste capítulo, apresentamos o processo de decodificação. Serão discutidos dois modelos de erros que podem ser assumidos para o sistema, o problema da decodificação ótima e suas dificuldades de implementação e o uso de decodificadores sub-ótimos para contornar esses problemas. Os resultados e comentários apresentados aqui são passíveis de generalização para os códigos de superfície como um todo. Espera-se que o texto a seguir dê ao leitor um vislumbre da teoria de decodificação em códigos tóricos e desperte a sua curiosidade para o tema.

4.1 Comentários iniciais

À semelhança do capítulo anterior, as análises traçadas aqui serão realizadas, em sua maioria, sobre o código tórico de Kitaev. Sendo assim, considere um código tórico definido em um reticulado quadrado de lado L, ou seja, o código $[[2L^2, 2, L]]$. Se a probabilidade de erro por qubit for, independentemente do tipo de erro, p, então o número de erros esperado para esse código é $2pL^2$. Perceba que, dependendo do valor de p, o número de erros esperados tenderá a ser maior que a distância do código, que é L. Entretanto, a performance do código tórico é muito melhor do que essa análise superficial sugere.

Suponha inicialmente que, dada uma síndrome, escolhe-se como operação de correção o conjunto das menores cadeias de operação que ligam todos os defeitos apontados pela síndrome. O termo defeito se refere aqui à presença de *anyons* e, intrinsecamente, aos vértices e faces em que eles estão localizados. Nestas condições, um total de $L/2$ erros já é suficiente para que, durante a correção, ocorra um erro sobre a informação armazenada.

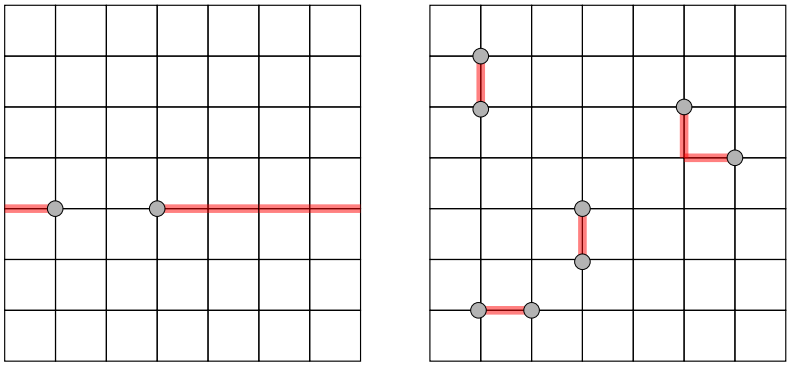
Veja a cadeia de erros da Figura 4.1(a). Dada a síndrome representada pelos

dois vértices destacados, se for utilizada a menor cadeia entre os defeitos como técnica de correção, então surgirá um ciclo não trivial no reticulado. O ponto crucial para entender que a capacidade de proteção de um código tórico é muito melhor do que inicialmente parece é perceber que esta pequena quantidade de erros só causará perda de informação se assumir uma distribuição espacial altamente atípica. Se a probabilidade de erro p for pequena o suficiente, as arestas onde ocorrem os erros tenderão a estarem diluídas por todo o reticulado, de forma que cadeias longas de erros serão pouco prováveis. Assim, uma distribuição de erros como a da Figura 4.1(b) é muito mais provável que a da Figura 4.1(a).

Nesta condição provável de pequenos erros esparsos, é relativamente fácil identificar como emparelhar os defeitos observados para aniquilá-los sem causar danos. Baseado nisso, espera-se que, se a probabilidade de erro por qubit p for pequena o suficiente, a probabilidade de recuperar o estado inicial corretamente tenda a um, conforme o tamanho do bloco de qubits aumente. De fato, há um limite de precisão para o armazenamento de informações quânticas usando um código tórico que estabelece o valor abaixo do qual p deve estar para que o aumento do tamanho do bloco aumente a probabilidade de recuperação da informação. Esse limite será abordado na Seção 4.4.

Figura 4.1 – Exemplos de cadeias de erros no código tórico

(a) Cadeia de erros incomum (b) Cadeia de erros comum



Fonte: adaptado de Dennis *et al.* [41].

Entretanto, existem problemas que precisam ser contornados. A medição da síndrome poderá não ser perfeita e, ocasionalmente, indicará a existência de erros que não existem, *erros fantasmas*. Em outros casos, a existência de um erro real poderá não ser notada. Nessas condições, os erros reais se misturam com os erros fantasmas, que possuem uma distribuição aleatória sobre o reticulado. A correção de erros deve ser realizada cautelosamente, pois a tentativa de aniquilar um erro fantasma introduzirá novos erros ao sistema, de forma que o que antes era apenas um erro na síndrome passará a ser um erro real.

Assim, uma estratégia natural para contornar a situação é repetir a medição

da síndrome a fim de verificar os resultados. Contudo, determinar o número de medições que devem ser feitas da síndrome, bem como formular um procedimento de recuperação que incorpore essas medições repetidas, é um processo sutil, uma vez que mais erros se acumulam conforme as medições são repetidas.

4.2 O problema da decodificação

Como os códigos tóricos são uma subclasse dos códigos estabilizadores, os erros detectáveis anticomutam com ao menos um elemento do seu estabilizador. Se for considerado que não ocorrem erros nas medições, então aqueles operadores em que for observado o autovalor -1 indicam a incidência de erros reais com certeza. Os geradores do estabilizador que anticomutam com um determinado erro revelam informações sobre a sua localização, mas não conseguem identificar o erro com exclusividade, uma vez que o código é degenerado.

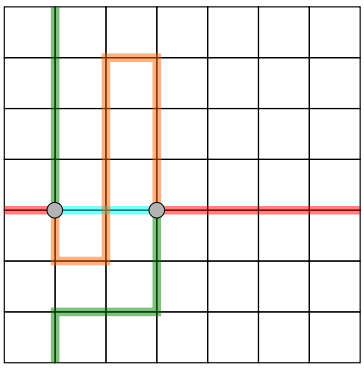
Considere a ocorrência de um erro E. Uma vez que seja realizada a medição de todos os geradores e a síndrome de E esteja estabelecida, será utilizado um algoritmo clássico chamado *decodificador* para decidir qual operador de correção E' será empregado. O decodificador tem o objetivo de escolher entre todas as cadeias de erro que apresentam síndrome igual à obtida, aquela cujo operador $C = E \cdot E'$ possua a máxima probabilidade de ser um ciclo homologicamente trivial. Portanto, uma falha no algoritmo de decodificação corresponde à criação de ciclos homologicamente não triviais e à corrupção da informação armazenada.

Watson [45] aponta dois requisitos importantes para um decodificador. O primeiro é que ele deve ser executado em um tempo eficiente e o segundo é que ele deve possuir um alto *limite de precisão* p_c. O primeiro requisito significa, basicamente, que o tempo que o decodificador leva para ser executado deve crescer no máximo polinomialmente em relação ao tamanho da sua entrada, em outras palavras, a síndrome. Aqui o tamanho da síndrome, por sua vez, depende do número de geradores do estabilizador, o que, por outro lado, depende diretamente da distância do código, que é L. O limite de precisão p_c, apontado no segundo requisito, determina um limite superior para as taxas de erro por qubit p para que a taxa de falha lógica do código decresça rapidamente, de preferência exponencialmente, quando L cresce. Ou seja, deseja-se que para $p < p_c$, a confiabilidade do código cresça com L.

Como foi dito, o código tórico é degenerado. De fato, tal código é altamente degenerado: existem 2^{2L^2} cadeias de erros que compartilham cada síndrome do código [45] (perceba que existem L^2 vértices e L^2 faces no reticulado e que multiplicar uma cadeia de erro por um subconjunto dos operadores associados a esses elementos não altera o bordo da cadeia); veja a Figura 4.2. Todas as cadeias, em diferentes cores, possuem o mesmo

bordo, os vértices destacados, e este é apenas um pequeno conjunto de cadeias com esse bordo específico. Identificar o erro e, consequentemente, a cadeia de correção adequada, é uma tarefa difícil.

Figura 4.2 – Cadeias de erros com a mesma síndrome no código tórico



Fonte: elaborado pelo autor.

Felizmente, não é necessário identificar exatamente a cadeia de erros que ocorreu para conseguir corrigir seu efeito. Se for aplicada uma correção E' que satisfaça $\partial E' = \partial E$ e $E \cdot E' \in S$, então a cadeia resultante $E \cdot E'$ possuirá efeito trivial sobre o espaço do código, em que S é o estabilizador do código e ∂ é o operador bordo, definido na Seção 3.2.1. Com isso, podemos dizer que o objetivo do decodificador é identificar a classe de homologia do erro que aconteceu e, a partir daí, escolher um operador de correção dentro dela.

A forma adequada para se utilizar o decodificador e o seu limite de precisão vão estar diretamente associados ao modelo de erros assumido para o sistema. Apresentamos a seguir dois possíveis modelos de erros que podem ser adotados: um em que não há erros de medição e outro em que se considera a existência desses erros.

4.3 Modelos de erros

O primeiro modelo de erros a ser apresentado aqui é o *modelo de ruído independente simples*, no qual se considera que os erros do tipo *bit-flip* e *phase-shift* não são correlacionados: acontecem de forma equiprovável com uma probabilidade p por qubit e não há erros na medição da síndrome. Apesar desse modelo provavelmente não ser particularmente fiel aos erros que acontecem em sistemas físicos, ele possui a vantagem de permitir que os erros X e Z, e suas correções, sejam tratados separadamente.

Considere que a cada intervalo de tempo se execute a medição de todos os operador face e vértice do código, e que durante esses intervalos de tempo novos erros possam surgir, seguindo o modelo de ruído independente simples. Os erros em cada intervalo de tempo sobre um qubit com estado ρ podem ser representados pelo canal quântico

$$\rho \to (1-p)^2 I\rho I + p(1-p)X\rho X + p(1-p)Z\rho Z + p^2 Y\rho Y. \tag{4.1}$$

A fidelidade da síndrome permite que, após cada medição, possa ser realizado o processo de decodificação. A Figura 4.3 mostra como é possível representar o resultado da medição da síndrome espacialmente sobre o reticulado do código. Foram destacados, em cinza, os vértices e as faces dos operadores de verificação que retornam o autovalor -1 para as determinadas cadeias de erros Z, representadas em vermelho, e X, em azul.

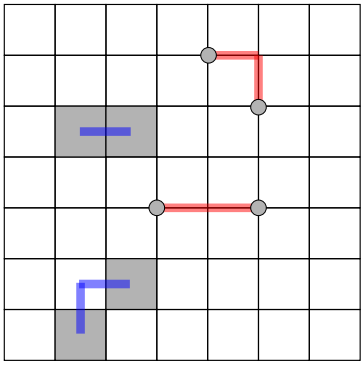


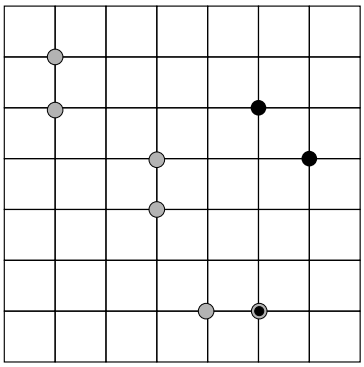
Figura 4.3 – Representação da síndrome de um código tórico

Infelizmente, também podem ocorrer erros nas medições. O segundo modelo de erros apresentado aqui, conhecido como *modelo fenomenológico*, incorpora a noção de ruído à medição da síndrome. Mais uma vez, é considerado que os erros X e Z não estão correlacionados e que acontecem a com probabilidade p. Porém, assume-se também que a probabilidade das medições dos operadores de verificação estarem incorretas é q e que os erros nos qubits e nas medições não são correlacionados no espaço, no tempo ou uns com os outros. Em geral, consideraremos que os valores das probabilidades de erros, p e q, são conhecidos.

A Figura 4.4 exemplifica o resultado de uma medição da síndrome do código tórico no modelo fenomenológico. Os vértices destacados em cinza representam operadores que identificaram a real existência de defeitos; em preto, os que identificaram defeitos fantasmas; e em cinza e preto, aquele que deveria ter identificado um erro, mas não o fez.

A estratégia desenvolvida por Dennis *et al.*[41] para lidar com os erros existentes nas medições consiste exatamente em repetir as medições a fim de obter maior confiança nas informações obtidas. Perceba, entretanto, que novos erros podem surgir tanto nos qubits quanto nas novas medições durante o processo. O procedimento formulado por Dennis *et al.*[41] contorna esse problema e consegue identificar e rejeitar erros fantasmas, como será apresentado na Seção 4.7.

Figura 4.4 – Exemplo de síndrome com erros de medição



Fonte: adaptado de Dennis *et al.*[41].

4.3.1 Modelos para representação de erros no espaço-tempo

Suponha que para um código tórico de tamanho arbitrariamente grande seja possível realizar a medição da síndrome de erro uma vez por intervalo de tempo. Considere também que o bloco de código seja monitorado por um tempo arbitrariamente longo e que todas as síndromes medidas sejam armazenadas. Uma questão que deve surgir é se essa informação armazenada sobre a síndrome será suficiente para permitir que a taxa de falha lógica diminua exponencialmente com o aumento do bloco do código ou, ainda, se será possível armazenar apenas um determinado número de síndromes anteriores e mesmo assim a taxa de falha decrescer. Felizmente, Dennis *et al.* [41] mostraram que a resposta para as duas perguntas é sim.

Para lidar com o modelo fenomenológico, Dennis *et al.*[41] sugerem a utilização de um modelo de representação do código tórico em um reticulado tridimensional. As duas primeiras dimensões correspondem ao reticulado do código tórico, enquanto a terceira, ortogonal ao plano formada pelas duas primeiras, refere-se a valores inteiros no tempo. Mais precisamente, é realizada uma discretização do tempo. Em [41] as duas dimensões relacionadas ao reticulado do código são denominadas horizontais, enquanto a terceira é denominada vertical. Manteremos essa nomenclatura, apesar de matematicamente não ser precisa.

Para fazer uso dessa representação, é considerado que os erros ocorrem nos valores inteiros de tempo t e que as medições da síndrome acontecem entre cada t e $t + 1$. Sendo a medição da síndrome realizada T vezes em sequência, o modelo será composto por T reticulados $L \times L$ empilhados verticalmente, iniciando em $t = 0$ e subindo sucessivamente até $t = T$. Conectam-se, então, os vértices equivalentes dos reticulados em camadas sucessivas. O resultado é um reticulado cúbico tridimensional.

Um erro em um qubit que ocorre no tempo t estará associado à aresta que corresponde ao respectivo qubit localizado na camada temporal t do reticulado. Por sua vez, a medição de um operador vértice $A(v)$ entre os tempos t e $t + 1$ está associada à

aresta que conecta os vértices v's das fatias de tempo t e $t+1$.

Uma representação equivalente vale para os operadores face $B(f)$, com o uso do reticulado dual. Como os erros X e Z são considerados independentes, então é possível tratar inicialmente os erros X e na sequência os erros Z, ou vice-versa. Como é considerada a existência de erros na medição dos operadores, então em algumas das arestas verticais a medida da síndrome estará incorreta. Os erros, sejam nos qubits ou na síndrome, são descritos por um conjunto de arestas marcadas no reticulado tridimensional.

Para apresentar visualmente o modelo, será utilizado um código de repetição de nove qubits que protege contra erros *phase-shift*, ao invés de um código tórico, assim como em [41]. A escolha é motivada em razão do modelo tridimensional do código tórico ser uma estrutura com $T \cdot L^2$ arestas, o que torna a sua representação visualmente sobrecarregada.

No código de repetição, os qubits serão considerados como residindo sobre um círculo. Para isso, considere nove vértices posicionados equidistantes sobre um círculo e nove arestas conectando-os. Cada qubit está associado a uma arestas e, portanto, possui dois vizinhos. Os erros de *phase-shift* nos qubits serão os únicos considerados. Os operadores de verificação são operadores vértices, da forma XX, atuando em qubits vizinhos. Para esse código, o modelo de representação dos erros no espaço-tempo é um reticulado com apenas duas dimensões, uma para o código e outra para o tempo.

As Figuras 4.5(a) e 4.5(b) são representações do modelo para esse código de repetição. Na Figura 4.5(a), as arestas verticais em verde representam a cadeia da síndrome Sdr, ou seja, os observáveis com autovalor -1 nas diversas medições ao longo do tempo. Esse é o resultado que se recebe ao medir o código repetidas vezes. Por sua vez, a Figura 4.5(b) representa, além da síndrome da Figura 4.5(a), também os erros que ocorreram, em vermelho. Como dito, os erros de medição são de dois tipos. Quando a síndrome indica a existência de um erro que na verdade não existe, ela é representada por uma aresta com as duas cores, verde e vermelho. Já uma aresta vertical apenas com o preenchimento vermelho representa uma medição que não indicou um erro que de fato existisse.

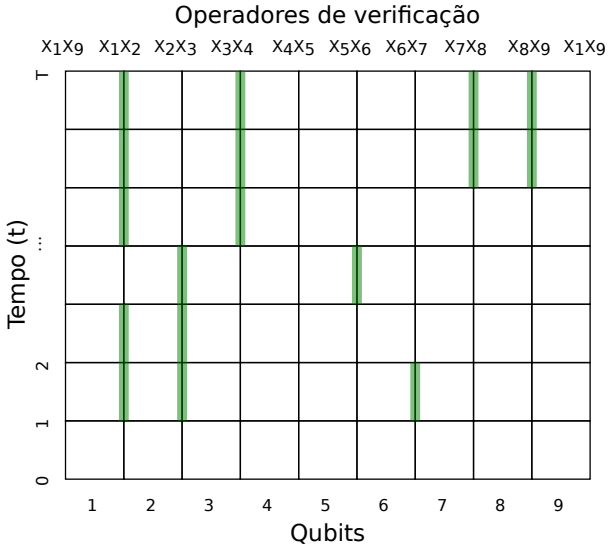
4.4 O limite de precisão e o decodificador ótimo

O limite de precisão p_c é a maior taxa de erros em um qubit p que pode ser tolerada por um código corretor. Se a probabilidade de erros p for tal que $p > p_c$, então a taxa de falha lógica do código crescerá juntamente com a distância do código, ao invés de decrescer. Por isso, é importante identificar o limite p_c para qualquer combinação de CQCE, modelo de erro e estratégia de decodificação.

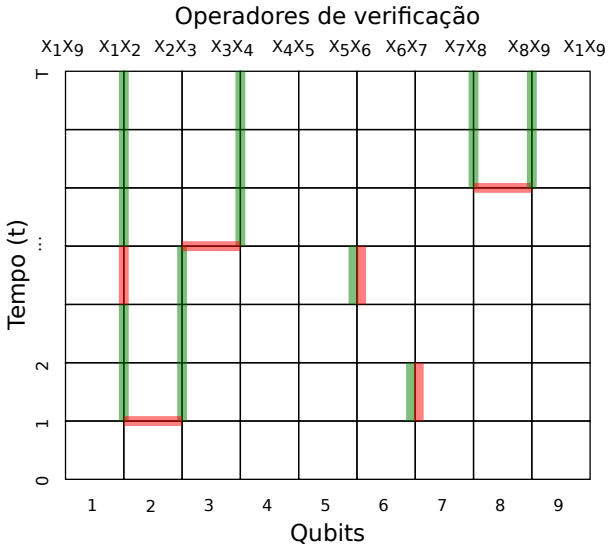
O maior limite de precisão possível para um determinado par de código e modelo de erro é chamado *limite ótimo* e é atingido por meio do uso de um decodificador

Figura 4.5 – Modelo fenomenológico para um código de repetição

(a) Síndrome de um código de repetição para o modelo fenomenológico.



(b) Síndrome e erros de um código de repetição para o modelo fenomenológico.



Fonte: adaptado de Dennis *et al.* [41].

específico, o *decodificador de máxima verossimilhança*. Para exemplificar, considere o modelo de ruído independente simples, ou seja, com a leitura da síndrome fidedigna e com taxa de erros por qubit p. Para um código tórico, a ideia que rege o decodificador de máxima verossimilhança é identificar a classe de homologia à qual há maior probabilidade de o erro ocorrido pertencer.

O primeiro passo do decodificador é identificar uma cadeia de operadores E' tal que $\partial E'$ seja igual à síndrome observada ∂E e não possua ciclos não triviais. Essa cadeia é uma candidata à cadeia de correção. Devido ao modelo de erros, a probabilidade

de ocorrência da cadeia E' é

$$P(E') = (1-p)^{n-|E'|}p^{|E'|},$$

em que $|E'|$ é o peso da cadeia E' e n é o total de qubits físicos do código.

Se a cadeia E' for multiplicada individualmente por cada um dos elementos do estabilizador S e a probabilidade correspondente a cada uma das multiplicações for somada, então é obtida a probabilidade de a cadeia de correção pertencer à classe de homologia trivial h_0:

$$P(E' \in h_0) = \sum_{M \in S} P(E' \cdot M).$$

Para calcular a probabilidade de a cadeia E' pertencer a outra classe de homologia h_i do toro, com $i = 1, 2, 3$, devemos multiplicar as cadeias $E' \cdot M$, com $M \in S$, por um elemento $C_i \in h_i$, de forma que

$$P(E' \in h_i) = \sum_{M \in S} P(E' \cdot M \cdot C_i).$$

Calcula-se, então,

$$P_{max}(E') = \max_{i \in 0,1,2,3} P(E' \in h_i).$$

Dado h_j tal que $P(E' \in h_j) = P_{max}(E')$, então devemos modificar E' multiplicando por C_j antes de aplicar a correção sobre o código, a fim de maximizar a probabilidade de recuperar corretamente o estado inicial. Esse procedimento pode ser generalizado facilmente para códigos de superfície, basta considerar todas as classes de homologia da superfície sobre a qual é construído o código.

O limite de precisão ótimo para o código tórico foi inicialmente calculado por Dennis *et al.* [41] usando um mapeamento do código tórico para um modelo mecânico estatístico chamado *modelo de Ising de ligação aleatória* (*random bond Ising model*) e os valores numéricos calculados por Honecker *et al.* [46]. O valor para o limite de precisão ótimo para o modelo de ruído independente encontrado utilizando tais técnicas foi $p_c = 0.1094 \pm 0.0002$. Em 2014, Bravyi *et al.* [47] descobriram um algoritmo exato eficiente para modelos de erro independente em códigos de superfície. A implementação desse decodificador ótimo encontrou um limite de precisão $0.109 \leq p_c \leq 0.11$, resultado consistente com aqueles provenientes do mapeamento do modelo de Ising de ligação aleatória.

Como dito anteriormente, deseja-se um limite de precisão elevado para um código corretor de erros, posto que, para todas as taxas de erro inferiores a esse limite, o

aumento do número de qubits físicos do código reduzirá a taxa de erros lógicos. Todavia, em um sistema real, o código deve operar em uma taxa bem abaixo do limite, já que o sistema não corresponderá perfeitamente aos modelos adotados.

4.5 Decodificadores sub-ótimos

No geral, o processo de decodificação ótima é muito custoso computacionalmente, devido ao grande número de probabilidades que precisam ser calculadas. Devido a isso, várias opções de decodificadores sub-ótimos foram criados, por exemplo: o decodificador de grupo de renormalização de decisão suave (*soft-decision renormalisation group*) [48], o decodificador de grupo de renormalização de decisão difícil (*hard-decision renormalisation group*) [49] e a versão de Edmonds do algoritmo de correspondência perfeita de peso mínimo (*minimum-weight perfect matching algorithm*) [50].

O algoritmo de Edmonds é citado como possível decodificador por Dennis *et al.* [41], enfatizando que ele pode ser executado em tempo polinomial dependente de L. Dados a ocorrência de uma cadeia de erros E e a cadeia da síndrome Sdr, esse algoritmo emparelha os defeitos na síndrome, que correspondem ao bordo ∂Sdr, por meio de uma cadeia de correção E' que possui o menor peso possível e satisfaz a condição $\partial Sdr = \partial E'$. Se a síndrome for fidedigna, então $Sdr = E$ e essa abordagem garante que o operador $C = E \cdot E'$ seja um ciclo.

Simulações numéricas feitas por Wang, Harrington e Preskill [51] sugerem que, para o algoritmo de Edmonds, o limite de precisão é $p_c = 0.1031 \pm 0.0001$. Quando $q = 0$, o algoritmo irá retornar a menor cadeia que conecta os defeitos apontados na síndrome, dois a dois. Porém, no caso em que $q \neq 0$, o resultado do algoritmo dependerá diretamente da relação de ordem entre p e q, pois as arestas do modelo tridimensional terão pesos diferentes. Na Seção 4.6 será apresentada uma função que pode ser utilizada para definir pesos para as arestas do reticulado tridimensional, em que o algoritmo de Edmonds opera.

Sempre que forem consideradas as medições da síndrome sem erros ($q = 0$), o processo de decodificação consistirá em aplicar o algoritmo decodificador sobre a síndrome. Definido o decodificador, se $p < p_c$, então sempre que se desejar diminuir a probabilidade de erros lógicos basta aumentar o tamanho do bloco de código. Uma estimativa para os limites de precisão pode ser obtida com o uso de simulações numéricas, como na tese de Watson [45]. A seguir, apresentaremos a prova dada por Dennis *et al.*[41] que garante a existência de limites de precisão para o código tórico, tanto no modelo de ruído independente simples quanto no fenomenológico, com tempo finito ou não.

4.6 Existência dos limites de precisão

A partir da síndrome do erro no modelo tridimensional é possível identificar a cadeia de erros que possui a menor "energia", ou seja, a maior probabilidade de ocorrência. Essa cadeia é denotada E_{min}. Dada a medição da síndrome, a cadeia de peso mínimo E_{min} pode ser determinada por um computador clássico em um tempo polinomial do número de vértices do reticulado com o uso do algoritmo de Edmonds [50].

A seguir, é apresentada a prova formulada por Dennis *et al.* [41] de que usar E_{min} como cadeia de correção é uma estratégia efetiva. Assim, o limite de precisão que calcularemos abaixo funciona como uma estimativa para o limite de precisão do algoritmo de Edmonds.

Uma cadeia de erros E que possua um total de $\mathcal{H}$ arestas horizontais e $\mathcal{V}$ arestas verticais ocorre com probabilidade proporcional à expressão:

$$\left(\frac{p}{1-p}\right)^{\mathcal{H}} \cdot \left(\frac{q}{1-q}\right)^{\mathcal{V}}, \tag{4.2}$$

onde p é a probabilidade de erro em um qubit e q é a probabilidade de erro na medição de um operador de verificação. No geral, $\left(\frac{p}{1-p}\right) \leq 1$ e $\left(\frac{q}{1-q}\right) \leq 1$.

Maximizar o valor da Expressão (4.2) é equivalente a minimizar o valor de

$$\mathcal{H} \log\left(\frac{1-p}{p}\right) + \mathcal{V} \log\left(\frac{1-q}{q}\right). \tag{4.3}$$

Então, deve-se escolher o E_{min} sob as condições de que $\partial E_{min} = \partial Sdr$ e minimize o valor da Equação (4.3).

Perceba que, na Expressão (4.3), os termos $\log\left(\frac{1-p}{p}\right)$ e $\log\left(\frac{1-q}{q}\right)$ funcionam como pesos para as arestas horizontais e verticais, respectivamente. Assim, se $p \neq q$, então os erros horizontais ou verticais possuirão pesos diferentes e a cadeia de peso mínimo E_{min} priorizará o tipo de aresta de menor peso. Se mais de uma cadeia possuir o peso mínimo, então uma delas será escolhida aleatoriamente.

A correção funcionará se o E_{min} for homologicamente equivalente ao erro E que aconteceu, e falhará em caso contrário. Considere um ciclo do reticulado tridimensional que possua $\mathcal{H}$ arestas horizontais, $\mathcal{V}$ verticais e pertença a $E + E_{min}$, onde $E + E_{min}$ denota a união disjunta dos conjuntos de arestas E e E_{min}. A cadeia $E + E_{min}$ é equivalente ao conjunto de arestas sobre o qual atua o operador $E_{min} \cdot E$. Essa é uma notação comum em teoria de grafos e será empregada com frequência ao longo desta e das próximas seções.

Denota-se por $\mathcal{H}_m$ e $\mathcal{V}_m$ o número de arestas do ciclo, horizontais e verticais, respectivamente, que estão contidas em E_{min}. E por $\mathcal{H}_e$ e $\mathcal{V}_e$ o número de arestas do mesmo ciclo contidas em E. Valem as seguintes relações:

$$\mathcal{H}_m + \mathcal{H}_e \geq \mathcal{H} \quad \text{e} \quad \mathcal{V}_m + \mathcal{V}_e \geq \mathcal{V}.$$

Segue diretamente das desigualdades acima que:

$$\left(\frac{p}{1-p}\right)^{\mathcal{H}_m} \cdot \left(\frac{q}{1-q}\right)^{\mathcal{V}_m} \cdot \left(\frac{p}{1-p}\right)^{\mathcal{H}_e} \cdot \left(\frac{q}{1-q}\right)^{\mathcal{V}_e} \leq \left(\frac{p}{1-p}\right)^{\mathcal{H}} \cdot \left(\frac{q}{1-q}\right)^{\mathcal{V}}. \tag{4.4}$$

Além disso, como E_{min} é uma cadeia de menor peso para o determinado bordo, valem que $\mathcal{H}_m \leq \mathcal{H}_e$ e $\mathcal{V}_m \leq \mathcal{V}_e$. Portanto:

$$\left(\frac{p}{1-p}\right)^{\mathcal{H}_e} \cdot \left(\frac{q}{1-q}\right)^{\mathcal{V}_e} \leq \left(\frac{p}{1-p}\right)^{\mathcal{H}_m} \cdot \left(\frac{q}{1-q}\right)^{\mathcal{V}_m}. \tag{4.5}$$

Combinando as Equações (4.4) e (4.5) tem-se:

$$\left(\frac{p}{1-p}\right)^{\mathcal{H}_e} \cdot \left(\frac{q}{1-q}\right)^{\mathcal{V}_e} \leq \left[\left(\frac{p}{1-p}\right)^{\mathcal{H}} \cdot \left(\frac{q}{1-q}\right)^{\mathcal{V}}\right]^{1/2}. \tag{4.6}$$

Dado um ciclo com $\mathcal{H}$ arestas horizontais e $\mathcal{V}$ verticais em $E + E_{min}$, existem $2^{\mathcal{H}+\mathcal{V}}$ possíveis formas de distribuir erros neste ciclo - cada aresta possui ou não um erro, ou seja, pertence ou não a E. Uma vez que as localizações dos erros estejam especificadas, a probabilidade de ocorrência da cadeia é:

$$p^{\mathcal{H}_e}(1-p)^{\mathcal{H}-\mathcal{H}_e} q^{\mathcal{V}_e}(1-q)^{\mathcal{V}-\mathcal{V}_e} = (1-p)^{\mathcal{H}}(1-q)^{\mathcal{V}} \left(\frac{p}{1-p}\right)^{\mathcal{H}_e} \cdot \left(\frac{q}{1-q}\right)^{\mathcal{V}_e}. \tag{4.7}$$

No entanto, uma vez que o ciclo está em $E + E_{min}$, é necessário que a Equação (4.6) seja satisfeita. Conclui-se que a probabilidade $P(\mathcal{H}, \mathcal{V})$ de um determinado ciclo com $\mathcal{H}$ arestas horizontais e $\mathcal{V}$ arestas verticais pertencer a $E + E_{min}$ é

$$
\begin{aligned}
P(\mathcal{H},\mathcal{V}) &= 2^{\mathcal{H}+\mathcal{V}}(1-p)^{\mathcal{H}}(1-q)^{\mathcal{V}}\left(\frac{p}{1-p}\right)^{\mathcal{H}_e}\left(\frac{q}{1-q}\right)^{\mathcal{V}_e} \\
&\leq 2^{\mathcal{H}+\mathcal{V}}(1-p)^{\mathcal{H}}(1-q)^{\mathcal{V}}\left[\left(\frac{p}{1-p}\right)^{\mathcal{H}}\left(\frac{q}{1-q}\right)^{\mathcal{V}}\right]^{1/2} \\
&= 2^{\mathcal{H}+\mathcal{V}}[p(1-p)]^{\mathcal{H}/2}[q(1-q)]^{\mathcal{V}/2} \\
&= (4\tilde{p})^{\mathcal{H}/2}(4\tilde{q})^{\mathcal{V}/2},
\end{aligned}
$$

em que

$$\tilde{p} = p(1-p) \quad \text{e} \quad \tilde{q} = q(1-q).$$

Desse modo, contando o número de arestas em um ciclo é possível definir um limitante superior para a probabilidade desse ciclo estar contido em $E + E_{min}$. De fato, esse limitante vale para a presença de qualquer cadeia com $(\mathcal{H},\mathcal{V})$ arestas em $E + E_{min}$, seja essa cadeia um ciclo ou não.

Uma outra interpretação para as cadeias de erros sobre o reticulado é uma espécie de passeio[1] sobre ele. O possível efeito de um desses passeios na informação codificada não depende apenas do seu tamanho, mas também do quão longe ele se desloca em relação ao ponto inicial. Em particular, o nosso interesse é verificar se um passeio fechado (que inicia e termina em um mesmo ponto) é homologicamente não trivial.

Cada um desses passeios visita cada aresta no máximo uma vez, mas será conveniente restringi-los mais. Denomina-se um passeio como uma *caminhada evasiva (CE)* (*self-avoiding walk*) se ele visitar cada vértice do reticulado no máximo uma vez (no caso de um ciclo, ele só revisita o ponto em que começa, e apenas para terminar o processo). Restringir a atenção às CE's não será prejudicial, pois, dado qualquer passeio que conecte dois vértices, sempre será possível encontrar uma CE eliminando ciclos do passeio.

Uma CE fechada é denominada como um *polígono auto-evitável (PA)* (*self-avoiding polygon*) caso não possua ciclos triviais. Similarmente às CE's, dado um passeio fechado homologicamente não trivial é possível obter um PA com a eliminação de algumas arestas.

Para calcular a probabilidade de erro por unidade de tempo (lembrando que foi assumido uma discretização do tempo no modelo de representação) no estado codificado é possível concentrar a atenção apenas em CE's que aconteçam entre duas fatias de tempo separadas por um tempo finito T. Na Seção 4.7 será visto por que é possível adotar $T \propto L$,

[1] Um passeio em um grafo é uma sequência de vértices, tal que, para vértices consecutivos sempre há uma aresta conectando-os no grafo.

onde é L é o lado do reticulado. Uma CE pode se iniciar em qualquer um dos L^2T vértices do reticulado tridimensional (novamente, é utilizada a independência entre os erros X e Z, permitindo tratar um reticulado, primal ou dual, por vez). Denotando por $\eta_{PA}(\mathcal{H},\mathcal{V})$ o número de PA's com $(\mathcal{H},\mathcal{V})$ arestas que iniciam em um determinado vértice, então a probabilidade $P_{PA}(\mathcal{H},\mathcal{V})$ de que $E+E_{min}$ contenha qualquer PA com $(\mathcal{H},\mathcal{V})$ arestas satisfaz:

$$\begin{aligned} P_{PA}(\mathcal{H},\mathcal{V}) &\leq L^2T\cdot\eta_{PA}(\mathcal{H},\mathcal{V})\cdot P(\mathcal{H},\mathcal{V}) \\ &= L^2T\cdot\eta_{PA}(\mathcal{H},\mathcal{V})\cdot(4\tilde{p})^{\mathcal{H}/2}(4\tilde{q})^{\mathcal{V}/2}. \end{aligned} \tag{4.8}$$

A informação quântica armazenada será danificada se $E+E_{min}$ contiver passeios homologicamente não triviais, os quais possuem no mínimo L arestas horizontais. Portanto, podemos determinar um limite para a probabilidade de falha P_{falha} no armazenamento a partir da Equação (4.8):

$$\begin{aligned} P_{falha} &\leq \sum_{V}\sum_{H\geq L} P_{PA}(\mathcal{H},\mathcal{V}) \\ &\leq L^2T\sum_{V}\sum_{H\geq L}\eta_{PA}(\mathcal{H},\mathcal{V})(4\tilde{p})^{\mathcal{H}/2}(4\tilde{q})^{\mathcal{V}/2}. \end{aligned} \tag{4.9}$$

É possível obter limites de precisão para o armazenamento de informação quântica com o uso de códigos tóricos através do estabelecimento de condições de garantam que com o aumento de L o limite superior da Equação (4.9) rapidamente se aproxime de zero e, portanto, a probabilidade de falha também se aproxime de zero. Para esta análise é necessário encontrar limites para o número de PA's com um determinado número de arestas verticais e horizontais, $\eta_{PA}(\mathcal{H},\mathcal{V})$.

O primeiro desses limites é obtido quando se desconsideram as diferenças entre arestas verticais e horizontais. De modo geral, dado um vértice inicial para um PA em um reticulado hipercúbico com d dimensões, o primeiro passo do passeio pode ser feito em qualquer uma das $2d$ direções. Para cada passo subsequente há, no máximo, $2d-1$ opções de direção, pois um passeio só pode visitar cada aresta uma única vez. Assim, para um passeio com um total de l arestas, vale que

$$\eta_{PA}^{(d)}(l) \leq 2d(2d-1)^{l-1},$$

onde $\eta_{PA}^{(d)}(l)$ é o número de PA's com l arestas em um reticulado hipecúbico com d dimensões. Para os casos bidimensional e tridimensional são conhecidos limites mais justos para $\eta_{PA}(l)$ [52, 53]:

$$\eta_{PA}^{(2)}(l) \leq P_2(l)(\mu_2)^l, \mu_2 \approx 2.638, \tag{4.10}$$

e

$$\eta_{PA}^{(3)}(l) \leq P_3(l)(\mu_3)^l, \mu_3 \approx 4.684, \tag{4.11}$$

onde $P_2(l)$ e $P_3(l)$ são polinômios [41].

Como um PA com $\mathcal{H}$ arestas horizontais e $\mathcal{V}$ arestas verticais possui um total de $l = \mathcal{H} + \mathcal{V}$ arestas, então é possível combinar as Equações (4.9) e (4.11) para obter uma nova cota superior para a probabilidade de falha:

$$\begin{aligned} P_{falha} &\leq L^2 T \sum_{\mathcal{V}} \sum_{\mathcal{H} \geq L} P_3(l)(\mu_3)^{\mathcal{H}+\mathcal{V}} (4\tilde{p})^{\mathcal{H}/2} (4\tilde{q})^{\mathcal{V}/2} \\ &= L^2 T \sum_{\mathcal{V}} \sum_{\mathcal{H} \geq L} P_3(l)(4\mu_3^2\tilde{p})^{\mathcal{H}/2} (4\mu_3^2\tilde{q})^{\mathcal{V}/2}. \end{aligned} \tag{4.12}$$

Caso tenhamos

$$\tilde{p} < (4\mu_3^2)^{-1} \quad \text{e} \quad \tilde{q} < (4\mu_3^2)^{-1}, \tag{4.13}$$

então vale para cada termo do somatório da Equação (4.12) que

$$(4\mu_3^2\tilde{p})^{\mathcal{H}/2} (4\mu_3^2\tilde{q})^{\mathcal{V}/2} \leq (4\mu_3^2\tilde{p})^{\mathcal{H}/2} \leq (4\mu_3^2\tilde{p})^{L/2},$$

já que $H \geq L$, $4\mu_3^2\tilde{q} < 1$ e $4\mu_3^2\tilde{p} < 1$. Como no reticulado existem $2L^2T$ arestas horizontais e L^2T arestas verticais, então o somatório sobre H e V pode ter no máximo $2L^2T \cdot L^2T = 2L^4T^2$ termos, de forma que

$$P_{falha} < Q_3(L,T) \cdot (4\mu_3^2\tilde{p})^{L/2}, \tag{4.14}$$

onde $Q_3(L,T)$ é um polinômio, pois é a soma finita dos polinômios $P_3(l)$.

Para que o código tórico consiga guardar informação quântica com confiabilidade arbitrariamente boa é suficiente que a probabilidade P_{falha} se torne arbitrariamente pequena quando L crescer, o que acontecerá na Equação (4.14), desde que T não cresça mais rápido que um polinômio em L.

Assim, obedecer às condições da Equação (4.13) é suficiente para que o armazenamento da informação no código tórico seja confiável. Numericamente, essas condições equivalem a:

$$p(1-p), q(1-q) \leq (87.8)^{-1} \approx 0.0113 \Longleftrightarrow p, q < 0.0114. \tag{4.15}$$

A Equação (4.14) estabelece um limite de precisão e garante que abaixo dele a probabilidade de falha decresça exponencialmente com o crescimento de L.

Nos casos em que q tente a zero, os únicos passeios que contribuem são PA's bidimensionais, pois ficam confinados em uma única fatia de tempo.

Partindo novamente da Equação (4.9), mas agora com a condição q tende a zero, pode-se realizar um desenvolvimento semelhante ao que resultou na Equação (4.14) utilizando a expressão da Equação (4.10). O resultado é

$$P_{falha} < Q_2(L,T) \cdot (4\mu_2^2 \tilde{p})^{L/2}, \tag{4.16}$$

onde $Q_2(L,T)$ é um polinômio. Devendo ser satisfeita a condição que

$$p(1-p) \leq (4\mu_2^2)^{-1} = (27.8)^{-1} \approx 0.0359 \Longleftrightarrow p < 0.0373. \tag{4.17}$$

Ou seja, se a medição da síndrome for garantida sem erros, será possível que p seja relaxado até 0.0373 e mesmo assim o aumento de L garantirá ganho de confiabilidade. É notável que o resultado da Equação (4.17) é consideravelmente menor do que os obtidos nas simulações numéricas realizadas por Wang *et al.* [51] ($p_c = 0.1031 \pm 0.0001$). Essa diferença não é uma surpresa, tendo em mente as grandes simplificações utilizadas na seção atual. Os cálculos acima, de Dennis *et al.* [41], funcionam mais como uma prova matemática da existência do limite de precisão do que como uma verdadeira estimativa para ele.

As simulações de Wang *et al.* sugerem que decodificar com base na cadeia de peso mínimo é muito mais eficiente do que a análise da seção atual e consideravelmente próximo do valor do limite de precisão ótimo obtido pelo modelo de Ising de ligação aleatória bidimensional, cujo valor é 0.1094 ± 0.0002.

É preciso, entretanto, ter em mente que, apesar de a performance do código aumentar rapidamente com o crescimento de L, o número total de qubits físicos também cresce, na ordem de L^2. Construir, armazenar e manipular recursos com essas dimensões não é uma tarefa trivial para a tecnologia atual. Então, deve-se levar em consideração o número de qubits físicos necessários para que se atinja o desempenho desejado pelo código.

4.7 A decodificação no modelo fenomenológico com tempo finito

Na seção anterior, quando o limite de precisão para o armazenamento de informação quântica no código tórico foi calculado, consideramos que as medições da síndrome fossem realizadas infinitas vezes, apesar de se buscar ciclos em um espaço confinado entre T fatias de tempo, sem estabelecer diretamente um início ou um fim para o processo. Ali, só foi necessário considerar os erros ocorridos durante o procedimento de medição da síndrome, desconsiderando-se a existência de qualquer erro presente quando o procedimento de correção se iniciou. A informação seria recuperada desde que $Sdr + E$ e $Sdr + E'$ fossem ciclos homologicamente equivalentes, em que a cadeia E é o conjunto de todas as arestas em que erros ocorreram, sejam de qubits ou de medição, e a cadeia E' é a estimativa obtida pelo decodificador.

Porém, essas simplificações não representam bem um sistema real. Por exemplo, durante uma interação entre qubits codificados em dois blocos de códigos tóricos, é possível que erros em um bloco se propaguem para outro bloco. Assim, para obter um histórico completo dos erros para qualquer um dos blocos de qubits, é preciso levar em conta a medição da síndrome dos dois blocos. Em princípio, seria possível levar todos os blocos em consideração. Porém, na prática, a realização da computação clássica necessária ao processo de correção seria complexa demais para ser realizada em tempo eficiente.

Para realizar computação tolerante a falhas com códigos tóricos é preciso realizar a correção de erros repetidamente. Uma vez que a medição da síndrome é imperfeita e a posição dos erros não pode ser determinada com exatidão, os erros que não foram corrigidos em uma certa etapa poderão causar problemas em etapas futuras. Algumas modificações no método de correção da seção anterior serão suficientes para que não seja preciso armazenar as síndromes por períodos de tempo muito longos e, ainda assim, conseguir corrigir a informação com sucesso.

Considere que a medição da síndrome é realizada T vezes seguidas, com início em $t = 0$. O resultado obtido é a cadeia de síndrome Sdr. Os erros que ocorrem durante esse processo, seja em qubits ou nas medições, fazem parte da cadeia E, a qual contém todos os erros que já estavam presentes quando o processo de medição da síndrome começou, sendo que tais erros podem ter sido deixados por etapas de correção anteriores. Assim, $Sdr + E$ conterá tanto cadeias fechadas quanto abertas, terminando na última etapa de tempo. Dizemos então que $Sdr + E$ é fechado relativamente à última etapa de tempo, ou simbolicamente $\partial_{rel}(Sdr + E) = 0$.

Os caminhos abertos contidos em $Sdr + E$ são de dois tipos:

(i) pares de defeitos criados anteriormente a $t = 0$ que persistiram até $t = T$;

(ii) pares de defeitos criados após $t = 0$ que se mantiveram até $t = T$.

A síndrome Sdr poderia ter sido gerada por qualquer erro E' com o mesmo bordo relativo de E. Portanto, o processo de correção deve selecionar uma cadeia de correção E' que seja provável, dada a síndrome observada Sdr. Um procedimento possível é escolher uma cadeia E' com $\partial_{rel}E' = \partial_{rel}Sdr$ que minimize a Equação (4.3). Novamente, é possível utilizar o algoritmo de Edmonds para alcançar esse objetivo.

Apesar de agora possuir uma estimativa para a cadeia de erros, ainda não há diretamente uma cadeia de correção: E' poderá apresentar a existência de erros de medição, ou mesmo de erros em qubits, que tenham se aniquilado naturalmente com o passar do tempo. Deve-se, então, projetar a cadeia $Sdr + E'$ na última fatia de tempo, $t = T$, operação denotada por $\prod(Sdr + E')$.

A projeção $\prod(Sdr+E')$ contém apenas as arestas horizontais que estão contidas em $Sdr + E'$ em um número ímpar de fatias de tempo. Perceba que E' possui a mesma projeção que $Sdr + E'$, pois a síndrome Sdr contém apenas arestas verticais e, portanto, sua projeção é trivial. A projeção $\prod(E')$ é a hipótese de quais qubits possuem erro no final das T medições. Após a construção de $\prod(E')$ devemos aplicar as portas X's e Z's nas arestas necessárias para compensar os erros presumidos. Perceba que na construção de E' não utilizamos diretamente a síndrome S, mas sim o seu bordo relativo $\prod(S)$.

Qualquer ciclo homologicamente trivial em $\prod(E')$ pode ser ignorado. Por outro lado, cada ciclo homologicamente não trivial modifica a informação codificada e corresponde a um operador lógico $\bar{X}$ ou $\bar{Z}$. Assim, após construir $\prod(E')$ é necessário compensar eventuais ciclos não triviais por meio de operadores $\bar{X}$ e/ou $\bar{Z}$. Fora os ciclos, as cadeias de erros sobreviventes na última fatia de tempo devem ser removidas por meio de cadeias de operadores X e Z que sejam homologicamente equivalentes às cadeias projetadas. Logo, essa etapa de recuperação consiste em emparelhar os defeitos, dois a dois, de forma que sua aniquilação não cause danos.

Claro que, mais uma vez, E' não irá necessariamente coincidir com E, de forma que $E + E'$ poderá conter cadeias fechadas relativas à última fatia de tempo. Os locais onde essas cadeias abertas encontram a última fatia de tempo corresponderão aos defeitos que o procedimento de correção não conseguiu remover.

4.7.1 Método de recuperação de sobreposição

Construir cadeias de peso mínimo E' com o mesmo bordo relativo de S' não parece ser um procedimento tão efetivo quanto continuar a medir a síndrome indefinidamente. Porém, é possível que um procedimento que guarda o histórico da síndrome por um tempo finito consiga resultados tão bons quanto aquele que a guarda por tempo infinito. A chave é reconhecer que as síndromes antigas são mais confiáveis que as síndromes recentes [41].

Percebendo isso, Dennis *et al.* [41] criaram o chamado *método de recuperação de sobreposição (overlapping recovery method)*, exposto a seguir.

Considere que a síndrome será medida $2T$ vezes em sequência, começando em $t = 0$. Constrói-se, então, E' com o mesmo bordo relativo que Sdr. No modelo tridimensional de representação, o bordo da síndrome representa o momento em que a medição de um determinado operador mudou de 1 para -1, e vice-versa. Cada vértice que compõe o bordo é denominado um *monopolo*.

A cadeia E' pode ser dividida em duas sub-cadeias disjuntas. A primeira parte consiste em todas as cadeias conectadas que começam e terminam em monopolos presentes no intervalo de tempo $0 \leq t < T$. Essa parte será denotada E'_v. O outro conjunto, formado pelas demais cadeias, será denotado por E'_m. Para recuperar a informação aplicam-se portas X's ou Z's sobre as arestas presentes na projeção $\prod(E'_v)$. Pode-se, então, apagar da memória o registro dos monopolos conectados por E'_v. Dessa forma, será necessário guardar apenas E'_m para realizar a próxima etapa de correção. De fato, apenas o bordo relativo de E'_m será mantido. Encerra-se, assim, a primeira etapa da correção.

A próxima etapa começa com a medição da síndrome por mais T vezes, ou seja, de $t = 2T$ até $t = 3T - 1$. Escolhe-se então uma cadeia de peso mínimo com bordo relativo à fatia de tempo $t = 3T - 1$, que novamente será denotada por E'. O bordo relativo de E' será a união do bordo relativo da síndrome Sdr no intervalo $2T \leq t < 3T$ e do bordo relativo de E'_m, da etapa anterior. De posse dessa nova cadeia E', a primeira etapa é novamente aplicada, corrigindo a síndrome mais antiga e mantendo a mais nova para a próxima etapa. Esse procedimento de correção é chamado método de recuperação de sobreposição, pois as cadeias de peso mínimo que são construídas em etapas sucessivas se sobrepõem em algumas regiões do espaço-tempo.

Se T for grande o suficiente, raramente um monopolo sobreviverá a mais de uma etapa de correção. Com isso, a quantidade de informação sobre a síndrome que precisará ser guardada será limitada. Além disso, para tal T, o método da recuperação de sobreposição possui performance próxima da que se teria com o armazenamento de uma quantidade indefinida de informação.

O procedimento de correção pelo método de recuperação de sobreposição irá diferir em eficácia do procedimento ótimo com memória infinita por uma quantidade insignificante, desde que se tenha [41]

$$T \gg \frac{L}{2} \cdot \frac{\log(\mu_3^2 p(1-p))^{-1}}{\log(\mu_3^2 q(1-q))^{-1}}. \tag{4.18}$$

Em particular, se $p \approx q$, é suficiente escolher $T \gg L$, onde L é o lado do reticulado. Portanto, o histórico de síndrome não precisa ser guardado infinitamente para que o processo de correção seja robusto. A chave para conseguir tolerância a falhas no

código tórico é não reagir imediatamente a informações da síndrome que potencialmente possam ser falsas. Se, durante a reconstrução dos possíveis erros ocorridos a partir da síndrome, forem encontradas cadeias abertas que pouco se estendem em direção ao passado, é perigoso aceitar a veracidade dessas cadeias e tentar corrigi-las. Por outro lado, cadeias que persistem por tempo comparável a L são confiáveis. O método de recuperação de sobreposição incorpora essa ideia, já que só se age sobre defeitos mais antigos, deixando os erros recentes para serem corrigidos na próxima etapa.

4.8 Exemplos

A fim de, esclarecer possíveis dúvidas que tenham surgido ao longo do capítulo, serão apresentados dois exemplos. O primeiro, é uma continuação daquele presente na Seção 3.5 do capítulo anterior. Para esse código, será considerado o modelo de ruído independente simples com a síndrome fidedigna. Na sequência, será considerado o código de repetição de nove qubits apresentado na Seção 4.3.1. Para ele, o modelo de erro considerado será o fenomenológico.

4.8.1 O código tórico [[50, 2, 5]] com síndrome fidedigna

Considere novamente o código tórico $[[50, 2, 5]]$, construído sobre um reticulado de lado $L = 5$. Considere, também, que as probabilidades de erros X e Z por qubit são equiprováveis e iguais a $p = 0.1$. A cadeia de correção utilizada será a de peso mínimo, como as obtidas pelo algoritmo de Edmonds, que, como dito, segundo Wang, Harrington e Preskill [51], possui um limite de precisão de $p_c = 0.1031 \pm 0.0001$. Nessas condições, se houver necessidade de aumentar a confiabilidade do sistema, basta aumentar o lado do reticulado L, já que $p < p_c$.

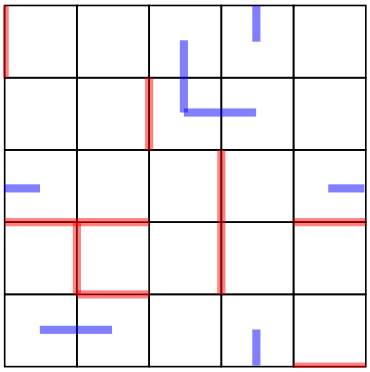
Utilizando o software R, foram gerados inicialmente 50 números aleatórios, seguindo uma distribuição de Bernoulli com parâmetro $p = 0.1$, o que representa a probabilidade de ocorrência de um erro Z em cada aresta do reticulado. Na sequência, foram gerados mais 50 números aleatórios, seguindo a mesma distribuição, associados a erros X. O resultado está representado na Figura 4.6: em vermelho, os erros Z no reticulado primal; e em azul, os erros X no reticulado dual.

Então, a síndrome do código é apresentada na Figura 4.7. Lembramos que durante a decodificação não se sabe qual a real cadeia de erros que ocorreu: a única informação que se tem em mãos é a síndrome.

Uma possível cadeia de correção E' de peso mínimo é a exemplificada na Figura 4.8.

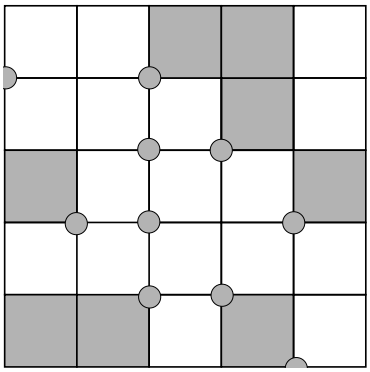
Por fim, a Figura 4.9 apresenta uma informação à qual durante a correção não

Figura 4.6 – Exemplo de erros no código tórico $[[50, 2, 5]]$



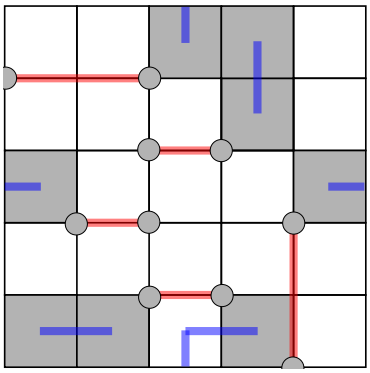
Fonte: elaborado pelo autor.

Figura 4.7 – Síndrome para a cadeia de erros do exemplo do código $[[50, 2, 5]]$



Fonte: elaborado pelo autor.

Figura 4.8 – Cadeia de correção para a síndrome para o exemplo do código $[[50, 2, 5]]$

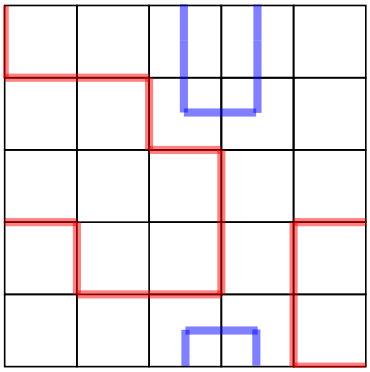


Fonte: elaborado pelo autor.

se possuía: o resultado da combinação da cadeia de erro E e da cadeia de correção E'. No exemplo, como não houve a criação de ciclos não triviais, o sistema retornou para o estado inicial, anterior aos erros.

Um exemplo construído com o código tórico com os erros seguindo o modelo fenomenológico seria sem dúvidas valioso. Porém, a representação visual, mesmo de um código pequeno, por exemplo, para $L = 5$, é uma tarefa ingrata. O grande número de

Figura 4.9 – Cadeia $C = E + E'$ para o exemplo do código $[[50, 2, 5]]$



Fonte: elaborado pelo autor.

arestas não permitiria a compreensão da imagem. Recorre-se, então, ao código de repetição de nove qubits que foi utilizado no Capítulo 4. Entretanto, a falta de resultados específicos para esse código não permitem maior rigor. Por exemplo, a Equação 4.18 que define o número de medições da síndrome no método de recuperação de sobreposição vale para o código tórico, mas não para o código de repetição. Apesar disso, imagina-se que o exemplo será benéfico à compreensão.

4.8.2 O código de repetição contra erros *phase-shift* com o modelo fenomenológico

Considere o código de repetição de nove qubits que protege contra erros *phase-shift*, mencionado na Subseção 4.3.1. Serão considerados apenas os erros *phase-shift*, ocorrendo a uma probabilidade de erros por qubit $p = 0.1$. A probabilidade de erro na síndrome q será igual à probabilidade p. O número de medições da síndrome por etapa T será tomado igual a 5, sem motivação específica.

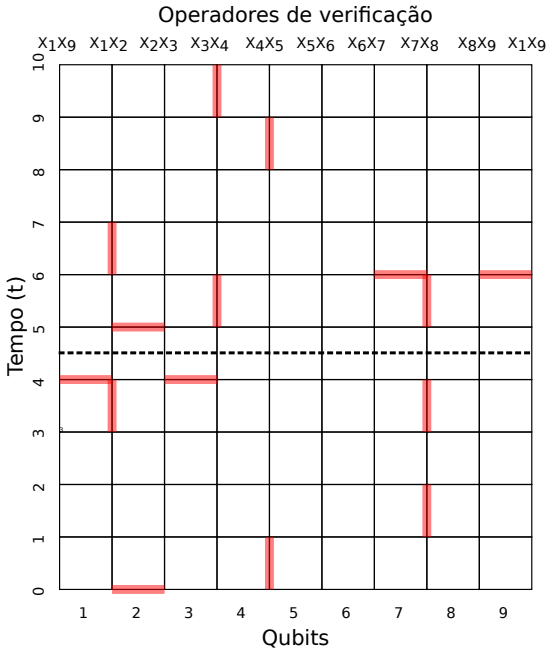
Com o uso do software R foram gerados, na primeira etapa, 90 números aleatórios seguindo uma distribuição de Bernoulli com parâmetro $p = 0.1$, correspondentes à ocorrência de erros Z durante as $2T$ etapas de tempo. Na segunda etapa, foram gerados mais 90 números aleatórios seguindo a mesma distribuição, correspondentes aos erros na síndrome. O resultado está apresentado na Figura 4.10. A linha tracejada separa as primeiras T fatias de tempo das últimas T. Os erros que estão presentes ao final das $2T$ medições estão nos qubits $1, 3, 7$ e 9, pois no qubit 2 ocorreram dois erros, que acabam se anulando.

Como resultado, a cadeia de síndrome Sdr é igual à Figura 4.11.

Os monopolos dessa cadeia de síndrome, ou seja, o seu bordo relativo $\partial_{rel} Sdr$, estão presentes na Figura 4.12.

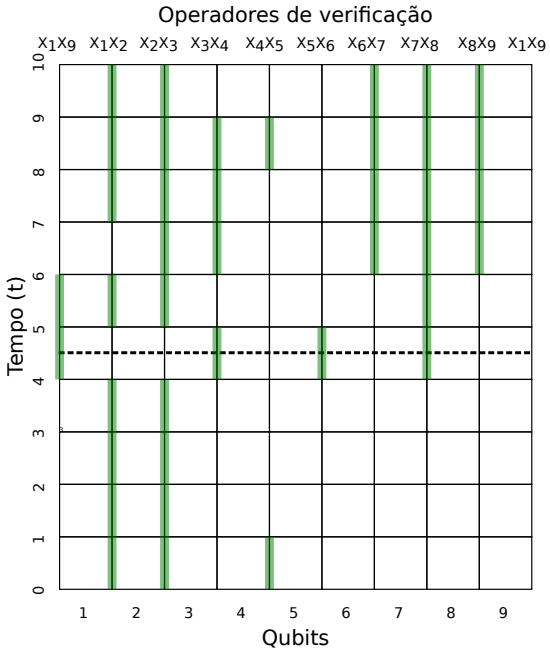
Como visto, a etapa de correção passa por encontrar uma cadeia E' com mesmo

Figura 4.10 – Exemplo de erros no código de repetição.



Fonte: elaborado pelo autor.

Figura 4.11 – Síndrome para a cadeia de erros do exemplo do código de repetição.

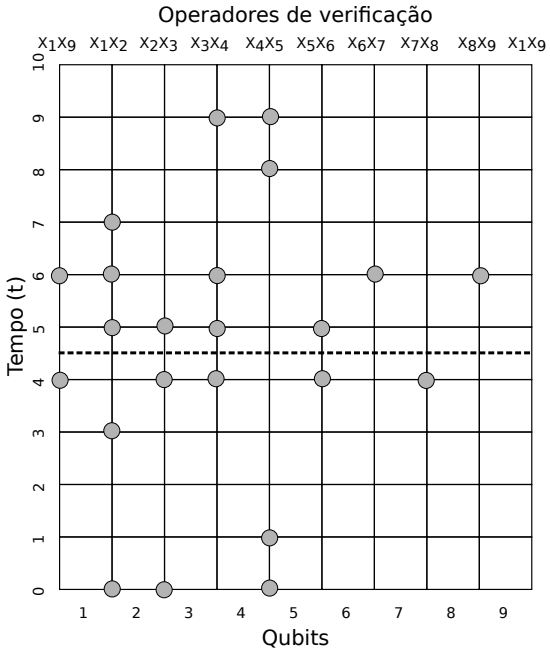


Fonte: elaborado pelo autor.

bordo relativo que Sdr. Uma possível escolha para E' está exposta na Figura 4.13.

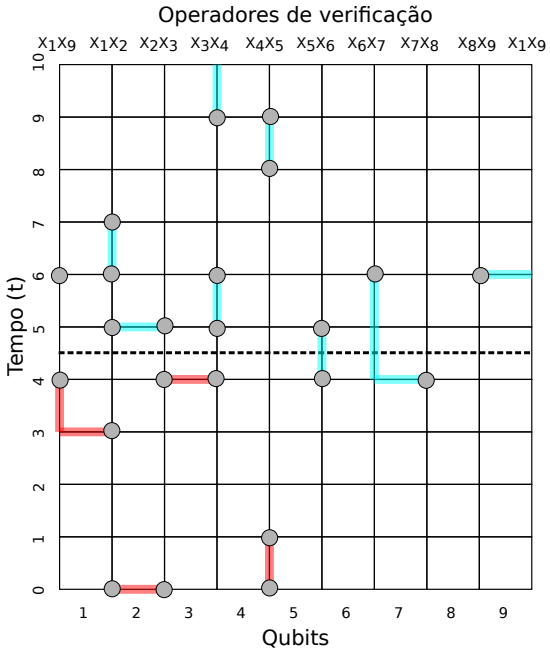
Na Figura 4.13 foram destacadas em vermelho as cadeias que ligam monopolos que se encontram em $0 \leq t < T$, ou seja, E'_v. Por sua vez, em azul celeste estão as cadeias pertencentes a E'_m. A cadeia de correção é dada pela projeção $\prod(E'_v)$, que corresponde aos qubits em que há uma incidência de um número ímpar de erros em E'_v. No exemplo, eles

Figura 4.12 – Monopolos para a cadeia de síndrome do código de repetição.



Fonte: elaborado pelo autor.

Figura 4.13 – Cadeia de correção para a síndrome do código tórico



Fonte: elaborado pelo autor.

correspondem aos qubits $1, 2$ e 3. Deve-se, então, aplicar a porta Z sobre esses qubits e continuar com o método de correção, fazendo mais T medições.

Perceba que os qubits que o método de recuperação de sobreposição indicou a aplicação das portas Z não correspondem exatamente aos erros observados na Figura 4.10, que eram $1, 3, 7$ e 9. Porém, a aplicação de Z sobre o segundo qubit não corresponde

necessariamente a um erro, lembre que E'_m é mantido para a próxima etapa de correção, e é provável que nessa nova etapa seja novamente aplicada uma operação sobre o segundo qubit e com isso não haja dano. Perceba que a projeção da cadeia $\prod(E')$ não corresponde diretamente aos qubits $1, 3, 7$ e 9, mas como a síndrome que reflete a existência dos erros nos qubits 7 e 9 é mais recente, o método decide por tardar essa correção, esperando que as próximas síndromes confirmem a existência desses erros.

5 Conclusões

Faremos neste capítulo um breve sumário do que foi abordado nesta dissertação.

No Capítulo 2, reunimos os tópicos necessários à compreensão do código tórico. Começamos definindo e enunciando conceitos fundamentais da mecânica quântica e apresentando os quatro postulados responsáveis por conectar o mundo físico ao formalismo da Mecânica Quântica. Em seguida, introduzimos alguns conceitos gerais da teoria de codificação quântica, tais como uma condição necessária e suficiente para que um código consiga corrigir um determinado conjunto de erros e o significado dos parâmetros na notação $[[n, k, d]]$.

Além dos conceitos gerais sobre códigos quânticos, também enfocamos nas propriedades de uma das classes mais abrangentes, os códigos estabilizadores. O restante do capítulo detalhou estruturas matemáticas específicas utilizadas na construção dos códigos tóricos: teoria de homologia e co-homologia, reticulados, tesselações e superfícies. Apesar de cada uma dessas áreas ser vasta e possuir por si só vários resultados interessantes, procuramos limitar cada uma delas ao necessário para os próximos capítulos.

No Capítulo 3 apresentamos formalmente o código tórico. Utilizamos duas abordagens, a primeira mediante a teoria de homologia/co-homologia e a segunda com ênfase nos *anyons*. Naturalmente, cada abordagem possui seus pontos fortes e fracos. A primeira enxerga a relação entre o código e os elementos homológicos, como cadeias, bordos, ciclos e grupos de homologia. Como consequência, as conclusões que surgem na teoria de homologia servem diretamente para o código tórico. Essa abordagem encara o código tórico mais como uma estrutura matemática do que como um sistema físico, fazendo uso da teoria matemática já bem fundamentada para compreender o seu funcionamento. Além disso, as relações entre operadores e reticulados primais e duais ficam claras a partir da ligação entre a homologia e a co-homologia.

Por sua vez, a apresentação com base nos *anyons* mostra de que modo os estados fundamentais do hamiltoniano do código tórico funcionam como os estados lógicos para o código e como a ocorrência de erros, além da consequente criação de pares de *anyons*, é penalizada energeticamente, fazendo com que o código tórico proteja a informação armazenada em nível físico. Entendemos que o uso da segunda abordagem permite uma compreensão física de como e porquê o código funciona. Assim, as duas apresentações se complementam e ajudam a entender o todo.

O Capítulo 3 trouxe ainda a versão do código tórico construída sobre um reticulado hexagonal e o código planar, originado de uma introdução de bordos sobre o reticulado do código tórico. Esses dois códigos mostram como os conceitos e ideias apresentados por

Kitaev[24] não se limitam ao código tórico construído sobre o reticulado quadrado. Ao longo do capítulo, buscamos expandir algumas demonstrações e comentários da literatura, tendo como objetivo tornar o entendimento mais fácil para o leitor. Construímos, ainda, um exemplo, a fim de apresentar os conceitos de forma menos teórica e mais fácil de absorver.

Por fim, no Capítulo 4, introduzimos o problema da decodificação, que em sua essência busca encontrar um operador de correção que possua máxima probabilidade de retornar o código para o estado inicial. Vimos que o algoritmo de decodificação e o modelo de erros adotado influenciam o limite de precisão de um código, valor abaixo do qual a taxa de erros precisa ser mantida a fim de que, com o aumento do número de qubits físicos, cresça também a confiabilidade do código. Uma das conclusões do capítulo foi que o código tórico consegue proteger bem a informação armazenada, mesmo com a sua distância sendo pequena em relação ao número de qubits físicos do sistema - devido à distribuição espacial dos erros. Introduzimos, também, o método formulado por Dennis *et al.*[41] para lidar com a existência de erros na síndrome. Para finalizar, formulamos dois exemplos, um para cada modelo de erros apresentado, indicando como seriam as etapas do processo de decodificação.

Assim como qualquer CQCE, o código tórico possui pontos positivos e negativos. Sem dúvida, pesa contra ele a impossibilidade de utilização de um conjunto de portas universais sobre a informação armazenada e as dificuldades tecnológicas atuais para manter tantos qubits agrupados. Por outro lado, esses sistemas possuem uma predisposição para serem utilizadas como memórias quânticas robustas. Além disso, o entendimento do código tórico funciona como uma porta de entrada para o estudo dos demais códigos topológicos. Esta pesquisa introduz e traz familiaridade a temas recorrentes ao estudo de códigos quânticos.

Esta dissertação teve como objetivo principal o estudo e a pesquisa do código tórico, assim como a consequente construção de um material que servisse como uma boa referência inicial ao tema em língua portuguesa. É digno de nota que o tema da decodificação em códigos quânticos corretores de erros é pouco abordado na literatura em língua portuguesa em geral, por isso, o capítulo dedicado ao assunto merece destaque. Outra contribuição dos autores são as diversas ilustrações apresentadas ao longo do texto, algumas como adaptações de imagens já apresentadas na literatura, com as devidas referências apontadas, e outras elaboradas pelos autores a fim de permitir melhor compreensão dos temas abordados. Assim, acreditamos que os objetivos tenham sido alcançados.

Referências

1 KASPAROV vs. Deep Blue: O Confronto Que Mudou a História. **CHESS.COM**, 24 out. 2018. Disponível em: <https://www.chess.com/pt/article/view/kasparov-vs-deep-blue-o-confronto-que-mudou-a-historia>. Acesso em: 04 jan. 2022.

2 FEYNMAN, R. P. Simulating physics with computers. **International Journal of Theoretical Physics**, v. 21, p. 467–488, 1982.

3 DEUTSCH, D. Quantum theory, the church-turing principle and the universal quantum computer. **Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences**, The Royal Society, v. 400, n. 1818, p. 97–117, 1985.

4 SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. **SIAM Journal on Computing**, Society for Industrial and Applied Mathematics, v. 26, n. 5, p. 1484–1509, 1997.

5 GROVER, L. K. A fast quantum mechanical algorithm for database search. In: **Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing**. New York: Association for Computing Machinery, 1996. p. 212–219.

6 JORDAN, S. **Quantum Algorithm Zoo**. 2011. Disponível em: <https://quantumalgorithmzoo.org/>. Acesso em: 30 nov. 2021.

7 GOOGLE quantum ai. **Google**. Disponível em: <https://quantumai.google/>. Acesso em: 03 jan. 2022.

8 IBM Quantum Experience. **IBM**. Disponível em: <https://quantum-computing.ibm.com/>. Acesso em: 03 jan. 2022.

9 QUANTUM Computing - Intel. **Intel**. Disponível em: <https://www.intel.com/content/www/us/en/research/quantum-computing.html>. Acesso em: 03 jan. 2022.

10 QUANTUM Computing - Microsoft Research. **Microsoft**. Disponível em: <https://www.microsoft.com/en-us/research/research-area/quantum-computing>. Acesso em: 03 jan. 2022.

11 QUANTUM Lab - DAMO Academy - Alibaba. **Alibaba**. Disponível em: <https://damo.alibaba.com/labs/quantum>. Acesso em: 03 jan. 2022.

12 ARUTE, F. et al. Quantum supremacy using a programmable superconducting processor. **Nature**, v. 574, n. 7779, p. 505–510, 2019.

13 PEDNAULT, E. et al. Leveraging secondary storage to simulate deep 54-qubit sycamore circuits. 2019. Disponível em: <https://arxiv.org/abs/1910.09534>.

14 GAMBETTA, J. **IBM's roadmap for scaling quantum technology**. Disponível em: <https://research.ibm.com/blog/ibm-quantum-roadmap>. Acesso em: 03 jan. 2022.

15 CHOW, J.; DIAL, O.; GAMBETTA, J. **IBM Quantum breaks the 100-qubit processor barrier**. Disponível em: <https://research.ibm.com/blog/127-qubit-quantum-processor-eagle>. Acesso em: 03 jan. 2022.

16 PACHOS, J. K. **Introduction to Topological Quantum Computation**. New York: Cambridge University Press, 2012.

17 NIELSEN, M. A.; CHUANG, I. L. **Quantum Computation and Quantum Information:**. 10. ed. New York: Cambridge University Press, 2011.

18 SHOR, P. W. Scheme for reducing decoherence in quantum computer memory. **Phys. Rev. A**, American Physical Society, v. 52, p. 2493–2496, 1995.

19 STEANE, A. M. Multiple-particle interference and quantum error correction. **Proceedings: Mathematical, Physical and Engineering Sciences**, The Royal Society, v. 452, n. 1954, p. 2551–2577, 1996.

20 CALDERBANK, A. R.; SHOR, P. W. Good quantum error-correcting codes exist. **Physical Review A**, American Physical Society (APS), v. 54, n. 2, p. 1098–1105, 1996.

21 STEANE, A. M. Error correcting codes in quantum theory. **Phys. Rev. Lett.**, American Physical Society, v. 77, p. 793–797, 1996.

22 KETKAR, A. et al. Nonbinary stabilizer codes over finite fields. **IEEE Transactions on Information Theory**, v. 52, n. 11, p. 4892–4914, 2006.

23 GOTTESMAN, D. Class of quantum error-correcting codes saturating the quantum hamming bound. **Physical Review A**, American Physical Society (APS), v. 54, n. 3, p. 1862–1868, 1996.

24 KITAEV, A. Fault-tolerant quantum computation by anyons. **Annals of Physics**, Elsevier BV, v. 303, n. 1, p. 2–30, 2003.

25 SATZINGER, K. J. et al. Realizing topologically ordered states on a quantum processor. **Science**, American Association for the Advancement of Science (AAAS), v. 374, n. 6572, p. 1237–1241, dec. 2021. ISSN 1095-9203. Disponível em: <http://dx.doi.org/10.1126/science.abi8378>.

26 PRESKILL, J. **Lecture notes for Physics 219: Quantum computation**. [S.l.]: California Institute of Technology, 2004.

27 CONWAY, J. H.; SLOANE, N. J. A. **Sphere-Packings, Lattices, and Groups**. New York: Springer-Verlag, 1998.

28 GHYKA, M. **The Geometry of Art and Life**. New York: Dover Publications, Inc., 1977.

29 FIRBY, P.; GARDINER, C. **Surface Topology**. Cambridge: Woodhead Publishing Limited, 2011.

30 LIMA, E. L. **Espaços métricos**. Rio de Janeiro: Instituto de Matemática Pura e Aplicada, 1983.

31 KAYE, P.; LAFLAMME, R.; MOSCA, M. **An Introduction to Quantum Computing**. 1st. ed. New York: Oxford University Press, 2007.

32 GOTTESMAN, D. **Stabilizer Codes and Quantum Error Correction**. 114 p. Tese (Doutorado) — California Institute of Technology, California, 1997.

33 BENNETT, C. H. et al. Mixed-state entanglement and quantum error correction. **Physical Review A**, American Physical Society, v. 54, n. 5, p. 3824–3851, 1996.

34 KNILL, E.; LAFLAMME, R.; VIOLA, L. Theory of quantum error correction for general noise. **Physical Review Letters**, American Physical Society, v. 84, n. 11, p. 2525–2528, 2000.

35 MUNKRES, J. R. **Topology / James R. Munkres.** 2nd ed. ed. Upper Saddle River, NJ: Prentice Hall, Inc., 2000.

36 CARMO, M. P. do. **Geometria diferencial de curvas e superfícies**. 1 ed. ed. Rio de Janeiro, RJ: Sociedade Brasileira de Matemática, 2005.

37 BOMBIN, H.; MARTIN-DELGADO, M. A. Homological error correction: Classical and quantum codes. **Journal of Mathematical Physics**, AIP Publishing, v. 48, n. 5, 2007. ISSN 1089-7658. Disponível em: <http://dx.doi.org/10.1063/1.2731356>.

38 BOMBIN, H. **An Introduction to Topological Quantum Codes**. 2013.

39 LIMA, E. L. **Homologia básica**. Rio de Janeiro: Instituto de Matemática Pura e Aplicada, 2014.

40 ARTIN, M. **Algebra**. [S.l.]: Pearson Prentice Hall, 2011.

41 DENNIS, E. et al. Topological quantum memory. **Journal of Mathematical Physics**, AIP Publishing, v. 43, n. 9, p. 4452–4505, 2002. ISSN 1089-7658. Disponível em: <http://dx.doi.org/10.1063/1.1499754>.

42 BRAVYI, S. B.; KITAEV, A. Y. **Quantum codes on a lattice with boundary**. 2008.

43 FREEDMAN, M. H.; MEYER, D. A. **Projective plane and planar quantum codes**. 2008.

44 Albuquerque, C. D.; Palazzo Jr., R.; Silva, E. B. Topological quantum codes on compact surfaces with genus g>=2. **Journal of Mathematical Physics**, v. 50, n. 2, p. 023513–023513, feb. 2009.

45 WATSON, F. H. E. **Performance of Topological Codes for Quantum Error Correction**. Tese (Doutorado) — Imperial College London, Londres, 2015.

46 HONECKER, A.; PICCO, M.; PUJOL, P. **Nishimori point in the 2D ± J random-bond Ising model**. 2001.

47 BRAVYI, S.; SUCHARA, M.; VARGO, A. Efficient algorithms for maximum likelihood decoding in the surface code. **Physical Review A**, American Physical Society (APS), v. 90, n. 3, sep. 2014. ISSN 1094-1622. Disponível em: <http://dx.doi.org/10.1103/PhysRevA.90.032326>.

48 DUCLOS-CIANCI, G.; POULIN, D. **A renormalization group decoding algorithm for topological quantum codes**. 2010.

49 BRAVYI, S.; HAAH, J. Quantum self-correction in the 3d cubic code model. **Physical Review Letters**, American Physical Society (APS), v. 111, n. 20, 2013. ISSN 1079-7114. Disponível em: <http://dx.doi.org/10.1103/PhysRevLett.111.200501>.

50 EDMONDS, J. Paths, trees, and flowers. **Canadian Journal of Mathematics**, Cambridge University Press, v. 17, p. 449–467, 1965.

51 WANG, C.; HARRINGTON, J.; PRESKILL, J. Confinement-higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory. **Annals of Physics**, Elsevier BV, v. 303, n. 1, p. 31–58, jan. 2003. ISSN 0003-4916. Disponível em: <http://dx.doi.org/10.1016/S0003-4916(02)00019-2>.

52 VANDERZANDE, C. **Lattice Models of Polymers**. [S.l.]: Cambridge University Press, 1998. (Cambridge Lecture Notes in Physics).

53 MADRAS, N.; SLADE, G. The self-avoiding walk. In: . [S.l.: s.n.], 1991.