UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE MATEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA
(Mestrado)

CELSO TOSHIO TAKAHASHI

CÓDIGOS COLORIDOS POLIGONAIS

Maringá-PR
2019

CELSO TOSHIO TAKAHASHI

Dissertação apresentada ao Programa de Pós-Graduação em Matemática do Departamento de Matemática, Centro de Ciências Exatas da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Matemática.

Área de concentração: Matemática Aplicada

Orientador: Dr. Eduardo Brandani da Silva

Maringá

2019

Dados Internacionais de Catalogação na Publicação (CIP)
(Biblioteca Setorial BSE-DMA-UEM, Maringá, PR, Brasil)

T136c

Takahashi, Celso Toshio
Códigos coloridos poligonais / Celso Toshio Takahashi. -- Maringá, 2019.
63 f. : il. color.

Orientador: Profº. Drº. Eduardo Brandani da Silva.
Dissertação (mestrado) - Universidade Estadual de Maringá, Centro de Ciências Exatas, Programa de Pós-Graduação em Matemática - Área de Concentração: Matemática Aplicada, 2019.

1. Códigos quânticos corretores de erros. 2. Códigos coloridos em superfícies compactas. 3. Mecânica quântica. 4. Códigos coloridos com bordos. I. Silva, Eduardo Brandani da, orient. II. Universidade Estadual de Maringá. Centro de Ciências Exatas. Programa de Pós-Graduação em Matemática - Área de Concentração: Matemática Aplicada. III. Título.

CDD 22.ed. 003.54

Edilson Damasio CRB9-1.123

CELSO TOSHIO TAKAHASHI

CÓDIGOS COLORIDOS POLIGONAIS

Dissertação apresentada ao Programa de Pós-Graduação em Matemática do Departamento de Matemática, Centro de Ciências Exatas da Universidade Estadual de Maringá, como parte dos requisitos necessários para a obtenção do título de Mestre em Matemática tendo a Comissão Julgadora composta pelos membros:

COMISSÃO JULGADORA:

Prof. Dr. Eduardo Brandani da Silva
DMA/Universidade Estadual de Maringá (Presidente)

Prof. Dr. Waldir Silva Soares Júnior
Universidade Tecnológica Federal do Paraná – Pato Branco

Prof. Dr. Emerson Vitor Castelani
Universidade Estadual de Maringá

Aprovada em 25 de fevereiro de 2019.
Local de defesa: Sala 107, Bloco F67, campus da Universidade Estadual de Maringá.

"Nada é suficientemente bom. Então vamos fazer o que é certo, dedicar o melhor de nossos esforços para atingir o inatingível, desenvolver ao máximo os dons que Deus nos concedeu, e nunca parar de aprender". (Ludwig van Beethoven)

À minha família

Agradecimentos

Agradeço à minha família, que mesmo de longe sempre me apoiaram e estavam disposto a me auxiliar a qualquer momento necessário. Sem seu apoio não haveria chegado aonde cheguei.

Agradeço aos amigos Anderson e Edilaine que se tornaram família, compartilhando momentos bons e ruins, definitivamente tornaram minha estadia em Maringá extremamente mais agradável. Agradeço também ao Roger, sua ajuda foi essencial durante os estudos e ao Evandro que compartilhou da jornada que foi desenvolver esta dissertação.

Claro que não posso deixar de agradecer ao meu orientador Eduardo, que não se mostrou ser somente um bom professor, mas também um ótimo amigo, sempre desenvolvendo um bom relacionamento com seus orientandos e se preocupando conosco. Sua paixão, dedicação e carinho são admiráveis.

Por fim, agradeço a CAPES e ao IMPA pelo apoio financeiro, essencial para a conclusão desta caminhada.

Resumo

Neste trabalho estudamos a construção dos chamados códigos coloridos poligonais. A ideia por trás desses códigos é expandir os códigos triangulares introduzidos por Bombín e Martin-Delgado, em 2007. Os códigos coloridos triangulares são construídos sobre uma superfície euclidiana de dimensão 2 com três bordos de cores distintas (vermelho, verde e azul) e têm a propriedade de implementar todo o grupo de Clifford, porém ele codifica apenas um único qubit. No caso dos códigos coloridos poligonais, utilizamos superfícies euclidianas de dimensão 2 com n bordos, $n \geq 3$, fazendo o número de qubits codificados aumentar sem perder a propriedade de implementar todo o grupo de Clifford.

Abstract

In this work we studied the construction of the called polygonal color code. The idea behind those codes are to expand the triangular codes introduced by Bombín and Martin-Delgado, in 2007. The triangular color codes are build over an euclidian surface of dimension 2 with three borders of distincts colors (red, green and blue) and has the property of implementing all the Clifford group but it only encodes a single qubit. In the case of the polygonal color codes we use euclidian surfaces of dimension 2 with n borders, $n \geq 3$, making the number of encoded qubits to increase without losing the property of implementing all the Clifford group.

Sumário

Introdução

Os códigos corretores de erros clássicos foram introduzidos por Claude Shannon em 1948 [19] e inspiraram a criação dos códigos quânticos corretores de erros análogos. O primeiro código exibido foi feito por Shor em 1995 [20]. Esse código era, na verdade uma concatenação de dois códigos de 3 qubits, um que protegia contra erros do tipo bit-flip e outro que protegia contra erros do tipo phase-flip, este último sem análogo clássico.

Em 1996 houve um grande avanço na codificação quântica pela criação de uma classe de códigos, hoje conhecida como códigos CSS, por Robert Calderbank, Peter Shor e Andrew Steane, como pode ser visto em [23], [9], que acabou gerando uma rica estrutura de códigos, que são os códigos quânticos estabilizadores [12].

Uma maneira alternativa de se fazer correção quântica de erros foi proposta por Kitaev [13], os códigos quânticos topológicos. Uma das grandes vantagens do código proposto por Kitaev é que seus operadores de verificação de paridade são geometricamente locais, ou seja, agem sempre em uma quantidade pequena de qubits em sua vizinhança. Devido a isso, as palavras quânticas são intrinsecamente resistentes aos ruídos locais, que não afetam as propriedades topológicas do sistema [6].

O código tórico de Kitaev foi generalizado e melhorado em alguns sentidos, como se pode ver nos trabalhos de Bombín e Martin-Delgado [5], nos de Albuquerque, Pallazo e Silva [5], [2], [3], nos de Delfosse, [10], Terhal, [8] dentre tantos outros.

Todos esses códigos supracitados implementam transversalmente as portas quânticas CNOT, $\overline{X}$ e $\overline{Z}$. No sentido de aumentar as portas quânticas implementáveis e assim aumentar a quantidade de tarefas executadas foi proposto em [4] uma classe diferente de códigos: os códigos coloridos. Sob as circusntâncias certas, os códigos coloridos podem implementar

transversalmente todo o grupo de Clifford. Para gerar todo o grupo de Clifford precisamos de apenas três operadores, que são H (Hadamard gate), K (phase shift gate, ou $\pi/4$ gate) e $\Lambda(X)$ (controlled-not gate ou CNOT gate):

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, K = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \Lambda(X) = \begin{pmatrix} I_2 & 0 \\ 0 & X \end{pmatrix}$$

A introdução dos códigos coloridos triangulares foi feita em 2007 e desde então não houve avanços no sentido de aumentar o número de bordos da superfície. Em [21], foi feito uma expansão dos códigos coloridos triangulares de Bombin e Martin-Delgado para os códigos coloridos poligonais $P(1, n, m)$. Foi-se realizado um aumento no número de bordos da superfície e avaliado o comportamento das strings, sejam elas strings diretas de uma única cor, ou as chamadas t-strings, que são as strings que são deformadas de modo equivalente combinando duas cores para formar uma terceira.

Ao fazer essa análise foi possível mostrar que, mesmo aumentando o número de qubits codificados tanto quanto se queira, os elementos geradores do grupo de homologia continuam sendo apenas as t-strings, o que possibilita a implementação do grupo de Clifford tal qual ocorria no código colorido triangular.

Capítulo 1

Preliminares da Mecânica Quântica

A mecânica quântica é a mais precisa e completa descrição do nosso mundo. É também a base para entendermos a computação quântica e a informação quântica. Neste capítulo buscamos fornecer toda a base de mecânica quântica necessária para que possamos entender a computação quântica e a informação quântica. Começaremos apresentando os conceitos básicos da álgebra linear necessários para os estudos da mecânica quântica, além de introduzirmos a notação usada por físicos para descrever a mecânica quântica, diferente da que estamos acostumados a usar na algebra linear. Em seguida, apresentamos os postulados básicos usados pela mecânica quântica. Nossas principais referências neste capítulo são [15] e [16].

1.1 Álgebra Linear

A álgebra linear consiste do estudo de espaços vetoriais e de operações lineares que agem sobre tais espaços. Uma boa base sobre álgebra linear é fundamental para se ter um entendimento sólido sobre a mecânica quântica.

1.1.1 Notação

Destacamos que, quando estudamos mecânica quântica, utilizamos algumas notações um pouco diferentes das quais estamos habituados na álgebra linear. Podemos citar como exemplo disso a notação padrão de um vetor na mecânica quântica, que é dada da forma

$$|\psi\rangle,$$

onde ψ é o rótulo para o vetor e a notação$|.\rangle$ é usada para indicar que o objeto é um vetor. Ao conjunto $|\psi\rangle$ chamamos de **ket**. Destacamos que ao vetor nulo representamos da forma 0.

Na Tabela 1.1.1 segue uma tabela com alguma das notações mais utilizadas na mecânica quântica.

Notação	**Descrição**
$\|\psi\rangle$	Um vetor. Também chamado de ket
$\langle\psi\|$	Vetor dual a $\|\psi\rangle$. Também chamado de bra.
$\langle\varphi\|\psi\rangle$	Produto interno entre os vetores $\|\varphi\rangle$ e $\|\psi\rangle$.
$\|\varphi\rangle \otimes \|\psi\rangle$, $\|\varphi\rangle\|\psi\rangle$, ou $\|\varphi\psi\rangle$	Produto tensorial entre os vetores$\|\varphi\rangle$ e $\|\psi\rangle$.
$A^\dagger$	Adjunta da matriz A.
$\langle\varphi\|A\|\psi\rangle$	Produto interno entre os vetores$\|\varphi\rangle$ e $A\|\psi\rangle$.

Tabela 1.1: Resumo das notações da teoria quântica, conhecida como notação de Dirac

1.1.2 Produto Interno

Definição 1.1. *Sejam V um espaço vetorial e $\mathbb{C}$ o seu corpo de escalares. Chamamos de* ***produto interno*** *a função*

$$\begin{array}{rccc} (.,.): & V \times V & \to & \mathbb{C} \\ & |v\rangle \times |w\rangle & \mapsto & (|v\rangle, |w\rangle) \end{array} \tag{1.1}$$

que satisfaz as seguintes condições:

1. $(.,.)$ *é linear no segundo argumento, ou seja,*

$$(|v\rangle, \sum_i \lambda_i |w_i\rangle) = \sum_i \lambda_i (|v\rangle, |w_i\rangle) \tag{1.2}$$

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

3. $(|v\rangle, |v\rangle) \geq 0$*, onde* $(|v\rangle, |v\rangle) = 0 \Leftrightarrow |v\rangle = 0$

Observação 1.2. *Esta não é a notação usual de produto interno que usaremos, esta é apenas uma forma mais conveniente de apresentar tal função. A notação usual utilizada é:*

$$\langle v|w\rangle,$$

onde $|v\rangle$ *e* $|w\rangle$ *são vetores em* V *e* $\langle v|$ *é a notação para o vetor dual de* $|v\rangle$*.*

Note que o fato de o produto interno ser linear no segundo argumento, temos que ele será conjugado-linear no primeiro argumento, ou seja,

$$(\sum_i \lambda_i |v_i\rangle, |w\rangle) = \sum_i \lambda_i^* (|v_i\rangle, |w\rangle). \tag{1.3}$$

Observação 1.3. *Como os estudos são focados na computação quântica e informação quântica, trabalhamos com espaços vetoriais de dimensão finita e, como todo espaço vetorial complexo de dimensão finita com produto interno é um espaço de Hilbert, a partir deste momento iremos referir ao nosso espaço vetorial como espaço de Hilbert.*

Dizemos que dois vetores $|v\rangle$ e $|w\rangle$ são **ortogonais** se seu produto interno é igual a zero, ou seja,

$$\langle v|w\rangle = 0.$$

Definimos a **norma** de $|v\rangle$ por

$$\||v\rangle\| = \sqrt{\langle v|v\rangle}.$$

Dizemos que um vetor $|v\rangle$ é **unitário** ou **normal** se ele possui norma 1, ou seja,

$$\||v\rangle\| = \sqrt{\langle v|v\rangle} = 1.$$

Dizemos que um conjunto $|i\rangle$ de vetores de índice i é **ortonormal** se cada um de seus vetores for unitário e seus vetores forem dois-a-dois ortogonais. Agora, suponha que $|w_1\rangle$,...,$|w_d\rangle$ é uma base para o espaço vetorial V com um produto interno. Defina

$$|v_1\rangle \equiv \frac{|w_1\rangle}{||w_1\rangle|}$$

e, para cada $1 \leq k \leq d-1$, definimos $|v_{k+1}\rangle$ por

$$|v_{k+1}\rangle \equiv \frac{|w_{k+1}\rangle - \sum_{i=1}^{k}\langle v_i|w_{k+1}\rangle|v_i\rangle}{||w_{k+1}\rangle - \sum_{i=1}^{k}\langle v_i|w_{k+1}\rangle|v_i\rangle\|}.$$

Então, o conjunto $|v_1\rangle$,...,$|v_d\rangle$ será uma base ortonormal de V. Este processo é chamado **Processo de Gram-Schmidt**. Logo, o processo de Gram-Schmidt nos permite encontrar uma base ortonormal para o nosso espaço vetorial.

Agora, daremos uma representação matricial para o produto interno. Sejam V um espaço de Hilbert e $|v\rangle = \sum_i v_i|i\rangle$ e $|w\rangle = \sum_j w_j|j\rangle$ as representações dos vetores $|w\rangle$ e $|w\rangle$ na base ortonormal $|i\rangle$. Então

$$\langle v|w\rangle = \left(\sum_i v_i|i\rangle, \sum_j w_j|j\rangle\right) = \sum_{ij} v_i^* w_j \delta_{ij} = \sum_i v_i^* w_i = [v_1^*, ..., v_n^*] \begin{bmatrix} w_1 \\ \vdots \\ w_n \end{bmatrix}. \tag{1.4}$$

Esta representação é muito interessante, pois com ela podemos ver que o vetor dual $\langle v|$ possui uma boa representação matricial como a matriz transposta conjugada da representação matricial de $|v\rangle$.

Através do produto interno podemos encontrar uma representação de operadores lineares muito útil, esta representação é chamada de **produto externo**.

Definição 1.4. *Sejam V e W espaços vetoriais com produto interno e sejam $|v\rangle \in V$ e $|w\rangle \in W$. Definimos o operador linear $|w\rangle\langle v| : V \longrightarrow W$ da forma*

$$(|w\rangle\langle v|)(|v'\rangle) \equiv |w\rangle\langle v|v'\rangle = \langle v|v'\rangle|w\rangle. \tag{1.5}$$

*A este operador chamamos de **produto externo***

A expressão dada em (1.5) pode ter dois tipos de interpretação: podemos denotá-la como o operador $|w\rangle\langle v|$ agindo no vetor $|v'\rangle$ ou como o vetor $|w\rangle$ multiplicado pelo número complexo $\langle v|v'\rangle$.

A utilidade do produto externo pode ser percebida, de certa forma, por meio de um resultado chamado **relação de completude**. Seja $|i\rangle$ uma base ortonormal para o espaço vetorial V. Então

$$\sum_i |i\rangle\langle i| = I. \tag{1.6}$$

De fato, dado $|v\rangle \in V$, temos que $|v\rangle = \sum_i v_i|i\rangle$, onde $v_i \in \mathbb{C}$, $\forall i$. Além disso, notemos que

$$\langle i|v\rangle = \langle i| \sum_j v_j|j\rangle = \sum_j v_j\langle i|j\rangle = v_i. \tag{1.7}$$

Logo,

$$\left(\sum_i |i\rangle\langle i|\right)|v\rangle = \sum_i |i\rangle\langle i|v\rangle \overset{1.7}{=} \sum_i v_i|i\rangle = |v\rangle. \tag{1.8}$$

Como $|v\rangle$ é abitrário, temos que a (1.6) é verdadeira.

1.1.3 Operadores Adjuntos e Hermitianos

Definição 1.5. *Seja A um operador linear em um espaço de Hilbert V. Então existe um único operador linear $A^\dagger$ em V tal que*

$$(|v\rangle, A|w\rangle) = (A^\dagger|v\rangle, |w\rangle), \tag{1.9}$$

para todo $|v\rangle$,$|w\rangle \in V$. Este operador linear é conhecido como ***adjunto ou conjugado Hermitiano*** *do operador A.*

Por convenção, se $|v\rangle$ é um vetor, definimos

$$|v\rangle^\dagger \equiv \langle v|. \tag{1.10}$$

Notemos que pela definição acima, temos que

$$(AB)^\dagger = B^\dagger A^\dagger \tag{1.11}$$

e, como caso particular para o resultado acima, temos

$$(A|v\rangle)^\dagger = \langle v|A^\dagger. \tag{1.12}$$

Existem outros resultados que usaremos com certa frequência neste trabalho que listamos em sequência. Seja V um espaço de Hilbert.

1. Sejam $|v\rangle$ e $|w\rangle \in V$ dois vetores arbitrários, então

$$(|w\rangle\langle v|)^\dagger = |v\rangle\langle w|; \tag{1.13}$$

2. (**Anti-linearidade do adjunto**). Sejam A_i operadores lineares em V e $a_i \in \mathbb{C}$, $\forall i$. Então

$$\left(\sum_i a_i A_i\right)^\dagger = \sum_i a_i^* A_i^\dagger; \tag{1.14}$$

3. Seja A um operador linear em V. Então

$$(A^\dagger)^\dagger = A. \tag{1.15}$$

Observação 1.6. *Quando trabalhamos com a representação matricial do operador A, seu adjunto será a matriz conjugada-transposta de A, ou seja,*

$$A^\dagger \equiv (A^*)^T. \tag{1.16}$$

Definição 1.7. *Um operador A cujo adjunto é o próprio operador A é chamado de* ***operador Hermitiano*** *ou* ***operador auto-adjunto****.*

Uma importante classe de operadores Hermitianos são os **projetores**. Sejam V um espaço vetorial de dimensão d e W um subespaço vetorial de V de dimensão k. Usando o processo de

Gram-Schmidt é possível construir uma base ortonormal $|1\rangle$,...,$|d\rangle$ para V tal que $|1\rangle$,...,$|k\rangle$ é uma base ortonormal para W. Definimos,

$$P \equiv \sum_{i=1}^{k} |i\rangle\langle i| \tag{1.17}$$

como sendo o **projetor** sobre o subespaço W.

Notemos que, pelas equações 1.13 e 1.14, temos que

$$P^\dagger = \left(\sum_{i=1}^{k} |i\rangle\langle i| \right)^\dagger = \sum_{i=1}^{k} (|i\rangle\langle i|)^\dagger = \sum_{i=1}^{k} |i\rangle\langle i| = P. \tag{1.18}$$

Portanto, P é Hermitiano.

Definição 1.8. *Um operador A é dito ser **normal** se*

$$AA^\dagger = A^\dagger A. \tag{1.19}$$

Claramente um operador Hermitiano é um operador normal.

Definição 1.9. *Um operador U é dito ser **unitário** se*

$$U^\dagger U = I = UU^\dagger. \tag{1.20}$$

Uma importante característica dos operadores unitários provém do fato que eles preservam o produto interno entre dois vetores.

De fato, sejam $|v\rangle$ e $|w\rangle$ dois vetores quaiquer. Então,

$$(U|v\rangle, U|w\rangle) = \langle v|U^\dagger U|w\rangle = \langle v|I|w\rangle = \langle v|w\rangle. \tag{1.21}$$

Logo, U preserva o produto interno entre vetores.

Existe uma subclasse dos operadores Hermitianos conhecida por **operadores positivos**.

Definição 1.10. *Um operador A é dito ser **positivo** se para todo $|v\rangle$ temos que $(|v\rangle, A|v\rangle)$ é um número real não negativo.*

Definição 1.11. *Seja A uma matriz quadrada. Definimos o **traço** de A como sendo a soma dos elementos da diagonal principal, ou seja,*

$$tr(A) \equiv \sum_i A_{ii}. \tag{1.22}$$

O traço satisfaz duas propriedades muito úteis. Sejam A,B operadores lineares e z um número complexo, então as seguintes propriedades são válidas.

1. **Ciclicidade**

$$tr(AB) = tr(BA). \tag{1.23}$$

2. **Linearidade**

$$tr(zA + B) = ztr(A) + tr(B). \tag{1.24}$$

Como consequência da ciclicidade do traço, temos que o traço de uma matriz é invariante sob uma operação transversal unitária, ou seja, seja A uma matriz e U uma matriz unitária, então

$$tr(UAU^\dagger) = tr(UU^\dagger A) = tr(A). \tag{1.25}$$

Definição 1.12. *Sejam A e B dois operadores lineares. Então definimos o* ***comutador*** *entre A e B da forma*

$$[A, B] \equiv AB - BA. \tag{1.26}$$

Se $[A, B] = 0$, ou seja, $AB = BA$, dizemos que A **comuta** com B.

Definição 1.13. *Sejam A e B dois operadores lineares. Então definimos o* ***anti-comutador*** *entre A e B da forma*

$$\{A, B\} \equiv AB + BA. \tag{1.27}$$

Se $\{A, B\} = 0$, ou seja, $AB = -BA$, dizemos que A **anti-comuta** com B

1.1.4 Matrizes de Pauli

Existem quatro matrizes que utilizamos com grande frequência nos estudos da mecânica quântica. A seguir expomos tais matrizes e suas respectivas notações.

$$\sigma_0 \equiv I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \sigma_1 \equiv \sigma_x \equiv X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$\sigma_2 \equiv \sigma_y \equiv Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad \sigma_3 \equiv \sigma_z \equiv Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Uma característica muito interessante das matrizes de Pauli é a de que elas são Hermitianas e unitárias.

1.1.5 Produto Tensorial

O **produto tensorial** é uma forma de juntar espaços vetoriais para criar um novo espaço vetorial maior e seu entendimento é essencial para os estudos relacionados aos sistemas de multipartículas. Suponha que V e W são espaços de vetoriais com produto interno de dimensão m e n, respectivamente, ou sejam, são espaços de Hilbert. Então $V \otimes W$(lê-se "V tensor W") é um espaço de Hilbert de dimensão mn.

Os elementos de $V \otimes W$ são combinações lineares de produtos tensoriais da forma $|v\rangle \otimes |w\rangle$, onde $|v\rangle \in V$ e $|w\rangle \in W$. Em particular, se $|i\rangle$ e $|j\rangle$ são bases ortonormais de V e W, respectivamente, então $|i\rangle \otimes |j\rangle$ é uma base ortonormal para $V \otimes W$. Além da notação de um elemento do produto tensorial já mostrada, existem outras três formas de representá-los:

$$|v\rangle|w\rangle, |v, w\rangle \ \ ou \ \ |vw\rangle, \tag{1.28}$$

da qual usaremos aquela que for mais adequada para a situação.

Por definição, o produto tensorial satisfaz as seguintes propriedades básicas:

1. Sejam $|v\rangle \in V$, $|w\rangle \in W$ e z um escalar. Então,

$$z(|v\rangle \otimes |w\rangle) = (z|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z|w\rangle). \tag{1.29}$$

2. Dados $|v_1\rangle, |v_2\rangle \in V$ e $|w\rangle \in W$, então

$$(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle. \tag{1.30}$$

3. Dados $|v\rangle \in V$ e $|w_1\rangle, |w_2\rangle \in W$, então

$$|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle. \tag{1.31}$$

Vejamos, agora, como representamos os operadores lineares que agem sobre $V \otimes W$. Sejam $|v\rangle \in V$ e $|w\rangle \in W$ e A e B operadores lineares sobre V e W, respectivamente. Então definimos um operador linear $A \otimes B$ sobre $V \otimes W$ da forma:

$$(A \otimes B)(|v\rangle \otimes |w\rangle) \equiv A|v\rangle \otimes B|w\rangle. \tag{1.32}$$

A fim de assegurar a linearidade destes operadores, esta definição é estendida para todos os elementos de $V \otimes W$ de forma natural, ou seja,

$$(A \otimes B)\left(\sum_i a_i|v_i\rangle \otimes |w_i\rangle\right) \equiv \sum_i a_i A|v_i\rangle \otimes B|w_i\rangle. \tag{1.33}$$

Esta ideia do produto tensorial de dois operadores pode ser estendida para o caso onde $A : V \to V'$ e $B : W \to W'$ possuem domínio e contra-domínio distintos. Dessa forma, um operador linear $C : V \otimes W \to V' \otimes W'$ é dado pela combinação linear dos produtos tensoriais dos operadores lineares de V em V' e de W em W', ou seja,

$$C = \sum_i c_i A_i \otimes B_i, \tag{1.34}$$

onde, por definição

$$\left(\sum_i c_i A_i \otimes B_i\right)|v\rangle \otimes |w\rangle \equiv \sum_i c_i A_i|v\rangle \otimes B_i|w\rangle. \tag{1.35}$$

A partir do produto interno nos espaços V e W podemos definir o produto interno em $V \otimes W$ da forma

$$\left(\sum_i a_i|v_i\rangle \otimes |w_i\rangle, \sum_j b_j|v'_j\rangle \otimes |w'_j\rangle\right) \equiv \sum_{ij} a_i^* b_j \langle v_i|v'_j\rangle\langle w_i|w'_j\rangle. \tag{1.36}$$

Dessa forma, o espaço $V \otimes W$ herda as propriedades de adjunto, unicidade e Hermiticidade.

Agora, mostraremos como é dada a representação matricial de um operador linear da forma $A \otimes B$, a esta representação chamamos de **produto de Kronecker**. Sejam A uma matriz $m \times n$ e B uma matriz $p \times q$. Então a matriz $A \otimes B$ será dada da forma

$$A \otimes B \equiv \begin{bmatrix} A_{11}B & A_{12}B & \cdots & A_{1n}B \\ A_{21}B & A_{22}B & \cdots & A_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1}B & A_{m2}B & \cdots & A_{mn}B \end{bmatrix}. \tag{1.37}$$

Nesta representação os termos da forma $A_{11}B$ denotam submatrizes $p \times q$, logo, $A \otimes B$ é uma matriz $mp \times nq$.

Por fim, denotamos $|\psi\rangle^{\otimes k}$ como sendo o produto tensorial de $|\psi\rangle$ consigo mesmo k vezes. A seguir listamos algumas propriedades muito úteis satisfeitas pelo produto tensorial:

1. O conjugado, a transposta e o adjunto se distribuem com relação ao tensorial, ou seja,

$$(A \otimes B)^* = A^* \otimes B^*; \quad (A \otimes B)^T = A^T \otimes B^T; \quad (A \otimes B)^\dagger = A^\dagger \otimes B^\dagger; \tag{1.38}$$

2. O produto tensorial de dois operadores unitários é também unitário, isto é, sejam A e B dois operadores lineares unitários então

$$(A \otimes B)(A \otimes B)^\dagger = I; \tag{1.39}$$

3. O produto tensorial de dois operadores hermitianos é também hermitiano, isto é, sejam A e B dois operadores lineares hermitianos então

$$(A \otimes B) = (A \otimes B)^\dagger; \tag{1.40}$$

4. O produto tensorial de dois operadores positivos é também positivo;

5. O produto tensorial de dois projetores é também um projetor.

1.2 Os Postulados da Mecânica Quântica

Nesta seção apresentamos os postulados básicos da mecânica quântica. Estes postulados foram desenvolvidos após vários processos de tentativa e erro e tem por finalidade realizar uma conexão entre o formalismo matemático da mecânica quântica e o mundo real.

1.2.1 Postulados

Postulado 1: A qualquer sistema físico isolado está associado um espaço vetorial complexo com produto interno(isto é, um espaço de Hilbert) conhecido como **espaço de estado** do sistema. O sistema é completamente descrito por seu **vetor de estado**, que é um vetor unitário no espaço de estado do sistema.

Observação 1.14. *Dado um sistema físico, a mecânica quântica não nos diz qual é o espaço de estado do sistema, nem os seus vetores de estado.*

O mais simples dos sistemas quânticos, e também o de maior interesse nosso, é o **qubit**. De forma semelhante ao **bit**, que pode estar nos estados 0 ou 1, o qubit também pode estar nos estados $|0\rangle$ ou $|1\rangle$, porém, no caso do qubit, estes não são os únicos estados possíveis. Qualquer combinação linear destes dois estados

$$|\psi\rangle = a|0\rangle + b|1\rangle, \tag{1.41}$$

onde α e β são números complexos tal que $|\alpha|^2 + |\beta|^2 = 1$, também são estados possíveis para o qubit. Dessa forma, se torna impossível dizer, com certeza, que um qubit está no estado $|0\rangle$ ou no estado $|1\rangle$. Aos outros estados chamamos de **superposições**.

Observação 1.15. *Para o caso de sistemas como o qubit, identificamos o estado* $|0\rangle$ *como o vetor* $(1, 0)$ *e o estado* $|1\rangle$ *como o vetor* $(0, 1)$.

Postulado 2: A evolução de um sistema quântico fechado é descrita por uma **transformação unitária**. Isto é, o estado $|\psi\rangle$ do sistema no instante t_1 está relacionado ao estado $|\phi'\rangle$ do sistema no instante t_2 por um operador unitário U que depende apenas dos instantes t_1 e t_2,

$$|\psi'\rangle = U|\psi\rangle. \tag{1.42}$$

Este postulado nos diz como ocorrem as mudanças de um estado ao longo do tempo. Da mesma forma que a mecânica quântica não nos diz qual o espaço de estado do sistema, ela não nos diz quais operadores unitários descrevem uma dinâmica quântica válida para o nosso

mundo. Mas destacamos que para o caso de um único qubit, qualquer operador unitário satisfaz tal condição.

Existem alguns operadores unitários que agem em um único qubit que são de grande importância para os estudos da computação quântica, alguns deles já foram citados neste capítulo, as matrizes de Pauli. As matrizes X e Z são, por vezes, chamadas como matrizes **bit flip** e **phase flip**, respectivamente, dado o fato que a matriz X leva o $|0\rangle$ no $|1\rangle$ e o $|1\rangle$ no $|0\rangle$ e a matriz Z mantém o $|0\rangle$ invariante e leva o $|1\rangle$ no $-|1\rangle$, onde chamamos o fator -1 que surge de **fator de fase**.

Outro operador unitário muito interessante é a **porta de Hadamard**(H), cujas ações são definidas da forma

$$H|0\rangle \equiv \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad H|1\rangle \equiv \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{1.43}$$

cuja representação matricial é dada da forma

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{1.44}$$

Observação 1.16. *Notemos que no Postulado 2 devemos ter um sistema quântico fechado, ou seja, um sistema que não interage com nenhum outro tipo de sistema, o que na realidade é algo inexistente, salvo a exceção de todo o universo. Porém, existem sistemas que chegam muito próximos desta condição e sua evolução pode ser descrita por meio de um operador unitário.*

Temos então, a evolução de um sistema quântico fechado que funciona bem, mas de nada adianta termos este sistema se não sabemos o que acontece em seu interior. Sendo assim, é necessário que o cientista, junto com seus equipamentos observem o interior deste sistema fechado, ou seja, teremos elementos exteriores interagindo com o sistema que, por consequência, deixa de ser um sistema fehado. Para explicar esta interação de observação, introduzimos o Postulado 3

Postulado 3: Medições quânticas são descritas por uma coleção M_m de **operadores de medição**. Estes são operadores agindo no espaço de estado do sistema que está sendo medido.

O índice m se refere aos resultados das medições que podem ocorrer no experimento. Se o estado do sistema quântico é $|\psi\rangle$ imediatamente antes da medição então a probabilidade que o resultado m ocorre é dada por

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle, \tag{1.45}$$

e o estado do sistema após a medição será

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}. \tag{1.46}$$

Os operadores de medição satisfazem a **equação de completude**,

$$\sum_m M_m^\dagger M_m = I. \tag{1.47}$$

A equação de completude expressa o fato que as probabilidades somam um:

$$1 = \sum_m p(m) = \sum_m \langle\psi|M_m^\dagger M_m|\psi\rangle. \tag{1.48}$$

Existe um importante caso especial deste Postulado 3, esta classe especial de medição é conhecida como **medições projetivas**. Para muitas aplicações de computação quântica e informação quântica nós estaremos interessados principalmente com medições projetivas. De fato, essas medições realmente serão equivalentes ao postulado geral de medições.

Medições Projetivas: Uma medição projetiva é descrita por um **observável**, M, que é um operador Hermitiano sobre o espaço de estado do sistema que está sendo observado. O observável tem decomposição espectral

$$M = \sum_m mP_m,$$

onde P_m é o projetor sobre o auto-espaço de M com autovalor m. As possíveis saídas da medição correspondem aos autovalores, m, do observável. Se a medição for realizada em um sistema descrito pelo estado $|\psi\rangle$, a probabilidade de obter o resultado m é dada por

$$p(m) = \langle\psi|P_m|\psi\rangle.$$

Dado que uma saída m ocorreu, o estado do sistema quântico imediatamente após a medição é

$$\frac{P_m|\psi\rangle}{\sqrt{p(m)}}.$$

Medições projetivas podem ser entendidas como um caso especial do Postulado 3. Suponha que os operadores de medição do Postulado 3, além de satisfazerem a relação $\sum_m M_m^\dagger M_m = I$, também satisfaçam as condições que M_m são projetores ortogonais, isto é, M_m são Hermitianos e $M_m M_{m'} = \delta_{m,m'} M_m$. Com estas restrições adicionais, o Postulado 3 se reduz às medições projetivas.

Medições projetivas possuem muitas propriedades boas. Em particular, é muito fácil calcular médias para medições projetivas. Por definição, o valor esperado de uma medição projetiva é:

$$E(M) = \sum_m m p(m) = \sum_m m\langle\psi|P_m|\psi\rangle = \langle\psi|(\sum_m m P_m)|\psi\rangle = \langle\psi|M|\psi\rangle.$$

Esta é uma fórmula útil, que simplifica muitos cálculos. A média do observável M é frequentemente escrita da forma $\langle M\rangle \equiv \langle\psi|M|\psi\rangle$.

Dessa fórmula para média, segue a fórmula para desvio padrão associado às observações de M

$$[\Delta(M)]^2 = \langle(M - \langle M\rangle)^2\rangle = \langle M^2\rangle - \langle M\rangle^2.$$

O desvio padrão é uma medição típica de dispersão dos valores observados após a medição M. Em particular, se fizermos um grande número de experimentos em que o estado $|\psi\rangle$ está preparado e o observável M é a medição, então o desvio padrão $\Delta(M)$ dos valores observados é determinado pela fórmula

$$\Delta(M) = \sqrt{\langle M^2\rangle - \langle M\rangle^2}.$$

Esta formulação de medição e desvio padrão em termos de observáveis dá origem, de um modo elegante, a resultados tais como as **Relações de Incerteza de Heisenberg** (o Princípio da Incerteza de Heisenberg).

Duas nomenclaturas amplamente utilizadas merecem ênfase. Ao invés de dar um observável para descrever uma medição projetiva, muitas vezes as pessoas simplesmente listam

um conjunto completo de projetores ortogonais P_m satisfazendo $\sum_m P_m = I$ e $P_m P_{m'} = \delta_{m,m'} P_m$. O observável correspondente neste uso é $M = \sum_m m P_m$. Outro termo amplamente utilizado é "medição em uma base $|m\rangle$", onde $|m\rangle$ forma uma base ortonormal, simplesmente significa realizar a medição projetiva com os projetores $P_m = |m\rangle\langle m|$.

Vamos olhar um exemplo de medição projetiva em um único qubit. Primeiro, a medição do observável Z. Este tem autovalores 1 e -1 com autovetores $|0\rangle$ e $|1\rangle$. Assim, por exemplo, a medição de Z no estado $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ dá resultado 1 com probabilidade

$$\langle\psi|0\rangle\langle 0|\psi\rangle = 1/2$$

e, similarmente, resulta em -1 com probabilidade também $1/2$.

Mais geralmente, suponha $\vec{v} = (v_1, v_2, v_3)$ um vetor real, unitário e 3-dimensional. Então podemos definir o observável

$$\vec{v}\vec{\sigma} \equiv v_1\sigma_1 + v_2\sigma_2 + v_3\sigma_3.$$

Medições desse observável são, às vezes, chamadas de "medição de spin ao longo do eixo $\vec{v}$", por razões históricas [16].

Suponhamos que estejamos interessados em um sistema quântico composto por dois (ou mais) sistemas físicos distintos. Como devemos descrever os estados do sistema composto? O quarto postulado a seguir descreve como o espaço de estado de um sistema composto é construído a partir dos espaços de estados dos sistemas componentes.

Postulado 4: O espaço de estados de um sistema físico composto é o produto tensorial dos espaços de estado dos sistemas físicos componentes. Mais ainda, se tivermos sistemas enumerados de 1 a n, e o sistema de número i está preparado no estado $|\psi_i\rangle$, então o estado conjunto do sistema total é $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$.

1.2.2 Fase

Fase é um termo comumente usado na teoria quântica, com alguns diferentes significados dependendo do contexto. Neste ponto é conveniente rever dois desses significados. Considere,

por exemplo, o estado $e^{i\theta}|\psi\rangle$ onde $|\psi\rangle$ é um vetor estado e θ é um número real. Dizemos que o estado $e^{i\theta}|\psi\rangle$ é igual a $|\psi\rangle$ a menos de um **fator de fase global** $e^{i\theta}$. É interessante observar que as estatísticas de medição fornecidas por estes dois estados são as mesmas. Para ver isso, suponha que M_m é um operador de medição associado a uma medição e note que

$$\langle\psi|M_m^\dagger M_m|\psi\rangle = \langle\psi|e^{-i\theta}M_m^\dagger M_m e^{i\theta}|\psi\rangle.$$

Portanto, de um ponto de vista observacional, estes dois estados são idênticos. Por esta razão podemos ignorar fatores de fase globais como sendo irrelevantes para observar propriedades de um estado físico. Existe um outro tipo de fase conhecido como **fase relativa**, que tem um significado bastante diferente. Considere os estados:

$$\frac{|0\rangle + |1\rangle}{\sqrt{2}} \text{ e } \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

No primeiro estado a amplitude de $|1\rangle$ é $\frac{1}{\sqrt{2}}$ e no segundo estado é $\frac{-1}{\sqrt{2}}$. Em cada caso a magnitude das amplitudes é a mesma, mas elas diferem em um sinal. Mais geralmente, dizemos que duas amplitudes a e b, diferem por fase relativa se existe um número real θ tal que $a = e^{i\theta}b$. Mais geralmente ainda, dois estados são ditos diferentes por uma fase relativa em alguma base, se cada amplitude nesta base está relacionada por um fator de fase. Por exemplo, os dois estados mostrados acima são iguais a menos de uma mudança de fase relativa pois a amplitude de $|0\rangle$ é idêntica (fator de fase relativa é 1) e a amplitude de $|1\rangle$ difere apenas por um fator de fase -1. A diferença de fase relativa para fase global é que o fator de fase pode variar de amplitude para amplitude. Isto faz a fase relativa um conceito base-dependente, o que não ocorre em fase global. Como resultado, estados que diferem apenas por fase relativa em alguma base dá origem a diferenças fisicamente observáveis nas medições estatísticas, e não é possível considerar esses estados como fisicamente equivalentes, o que fazemos com estados que diferem por um fator de fase global.

Capítulo 2

Códigos Corretores de Erros Quânticos

Neste capítulo, iniciamos explicando as ideias básicas da Teoria de correção de erros clássica. Iremos abordar alguns desafios conceituais que devem ser superados para que a correção de erros quântica seja possível e apresentamos um exemplo simples de um código corretor de erros quânticos, o código de Shor. Em seguida discutimos algumas ideias da teoria de códigos lineares clássicos e como ele pode dar origem a uma interessante classe de códigos quânticos conhecida como códigos de Calderbank-Shor-Steane (CSS). E, por fim concluímos introduzindo o conceito de códigos estabilizadores, uma classe de códigos muito bem estruturada com uma grande conexão com os códigos corretores de erros clássicos. A principais referências pra este capítulo são [1], [16] e [22].

2.1 Introdução

O ruído é um grande problema de sistemas de processamento de informação. Sempre que possível, construímos sistemas para evitar o ruído completamente ou, quando não é possível evitá-lo, tentamos protegê-lo dos efeitos do ruído. Os detalhes das técnicas usadas para proteger o sistema contra o ruído é, na prática, um pouco complicado, mas os seus princípios básicos são facilmente compreensíveis. A ideia é que se desejamos proteger uma mensagem contra os efeitos do ruído, então nós **codificamos** a mensagem adicionando algumas redundâncias à sua informação. Desta forma, mesmo que algumas das informações da mensagem codificada sejam corrompidas pelo ruído, teremos redundância suficiente para que

seja possível recuperar ou **decodificar** a mensagem de tal forma que toda a informação na mensagem original possa ser recuperada.

Como no exemplo dado em [16]. Suponhamos que Alice deseja enviar uma mensagem para Bob por meio de um canal de comunicação clássico ruidoso. O efeito do ruído no canal é o de trocar o bit sendo transmitido com probabilidade $p > 0$, enquanto que com a probabilidade $1 - p$, o bit é transmitido sem erro. Uma forma simples de proteger o bit contra os efeitos do ruído deste canal é substituir o bit que desejamos proteger com três cópias dele mesmo:

$$0 \rightarrow 000 \tag{2.1}$$

$$1 \rightarrow 111 \tag{2.2}$$

Os bit strings 000 e 111 são, por vezes, chamados de **0 lógico** e **1 lógico**, já que atuam como sendo o 0 e o 1, respectivamente. Agora, Alice pode enviar todos os três bits através do canal de tal forma que Bob irá receber três bits e irá decidir qual o valor do bit original enviado por Alice. Suponhamos que Bob receba o bit string 010 e que a probabibilidade p de ocorrer a troca de bit não seja tão alta, então é bem provável que apenas o segundo bit tenha sofrido uma troca de bit e, portanto, Bob concluirá que o bit original era 0. Este tipo de código é conhecido como **código de repetição** já que codificamos a mensagem repetindo-a várias vezes.

2.1.1 O Código Bit Flip de Três Qubits

Para proteger a inormação quântica contra os efeitos do ruído é, teoricamente, possível fazer uso de códigos quânticos corretores de erros (QECC) que, em sua essência, seguem princípios semelhantes aos códigos corretores de erros clássicos. Porém, existem algumas diferenças importantes entre a informação quântica e informação clássica que devem ser levadas em conta [16]:

- Teorema da não-clonagem: tentar implementar quanticamente o código de repetição já discutido é impossível, segundo o Teorema da não-clonagem. Esse teorema afirma que é impossível copiar um estado quântico arbitrário desconhecido.

- Os erros são contínuos: um contínuo de diferentes erros podem ocorrer em um único qubit. Determinar qual erro ocorreu para corrigi-lo iria requerer precisão infinita e, portanto, recursos infinitos.

- Medições destroem a informação quântica: na correção de erros clássica observa-se a saída do canal e, com base nisso, decidimos o procedimento a ser adotado. Observação na teoria quântica geralmente destrói o estado quântico sob observação, como foi visto no Postulado 3.

Felizmente nenhum desses problemas é incontornável, como veremos no decorrer deste capítulo. Vamos supor que enviamos qubits através de um canal que troca o qubit com uma probabilidade p e o mantém intocado com probabilidade $1-p$. Isto é, com probabilidade p o estado $|\psi\rangle$ é levado para o estado $X|\psi\rangle$, onde X é uma matriz de Pauli. Nós chamamos este canal de **canal de bit flip**.

Agora, suponhamos que codificamos um único estado de qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, onde α e β são números complexos satisfazendo $|\alpha|^2 + |\beta|^2 = 1$, em outros três qubits da forma $|\psi_L\rangle = \alpha|0_L\rangle + \beta|1_L\rangle$. Onde a codificação realiazada é a seguinte

$$\begin{aligned} |0\rangle &\rightarrow |0_L\rangle := |000\rangle \\ |1\rangle &\rightarrow |1_L\rangle := |111\rangle \end{aligned}$$

onde os qubits $|0_L\rangle$ e $|1_L\rangle$ são chamados de qubits lógicos. Um circuito que realiza este tipo de codificação é ilustrada na Figura 2.1.

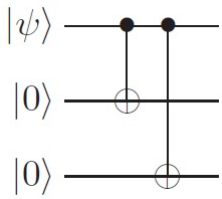

Figura 2.1: Circuito de codificação para o código de três qubits que protege contra erros do tipo bit-flip. Os dados a serem codificados entram no circuito na linha superior. [16]

Primeiramente devemos descobrir se ocorreu algum tipo de erro, caso tenha ocorrido, onde aconteceu tal erro. Para isso realizaremos uma medição sobre o estado quântico e essa

medição resultará na **síndrome de erro**. Como estamos no canal com apenas o erro bit-flip possível, temos que as medições projetivas que utilizaremos são:

$$P_0 = |000\rangle\langle 000| + |111\rangle\langle 111| \tag{2.3}$$

$$P_1 = |100\rangle\langle 100| + |011\rangle\langle 011| \tag{2.4}$$

$$P_2 = |010\rangle\langle 010| + |101\rangle\langle 101| \tag{2.5}$$

$$P_3 = |001\rangle\langle 001| + |110\rangle\langle 110| \tag{2.6}$$

Agora, vamos entender como funciona a detecção de erro. Vamos supor que ocorreu um erro bit-flip no primeiro qubit, ou seja, transmitimos o estado $|\psi\rangle = \alpha|000\rangle + \beta|111\rangle$ e foi recebido o estado $|\psi'\rangle = \alpha|100\rangle + \beta|011\rangle$. A medição da síndrome é dada por $\langle\psi'|P_i|\psi'\rangle$, para $i = 0, 1, 2, 3$.

Nesse caso em particular, temos que $\langle\psi'|P_1|\psi'\rangle = 1$ e $\langle\psi'|P_i|\psi'\rangle = 0$ para $i = 0, 2, 3$.

Essa síndrome está associada ao primeiro qubit. É importante ressaltar que essa medição de síndrome não altera o estado pois contém informação apenas de onde o erro ocorreu mas nada diz a respeito das amplitudes α e β, ou seja, medindo a síndrome não estamos destruindo a informação contida no estado quântico.

Após fazer a medição da síndrome, o passo seguinte é criar um protocolo para recuperação do estado inicial. Assim, para cada síndrome de erro recebida, uma ação deve estar associada a ela de modo a recuperar esse estado inicial.

Caso a síndrome seja 0, significa que não ocorreram erros, logo, nenhuma providência deve ser tomada. Se a síndrome for 1, como no exemplo anterior, significa que houve um erro no primeiro qubit. Dessa maneira devemos inverter o primeiro qubit e , para isso, aplicamos o operador $X_1 = X \otimes I \otimes I$ e recuperamos exatamente o estado inicial. Analogamente, se a síndrome for 2, significa que houve um erro no segundo qubit e devemos aplicar o operador $X_2 = I \otimes X \otimes I$ para invertê-lo novamente e caso a síndrome seja 3, o erro foi no terceiro qubit e a ação que devemos tomar é aplicar o operador $X_3 = I \otimes I \otimes X$ para recuperar o estado inicial.

Esse procedimento de correção de erro funciona perfeitamente se houver erro, em no máximo, um dentre os três qubits. Isso ocorre com probabilidade exatamente igual ao código

clássico de repetição de 3 bits.

Existe uma maneira alternativa de detecção e correção de erros do tipo bit-flip. Ao invés de utilizar as medições projetivas P_0, P_1, P_2 e P_3, podemos utilizar apenas uma sequência de duas medições, fornecidas pelos observáveis $Z_1Z_2 = Z \otimes Z \otimes I$ e $Z_2Z_3 = I \otimes Z \otimes Z$. Cada um desses observáveis tem autovalores ± 1 e, portanto, cada medição fornece apenas um bit de informação para um total de dois bits de informação (esses dois bits combinados geram quatro possíveis síndromes de erro, assim como no método anterior).

A primeira medição verifica se o primeiro e o segundo qubit são iguais, caso sejam resulta em $+1$, caso não sejam resulta em -1. Para ver isso basta notar que

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

De maneira análoga, a medição Z_2Z_3 compara o segundo e o terceiro qubits, novamente resultando em $+1$ se eles forem iguais e em -1 se forem diferentes. Em posse dos dois resultados é possível determinar em qual dos qubits ocorreu o erro e o protocolo de correção é identico à técnica anterior.

É claro que esse código simples não é muito robusto e não oferece proteção alguma, por exemplo, para erros do tipo phase-shift.

2.1.2 O Código Phase Flip de Três Qubits

Os canais com erros phase-shift não tem análogo clássico, o que nos leva a pensar que não existiria uma adaptação simples de códigos clássicos que nos levaria a resolver esse tipo de erro. Porém, uma simples adaptação do código de 3 qubits para erros do tipo bit-flip nos leva a um código que protege contra um erro de inversão de fase de um único qubit.

Um canal com erro phase-shift inverte a fase relativa entre os estados $|0\rangle$ e $|1\rangle$ com probabilidade p, ou seja, leva o estado $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ para o estado $|\psi'\rangle = \alpha|0\rangle - \beta|1\rangle$.

Para essa codificação, usaremos a base conjugada $\{|+\rangle, |-\rangle\}$, onde $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ e $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. Repare que o operador phase-shift Z leva $|+\rangle$ em $|-\rangle$ e vice versa, ou seja, Z age exatamente como um erro do tipo bit-flip, com respeito aos rótulos $+$ e $-$.

Dessa maneira, vemos que a codificação para detectar e corrigir um erro do tipo phase-

shift é realizada em dois passos: primeiro faz-se a codificação exatamente como no caso do bit-flip e depois aplica-se a porta de Hadamard H em cada qubit, como ilustrado na Figura 2.2.

Assim:

$$\begin{aligned} |0\rangle &\rightarrow |0_L\rangle := |+++\rangle \\ |1\rangle &\rightarrow |1_L\rangle := |---\rangle \end{aligned}$$

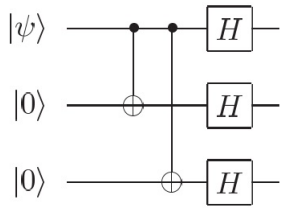

Figura 2.2: Circuito de codificação para o código de três qubits que protege contra erros do tipo phase-shift. Os dados a serem codificados entram no circuito na linha superior.[16]

Agora os procedimentos de detecção e correção de erros são feitos de maneira semelhante ao caso do bit-flip, porém como vamos utilizar a base conjugada, as medições projetivas que geram as síndromes de erro são dadas pelas medições definidas em (2.3) conjugadas pela porta de Hadamard H,ou seja, $P_i' = H^{\otimes 3} P_i H^{\otimes 3}$ para cada $i = 0, 1, 2, 3$.

Dessa maneira, se recebermos um certo estado $|\psi'\rangle$ podemos calcular a medição da síndrome ao aplicar operador $\langle\psi'|P_i'|\psi'\rangle$, para $i = 0, 1, 2, 3$.

Exatamente como no caso dos erros bit-flip, as possíveis síndromes (0, 1, 2 ou 3) referem-se a onde ocorreu o erro. Se for 1 o erro foi no primeiro qubit, 2 no segundo, 3 no terceiro e 0 significa que não houve erro. E o protocolo de correção de erros também é semelhante, porém o operador utilizado na recuperação do estado inicial é o operador $Z_i = HX_iH$, $i = 1, 2, 3$.

Equivalentemente poderíamos ter utilizado a sequência de observáveis $H^{\otimes 3} Z_1 Z_2 H^{\otimes 3} = X_1 X_2$ e $H^{\otimes 3} Z_2 Z_3 H^{\otimes 3} = X_2 X_3$. É interessante notar que os observáveis X_1X_2 e X_2X_3 definidos fazem um trabalho semelhante aos observáveis Z_1Z_2 e Z_2Z_3 do código para bit-flip. Enquanto Z_1Z_2 e Z_2Z_3 comparavam se um certo par de qubits eram iguais, os observáveis X_1X_2 e X_2X_3 faz o mesmo, mas em relação à base conjugada.

Após feito isso, a sequência de autovalores indicam onde ocorreu o erro e o protocolo de correção, para a recuperação da informação original, é semelhante ao que foi feito para as medições projetivas.

2.2 O Código de Shor

Existe um código quântico simples que pode proteger contra os efeitos de um erro arbitrário em um único qubit. Este código é uma combinação dos códigos bit flip e phase flip de três qubits e é conhecido como **Código de Shor**.

Primeiramente, codificamos o qubit utilizando o código phase-shift:

$$|0\rangle \rightarrow |+++\rangle \tag{2.7}$$

$$|1\rangle \rightarrow |---\rangle. \tag{2.8}$$

Em seguida, cada um desses qubits é codificado usando o código bit-flip:

$$|+\rangle \rightarrow \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \tag{2.9}$$

$$|-\rangle \rightarrow \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle). \tag{2.10}$$

O resultado é um código de nove qubits, com estados lógicos dados por:

$$|0\rangle \rightarrow |0_L\rangle \equiv \frac{1}{2\sqrt{2}}[(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle)] \tag{2.11}$$

$$|1\rangle \rightarrow |1_L\rangle \equiv \frac{1}{2\sqrt{2}}[(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle)] \tag{2.12}$$

O circuito quântico que codifica o Código de Shor é mostrado na Figura 2.3. Como descrito, a primeira parte do circuito codifica o qubit usando o código de phase-shift de três qubits, e se compararmos com a Figura 2.2 podemos ver que são realmente idênticos. A segunda parte do circuito codifica cada um desses três qubits usando o código bit-flip, que acaba sendo três cópias do circuito do código de bit-flip mostrado na Figura 2.1. Esse método de codificação usa níveis organizados hierarquicamente, e é conhecido como **concatenação**.

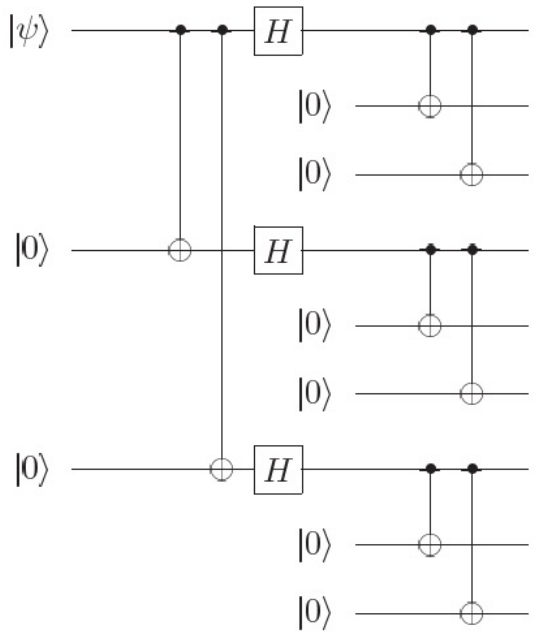

Figura 2.3: Circuito de codificação para o Código de Shor.[16]

Essa técnica é um bom truque utilizado para se criar novos códigos a partir de códigos existentes.

O código de Shor é capaz de proteger contra erros de bit flip e phase flip em qualquer qubit. Suponhamos que ocorreu um bit flip no primeiro qubit. Como o primeiro qubit pertence ao primeiro bloco de três qubits, basta que realizemos medições nos qubits do primeiro bloco. Desta forma, realizamos a medição de Z_1Z_2 e comparamos os dois primeiros qubits e descobrimos que são diferentes. Em seguida realizamos uma medição de Z_2Z_3 no segundo e terceiro qubit e descobrimos que são iguais. Logo, houve um bit flip no primeiro qubit. De forma análoga, se quisermos saber se houve um bit flip no quarto, quinto ou sexto qubit, que estão no segundo bloco, apenas realizaremos as medições Z_4Z_5 e Z_5Z_6, ou se quisermos saber se houve um bit flip no sétimo, oitavo ou nono qubit, que estão no terceiro bloco, apenas realizaremos as medições Z_7Z_8 e Z_8Z_9. E, assim, caso seja econtrado erro no i-ésimo qubit, basta aplicarmos o operador X_i para recuperarmos a informação.

Agora, suponha que ocorreu um erro de phase flip no primeiro qubit. Este tem o efeito de mudar o sinal do primeiro bloco de qubits. Na realidade, um erro de phase flip em qualquer um dos três primeiros qubits tem esse mesmo efeito. Dessa forma, para verificarmos se

houve um erro de phase flip realizamos uma comparação entre os blocos e não em um único qubit. Assim, as medições que devem ser usadas são: $X_1X_2X_3X_4X_5X_6$ e $X_4X_5X_6X_7X_8X_9$. O observável $X_1X_2X_3X_4X_5X_6$ compara o sinal entre o primeiro e o segundo bloco, enquanto o observável $X_4X_5X_6X_7X_8X_9$ compara o sinal entre o segundo e o terceiro bloco. Com essas duas medidas podemos avaliar qual bloco tem sinal diferente e então aplicamos o operador Z sobre qualquer um dos qubits desse bloco.

Com esse código ainda é possível corrigir um erro bit flip e phase flip ocorrendo no mesmo qubit, ou seja, o operador ZX é aplicado a esse qubit corrompido. Basta aplicar os dois procedimentos descritos acima. Ressaltamos que o código de Shor é melhor que os códigos clássicos de repetição no sentido de fazer uma busca por erros mais apurada. Na verdade, o código de Shor além de proteger contra inversões de bit e de fase sobre um qubit, ele também protege estados quânticos contra a ação de erros completamente arbitrários, desde que apenas um qubit seja afetado, [16].

2.3 Construindo Códigos Quânticos

Agora, possuímos uma estrutura para estudar os códigos corretores de erros quânticos, porém não temos muitos exemplos destes códigos. Para solucionarmos este problema trabalharemos brevemente com os códigos lineares clássicos e em seguida mostraremos como as ideias do código linear clássico pode ser usado para construir uma classe de códigos quânticos conhecida como Códigos de Calderbank-Shor-Steane(CSS).

2.3.1 Códigos Lineares Clássicos

Um **código linear** C que codifica k bits em um espaço de código com n bits é especificado por uma **matriz geradora** $n \times k$, G, cujas entradas são todos elementos em $\mathbb{Z}_2$. A matriz G leva as mensagens em seus equivalentes codificados. Assim, a mensagem x de k bits é codificada como Gx, onde x é um vetor coluna. Um exemplo simples, encontrado em [16], é o código de repetição que leva um único qubit em três repetições suas. Sua matriz geradora é:

$$G = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \tag{2.13}$$

já que G leva as possíveis mensagens, 0 e 1, para suas formas codificadas, $G[0] = (0, 0, 0)$ e $G[1] = (1, 1, 1)$. Dizemos que um código que usa n bits para codificar k bits é um código $[n, k]$. Logo, este exemplo é um código $[3, 1]$. Um exemplo um pouco mais complexo é codificar dois bits usando as três repetições de cada bit, ou seja, um código $[6, 2]$. Sua matriz geradora será:

$$G = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}, \tag{2.14}$$

da qual temos

$$G(0,0) = (0,0,0,0,0,0); G(0,1) = (0,0,0,1,1,1) \tag{2.15}$$

$$G(1,0) = (1,1,1,0,0,0); G(1,1) = (1,1,1,1,1,1) \tag{2.16}$$

como se era de se esperar. O conjunto de possíveis palavras códigos para o código corresponde ao espaço vetorial gerado pelas colunas de G, assim, para garantirmos que todas as mensagens são codificadas de forma única, exigimos que as colunas de G sejam linearmente independentes.

Realizar correção de erros não é tão clara. A correção de erros linear é mais facilmente compreendida se introduzirmos uma formulação alternativa de códigos lineares em termos das **matrizes de verificação de paridade**. Nesta definição, um código $[n, k]$ é definido por consistir de todos os vetores x de n-elementos sobre $\mathbb{Z}_2$ tal que

$$Hx = 0, \tag{2.17}$$

onde Hé uma matriz $n-k \times n$, conhecida como **matriz de verificação de paridade**.

Existe um processo que nos permite converter a matriz de verificação paridade na matriz geradora e vice-versa. Para convertermos a matriz de verificação de paridade na matriz geradora selecionamos k vetores independentes $y_1, ...y_k$ que geram o núcleo de H e construimos a matriz G possuindo as colunas y_1 até y_k. Para o processo inverso, selecionamos $n-k$ vetores independentes $y_1, ...y_{n-k}$ ortogonais as colunas de G e construímos a matriz H, onde suas linhas serão as matrizes $y_1^T, ...y_k^T$. Por exemplo, considere o código de repetição $[3, 1]$ definida pela matriz geradora dada em (2.14). Para construir a matriz H devemos selecionar dois vetores independetes ortogonais às colunas de G, digamos $(1, 1, 0)$ e $(0, 1, 1)$ e definimos a matriz de paridade como sendo

$$H = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{2.18}$$

Para termos uma melhor visão sobre como uma correção de erros deve ser realizada em um código linear, introduziremos o conceito de **distância**. Sejam x e y duas palavras com n bits cada. A **distância de Hamming** entre x e y é definida como o número de bits que x difere de y. Por exemplo,

$$d((1, 1, 0, 0), (0, 1, 0, 1)) = 2. \tag{2.19}$$

O **peso de Hamming** de uma palavra x é definida como a distância da palavra ao valor nulo, ou seja, $wt(x) \equiv d(x, 0)$, ou ainda, o número de entradas não nulas em x. Note que $d(x, y) = wt(x + y)$

Definimos a **distância** de um código como sendo a distância mínima entre quaisquer duas palavras,

$$d(C) \equiv \min_{x,y \in C, x \neq y} d(x, y). \tag{2.20}$$

Mas, $d(x, y) = wt(x + y)$. Como o código é linear, $x + y$ é uma palavra código se x e y são palavras códigos, então

$$d(C) \equiv \min_{x \in C, x \neq 0} wt(x). \tag{2.21}$$

Fixando $d \equiv d(C)$, dizemos que C é um código $[n, k, d]$. A importância da distância é que um código com distância de pelo menos $2t + 1$, para algum inteiro t é capaz de corrigir erros em até t bits.

Para concluirmos nossos estudos sobre códigos lineares clássicos explicaremos uma importane construção para os códigos conhecida como construção **dual**. Seja C um código $[n, k]$ com matriz geradora G e matriz de verificação de paridade H. Então podemos definir outro código, o **dual** de C, denotado por $C^{\perp}$, como sendo o código com matriz geradora H^T e matriz de verificação de paridade G^T. Equivalentemente, o dual de C consiste de todas as palavras códigos y tal que y é ortogonal a todas as palavras códigos em C.

2.3.2 Códigos de Calderbank-Shor-Steane

Nosso primeiro exemplo de uma ampla classe de códigos corretores de erros quânticos são os códigos de **Calderbank-Shor-Steane**, mais conhecidos como códigos CSS. Os códigos CSS são uma importante subclasse da classe mais geral de códigos estabilizadores.

Sejam C_1 e C_2 códigos binários lineares clássicos $[n, k_1]$ e $[n, k_2]$ tais que $C_2 \subset C_1$ e C_1 e $C_2^{\perp}$ corrigem ambos t erros. Considere $x \in C_1$, o estado quântico $|x + C_2\rangle$ é definido por:

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

onde a soma é realizada módulo 2. O estado $|x + C_2\rangle$ depende somente da classe lateral C_1/C_2 a qual x pertence. Além disso, se x e x' pertencem a classes laterais distintas então os estados $|x + C_2\rangle$ e $|x' + C_2\rangle$ são estados ortogonais. O código quântico $\mathrm{CSS}(C_1, C_2)$ 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 $|C_1|/|C_2|$, logo a dimensão de $\mathrm{CSS}(C_1, C_2)$ é $|C_1|/|C_2| = 2^{k_1 - k_2}$. Portanto, o código $\mathrm{CSS}(C_1, C_2)$ tem parâmetros $[n, k_1 - k_2]$.

Um importante exemplo de código CSS é o código de Steane, construído a partir do código de Hamming $[[7, 4, 3]]$, cuja matriz de verificação de paridade é dada por:

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{2.22}$$

Vamos denotar o código de Hamming por C_1 e seja $C_2 = C_1^{\perp}$. Para que tenhamos um código CSS é necessário que $C_2 \subset C_1$. Por definição, a matriz de verificação de paridade H de um código é igual à transposta da matriz geradora do seu código dual. Assim

$$H_{[C_2]} = G_{[C_1]}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Observe que o espaço gerado pelas linhas de $H_{[C_2]}$ contém estritamente o espaço gerado pelas linhas de $H_{[C_1]}$ e, como os códigos são os núcleos de $H_{[C_1]}$ e $H_{[C_2]}$, conclui-se que $C_2 \subset C_1$. Além disso, $C_2^{\perp} = (C_1^{\perp})^{\perp} = C_1$ e portanto C_1 e $C_2^{\perp}$ são códigos de distância 3, capaz de corrigir um bit. Como C_1 é um código $[7, 4]$ e C_2 um código $[7, 3]$ conclui-se que $\mathrm{CSS}(C_1, C_2)$ é um código quântico $[[7, 1, 3]]$.

2.4 Códigos estabilizadores

Os códigos **estabilizadores**, algumas vezes conhecidos como códigos quânticos **aditivos**, são uma importante classe de códigos quânticos cuja construção é análoga a dos códigos lineares clássicos. A fim de compreendermos os códigos estabilizadores é mais eficiente desenvolvermos, primeiro, o **formalismo estabilizador**, um método poderoso para entender uma vasta classe de operações na mecânica quântica.

2.4.1 O Formalismo Estabilizador

Considere o estado de dois qubits

$$|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}. \tag{2.23}$$

Facilmente podemos verificar que $X_1X_2|\psi\rangle = |\psi\rangle$ e $Z_1Z_2|\psi\rangle = |\psi\rangle$, neste caso, dizemos que $|\psi\rangle$ é **estabilizado** pelos operadores X_1X_2 e Z_1Z_2. Além disso, o estado $|\psi\rangle$ é o **único** estado quântico, a menos de uma fase global, que é estabilizado pelos operadores X_1X_2 e Z_1Z_2. A ideia básica do formalismo estabilizador é a de que vários estados quânticos podem ser descritos mais facilmente se trabalharmos com os operadores que os estabilizam ao invés de trabalharmos com o próprio estado.

A chave para o poder do formalismo estabilizador se encontra no uso inteligente da **teoria de grupos**. O grupo de principal interesse é o **grupo de Pauli** G_n de n qubits. Para um único qubit, o grupo de Pauli é definido por consistir todas as matrizes de Pauli juntamente com os fatores multiplicativos ± 1, $\pm i$:

$$G_1 \equiv \{\pm I, \pm iI, \pm X, \pm iX, \pm Y, \pm iY, \pm Z, \pm iZ\}. \tag{2.24}$$

Este conjunto de matrizes forma um grupo sob a operação de multiplicação entre matrizes. O grupo de Pauli geral de n qubits é definido por consistir todas as n-uplas de produto tensorial das matrizes de Pauli, juntamente com os fatores multiplicativos ± 1, $\pm i$.

Definição 2.1. *Seja S um subgrupo de G_n e defina V_S como sendo o conjunto de estados de n qubits que são fixos por cada elemento de S. V_S é o* ***espaço vetorial estabilizado*** *por S, e S é dito ser o* ***estabilizador*** *do espaço V_S, já que cada elemento de V_S é estável sob as ações dos elementos de S. Além disso, a combinação linear de quaisquer dois elementos de V_S também está em V_S.*

Vejamos um exemplo simples do formalismo estabilizador demonstrado em [16]. Considere um espaço de 3 qubits e $S \equiv \{I, Z_1Z_2, Z_2Z_3, Z_1Z_3\}$. O subspaço fixo por Z_1Z_2 é gerado por $|000\rangle$, $|001\rangle$, $|110\rangle$ e $|111\rangle$ e o subspaço fixo por Z_2Z_3 é gerado por $|000\rangle$, $|100\rangle$, $|011\rangle$ e $|111\rangle$. Note que os elementos $|000\rangle$ e $|111\rangle$ são comuns em ambas as listas. Na realidade, é possível perceber que V_S é o subespaço gerado por estes dois estados.

Neste exemplo determinamos V_S apenas analisando os espaços estabilizados por apenas dois dos operadores em S. Isto acontece por causa da descrição de um grupo por seus **geradores**. Um conjunto de elementos $g_1, ..., g_n$ em um grupo G é dito que **gera** o grupo G se cada elemento de G pode ser escrito como um produto dos elementos da lista $g_1, ..., g_n$

e escrevemos $G = \langle g_1, ..., g_n \rangle$. No exemplo temos que $Z_1 Z_3 = (Z_1 Z_2)(Z_2 Z_3)$, logo $S = \langle Z_1 Z_2, Z_2 Z_3 \rangle$. Sendo assim, para verificarmos se um vetor é estabilizado por um grupo S, basta verificar se ele é estabilizado pelos seus geradores.

Nem todo subgrupo S do grupo de Pauli pode ser usado como um estabilizados para um espaço vetorial não trivial. De fato, temos que

$$-I|\psi\rangle = -|\psi\rangle, \tag{2.25}$$

para qualquer $|\psi\rangle$, logo qualquer subgrupo S que contenha o elemento $-I$ irá apenas estabilizar o espaço vetorial trivial. Em verdade, existem duas condições que devem ser satisfeitas por S para que este estabilize um espaço vetorial não trivial, são elas:

1. Os elementos de S comutam;

2. $-I$ não é um elemento de S.

De fato, suponha que o grupo S não é abeliano, considere um estado não nulo $|\psi\rangle$ arbitrário do autoespaço $+1$ de S e dois elementos $M, N \in S$ que não comutam, portanto eles anticomutam. Assim,

$$|\psi\rangle = MN|\psi\rangle = -NM|\psi\rangle = -|\psi\rangle, \tag{2.26}$$

o que resultaria que $|\psi\rangle = 0$. O caso 2 já foi demonstrado acima.

Um belo exemplo do formalismo estabilizador é dado pelo código de Steane de sete qubits. Acontece que os seis geradores $g_1, ..., g_6$ listados na Tabela **??** geram um estabilizador para o espaço de código do código de Steane. Note a semelhança na estrutura entre os geradores da Tabela **??** e a matriz de verificação de paridade dos códigos lineares clássicos usados na construção do código de Steane.

Considere um espaço de Hilbert H com dimensão 2^n e o grupo de Pauli G_n. Denote $S \subset G_n$ um grupo estabilizador, assim:

Definição 2.2. *Um **código estabilizador** $C_S \subset H$ associado ao grupo estabilizador S é o autoespaço simultâneo, com autovalor $+1$, de todos os elementos de S. Ou seja:*

$$C_S = \{|\psi\rangle; M|\psi\rangle = |\psi\rangle, \forall M \in S\}.$$

Nome	**Operador**
g_1	$IIIXXXX$
g_2	$IXXIIXX$
g_3	$XIXIXIX$
g_4	$IIIZZZZ$
g_5	$IZZIIZZ$
g_6	$ZIZIZIZ$

Tabela 2.1: Geradores do estabilizador para o código de Steane de sete qubits. As entradas representam produtos tensoriais nos respectivos qubits, por exemplo, $XIXIXIX = X \otimes I \otimes X \otimes I \otimes X \otimes I \otimes X = X_1X_3X_5X_7$

O código é chamado de estabilizador por ser gerado pelo grupo estabilizador.

Proposição 2.3. *Se S tem $n-k$ geradores então o espaço do código tem dimensão 2^k, ou seja, C_S codifica k qubits.*

Demonstração. Sejam $\{M_1, M_2, ...M_{n-k}\}$ o conjunto dos geradores de S. Para cada $M \in S$ tem-se que $M^2 = I$, pois caso contrário M não teria autovalor $+1$. Para cada $M \neq \pm I$ existe pelo menos um $N \in G_n$ que anticomuta com M. Daí os autovalores $+1$ e -1 ocorrem com multiplicidades iguais. Vamos supor que $M = M_1$. Então $M_1|\psi\rangle = |\psi\rangle$ se, e somente se, $M_1N|\psi\rangle = -NM_1|\psi\rangle = -N|\psi\rangle$. Isto é, N leva autoestados associados ao autovalor $+1$ e autoestados associados ao autovalor -1 de M_1, com probabilidades iguais. Portanto temos $\frac{1}{2}(2^n) = 2^{n-1}$ estados mutuamente ortogonais tais que $M_1|\psi\rangle = |\psi\rangle$.

Agora, seja $M_2 \in G_n$ tal que $M_2 \neq \pm I, \pm M_1$ e comuta com M_1. Existe um $N \in G_n$ que comuta com M_1 e anticomuta com M_2. Assim, N preserva o autoespaço associado ao autovalor $+1$ de M_1 e, dentro desse espaço, altera na mesma proporção os autoestados associados aos autovalores $+1$ e -1 de M_2. Logo, o espaço satisfazendo $M_1|\psi\rangle = M_2|\psi\rangle = |\psi\rangle$ tem dimensão 2^{n-2}.

Seguindo esse processo, cada vez que adicionamos um gerador a dimensão é dividida por dois, então, com $n-k$ geradores a dimensão do espaço é $\left(\frac{1}{2}\right)^{n-k} 2^n = 2^k$.

□

Os $n-k$ operadores do estabilizador S funcionam como os operadores de verificação de paridade de um código. Ou seja, são os observáveis que medimos para identificar os erros.

O peso de um operador de Pauli é o número de fatores no tensor que difere de I. Um código estabilizador com distância mínima d tem a propriedade que cada $E \in G_n$ com peso menor que d, pertence ao estabilizador ou anticomuta com algum elemento dele. Sob esse aspecto, se o estabilizador não contém elementos de peso menor que d, então o código é não-degenerado. Para que um código corrija t erros, sua distância deve ser pelo menos $d = 2t+1$, e um código com distância $s+1$ pode detectar s erros ou corrigir s erros em locais conhecidos, [18].

Capítulo 3

Códigos Quânticos Topológicos

Neste capítulo discutimos uma restrição que é relevante em vários cenário físicos, a **localidade**. Em particular, estamos interessados em situações onde a localidade geométrica é relevante. Em geral, isso significa que os qubits físicos que compoem o código são colocados em uma tesselação e apenas interações entre qubits próximos são possíveis. Códigos topológicos oferecem uma solução natural para restrições de localidade, já que possuem geradores do estabilizador com suporte local..

Nos códigos topológicos a informação é armazenada em graus de liberdade globais, dessa forma, tesselações maiores fornecem maiores distâncias de código. A natureza por trás destes graus de liberdade globais é ilustrada na Figura 3.1, onde várias curvas fechadas em um toro são comparadas. Considere as curvas a e b. Se as examinarmos na região limitada pela curva pontilhada, elas parecem semelhantes. Porém, a curva a é a fronteira de uma região, mas a curva b não é. Para que possamos dizer se uma curva é uma fronteira de uma região precisamos de informação global.

Os estudos realizados neste capítulo seguem principalmente as referências [4], [14] e [22] e iniciamos dando uma breve introdução aos conceitos de códigos locais e de homologia, o que é suficiente para os estudos futuros. Em seguida, estudaremos os exemplos mais básicos de códigos quânticos topológicos, os códigos de superfície, bem como suas principais características.

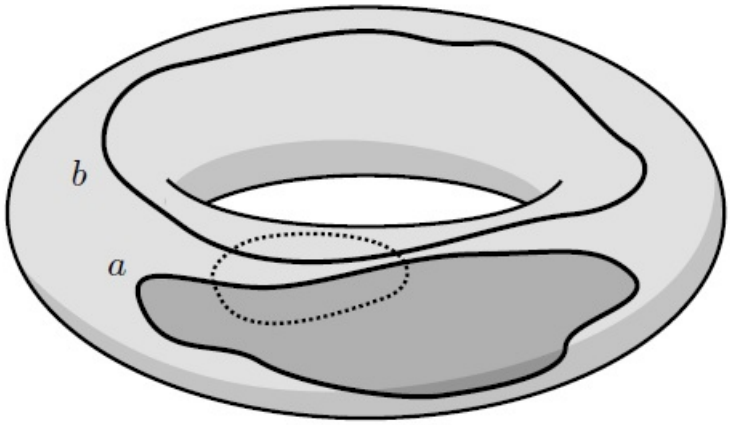

Figura 3.1: Curvas fechadas em um toro podem ser a fronteira de uma região, como a curva a. Entretanto, a curva b não a é fronteira de uma região. A diferença não pode ser notada apenas olhando para um região local tal como a região delimitada pela curva pontilhada.[14]

3.1 Códigos Locais

Intuitivamente, um **código estabilizador local** é um código onde todos os geradores do estabilizador agem apenas em alguns qubits próximos. Para formalizar esta ideia, podemos falar sobre códigos n-locais, cujo suporte de cada gerador estabilizador é limitado a n qubits.

Uma família de códigos estabilizadores é **local** se podemos escolher os geradores do estabilizador tal que [14]

1. a família contém códigos com uma grande ditância arbitraria;
2. o número de qubits no suporte dos geradores do estabilizador é limitado;
3. o número de estabilizadores com suporte contendo qualquer qubit dado é limitado.

A noção de localidade pode ser colocada em termos de conectividade gráfica sem qualquer outra estrutura. Uma família de códigos é **local em D dimensões** quando

1. os qubits são colocados em uma região de dimensão D;
2. o suporte de qualquer gerador estabilizador está contido em um hipercubo de tamanho limitado.

Isto está ilustrado na Figura 3.2. Note que um código que é local no sentido geométrico é também local no sentido geral apresentado acima.

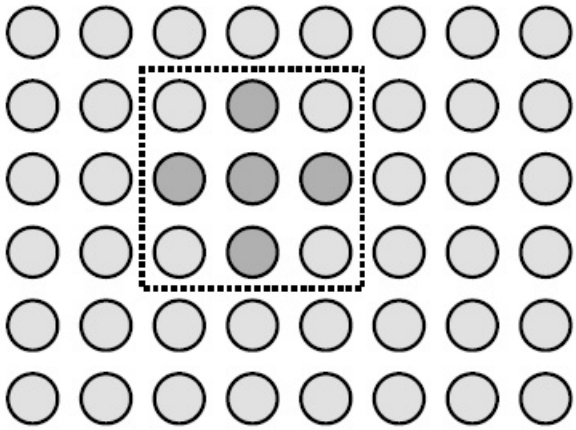

Figura 3.2: Em códigos locais 2D os qubits são organizados em uma região 2D. O suporte de qualquer gerador estabilizador deve estar contido em uma caixa de tamanho fixo, aqui uma caixa 3×3. Os círculos representam qubits e os círculos escuros formam o suporte de um gerador.[14]

3.2 Homologia

Os códigos quânticos topológicos que vamos trabalhar dependem da homologia da superfície considerada, por isso vamos passar pelos conceitos introdutório de homologia e do primeiro grupo de homologia de um espaço ou uma superfície. Para mais detalhes ver [7], [11] e [24].

Como vamos trabalhar em um caso bem específico, não vamos tratar das definições de homologia em um espaço topológico qualquer, mas sim nas superfícies bidimensionais.

Mergulhando um grafo G em uma superfície S dizemos que estamos fazendo uma "celulação"da superfície. Comumente se denominam os vértices por 0-cells, as arestas por 1-cells e as faces por 2-cells.

Definição 3.1. *Dado uma superfície S com um grafo G mergulhado, considere P como o*

conjunto de todas as p-cells. Para cada $P' \subset P$, definimos como p-chain em S a soma finita

$$c = \sum_i c_i p_i \text{ tal que } c_i = \begin{cases} 0; p_i \notin P' \\ 1; p_i \in P' \end{cases},$$

com $P' \subset P$.

Em particular, se rotularmos as arestas da tesselação da superfície por $\{e_i\}_1^E$ podemos representar as 1-chains definidas acima pelas seguintes somas formais:

$$c = \sum_i c_i e_i \text{ tal que } c_i = \begin{cases} 0; e_i \notin E' \\ 1; e_i \in E' \end{cases}$$

onde E' é um conjunto qualquer de arestas de G.

Essas 1-chains podem ser somadas, módulo 2, nos levando a gerar um grupo abeliano aditivo C_1. De forma análoga podem ser representadas as 0-chains e as 2-chains, bem como seus grupos abelianos aditivos, para os vértices C_0 e as faces C_2 de G, respectivamente.

Podemos relacionar tais grupos de p-chains utilizando os homeomorfismos conhecidos como operadores bordo, que denotaremos por $\partial_i : C_i \to C_{i-1}$, no nosso caso para $i = 0, 1, 2$.

Tais homeomorfismos levam um p-cell em sua fronteira e leva uma p-chain na soma (módulo 2) das fronteiras de todas as p-cells que a compõe. Por exemplo, considere um certa face f de G cuja fronteira sejam as arestas $\{e_1, ..., e_k\}$, então $\partial_2(f) = e_1 + ... + e_k$. Uma 1-chain c é chamada de ciclo se $\partial_1 c = 0$.

Pode-se então construir uma cadeia

$$C_2 \overset{\partial_2}{\to} C_1 \overset{\partial_1}{\to} C_0 \overset{\partial_0}{\to} \{0\}.$$

É fácil notar que $Im(\partial_{(i+1)}) \subset Ker(\partial_i)$.

Vamos então definir dois subgrupos de C_1, os quais denotaremos por Z_1 e B_1. O subgrupo Z_1 é definido exatamente como sendo o núcleo de ∂_1, ou seja, os conjuntos de arestas (1-chains) que formam um ciclo, uma curva fechada. Já o subgrupo B_1 é definido como a imagem de ∂_2, ou seja, o conjunto de arestas que formam a fronteira de uma 2-chain.

Com esses elementos em mãos podemos definir o primeiro grupo de homologia de uma superfície:

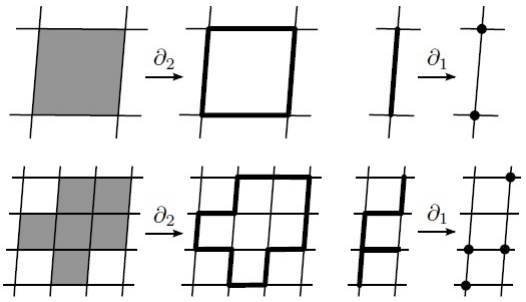

Figura 3.3: As ações dos operadores bordo ∂_2 e ∂_1. Note que como a soma é feita módulo 2, as arestas que pertencem a um número par de faces da 2-chain não estão na imagem daquela 2-chain pelo operador ∂_2. Analogamente, apenas os vértices que pertencem a um número ímpar de arestas da 1-chain estão na imagem daquela 1-chain pelo operador ∂_1.[14]

Definição 3.2. *O primeiro grupo de homologia de uma supercície S é definido pelo quociente*

$$H_1 = \frac{Z_1}{B_1}. \tag{3.1}$$

Assim, os elementos de H_1 são classes laterais da forma $\overline{z} := \{z + b | b \in B_1\}$ para algum $z \in Z_1$, a adição é a herdada de Z_1, nomeada $\overline{z} + \overline{z}' = \overline{z + z'}$, e o elemento nulo é a classe lateral $\overline{0} = B_1$. Tecnicamente, (3.1) define $H_1(S; \mathbb{Z}_2)$, o primeiro grupo de homologias da superfície S sobre $\mathbb{Z}_2$.

A topologia algébrica nos ensina que o grupo H_1 depende apenas, a menos de isomorfismo, da topologia da superfície. De fato:

$$H_1 \simeq \mathbb{Z}_2^{2g}. \tag{3.2}$$

De modo geral, o primeiro grupo de homologia é um grupo abeliano finitamente gerado e, independentemente do grafo o qual foi utilizado no mergulho, o número de geradores do primeiro grupo de homologia tem muito a dizer sobre a natureza topológica dessa superfície, a saber, como pode ser visto em [11], [24], o número de geradores do primeiro grupo de homologia é o dobro do gênero da superfície considerada.

Em um toro, por exemplo, existem dois geradores para o primeiro grupo de homologia, chamemos-os de b e d. Claro que no toro existem infinitas curvas fechadas (ciclos) diferentes, porém, todas elas podem ser deformadas continuamente para a curva b, para a curva d, para

uma concatenação de b com d ou até para um único ponto (que seria o elemento neutro 0 do grupo), a qual chamamos de curva homologicamente trivial, como podemos ver na Figura 3.4

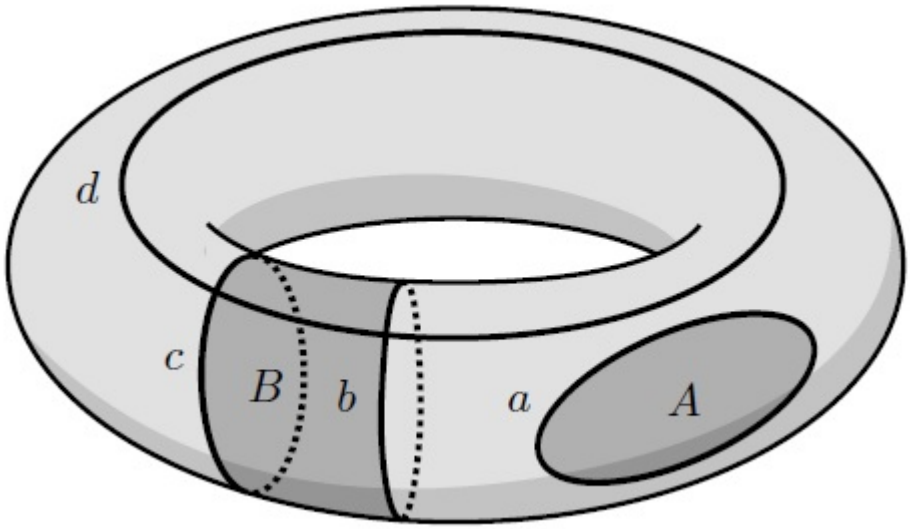

Figura 3.4: Algumas curvas fechadas em um toro. A curva a é a fronteira de uma região A, então é homologicamente trivial. As curvas b e c não são homologicamente triviais, mas, juntas formam uma fronteira para a região B e, portanto, são homologicamente equivalentes. A curva d homologicamente não trivial e não é equivalente as curvas b e c. As curvas b e d são geradoras do primeiro grupo de homologia do toro.[14]

3.3 Códigos de Superfície

Os **códigos de superfície** são os exemplos mais básicos de códigos topológios. A ideia por trás dos códigos de superfície é a de codificar a informação em "graus de liberdade homológicos". Primeiramente fixamos uma tesselação mergulhada em uma superfície fechada. Nós fixamos um qubit a cada uma das arestas, então cada elemento da base computacional pode ser interpretado como um 1-chain $c \in C_1$ da maneira mais óbvia:

$$|c\rangle := \bigotimes_i |c_i\rangle, \quad c \in C_1. \tag{3.3}$$

O índice i percorre os qubits físicos ou, equivalentemente, as arestas da tesselação e_i. Para uma maior conveniência, rotulemos os produtos das matrizes de Pauli X e Z com as 1-chains:

$$X_c := \bigotimes_i X_i^{c_i}, \quad Z_c := \bigotimes_i Z_i^{c_i}, \quad c \in C_1. \tag{3.4}$$

Note que estas rotulações se tratam de homomorfismos de grupo de C_1 para G_n, pois

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

A definição de um código de superfície é bem natural. Ela tem uma base com os elementos

$$|\overline{z}\rangle := \sum_{b \in B_1} |z + b\rangle, \quad \overline{z} \in H_1, \tag{3.6}$$

que são somas de todos os ciclos que forma uma classe homológica dada. É fácil ver que $\langle \overline{z} | \overline{z}' \rangle = 0$, para $\overline{z} \neq \overline{z}'$. Então, de 3.1 temos $|H_1| = 2^{2g}$ e o número de qubits codificados é $k = 2g$.

Para termos uma prova de como funcionam os códigos de superfície, devemos estudar os efeitos dos erros de bit flip. Note que $X_c|b\rangle = |b + c\rangle$ e, assim, $X_c|\overline{z}\rangle = |\overline{z} + \overline{c}\rangle$. Sejam $z, z' \in Z_1$. Se $\partial c \neq 0$ então $\partial(z + c) \neq 0$ e $\overline{z} + \overline{c} \neq \overline{z}'$. Isto implica que $\langle \overline{z}' | X_c | \overline{z} \rangle = 0$, então X_c pode levar o código nele mesmo apenas se c for cíclico. Mas erros X_b, com b uma fronteira, não fazem nada, já que $X_b|\overline{z}\rangle = |\overline{z} + \overline{b}\rangle = |\overline{z}\rangle$. Dessa forma, apenas erros de bit flip X_z, com $z \in Z_1$ não são detectáveis. Portanto, a distância de um código de superfície para erros bit flip é o comprimento do menor ciclo não trivial da tesselação.

3.3.1 Grupo Estabilizador

Dado um vértice v e uma face f, considere os operadores de Pauli da face e do vértice

$$X_f := \prod_{e \in \partial_2 f} X_e, \quad Z_v := \prod_{e | v \in \partial_1 e} Z_e, \tag{3.7}$$

onde ∂f e ∂e são vistos como conjuntos. Estes operadores são retratados na Figura 3.5, onde podemos ver que um operador vértice tem suporte nas arestas que se encontram em um vértice e um operador face tem suporte nas arestas que formam a face.

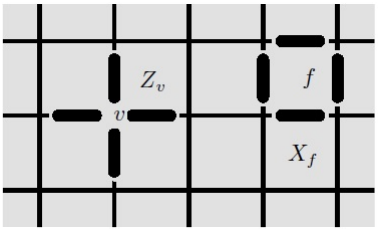

Figura 3.5: Os suportes dos geradores dos estabilizadores do vértice e da face.[14]

Os operadores face e vértice comutam entre sí e afirmamos que eles geram o estabilizador dos códigos de superfície, como definido em (3.6). Para verificar isso, considere os estados

$$|c'\rangle := \prod_f \frac{1+X_f}{2} \prod_v \frac{1+Z_v}{2}|c\rangle, \quad c \in C_1. \tag{3.8}$$

Estes estados abrangem o código definido por 3.7, pois $|c'\rangle$ é a projeção de $|c\rangle$ sobre o subespaço do código. Note que $Z_v|c\rangle = -|c\rangle$ se $v \in \partial c$, de outra forma $Z|c\rangle = |c\rangle$. Assim, $|c'\rangle = 0$ se $\partial c \neq 0$. Ainda, o primeiro produtório pode ser expandido como uma soma de subconjuntos de face f_i:

$$\prod_f (1+X_f) = \sum_{\{f_i\}} \prod_i X_{\partial_2 f_i} = \sum_{\{f_i\}} X_{\partial_2(\sum_i f_i)} = \sum_{c_2 \in C_2} X_{\partial_2 c_2}. \tag{3.9}$$

Como ∂_2 é um homomorfismo de grupos, podemos substituir a soma sobre as 2-chains por uma soma sobre as 1-chains na imagem de ∂_2, a menos de um fator. Sendo assim, para $z \in Z_1$ temos

$$|z'\rangle := \prod_f \frac{1+X_f}{2}|z\rangle \propto \sum_{b \in B_1} X_b|z\rangle = |\overline{z}\rangle. \tag{3.10}$$

Logo, temos o mesmo subespaço do código dos estabilizadores (3.7) e a expansão da base (3.6).

Os geradores do estabilizador (**??** não são todos indepedentes. Eles estão sujeitos as duas condições seguintes:

$$\prod_f X_f = 1 \qquad \prod_v Z_v = 1. \tag{3.11}$$

Sendo assim, existem $V + F - 2$ geradores independentes. Segue-se da teoria de códigos estabilizadores que o número de qubits codificados é

$$k = E - (V + F - 2) = 2 - \chi = 2g, \tag{3.12}$$

onde $\chi = V - E + F$ é a **característica de Euler** e, para superfícies orientáveis fechadas, temos $\chi = 2(1 - g)$. Corcordando com o valor obtido através da equação 3.6.

3.3.2 Tesselação Dual

Dada uma tesselação mergulhada em uma superfície nós podemos construir sua tesselação dual, como podemos ver na Figura 3.6. A ideia é que as faces da tesselação original são levadas em vértices na tesselação dual, arestas em arestas no dual e vértices em faces no dual. Denotamos com um $*$ os vértices f^*, as arestas e^* e as faces v^* no dual relativas as faces f, arestas e e vértices e da tesselação original, respectivamente.

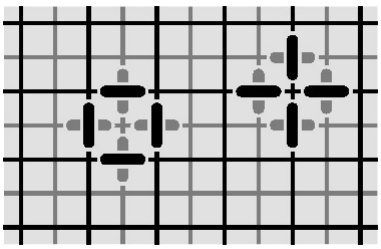

Figura 3.6: Uma tesselação e seu dual. Sob dualidade, vertíces e faces são trocados, bem como os operadores vértice e face.[14]

De forma semelhantte, temos operadores de bordo dual

$$\partial_1^* : C_0^* \to C_1^*, \qquad \partial_2^* : C_1^* \to C_2^* \tag{3.13}$$

agindo nas chains dual $c*$. Os operadores de bordo ∂^* produzem o grupo cíclico dual Z_1^* e as fronteiras dual B_1^* e, assim, um grupo homológico

$$H_1^* = \frac{Z_1^*}{B_1^*} \simeq H_1. \tag{3.14}$$

Comparando as ações de ∂ e ∂^* em suas respectivas tesselações, observamos que

$$e^* \in \partial_1^* v^* \Leftrightarrow v \in \partial_1 e, \qquad f^* \in \partial_2^* e^* \Leftrightarrow e \in \partial_2 f. \tag{3.15}$$

Considere, agora, o efeito causado ao aplicarmos a porta de Hadamard em todos os qubits

do código de superfície. O código é levado em outro subespaço descrito pelos estabilizadores

$$W^{\otimes E} X_f W^{\otimes E} = \prod_{e \in \partial_2 f} Z_e = \prod_{e^* | f^* \in \partial_2^* e^*} Z_e =: Z_f^* \tag{3.16}$$

$$W^{\otimes E} Z_v W^{\otimes E} = \prod_{e \in \partial_1 e} X_e = \prod_{e^* \in \partial_1^* v^*} X_e =: X_v^*. \tag{3.17}$$

Este é o código de superfície definido na tesselação dual. Logo, podemos lidar com erros de phase flip como já fizemos com os erros de bit flip, mas trabalhando na tesselação dual. Sendo assim, temos que a distância de um código de superfície é o tamanho do menor ciclo não trivial na tesselação original ou no seu dual.

3.4 Códigos Coloridos

Os códigos de superfície não são tão bons em termos das portas que eles permitem implementar através de operações transversais. Como eles são códigos CSS, eles permitem a implementação das portas CNot e a implementação dos operadores $\overline{X}$ e $\overline{Z}$. Mas são somente essas.

Para ir além destas portas quânticas preisamos considerar uma classe diferente de códigos topológicos, os **códigos coloridos**.

3.4.1 Tesselação e Grupo Estabilizador

Os códigos de superfície podem ser construidos em qualquer tesselação mergulhada em uma variedade fechada. No caso dos códigos coloridos, precisamos considerar um tipo particular de tesselação: aqueles que são 3-valentes e tem faces 3-coloríveis. Ou seja, a tesselação deve ser da forma

1. três arestas se encontram em um vértice;
2. é possível atribuir três rótulos a cada face de tal forma que faces vizinhas tenham rótulos diferentes.

É comum escolher as cores vermelho, verde e azul (RGB), como sendo os rótulos. O exemplo mais básico para este tipo de tesselação é a tesselação de **favo de mel**, do qual pode ser mergulhado em um toro, veja a Figura 3.7. Note que em uma tesselação de um código colorido também é possivel atribuir cores às arestas: atribuímos a cor vermelha as arestas que não fazem parte de uma face vermelha e de forma semelhante para as cores verde e azul.

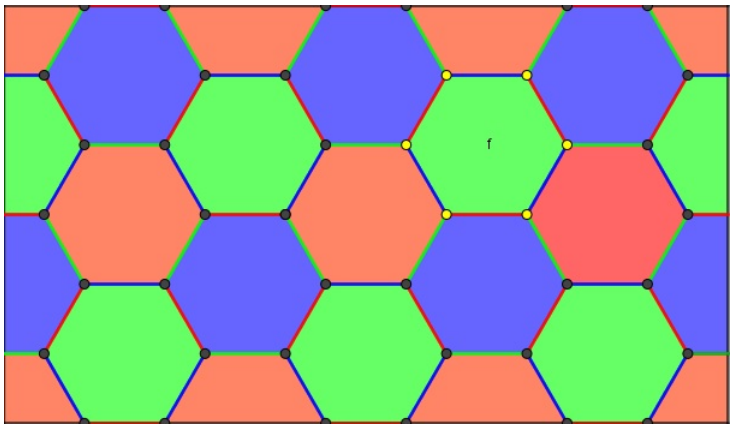

Figura 3.7: Códigos coloridos são definidos em tesselações 3-valentes e com faces 3-coloríveis, como o favo de mel. Rotulamos as faces como vermelho, verde e azul (RGB), usualmente. Os qubits são colocados nos vértices. Há dois geradores do estabilizador por face, X_f e Z_f, com o suporte mostrado em amarelo.

O primeiro passo para construirmos um código colorido a partir de uma tesselação é atribuir um qubit a cada um de seus vértices, diferentemente dos códigos de superfície, onde atribuíamos o qubit às arestas. Em seguida, definimos os geradores do estabilizador, da qual temos dois geradores por face f:

$$X_f := \prod_{v \in f} X_v, \qquad Z_f := \prod_{v \in f} Z_v, \tag{3.18}$$

onde X_v e Z_v são os operadores de Pauli X e Z agindo no qubit do vértice v e $v \in f$ é uma notação simbólica para denotar que v é parte de uma face f. Assim como em 3.11, estes geradores não são independentes. Eles estão sujeitos a quatro restrições. Denotemos por F_r, F_g e F_b o conjunto de faces vermelhas, verdes e azuis, respectivamente. Então, as restrições

são:

$$\prod_{f\in|F_r} X_f = \prod_{f\in|F_g} X_f = \prod_{f\in|F_b} X_f, \tag{3.19}$$

$$\prod_{f\in|F_r} Z_f = \prod_{f\in|F_g} Z_f = \prod_{f\in|F_b} Z_f. \tag{3.20}$$

Assim, o número de geradores independentes do estabilizador é

$$g = 2(|F_r| + F_g| + F_b|) - 4. \tag{3.21}$$

3.4.2 Tesselações Reduzidas

A partir de uma tesselação de um código colorido podemos construir outras três tesselações "reduzidas", rotuladas a partir da cor das faces a serem reduzidas. Vamos estudar a tesselção reduzida vermelha, a verde e azul são análogas.

Os vértices da nova tesselação correspondem as faces vermelhas que são reduzidas a um único ponto. Ao conectarmos estes novos vértices, temos que as novas arestas passam exatamente por cima das arestas vermelhas da tesselação original, dessa forma, cada uma dessas arestas novas corresponde a dois vértices da tesselação original. Com relação as novas faces, teremos uma para cada face verde e azul da tesselação original, como pode ser visto na Figura 3.8

Agora, denote V^r, E^r e F^r como o número de vértices, arestas e faces da tesselação reduzida vermelha. Sendo assim, temos que

$$n = 2E^r, V^r = |F_r|, F^r = |F_g| + |F_b| \tag{3.22}$$

Portanto, o número de qubits codificados é

$$k = n - g = 2E^r - 2(V^r + F^r - 2) = 2(E^r - V^r - F^r + 2) = 2(2 - \chi) = 4g. \tag{3.23}$$

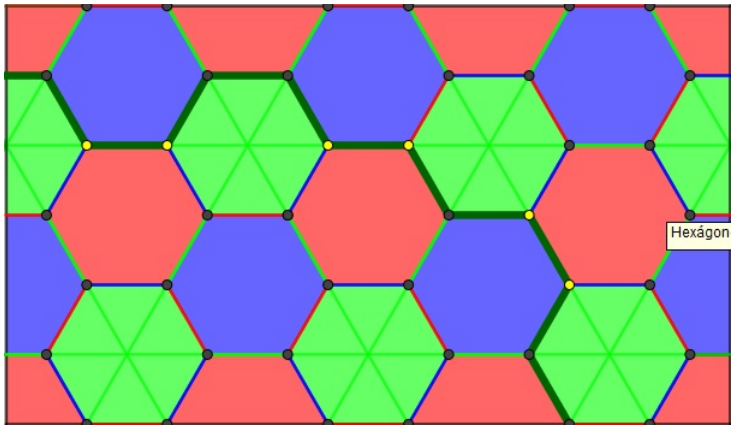

Figura 3.8: Qualquer tesselação reduzida de uma tesselação de favo de mel é triangular. Aqui temos uma tesselação reduzida verde e uma string γ. Os qubits no suporte de X_γ^g e Z_γ^g estão marcados com círculos ao longo da string. Eles aparecem aos pares, com cada par relacionado a uma aresta da tesselação reduzida verde.

3.4.3 Operadores String e String-Nets

Para ficar claro como funcionam os códigos coloridos, devemos entender como funcionam os operadores string. Assim como nos códigos tóricos, a homologia dessas strings estão definidas sobre $\mathbb{Z}_2$, pois estamos tratando de sistemas quânticos de dois níveis.

Essas strings podem ser verdes, azuis ou vermelhas dependendo de qual tesselação reduzida estamos considerando e, independentemente da cor, elas podem ser do tipo X ou do tipo Z. Vamos denotar essses operadores strings por $S_\mu^{C\sigma}$ onde C é uma cor, σ é Z ou X e μ é um rótulo da classe de homologia.

Em [4] é provado que:

$$S_\mu^{R\sigma} S_\mu^{G\sigma} S_\mu^{B\sigma} \sim 1. \tag{3.24}$$

A equação (3.24) nos mostra que tem-se apenas duas cores independentes.

Uma propriedade essencial para os códigos coloridos é que além de um operador string de uma certa cor poder ser deformado a um outro operador de string homólogo a ele de forma equivalente, também é possível combinar duas strings de cores diferentes para produzir um terceiro operador equivalente de cor diferente das duas combinadas, as chamadas t-strings

(ou string-nets).

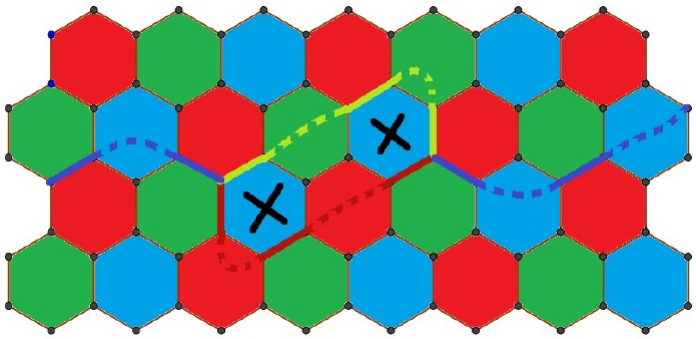

Figura 3.9: Um exemplo de uma t-string (ou string-net). Note que a string azul se ramifica em duas partes, sendo uma vermelha e uma verde, e depois essas duas partes se combinam e voltam a ser uma string azul. Essa string net pode ser transformada de forma equivalente em uma string toda azul fazendo o produto com operadores face das faces marcadas.[21]

3.4.4 Códigos Coloridos Triangulares

Para lidar com toros de gênero arbitrário as strings são suficientes. Os códigos coloridos começam a ficar mais interessantes se considerarmos superfícices orientadas com bordo, da qual podemos obtê-las abrindo buracos em uma superfície fechada. Em particular, vamos introduzir buracos em nossa superfície, tirando faces. Se removermos, por exemplo, uma face verde, strings verdes podem ter um ponto final nelas, mas não as strings vermelhas ou azuis. Então as bordas tem cores e apenas strings verdes podem terminar em um bordo verde, como ilustrado na Figura 3.10. Um importante caso de tais códigos coloridos é o **código colorido triangular**.

A construção deste código colorido começa com um código colorido em uma esfera onde um vértice e suas três arestas e faces adjacentes são removidas. Pelas restrições dadas em (3.19) podemos observar que dois geradores do estabilizador são removidos no processo. Como um código colorido na esfera codifica zero qubits, um código triangular irá codificar um único qubit. Um exemplo de código triangular pode ser visto na Figura 3.11

As t-strings existentes nos códigos triangulares são a chave para a implementação do

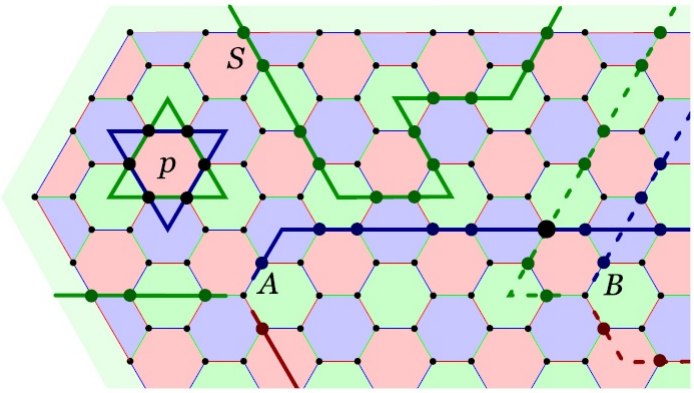

Figura 3.10: Uma tesselação de favo de mel com bordo verde. A string verde S é homóloga a parte do bordo e, assim, é equivalente a identidade. Há também um par de operadores 3-string, A e B, que são equivalentes.[4]

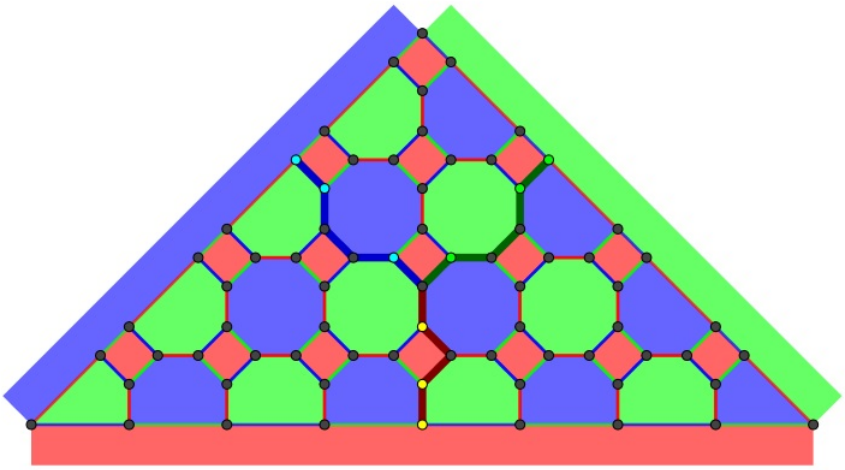

Figura 3.11: Um código colorido triangular [73,1,9] baseado em uma tesselação 4-8-8. Existe apenas uma classe de string-nets não trivial.

grupo de Clifford. A diferença para os códigos de superfície é que, geometricamente, as t-strings X e as t-strings Z são exatamente as mesmas e assim, se denotarmos duas t-strings T^X e T^Z temos que $\{T^X, T^Z\} = 0$.

Com isso, dados os operadores codificados $\overline{X} = X^{\otimes n}, \overline{Z} = Z^{\otimes n}, \overline{K} = K^{\otimes n}$ e $\overline{H} = H^{\otimes n}$, onde n é o número de qubits do código, temos que

$$\overline{H}.X_f.\overline{H} = Z_f, \qquad \overline{H}.Z_f.\overline{H} = X_f, \qquad \overline{H}.\overline{X}.\overline{H} = \overline{Z} \quad \text{e} \quad \overline{H}.\overline{Z}.\overline{H} = \overline{X}.$$

Como já vimos, os operadores X_f e Z_f são geometricamente iguais. Logo a mudança causada de X para Z, e vice-versa, pela porta de Hadamard não é um problema para a implementação transversal desta porta, o que não acontece nos códigos não coloridos, aqueles onde os qubits estão relacionadas as arestas. Como os operadores face X_f e vértice Z_v são geometricamente diferentes, a implementação transversal da porta de Hadamard não é possível nestes códigos.

Além disso,

$$\overline{K}.Z_f.\overline{K} = Z_f \text{ e } \overline{K}.X_f.\overline{K} = (-1)^{t/2}X_f.Z_f,$$

onde t é o número de vértices da face. Pela mesma justificativa da porta de Hadamard, a implementação transversal da porta phase-shift não é possível nos códigos não coloridos. Notemos que quando implementamos a porta phase-shift K surge um fator de fase $(-1)^{t/2}$. Dessa forma, devemos usar uma tesselação onde todas as faces tem um número de vértices múltiplo de 4, como a $\{4, 8, 8\}$ (por sinal, a única possível no plano euclideano) todos os operadores geradores do grupo de Clifford podem ser implementados transversalmente, visto que a porta CNOT é sempre implementada transversalmente em todo código CSS [4], [6], [14].

Capítulo 4

Múltiplos Bordos

Este capítulo se baseia na referência [21] e trás um procedimento para a criação de novas classes de códigos coloridos com base nos códigos coloridos triangulares, os códigos poligonais.

Os códigos coloridos são códigos corretores de erros com bons parâmetros quando aplicados em superfícies compactas, mas uma boa característica é a possibilidade de implementar todo o grupo de Clifford, o que permite realizar várias tarefas importantes na computação quântica, quando aplicado nas superfícies com bordo gerando os conhecidos "códigos triangulares".

Os códigos triangulares codificam apenas um qubit e aumentar o número de qubits codificados melhoraria os parâmetros dos códigos, além de ser uma economia de recursos para a sua implementação.

Para que isso aconteça, vamos considerar polígonos com uma quantidade de lados maiores que três, vamos avaliar um modo de distribuir os bordos de cada uma das cores, e verificar quais são as strings de homologia não-trivial que geram as outras, em cada um dos casos analisados.

Neste trabalho vamos manter apenas um bordo de uma certa cor e ir aumentando alternadamente a quantidade de bordos de cada uma das outras duas cores consideradas. Sem perda de generalidade, vamos supor que a cor azul sempre estará fixada em apenas um bordo e vamos aumentando a quantidade de bordos verdes e vermelhos, de um em um, e nesta ordem. A título de notação, vamos considerar o primeiro caso sendo o caso dos códigos triangulares de Bombim e Martin-Delgado, que denotaremos por $P(1,1,1)$ (1 azul - 1 verde - 1 vermelho).

O segundo caso seria um quadrilátero, onde temos apenas um bordo azul (sempre), dois verdes e um vermelho, que denotaremos por $P(1, 2, 1)$, ou Código Poligonal $1 - 2 - 1$.

Seguindo as regras das strings de uma superfície com bordos rotulados, no nosso caso por cores, um ciclo pode ser homologicamente não-trivial se ele inicia e termina em bordos distintos da mesma cor, ou no caso de uma t-string ela inicia em um bordo de uma certa cor, passa por um certo ponto de bifurcação, e desse ponto cada uma das duas string geradas finaliza em um bordo da sua cor correspondente. Desse modo, em um quadrilátero conforme descrito anteriormente teríamos, a menos de deformações contínuas, três tipos de ciclos de homologia não-trivial: um ciclo que parte de um bordo verde e finaliza em outro bordo verde, e duas t-strings que iniciam em um dos bordos verdes se ramifica e finaliza nos bordos correspondentes como pode ser visto na Figura 4.1.

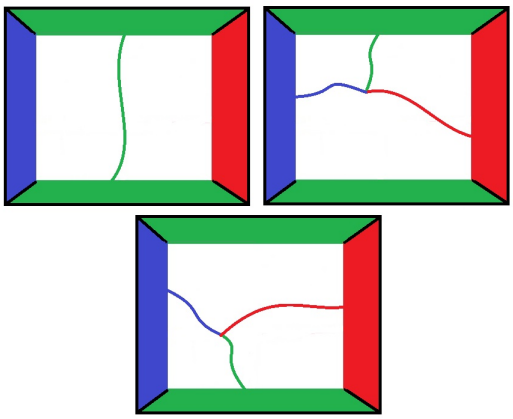

Figura 4.1: Strings homologicamente não-triviais em um polígono de 4 lados.[21]

Porém, podemos observar que uma dessas strings pode ser gerada pelo produto (concatenação) das duas outras. A string verde pode ser gerada pelas duas t-strings, como fica ilustrado na Figura 4.2. Isso significa que as duas t-strings em questão geram qualquer ciclo homologicamente não-trivial dessa superfície, e assim o código $P(1, 2, 1)$ codifica dois qubits com uma ótima propriedade: como ambos os ciclos geradores são t-strings, o código gerado nessa superfície ainda vai implementar todo o grupo de Clifford transversalmente, pois tem as mesmas propriedades essenciais dos códigos triangulares, como foi descrito na Seção 3.4.4.

Seguindo a técnica estabelecida para o crescimento do número de lados de polígonos, para o terceiro caso considere um polígono de 5 lados da forma $P(1, 2, 2)$. Vamos enumerar

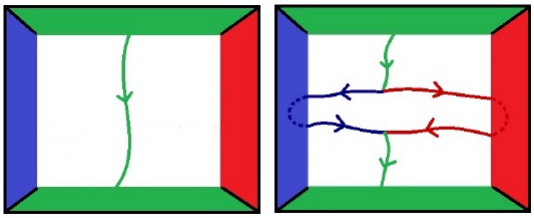

Figura 4.2: A string verde de homologia não-trivial pode ser gerada pela concatenação de duas t-strings também homologicamente não-triviais.[21]

os bordos a fim de facilitar a notação, tomando o bordo azul como base e percorrendo o polígono no sentido horário temos os bordos verdes G_1 e G_2 e os lados vermelhos R_1 e R_2. Analogamente ao que foi considerado no caso do polígono de quatro lados, existem ao todo seis ciclos de homologia não-trivial: um verde (g_{12}) que inicia em G_1 e finaliza em G_2, um vermelho (r_{12}) que inicia em R_1 e finaliza em R_2 além de quatro t-strings, $(T_{11}, T_{12}, T_{21}, T_{22})$ onde denotamos por T_{ij} a string que inicia no bordo azul, bifurca e finaliza nos lados G_i e R_j, como ilustrado na Figura 4.3.

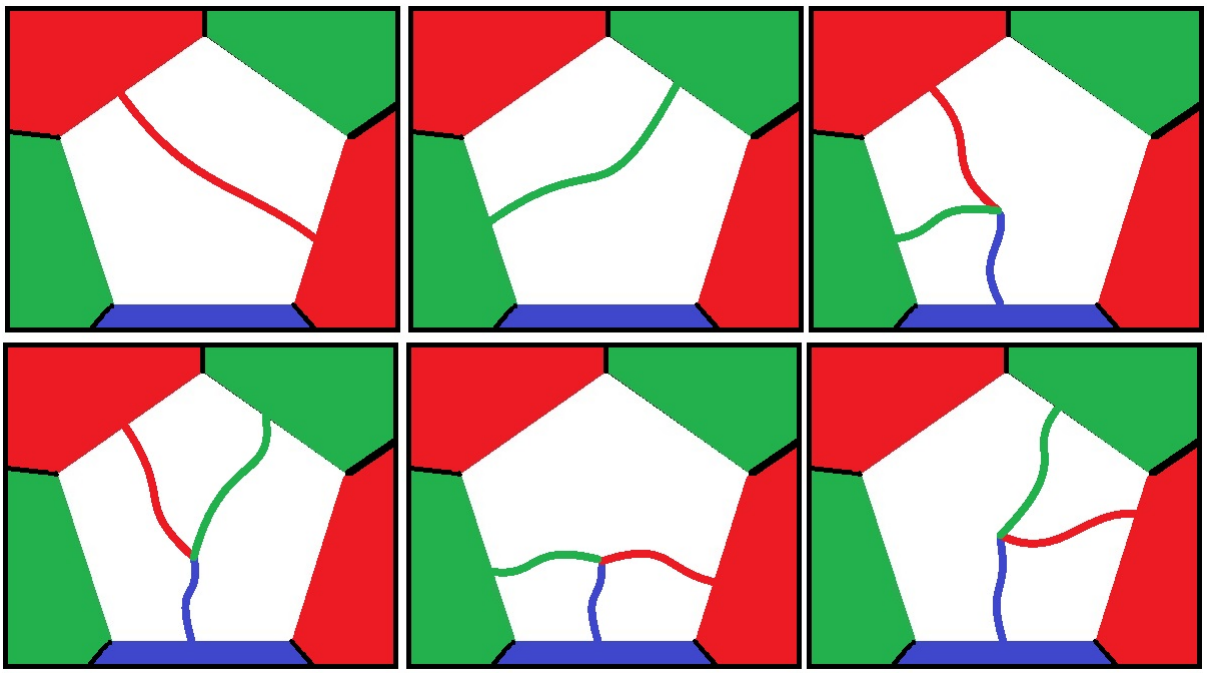

Figura 4.3: Strings homologicamente não-triviais em um polígono de 5 lados[21]

Novamente, apesar de termos seis possibilidades de strings de homologia não-trivial, al-

gumas dessas strings podem ser geradas pelas outras. Observe as seguintes relações:

$$\begin{cases} T_{11} + T_{21} = g_{12} \\ T_{12} + T_{22} = g_{12} \end{cases} \text{ e } \begin{cases} T_{11} + T_{12} = r_{12} \\ T_{21} + T_{22} = r_{12} \end{cases} . \tag{4.1}$$

Podemos observar que, assim como no caso anterior, as strings diretas de uma única cor podem ser geradas por t-strings. Além disso, podemos também gerar uma t-string em função de outras. Por exemplo, do primeiro sistema temos que $T_{11} + T_{21} = T_{12} + T_{22}$ o que implica que $T_{11} + T_{21} + T_{12} = T_{22}$, conclusão essa que seria idêntica ao que chegaríamos pelo segundo sistema.

Portanto, das seis classes de homologia não-triviais, temos que apenas três delas geram todas as outras, neste caso T_{11}, T_{12}, T_{21}.

Claramente poderíamos ter escolhido uma outra combinação de geradores, inclusive alguma combinação que incluísse uma ou mais strings diretas, porém uma escolha apenas com t-strings é melhor para os nossos propósitos.

Generalizando a ideia tem-se o seguinte resultado:

Teorema 4.1. *Seja um polígono com $n+m+1$ lados com $n = m$ ou $m = n+1$. Então pode-se gerar o código $P(1, m, n)$ que codifica $(m+n-1)$ qubits de modo que os qubits codificados são sempre representados por t-strings.*

O fato de $m = n$ ou $m = n + 1$ nos garante que é possivel que o polígono seja construído sem que lados adjacentes tenham a mesma cor. Se tivermos no caso $m = n$, estamos em um polígono com um número ímpar de lados, assim posicionamos como "base"o lado de cor azul (que só tem um). De um dos vértices dessa aresta parte uma sequência de n arestas cujas cores alternam entre verde e vermelha, iniciando pela vermelha. Do outro vértice do lado azul, parte uma outra sequência de arestas cujas cores alternam entre vermelha e verde, iniciando pela verde. Como ambas as sequências tem o mesmo número de arestas e começaram com cores diferentes, então elas terminam com cores diferentes, e então estes dois últimos lados de cada sequência se unem fechando o polígono.

Se tivermos $m = n + 1$, estamos em um polígono com número par de lados. Novamente começamos posicionando o lado azul como base. De cada um dos vértice do lado azul

partirá um sequência de n arestas, ambas as sequências iniciando com o lado verde. Se n for par, existe o mesmo número de arestas vermelhas e verdes em cada sequência e ambas terminam com a cor vermelha, assim usamos o último lado verde para conectar essas duas arestas vermelhas para fechar o polígono. Se n for ímpar existe uma aresta verde a mais que vermelha em cada sequência de arestas, e ambas as sequências terminam com a cor verde. Usamos então a última aresta vermelha conectando essas duas últimas verdes para fechar o polígono ficando oposta à aresta azul.

Em relação ao número de qubits codificados, vamos utilizar a mesma estratégia utilizada no polígono de 5 lados. Tomando o bordo azul como base e percorrendo o polígono no sentido horário temos os bordos verdes $(G_1, ..., G_m)$ e os bordos vermelhos $(R_1, ..., R_n)$. Denotemos por (g_{ij}) a string verde que inicia no bordo G_i e termina no bordo G_j e por (r_{ij}) a string vermelha que inicia no bordo R_i e termina no bordo R_j. Note que $g_{ij} = g_{ji}$ e $r_{ij} = r_{ji}$. Temos também as t-strings (T_{ij}), que são strings que iniciam no bordo azul e terminam nos bordos G_i e R_j.

Novamente, como foi realizado no polígono de 5 lados, observando que as strings podem ser geradas pelas t-strings, podemos contruir sistemas a fim de encontrar nossos geradores do grupo estabilizador do código. Partindo do bordo G_1, temos os seguintes sistemas:

$$\begin{cases} T_{11} + T_{21} = g_{12} \\ T_{12} + T_{22} = g_{12} \\ \vdots \\ T_{1n} + T_{2n} = g_{12} \end{cases} ; \begin{cases} T_{11} + T_{31} = g_{13} \\ T_{12} + T_{32} = g_{13} \\ \vdots \\ T_{1n} + T_{3n} = g_{13} \end{cases} \cdots \begin{cases} T_{11} + T_{m1} = g_{1m} \\ T_{12} + T_{m2} = g_{1m} \\ \vdots \\ T_{1n} + T_{mn} = g_{1m} \end{cases}$$

Para o bordo G_2 obtemos os sistemas:

$$\begin{cases} T_{21} + T_{31} = g_{23} \\ T_{22} + T_{32} = g_{23} \\ \vdots \\ T_{2n} + T_{3n} = g_{23} \end{cases} ; \begin{cases} T_{21} + T_{41} = g_{24} \\ T_{22} + T_{42} = g_{24} \\ \vdots \\ T_{2n} + T_{4n} = g_{24} \end{cases} \cdots \begin{cases} T_{21} + T_{m1} = g_{2m} \\ T_{22} + T_{m2} = g_{2m} \\ \vdots \\ T_{2n} + T_{mn} = g_{2m} \end{cases}$$

E assim por diante até o bordo G_{m-1}:

$$\begin{cases} T_{(m-1)1} + T_{m1} = g_{(m-1)m} \\ T_{(m-1)2} + T_{m2} = g_{(m-1)m} \\ \vdots \\ T_{(m-1)n} + T_{mn} = g_{(m-1)m} \end{cases}$$

Fazendo o mesmo processo para as strings vermelhas. Partindo do bordo R_1:

$$\begin{cases} T_{11} + T_{12} = r_{12} \\ T_{21} + T_{22} = r_{12} \\ \vdots \\ T_{m1} + T_{m2} = r_{12} \end{cases} ; \begin{cases} T_{11} + T_{13} = r_{13} \\ T_{21} + T_{23} = r_{12} \\ \vdots \\ T_{m1} + T_{m3} = r_{13} \end{cases} \cdots \begin{cases} T_{11} + T_{1n} = r_{1n} \\ T_{21} + T_{2n} = r_{1n} \\ \vdots \\ T_{m1} + T_{mn} = r_{1n} \end{cases}$$

Para o bordo R_2:

$$\begin{cases} T_{12} + T_{13} = r_{23} \\ T_{22} + T_{23} = r_{23} \\ \vdots \\ T_{m2} + T_{m3} = r_{23} \end{cases} ; \begin{cases} T_{12} + T_{14} = r_{24} \\ T_{22} + T_{24} = r_{24} \\ \vdots \\ T_{m2} + T_{m4} = r_{24} \end{cases} \cdots \begin{cases} T_{12} + T_{1n} = r_{2n} \\ T_{22} + T_{2n} = r_{2n} \\ \vdots \\ T_{m2} + T_{mn} = r_{2n} \end{cases}$$

E assim por diante até o bordo R_{n-1}:

$$\begin{cases} T_{1(n-1)} + T_{1n} = r_{(n-1)n} \\ T_{2(n-1)} + T_{2n} = r_{(n-1)n} \\ \vdots \\ T_{m(n-1)} + T_{mn} = r_{(n-1)n} \end{cases}$$

Considere o primeiro conjunto de sistemas, os sistemas resultantes das strings que partem do bordo G_1. Para o primeiro sistema, em cada uma das linha podemos obter as equações:

$$\begin{gathered} T_{22} = T_{11} + T_{21} + T_{12} \\ T_{23} = T_{11} + T_{21} + T_{13} \\ \vdots \\ T_{2n} = T_{11} + T_{21} + T_{1n} \end{gathered}$$

Logo,

$$T_{2j} = T_{11} + T_{21} + T_{1j} \tag{4.2}$$

onde $j = 2, ..., n$. Para o segundo sistema, em cada uma das linhas podemos obter a equação:

$$\begin{aligned} T_{32} &= T_{11} + T_{31} + T_{12} \\ T_{33} &= T_{11} + T_{31} + T_{13} \\ &\vdots \\ T_{3n} &= T_{11} + T_{31} + T_{1n} \end{aligned}$$

Assim,

$$T_{3j} = T_{11} + T_{31} + T_{1j}, \tag{4.3}$$

onde $j = 2, ..., n$. Obtemos uma expressão semelhante para cada um dos restantes sistemas. Portanto, obtemos as equações:

$$\begin{aligned} T_{2j} &= T_{11} + T_{21} + T_{1j} \\ T_{3j} &= T_{11} + T_{31} + T_{1j} \\ &\vdots \\ T_{mj} &= T_{11} + T_{m1} + T_{1j}. \end{aligned}$$

Ou seja,

$$T_{ij} = T_{11} + T_{i1} + T_{1j}, \tag{4.4}$$

onde $i = 2, ..., m$ e $j = 2, ..., n$. Além disso, temos que se $i = 1$, temos

$$T_{1j} = T_{11} + T_{11} + T_{1j} = T_{1j} \tag{4.5}$$

e, se $j = 1$, temos

$$T_{i1} = T_{11} + T_{i1} + T_{11} = T_{i1}. \tag{4.6}$$

Portanto, podemos gerar qualquer uma das t-strings usando a equação 4.4. Dessa forma, temos que as t-strings $(T_{11}, T_{12}, ..., T_{1n}, T_{21}, T_{31}, ..., T_{m1})$ geram todas as t-strings e o código codifica $m + n - 1$ qubits.

4.1 Nova família de códigos coloridos

Nesta seção focaremos na construção dos códigos do tipo $P(1,2,1)$. Estes códigos possuem as mesmas propriedades dos códigos triangulares, mas apresentam melhores parâmetros.

Iniciamos considerando um quadrilátero e usamos a tesselação semirregular $\{4,8,8\}$. No exemplo da Figura 4.4 usamos um retângulo de medidas $(2+\sqrt{2}) \times (1+\sqrt{2})$ e os polígonos da tesselação possuem lados 1 gerando, assim, um parâmetro $[12,2,3]$.

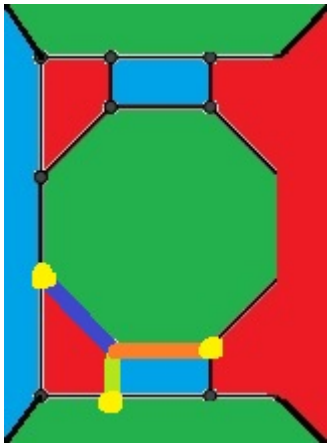

Figura 4.4: Código $P(1,2,1)$, com parâmetros $[12,2,3]$, gerado pela tesselação $\{4,8,8\}$ em um quadrilátero. Destacamos uma t-stirng com distância de código mínima, com os suportes em amarelo.[21]

Comparando este código com o código triangular em [4] podemos ver que ambos tem distância 3, mas a taxa de codificação de $P(1,2,1)$ é $\frac{1}{6}$ enquanto no triangular temos uma taxa de codificação de $\frac{1}{7}$.

Aumentando tanto a largura e o comprimento do retângulo por $2+\sqrt{2}$ a distância mínima do código vai aumentando de 2 em 2. Então é possível ter um controle de quantos qubits o código contem. A Figura 4.5 exemplifica dois desses passos. Assim, seguindo esse procedimento conseguimos gerar uma família de códigos do tipo $P(1,2,1)$ com parâmetros $[8m^2+2m+2, 2, 2m+1]$, $m \geq 1$.

Comparando com o código triangular (3.11), o código $P(1,2,1)$ da família descrita acima com a mesma distância mínima é de $[138,2,9]$, cuja taxa de codificação é $\frac{1}{69}$ enquanto que a taxa de codificação do triangular é $\frac{1}{73}$, mostrando que o uso dos códigos poligonais pode

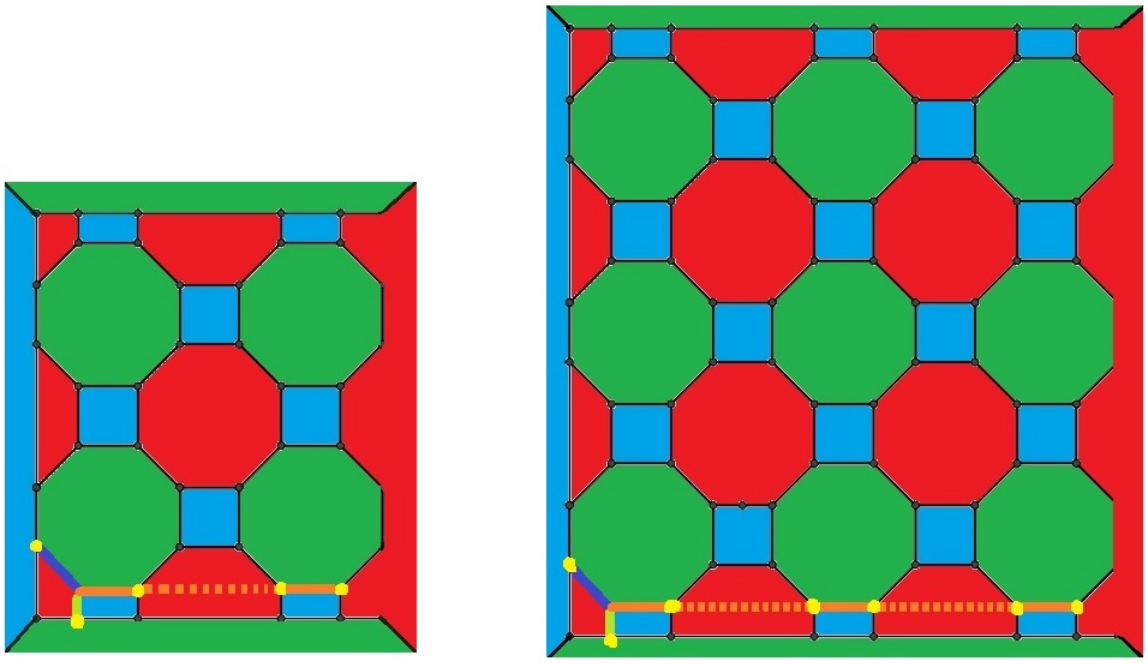

Figura 4.5: Códigos poligonais do tipo $P(1, 2, 1)$, com parâmetros $[38, 2, 5]$ e $[80, 2, 7]$, gerado pela tesselação $\{4, 8, 8\}$ que pertencem a família de códigos $[8m^2 + 2m + 2, 2, 2m + 1]$.[21]

apresentar uma melhoria com relação aos códigos triangulares.

Considerações Finais

Concluímos que os códigos coloridos poligonais apresentados em [21] apresentam uma melhoria com relação aos códigos coloridos triangulares apresentados em [4], pois além de conseguirmos aumentar o número de qubits codificados sem perder a propriedade de implementar todo o grupo de Clifford, conseguimos também melhorar a taxa de codificação do código, como visto na Seção 4.1.

Destacamos ainda que já está sendo realizado um estudo para uma possível continuação a ser realizada no doutorado, onde o professor Eduardo Brandani da Silva, o aluno Evandro Mazetto Brizola e eu, já iniciamos estudos relacionados a ruídos quânticos para que possamos adentrar na parte de decodificação quântica.

Referências

[1] Albuquerque, C. D., *Análise e construção de códigos quânticos topológicos sobre variedades bidimensionais*, Ph.D. Tese, Universidade Estadual de Campinas, 2009.

[2] Albuquerque, C. D., Palazzo Jr., R. , Silva. E. B., *On toric quantum codes*, Int. J. Pure Appl. Math, 50, pp 221, 2009.

[3] Albuquerque, C. D., Palazzo Jr., R. and Silva, E. B., *Topological quantum codes on compact surfaces with genus* $g \geq 2$, J. Math. Phys., 50, 2009

[4] Bombin, H. , Martin-Delgado, M. A., *Topological quantum distillation*, Physical Review Letters 97, 2006.

[5] Bombin, H., Martin-Delgado, M. A., *Topological Quantum Error Correction with Optimal Encoding Rate*, Phys. Rev., 73, 2007.

[6] Bombin, H., Martin-Delgado, M. A. (2007), *Computacion cuántica topológica y sistemas fuertemente correlacionados*, Revista Espanola de Fisica. 21, pp. 31, 2007.

[7] Bredon, G. E., *Topology and Geometry*, Springer-Verlag, 1993.

[8] Breuckmann, N. P. , Terhal, B. M., *Constructions and noise threshold of hyperbolic surface codes*, IEEE transactions on Information Theory, 62, 2016

[9] Calderbank, A. R. and Shor, P. W., *Good quantum error-correcting codes exist*, Phys. Rev. A 54, pp. 1098, 1996

[10] Delfosse, N., *Tradeoffs for reliable quantum information storage insurface codes and color codes*, IEEE International Symposium on Information Theory, pp. 917, 2013.

[11] Giblin, P., *Graphs, Surfaces and Homology*, Cambridge University Press, 2010.

[12] Gottesman, D., *Class of quantum error-correcting codes saturating the quantum Hamming bound*, Phys. Rev. A 54, pp. 1862, 1996.

[13] Kitaev, A. Y., *Fault-tolerant quantum computation by anyons*, Annals of Physics, 303, pp. 2, 2003.

[14] Lidar, D, Brun, T., *Quantum Error Correction*, Cambridge University Press, 2013.

[15] Lima, E. L., *Álgebra Linear*. 7ª.ed. Rio de Janeiro: IMPA, 2006

[16] Nielsen, M. A. and Chuang, I. L. *Computação Quântica e Informação Quântica.* Bookman, 2003.

[17] Nigg, D. , Mueller, M. , Martinez, E. A. , Schindler, P. , Henrich, M. , Monz, T. , Martin-Delgado, M. A. , Blatt, R., *Quantum Computations on a topologically encoded qubit*, Science, 345, pp. 302, 2014.

[18] Preskill, J., *Lectures notes for physics 219: quantum error correction*, California Institute of Technology, 1999.

[19] Shannon, C.E., *A Mathematical Theory of communication*, The Bell System Technical Journal, vol. XXVIII, april, 1950.

[20] Shor, P. W., *Scheme for reducing decoherence in quantum computer memory*, Phys. Rev., 52, pp. 2493, 1995.

[21] Silva. E. B.; Soares, W. S., *Construction of color codes from polygons.* Journal of Physics Communications , v. 2, p. 095011, 2018.

[22] Soares, W.S., *Novos Métodos de Construção de Códigos Quânticos Coloridos sobre Superfícies Bidimensionais*. Tese, Universidade Estadual de Maringá, 2017.

[23] Steane, A. M., *Simple Quantum Error Correction Codes*, Phys. Rev. Letters A 54, pp. 4741, 1996.

[24] Vick, J. W. , *Homology Theory: An Introduction to Algebraic Topology*, Graduate Texts in Mathematics, Springer, 2012.