The Hebrew University of Jerusalem
The Rachel and Selim Benin School of Computer Science and Engineering

Understanding Quantumness And Testability.

David Ponarovsky

Thesis submitted in partial fulfillment of the requirements
for the Master of Sciences degree
in Computer Science

Under the supervision of **Prof. Michael Ben Or**

August 2023

אבחון בתורה קוונטית

מוגש על ידי
דויד פונרובסקי

עבודת גמר לתואר מוסמך במדעי המחשב

עבודה זו הונחתה על ידי
פרופ׳ מיכאל בן אור

אוגוסט 2023

תקציר

מחשוב קוונטי הוא מודל חישוב מבטיח, הקוונצזיוס הרווח הוא שקימות בעיות ניתן לפתור באופן יעיל בהמצאות מחשבים קוונטים אך הן קשות בכדי לפתור אותן עם המחשבים הקלאסים. הדוגמא הבולטת היא הסימולציה הכימית, בעוד שהאנושות היום יודעת לסמלץ ריצה של מעגלים חשמלים, חזק של מבנים וטיסות של מטוסים אינ־נו מסוגלים, גם בעזרת מחשב העל החזק ביותר בכדו"א, לסמלץ את התפתחות בזמן של מלקולה בת 80 אטומים. השיטה הנאיבית לסימלוץ קלאסי תדרוש מאיתנו לעקוב אחר מס' אקספוננציאלי במספר האטומים של משתנים בזכרון. לעומת זאת, חומרה קוונטית ־או מכניקת הקוונטים־ מאפשרת קידוד יעיל של המלקולה באופן זהה למבנה שלה בטבע. אומנם איננו יודעים להוכיח כי אין אלגוריתמים קלאסים שמסמלצים מלקולות בזמן מהיר אך אנו יודעים להגיד בוודאות שהסימולציה הקוונטית היא קלה למימוש, רצה בזמן מהיר ויעלה בשימוש של משאבי זכרון. על כן החזון העומד לנגד עיני רוב המומחים הוא שבעתיד תהליך פיתוח התרופות והנדסת החומרים יהפוך לשיטתי, אם היום, מרגע שחוקר מציע מועמדים לתרופות או חיסונים, נדרשים לחכות זמן רב לאישור הסימולציה בעתיד החוקר יקבל תשובה בזמן אמת על מסך. בדומה לקלות שהיום אנשים מסוגלים לשתף ספריות של קוד או חומרה, להקים חברות ולשחרר מוצרים כך שבעתיד ניראה תעשיה מורכבת ומקושרת המתעסקת בפיתוח התרופות, חומרים, הנדסה גנטית ועוד.

ישנה רק בעיה קטנה, אומנם אין ויכוח כי מודל החישוב הקוונטי חזק מהמודל הקלאסי אך כלל לא ברור שמודל החישוב הקוונטי בכלל ניתן למימוש, זאת מאחר והרכבים הבסיסים בחומרה הקוונטית נוטים לסבול משגיאות רבות, יותר מכך, נכון להיום אין לנו מחשבים קוונטים שמתגברים על השגיאות וכל חישוב ארוך מעט צובר מספיק שגיאות עד כי תוצאת החישוב הסופית שקולה לג'בריש. מול בעיה דומה עמד וון־פון ניומאן, בין ממציאי המחשב הקלאסי, לפני קצת פחות ממאה, פון ניומאן שאל האם חישוב קלאסי בכלל אפשרי? האם ניתן להתגבר על הרעש? הוא ענה על השאלה בחיוב. ראוי לציין שמאז גם החומרה התקדמה ורוב הרעיונות של "חישוב חסין לשגיאות" לא ממושו, אבל בהחלט חלק גדול מהרעיונות מומשו ומהווים רכיבים קריטים במוצרים יומיומים בהם אנו משתמשים, דוגמא בולטת היא תקשורת סוללרית בפרוטוקול 5G המשתמש בקודי LDPC (עליהם נרחיב בעבודה) כדאי לתקן שגיאות. באופן דומה חוקרים הראו כי ניתן להכליל את הרעיונות הקלאסים למקרים הקוונטים בכדי לאפשר "חישוב קוונטי חסין לשגיאות". בבסיס הרעיונות עומד קידוד מיוחד בקוד המאפשר לזהות ולתקן שגיאות בקלות.

נוסף על קודים שמתקנים שגיאות קיימים גם קודים הניתנים לבדיקה באופן יעיל.כדאי להבין מהם קודים ניתנים לבדיקה כדאי לדמיין בראש פאזל בו כל החתיכות צבועות בדיוק באותו הצבע. פאזל שכזה יקרא לא ניתן לבדיקה אם קיימת השמה של חיתיכות הפאזל כך שמצד אחד כדאי להגיע אליה יש להחליף חצי מהחלקים ב"ההשמה האמית" ־כלומר הדרך הנכונה לחבר אותו־ אך מצד שני קיימות מספר בודד של אי התאמות בין חיבורי החלקים בשונים. כלומר צריך לעבוד הרבה כדאי להגיע ל"ההשמה האמיתית" אך אם דוגמים חיבור שרירותי, בסיכוי אפסי ניראה סתירה. פאזלים הניתנים לבדיקה הם בדיוק ההפך הגמור, אם השמה רחוקה מאוד מלהיות "ההשמה האמיתית" אז בהסתברות גבוהה בדיקה שרירותית תחשוף זאת.קיימם של קודים קוונטים טובים וקודים קלאסים הניתנים לבדיקה יעילה היו שאלות פתוחות במשך זמן רב, ולמרות שניראה שאין קשר ישיר בייניהם שניהם נפתרו באיבחה אחת בבניה מתקדמת שפותחה רק לפני שנתיים (2021). מעניין לשאול האם מדובר בצירווף מקרים,או שיש קשר עמוק יותר בין השתיים שאיננו מבינים עוד.

בעבודה זאת, אנחנו סוקרים, מקצה לקצה, קודי מתקני השגיאות הקוונטים. אחל מסקירה רחבה של קודים קלאסים, שיטות לבנייה של קודים טובים, אפיון של רעש קוונטי וקודים קוונטים פרמטיבים ועד לבניה המתקדמת של קודים קוונטים LDPC טובים. כל זאת לצד סקירה של קודים הניתנים לבדיקה. לבסוף אנחנו מציגים ניסיונות, כושלים, במחקרי המשך, בפרט ניסיון להגיעה לבניה של

קודים קלאסים הניתנים לבדיקה מבלי לפתח במקביל גם קודים קוונטים טובים. אם היינו מצליחים לבודד את התוצאות הדבר היה מצביע על כך שהעובדה ששניהם פותחו באותה בניה היא מקרית לחלוטין. יחד עם זאת, לא התקדמנו מספיק רחוק כדאי לעלות חשד סביר שקיים קשר הכרחי.

העבודה מניחה רק ידע בסיסי באלגברה לינארית וקומבינטוריה, ואנו סבורים שכל בוגר מדעי המחשב יוכל להנות מקריאתה, להבין באופן טוב מאוד את הנושא, ולהעזר בה כדאי להתחיל מחקר בתחום.

Contents

List of Figures

Chapter 1

Introduction

Many experts believe that quantum computing is a highly effective computation model. The consensus is that certain problems that are difficult for classical computers to solve can be effectively solved with quantum computers. One of the biggest challenges in simulation is chemical simulation. Although we have the ability to simulate the workings of electrical circuits, the durability of structures, and the flight of aircrafts, simulating the evolution of an 80-atom molecule over time is still beyond our capabilities, even with the assistance of the world's most powerful supercomputer. When it comes to simulating molecules, the classical method can be inefficient as it requires tracking a large number of variables in memory. However, quantum hardware, based on the principles of quantum mechanics, allows for a more efficient encoding of the molecule's structure, mimicking the way it exists in nature. Although it is unclear how to prove that classical algorithms cannot efficiently simulate molecules, we do know that quantum simulation is easily achievable and uses memory resources efficiently. Therefore, experts envision that in the future, drug development and materials engineering will become more systematic. Today, researchers proposing candidates for drugs or vaccines may have to wait a long time for approval, but in the future, simulations will provide real-time answers on a screen. This will be similar to the ease with which people share libraries of code or hardware, establish companies, and release products. As a result, we will see a complex and interconnected industry dealing with the development of drugs, materials, genetic engineering, and more.

Despite the undeniable superiority of the quantum computation model over the classical model, there remains a significant challenge: the implementation of the quantum computation model. The essential components of quantum hardware are prone to numerous errors, and currently, no quantum computers can overcome these errors. Even a slightly lengthy computation accumulates enough errors to render the final computation result incomprehensible. This issue is reminiscent of a problem faced by Von Neumann, one of the inventors of the classical computer, nearly a century ago. He asked if classical computation was possible despite the noise and answered affirmatively. It is worth noting that hardware has advanced considerably since then, but most of the ideas of "fault tolerance computation" have yet to be realized. However, many ideas are critical components of everyday products, such as the LDPC codes used for error correction in cellular communication in the G5 protocol. Researchers have also demonstrated that classical ideas can be generalized to quantum cases to enable "quantum fault tolerance computation." The ideas are based on a unique code encoding system that facilitates easy identification and correction of errors.

In addition to error correction codes, there are also codes that desinged to be tested effectively. To understand what testable codes are, Imagine in your head a puzzle in which all the pieces are

painted in exactly the same color. Such a puzzle will be called untestable if there is a placement of the pieces of the puzzle so that on the one hand it is worthwhile to reach it, half of the pieces must be swapped for obtaining the "true placement" of the puzzle - that is, the correct way to put it together - but on the other hand there are a few discrepancies between the connections of the different parts. That is, you have to work a lot to get to the "real placement," but if you sample an arbitrary connection, there is zero chance that a contradiction will appear. Testable puzzles are the exact opposite; if a placement is very far from the "true placement," then an arbitrary test will most likely reveal it. The existence of good quantum codes and efficiently testable classical codes have been open questions for a long time. Although, it seems like, there is no direct connection between them, both have been obtained by the same one advanced construction that was developed only two years ago (2021). Hence, it is interesting to ask if this is a coincidence or if there is a deeper connection between the two that we no longer understand.

In this work, we review, end to end, the quantum error correction codes. Starting from an overview of classical codes, methods for constracting good codes, characterization of quantum noise and permutative quantum codes, to the advanced construction of good LDPC quantum codes. All this is alongside a review of testable codes. Finally, we present failed attempts in follow-up research, in particular, an attempt to reach the construction of testable classical codes without simultaneously developing good quantum codes. If we were able to isolate the results, it would indicate that the fact that both were obtained in the same constraction is completely coincidental. At the same time, we haven't progressed far enough to suspect that there is a necessary connection reasonably.

The work assumes only a basic knowledge of linear algebra and combinatorics. So we believe that every computer science graduate will be able to enjoy reading it, understand the subject very well, and use it as a gateway for starting research in the field.

Chapter 2

Codes

2.1 Introduction

Coding theory has emerged due to the need to transfer information in noisy communication channels. By embedding a message in a higher-dimensional space, one can guarantee robustness against possible faults. The ratio of the original content length to the transmitted message *length* is the *rate* of the code, and it measures how consuming our communication protocol is. Additionally, the *distance* of the code quantifies how many faults the scheme can absorb such that the receiver can recover the original message. We can consider the code as a collection of all strings that satisfy specified restrictions.

Non-formally, a code is good if its distance and rate scale linearly with the encoded message length. In practice, one is also interested in implementing these checks efficiently. We say that a code is an LDPC if any bit is involved in a constant number of restrictions, each of which is a linear equation, and if any restriction contains a fixed number of variables.

Moreover, another characteristic of the code is its testability, which is the complexity of the number of random checks one must do to verify that a given candidate is in the code. Besides being considered efficient in terms of robustness and overhead, good codes are also vital components in establishing secure multiparty computation [BGW19] and have a deep connection to probabilistic proofs.

In Section 2, we state the notations, definitions, and formal theorem. Then, in Sections 3 and 4, we review past results and provide their proofs to make this paper self-contained. Readers familiar with the basic concepts of LDPC, Tanner, and Expanders codes construction may consider skipping directly to Section 5, in which we provide our proof. Readers familiar with the basic concepts of LDPC, Tanner, and Expanders codes construction may skip Sections 2, 3, and 4 and proceed directly to Section 5, where we provide our proof.

2.2 Codes in General. Notations and Definitions.

Here we focus only on linear binary codes, which one could think about as linear subspaces of $\mathbb{F}_2^n$. A common way to measure resilience is to ask how many bits an evil entity needs to flip such that the corrupted vector will be closer to another vector in that space than the original one. Those ideas were formulated by Hamming [Ham50], who presented the following definitions.

Definition 2.2.1. *Let* $n \in \mathbb{N}$ *and* $\rho, \delta \in (0,1)$. *We say that* C *is a **binary linear code** with parameters* $[n, \rho n, \delta n]$. *If* C *is a subspace of* $\mathbb{F}_2^n$, *and the dimension of* C *is at least* ρn. *In addition, we call the vectors belong to* C *codewords and define the distance of* C *to be the minimal number of different bits between any codewords pair of* C.

From now on, we will use the term code to refer to linear binary codes, as we don't deal with any other types of codes. Also, even though it is customary to use the above parameters to analyze codes, we will use their percent forms called the relative distance and the rate of code, matching δ and ρ correspondingly.

Definition 2.2.2. *A **family of codes** is an infinite series of codes. Additionally, suppose the rates and relative distances converge into constant values* ρ, δ. *In that case, we abuse the notation and call that family of codes a code with* $[n, \rho n, \delta n]$ *for fixed* $\rho, \delta \in [0,1)$, *and infinite integers* $n \in \mathbb{N}$.

Notice that the above definition contains codes with parameters attending to zero. From a practical view, it means that either we send too many bits, more than a constant amount, on each bit in the original message. Or that for big enough n, adversarial, limited to changing only a constant fraction of the bits, could disrupt the transmission. That distinction raises the definition of good codes.

Definition 2.2.3. *We will say that a family of codes is a **good code** if its parameters converge into positive values.*

2.3   Singleton Bound

To get a feeling of the behavior of the distance-rate trade-of, Let us consider the following two codes; each demonstrates a different extreme case. First, define the repetition code $C_r \subset \mathbb{F}_2^{n \cdot r}$, In which, for a fixed integer r, any bit of the original string is duplicated r times. Second, consider the parity check code $C_p \subset \mathbb{F}_2^{n+1}$, in which its codewords are only the vectors with even parity. Let us analyze the repetition code. Clearly, any two n-bits different messages must have at least a single different bit. Therefore their corresponding encoded codewords have to differ in at least r bits. Hence, by scaling r, one could achieve a higher distance as he wishes. Sadly the rate of the code decays as $n/nr = 1/r$. In contrast, the parity check code adds only a single extra bit for the original message. Therefore scaling n gives a family which has a rate attends to $\rho \to 1$. However, flipping any two different bits of a valid codeword is conversing the parity and, as a result, leads to another valid codeword.

To summarize the above, we have that, using a simple construction, one could construct the codes $[r, 1, r], [r, r-1, 2]$. Each has a single perfect parameter, while the other decays to the worst.

Besides being the first bound, Singleton bound demonstrates how one could get results by using relatively simple elementary arguments. It is also engaging to ask why the proof yields a bound that, empirically, seems far from being tight.

Theorem (Singleton Bound.)**.** *For any linear code with parameter* $[n, k, d]$, *the following inequality holds:*

$$k + d \le n + 1$$

Proof. Since any two codewords of C differ by at least d coordinates, we know that by ignoring the first $d - 1$ coordinate of any vector, we obtain a new code with one-to-one corresponding to

the original code. In other words, we have found a new code with the same dimension embedded in $\mathbb{F}_2^{n-d+1}$. Combine the fact that dimension is, at most, the dimension of the container space, we get that:

$$\dim C = 2^k \leq 2^{n-d+1} \Rightarrow k + d \leq n + 1$$

$\square$

It is also well known that the only binary codes that reach the bound are: $[n, 1, n], [n, n-1, 2], [n, n, 1]$ [AF22]. In particular, there are no good binary codes that obtain equality(And no binary code which get close to the equality exits). Let's review the polynomial code family [RS60], which is a code over none binary field that achieve the Singleton Bound.

Next, we will review Tanner's construction, that in addition to being a critical element to our proof, also serves as an example of how one can construct a code with arbitrary length and positive rate.

2.4 Tanner Code

The constructions require two main ingredients: a graph Γ, and for simplicity, we will restrict ourselves to a Δ regular graph, Yet notice that the following could be generalize straightwardly for graphs with degree at most Δ. The second ingredients is a ;small' code C_0 at length equals the graph's regularity, namely $C_0 = [\Delta, \rho\Delta, \delta\Delta]$. We can think about any bit string at length Δ as an assignment over the edges of the graph. Furthermore, for every vertex $v \in \Gamma$, we will call the bit string, which is set on its edges, the local view of v. Then we can define, [Tan81]:

Definition 2.4.1. *Let $C = \mathcal{T}(\Gamma, C_0)$ be all the codewords which, for any vertex $v \in \Gamma$, the local view of v is a codeword of C_0. We say that C is a **Tanner code** of Γ, C_0. Notice that if C_0 is a binary linear code, So C is.*

Example 2.4.1. *Consider the Petersen graph Γ, which is a regular graph with degree 3. Let C_0 be the set of all words with even parity. It follows that C_0 contains all even-length binary strings of length 3: 000, 110, 101, and 011. However, the size of $\mathcal{T}(\Gamma, C_0)$ is significantly larger, as shown in Figure fig. 2.1. Specifically, any rotation of the inner and outer cycles simultaneously gives rise to another valid codeword, so any assignments that are not invariant under these rotations would produce five additional valid codewords.*

Lemma 2.4.1. *Tanner codes have a rate of at least $2\rho - 1$.*

Proof. The dimension of the subspace is bounded by the dimension of the container minus the number of restrictions. So assuming non-degeneration of the small code restrictions, we have that any vertex count exactly $(1 - \rho)\Delta$ restrictions. Hence,

$$\dim C \geq \frac{1}{2}n\Delta - (1 - \rho)\Delta n = \frac{1}{2}n\Delta(2\rho - 1)$$

Clearly, any small code with rate $> \frac{1}{2}$ will yield a code with an asymptotically positive rate $\square$

Figure 2.1: Peterson Graph.

Based on Lemma 2.4.1, we can obtain a recipe for constructing codes with a almost non-vanishing rate for arbitrarily large lengths and dimensions. This recipe involves concatenating a series of Tanner codes over complete graphs. To be more precise, we can define a family of codes as follows:

$$C_{i+1} = \mathcal{T}\left(K_{n(C_i)+1}, C_i\right)$$
$$C_0 = \text{ Some simple } \Delta[1, \rho_0, \delta_0] \text{ code.}$$

Where $n(C_i)$ represents the code length of the ith code. Repeating the process described above $\log_\Delta^*(n)$ times allows us to extend the initial code $\Delta[1, \rho_0]$ to $n[1, \sim 2\rho^{\log_\Delta^*(n)}]$. Interestingly, any family of finite groups generated by a constant-size generator set can define a family of codes by utilizing their Cayley graphs as a basis for Tanner codes.

Once we have seen that Tanner codes enable us to achieve rates, the next natural question to ask is about the distance of the codes. Achieving a linear distance requires a little bit more from the graphs, but to understand this idea better, let us return to the repetition code. For instance, the repetition code can be presented as a Tanner code over the cycle graph.

Example 2.4.2. *In this representation, each vertex checks if the bits on its edges are equal. A valid codeword is an assignment in which all the bits are equal, since otherwise, there would be an edge with no supporting vertex. An illustration of a legal assignment is provided in section 2.4.*

Recall that the distance of a linear code is the minimal weight of the non-zero codewords. Consider a codeword $c \in C$ and group the vertices by four sets V_i such that V_i is the set of vertices that see $i \in \{0, 1\}^2$. Since $c \in C$, we have that $|V_{10}| = |V_{01}| = 0$. Additionally, any vertex in V_{00} is not connected to V_{11}, which gives us two possible cases: either all the vertices in V_{11} are isolated, or the graph is not connected. Hence, the distance of the code is equal to $\frac{1}{2} \sum |V_i| \cdot |i| = \frac{1}{2} 2 \cdot n = n$.

Figure 2.2: The 1^n assignment on the cycle graph. Any vertex compute parity $1 + 1 = 0$, therefore all the restrictions are satisfied and $1^n \in \mathcal{T}$ (cycle , parity).

It is worth mentioning that, in the literature, the repetition code is not usually given as an example of a Tanner code. However, this example will come up again later in the chapter on quantum codes, when we discuss the Toric code, its relation to the hyperproduct code, and how it can be seen as a hyperproduct of two cycle codes.

Furthermore, analyzing the repetition code gives a clue as to how, in certain cases, one might prove a lower bound on the code distance. We would like to say that, if the weight of the code word is below the distance, then it must be that there is at least one vertex that has a non-trivial local view which is not a codeword in C_0. Put differently, we cannot spread a small weight codeword over $\{V_i\}$, defined above, without expanding into subsets corresponding to low $|i|$. Next, we are going to present the Expander codes, which are Tanner codes constructed from graphs with good algebraic expansion.

2.5   Expander Codes

We saw how a graph could give us arbitrarily long codes with a positive rate. We will show, Sipser's result [SS96] that if the graph is also an expander, we can guarantee a positive relative distance.

Definition 2.5.1. *Denote by λ the second eigenvalue of the adjacency matrix of the Δ-regular graph. For our uses, it will be satisfied to define expander as a graph $G = (V, E)$ such that for any two subsets of vertices $T, S \subset V$, the number of edges between S and T is at most:*

$$\mid E\,(S, T) - \frac{\Delta}{n}|S||T| \mid \,\leq \lambda \sqrt{|S||T|}$$

This bound is known as the Expander Mixing Lemma. We refer the reader to [HLW06] for more

deatilied survery.

Figure 2.3: Expander Graph. Any small set of vertices, like the isolated set on the left, has more edges leaving it than intermediate edges. The exact amount is controlled by the expansion factor.

Theorem. *Theorem, let C be the Tanner Code defined by the small code $C_0 = [\Delta, \delta\Delta, \rho\Delta]$ such that $\rho \geq \frac{1}{2}$ and the expander graph G such that $\delta\Delta \geq \lambda$. C is a good LDPC code.*

Proof. We have already shown that the graph has a positive rate due to the Tunner construction. So it's left to show also the code has a linear distance. Fix a codeword $x \in C$ and denote By S the support of x over the edges. Namely, a vertex $v \in V$ belongs to S if it connects to nonzero edges regarding the assignment by x, Assume towards contradiction that $|x| = o(n)$. And notice that $|S|$ is at most $2|x|$, Then by The Expander Mixining Lemma we have that:

$$\frac{E(S,S)}{|S|} \leq \frac{\Delta}{n}|S| + \lambda$$
$$\leq_{n \to \infty} o(1) + \lambda$$

Namely, for any such sublinear weight string, x, the average of nontrivial edges for the vertex is less than λ. So there must be at least one vertex $v \in S$ that, on his local view, sets a string at a weight less than λ. By the definition of S, this string cannot be trivial. Combining the fact that any

nontrivial codeword of the C_0 is at weight at least $\delta\Delta$, we get a contradiction to the assumption that v is satisfied, videlicet, x can't be a codeword $\qquad\square$

Now, we have an explicit construction for good classical LDPC codes. Next, we present a randomized construction which gives us non-LDPC, but good, codes of arbitrary length. We have to present this construction rather than stopping here, as for proving good quantum codes, we have to rely on assumptions about the small code which hold only for codes of sufficient length.

2.6   Randomized Constructions.

Randomized constraction reffers to taking as eithr generator or pairty check matrix a matrix that sampled uniformly random. That is, we are going show that if the generator matrix A is a random matrix in $M(\mathbb{F}_2^{k\times n})$ then Im A is a good code.

Claim 2.6.1. *Let $n, k \in \mathbb{N}$ such $k = \Theta(n)$ and let A be a random matrix in $M(\mathbb{F}_2^{k\times n})$ then the code generated by A is good code with high probability.*

Let's start with the following claim:

Claim 2.6.2. *Let A be a random matrix in $M(\mathbb{F}_2^{k\times n})$. For any non-zero $x \in \mathbb{F}$, we have that Ax is distributed uniformly.*

Proof. By the fact that $x \neq 0$, there exists at least one coordinate $i \in [k]$ such that $x_i \neq 0$. Thus, we have

$$(Ax)_j = \sum_k A_{jk}x_k = \sum_{i\neq k} A_{jk}x_k + A_{ji}x_i$$
$$= \sum_{i\neq k} A_{jk}x_k + A_{ji}$$

Notice that due to the fact that $\mathbb{F}_2$ is a field, there is exactly one assignment that satisfies the equation conditioned on all the values A_{jk} where $j \neq k$.

$$\mathbf{Pr}\left[(Ax)_j = 1\right] = \sum_{A_{jk};k\neq i} \mathbf{Pr}\left[(Ax)_j = 1 \mid A_{jk}; k \neq i\right] \mathbf{Pr}\left[A_{jk}; k \neq i\right]$$
$$= \frac{1}{2}$$

Therefore, any coordinate of Ax is distributed uniformly $\Rightarrow Ax$ is distributed uniformly. $\qquad\square$

proof of 2.6.1. By the uniformity of Ax, we obtain that the expected Hamming weight of Ax is:

$$\mathbf{E}\left[|Ax|\right] = \mathbf{E}\left[\sum_i^n (Ax)_i\right] = \frac{1}{2}n$$

As the coordinates of A_x are independent (each row of A is sampled separately), we can use the Hoff's bound to conclude that:

$$\mathbf{Pr}\left[||Ax| - \mathbf{E}\left[|Ax|\right]| \geq \left(\frac{1}{2} - \delta\right)n\right] \leq e^{-n\left(\frac{1}{2}-\delta\right)^2}$$

Now, we will use the union bound to show that any $x \in \mathbb{F}_2^k$, Ax is of weight at least δ.

$$\mathbf{Pr}\left[|Ax| \geq \delta : \forall x \in \mathbb{F}_2^k\right] \geq 1 - |\mathbb{F}_2^k| \cdot e^{-n\left(\frac{1}{2}-\delta\right)^2}$$

Let $k = \rho n$ and note that the statement holds when $\rho \geq \left(\frac{1}{2} - \delta\right)^2$. In other words, for any $k = \Theta(n)$, there exists a positive δ such that with high probability, the distance of Im is at least δn. $\qquad\square$

2.7 Locally Testable Codes.

In addition to distance and rate, we are also interested in ensuring that the checking process is robust. Specifically, we want to guarantee that a single missed check will only have a tiny probability of compromising the entire computation, even in the presence of significant errors.

Definition 2.7.1. *Consider a code C a string x, and denote by $\xi(x)$ the fraction of the checks in which x fails. C will be called a **local-testabile** $f(n)$ If there exists $\kappa > 0$ such that*

$$\frac{d(x, C)}{n} \leq \kappa \cdot \xi(x) f(n)$$

Next, we will introduce the polynomial code, a testable code. We will then demonstrate a proof that emphasizes the concept of restriction degeneration, meaning that if more than a certain number of restrictions are satisfied, then almost all the remaining restrictions must also be satisfied. It is also worth mentioning that the polynomial code has a nice quantum variant which we will present in the next chapter.

2.8 Polynomial Code.

Consider the field $\mathbb{F}_q$ for an arbitrary prime power q greater than n. The polynomial codes rely on the fact that any two different polynomials in the ring $\mathbb{F}_q[x]$ of degree at most d differ by at least $q - d + 1$ points. For example, consider a pair of polynomials of degree 1, namely two linear straight lines. If they are not identical, then they have at most one intersection point, and they disagree on each of the $n-1$ remaining points. By defining the code to be the subspace containing all polynomials of degree at most d, such that any codeword is an image of such a polynomial encoded by n numbers, we can guarantee a lower bound on the code's distance. Formally, we define:

Definition 2.8.1 (Polynomial Code. [RS60])**.** *Fix $m > n$ to be a prime power and let $a_0, a_1, a_2, \ldots a_n$ distinct points of the field $\mathbb{F}_q$ and define the code C as follows:*

$$C = \{p(a_0), p(a_1), p(a_2), \cdots p(a_n) : p \text{ is polynomial at degree at most } d\}$$

Observe that C is a linear code of length n over the alphabet $\mathbb{F}_q$. The following Lemma states the relation between the maximal degree of the polynomials and the properties of the code.

Lemma 2.8.1. *Fix the degree of the polynomial code to be at most d. Then the parameters of the code are $[n, d+1, n-d]$.*

Proof. The dimension of the code is equal to the dimension of the polynomial space of degree at most d, which is spanned by the monomial basis $e_0, e_1, e_2, \ldots, e_d = 1, x, \ldots, x^d$, and is therefore $d+1$. Furthermore, suppose that f and g are two different polynomials, i.e. $f \neq g$. Thus, $h = f - g$ is a non-zero polynomial of degree at most d, and therefore has at most d roots. This implies that there are at most d points in which f and g are equal, and at least $n - d$ points in which they disagree. In other words, the distance between any two different codewords of the code is at least $n - d$. $\square$
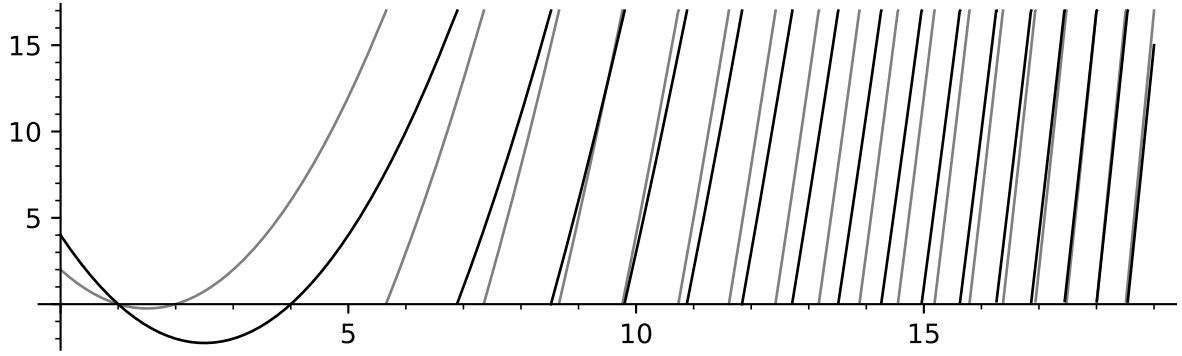
Figure 2.4: The plot $x \mapsto (x - 1)(x - 2)$ and $x \mapsto (x - 1)(x - 4)$ presents the extension of the polynomials

Now we are about to prove that the polynomial code is a testable code. The next claim will enable us to design a random test to certify polynomials.

Claim 2.8.1. *Consider the finte field $\mathbb{F}_q$ for prime q, denote by w the d root of 1, namely $w^d = 1$ then for any polynomial f with degree at most d it holds:*

$$f(x) = \frac{1}{d} \sum_t^d f(x + w^t y) \quad \forall x, y \in \mathbb{F}_q \tag{2.1}$$

Proof. Denote by a_j the jth coffecicient of f. And recall that for any fiexd $l \in \mathbb{F}_q$ the summtiom $\sum_t^d w^{t \cdot l}$ equals:

$$\sum_t^d w^{t \cdot l} = \begin{cases} \frac{w^{l \cdot d} - 1}{w^l - 1} = 0 & l \neq 0 \\ d & l = 0 \end{cases}$$

Thus:

$$\sum_t f\left(x + w^t y\right) = \sum_{t,j} a_j \left(x + w^t y\right)^j = \sum_{t,j} a_j \left(\sum_l x^{j-l} w^{t \cdot l} y^l \binom{j}{l}\right)$$

$$= \sum_j a_j \sum_l x^{j-l} y^l \binom{j}{l} \sum_t w^{t \cdot l}$$

$$= \sum_j a_j x^j \cdot d = f(x) \cdot d$$

□

Example 2.8.1. *For example consider a linear function over $\mathbb{F}_q$, then it $w^2 = 1 \Rightarrow w = -1$ and it clears that $f(x) = \frac{1}{2}(f(x - y) + f(x + y))$.*

Hence, we can consider the following tester: randomly select $x, y \in \mathbb{F}_q$ and accept if eq. (2.1) is satisfied for a given function f.

Claim 2.8.2. *Suppose that $f : \mathbb{F}_q \to \mathbb{F}_q$ satisfies eq. (2.1) than, ethier f is polyonomial at degree at most d or that f is the zero function.*

Proof. Assume by contradiction that f has more then $d+1$ zeros (w.l.g). Denote them by $x_1, x_2..x_{d+1}$. We will show that for any $x \in \mathbb{F}_q$ one can find $y \in \mathbb{F}_q$ such that $x + w^t y = x_t$. Those equations compose a linear diophantine system which can reduced to the following system:

$$\begin{bmatrix} w^0 & 0 & \ldots & 0 & 0 \\ w^1 & 1 & \ldots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ w^{d-1} & 0 & \ldots & 1 & 0 \\ w^d & 0 & \ldots & 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ k_2 \\ k_3 \\ \vdots \\ k_d \end{bmatrix} = \begin{bmatrix} x_1 - x \\ x_2 - x \\ \vdots \\ x_d - x \\ x_{d+1} - x \end{bmatrix}$$

The determinant of the above matrix is $w \neq 0$ and therefore there exists a y satisfies the equations. Hence for any x we have $f(x) = \frac{1}{d}(f(x_1) + f(x_2) + .. + f(x_{d+1})) = 0$. □

Consider the following decoding process. Given a candidate $f : \mathbb{F}_q \to \mathbb{F}_q$, we will set for any point $x \in \mathbb{F}_q$ the plurality of eq. (2.1) over the $y \in \mathbb{F}_q$. Denote the output by g, and we have that:

$$g(x) = \arg_{z \in \mathbb{F}_q} \max \mathbf{Pr}_y \left[\sum_i^d f(x + w^i y) = z \right]$$

Claim 2.8.3. *If f pass the test with probability $\geq 1 - \varepsilon$ then the relative distance between f and g is at most 2ε.*

Proof. Notice that for any x for which there exists z such $\mathbf{Pr}_y \left[\sum_i^d f(x + w^i y) = z \right] \geq \frac{1}{2}$ it must hold that $g(x) = f(x)$. Denote by A that set and by α the probability to hit such x (namely $|A|/|\mathbb{F}_q|$). The probability to pass the test can be bounded by:

$$1 - \varepsilon \leq \mathbf{Pr}[f \text{ pass}] = \mathbf{Pr}[f \text{ pass} \cap x \in A] + \mathbf{Pr}[f \text{ pass} \cap x \notin A]$$
$$\leq 1 \cdot \alpha + \frac{1}{2} \cdot (1 - \alpha)$$
$$\Rightarrow \alpha \geq 1 - 2\varepsilon$$

Thus, at most 2ε of the points do not agree with g. □

Claim 2.8.4. *Let f be a function that pass the test with probability $1 - \varepsilon$ and fix $x \in \mathbb{F}$ then:*

$$\mathbf{Pr}_{u,v} \left[\sum_i^d f(x + w^i u) = \sum_i^d f(x + w^i v) \right] \geq 1 - 2\varepsilon \cdot d$$

Proof. Recall that for any i the random variabile $x + w^i u$ is distributed uniformaly random regardless x. Therefore we have that for any i the following equation holds with probability at least $1 - \varepsilon$:

$$f\left(x + w^i u\right) = \sum_{j}^{d} f\left(x + w^i u + w^j v\right)$$

Thus, by the union bound, those equastions satisfied simulatnusly for all the i with probability $1 - d \cdot \varepsilon$. By reappting on the same arguments but swapping u and v we have that

$$f\left(x + w^i v\right) = \sum_{j}^{d} f\left(x + w^i v + w^j u\right)$$

also happens with probability at least $1 - d \cdot \varepsilon$ and therefore, again by the union bound, both of the equations sets hold with probability at least $1 - 2d \cdot \varepsilon$. But that is exactly the event in which:

$$\sum_{i} f\left(x + w^i u\right) = \sum_{i}^{d}\sum_{j}^{d} f\left(x + w^i u + w^j v\right) = \sum_{j} f\left(x + w^j v\right)$$

$\square$

Claim 2.8.5. *For any x the probability over v that $g\left(x\right) = \sum_i f(x + w^i v)$ is grater than $1 - \Theta\left(d \cdot \varepsilon\right)$.*

Proof. Fix arbitrary x and denote by α the probability to hit v such that $g(x) = \sum_i^d f\left(x + w^i v\right)$ and by A the set of v's satisfy the relation. Notice that the probabiliy to pick a pair v, u such that $\sum_i^d f\left(x + w^i v\right) = \sum_i^d f\left(x + w^i u\right)$ is lower than the prbability that either v and u are both belong to A or they both belong to his complentarty. Namely: $\alpha^2 + (1 - \alpha)^2 \geq 1 - 2\varepsilon \cdot d$. Hence we get $\alpha \geq \frac{1}{2}\left(1 + \sqrt{1 - 4\varepsilon d}\right) = 1 - \Theta\left(\varepsilon \cdot d\right)$. $\square$

Claim 2.8.6. *For $\varepsilon = \Theta(1/d^2)$ g is a polynomial at degree at most d.*

Proof. Fix $x, y \in \mathbb{F}_q$. From claim 2.8.5 with probability at least $1 - \Theta\left(d \cdot \varepsilon\right)$ it holds that $g\left(x + w^j y\right) = \sum_j^d f\left(x + w^i v + w^j y\right)$. Moreover, as $v + w^j y$ distributed uniformly for any j, it follows that one can fix j and has with probability at least $1 - \Theta\left(\varepsilon d\right)$ that $g\left(x\right) = \sum_i^d \left(f\left(x + w^i \left(v + w^j y\right)\right)\right)$. So, the raltion holds for all the j's with probability $1 - \Theta\left(\varepsilon d^2\right)$ combining all togeter obtains:

$$g\left(x\right) = \sum_{i,j}^{d} f\left(x + w^i \left(v + w^j y\right)\right) = \sum_{j}^{d}\sum_{i}^{d} f\left(x + w^i v + w^j y\right) = \sum_{j}^{d} g\left(x + w^j y\right)$$

With probability $1 - \Theta\left(\varepsilon \cdot d^2\right)$. However, the probability is about sampling v such those transimitions are vaild. But as the value of g is indepanded on the choice of v it's enough to have just a single v to garuntee $g\left(x\right) = \sum_j^d g\left(x + w^j y\right)$ namely that the probability is positive. So if $\varepsilon = \Theta(1/d^2)$ we have that for any x, y there exists v such section 2.8 is satisfied and therfore g is satisfies eq. (2.1) for any x, y. $\square$

Chapter 3

Quantum Error Correction Codes.

3.1   Introduction.

It is widely believed that quantum machines have a significant advantage over classical machines in a wide range of computational tasks [Gro96], [AK99]. Simple algorithms which can be interpreted as the quantum version of scanning all the options, reducing the running time by the square root of the classical magnitude. Nevertheless, Shor has demonstrated a polynomial-depth quantum circuit that solves the hidden abelian subgroup problem [Sho97], which is considered a breakthrough, as it made the computer science community believe that a quantum computer could offer an exponential advantage.

Despite a consensus on the superiority of the ideal quantum computing model, it is still uncertain whether it is possible to implement such a machine in a noisy environment. Still, simply pointing out the existence of noise is not sufficient to negate the feasibility of computation. Evidence of this is that classical computers also experience a certain rate of errors. Therefore, to fully comprehend the difficulty, let us compare two main factors that made it a challenging task.

First is the magnitude of the error rate; classical computers also have errors, and sometimes we witness system failures (e.g. the blue screen). The error rate of modern computers is so low that the probability of errors propagating stays negligible, even if the length of the computation is polynomial in the scale of what is considered a reasonable input size. It's worth mentioning that in exascale computing, when supercomputers perform around 10^{18} operations per second, it is difficult to overlook faults. In quantum computing, we become aware of their existence much earlier.

The second difference, which is a subtle point, is that quantum states are susceptible to additional types of errors. In addition to the possibility of bit-flip errors, a quantum state may also experience a change in phase. For example, consider the initial state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and suppose that due to noise the state is transformed into $\frac{1}{\sqrt{4}}(\sqrt{3}|0\rangle + |1\rangle)$. Classical circuits are oblivious to such faults, meaning that their operation would remain unchanged as no error has occurred. Quantum circuits, however, are usually affected and may fail. Furthermore, when designing a decoder for quantum error correction codes, one must ensure that the decoding process does not introduce bit-flip errors if a classical code is used to protect against phase flips.

3.2 Quantum Noise.

Definition 3.2.1 (Bit and pahse flip.)**.** *Consider a quantum state $|\psi\rangle$ encoded in the computation base. We will say that a bit-flip occurs in a scenario the operator Pauli X is applied on one of our state's qubits. The bit-flip event could be considered as exactly as the standard bit-flip error in the classical regime. Similarly, phase-flip occurs when the Pauli Z is applied on one of the qubits.*

However, even though quantum noise is so violent, it has been proven that any ideal circuit of polynomial depth can be transformed into a robust circuit at poly-logarithmic cost [AB99]. In other words, there is a threshold: if physicists provide qubits and a finite gate set with a noise rate below that threshold, then BQP, the class of polynomial-time ideal quantum computation, is feasible and can be computed on a realistic machine.

The basic idea in [AB99] was to show the existence of quantum error correction codes, which would enable logic operations to be performed in a way that prevents errors from propagating. This process involves separating the operation into two stages: the operation itself and an error correction stage. This comes with an additional cost in terms of space and time, but it can reduce the probability of the final state being faulty. The trade-off between the resources needed and the rate of decrease defines the threshold. If the balance is positive, then the process can be repeated in a recursive manner, and after log-log iterations, the failure probability will decay to zero. At the same time, the circuit will scale to a maximum of poly-logarithmic width and depth.

Let's return to the repetition code presented in Chapter 2. We would like to have an analog; a first and natural attempt might consider duplicating copies of the state. Unfortunately, copying a general state is not a linear operation and therefore cannot be done in the circuit model (or any other believed to be feasible). In particular, there is no circuit U which can simultaneously duplicate the states $|0\rangle, |1\rangle, |+\rangle, |-\rangle$.

To overcome the issue, Shor came up with the nine-qubit code [Sho95], which at first glance might seem a naive straightforward implementation of "duplication", but instead uses a clever insight about quantumness in general. Any operation can be seen as a linear (and even unitary) operation over a subspace embedded in a large enough dimension. The encoding is given as follows:

$$|\overline{0}\rangle = \frac{1}{2\sqrt{2}} \left(|000\rangle + |111\rangle\right)^{\otimes 3}$$

$$|\overline{1}\rangle = \frac{1}{2\sqrt{2}} \left(|000\rangle - |111\rangle\right)^{\otimes 3} .$$

For convenience, let us use the notation $|\mathbf{GHZ}^{\pm}\rangle = |0^m\rangle \pm |1^m\rangle$. We can also consider the Shor code over m^2 qubits, which is defined as above, such that any logical state contains m products over m qubits. Therefore, the state $|\overline{0}\rangle$ over m^2 qubits can be written as $|\mathbf{GHZ}^+\rangle^m$. We are now ready to prove a statement regarding the robustness.

Lemma 3.2.1. *The Shor code over 9 qubits enable to correct a single either bit or phase flip.*

It is evident that a single bit-flip error can be handled in the same way as in the conventional case. The decoder will check if any of the triples have the same value, and if not, it will correct it by majority. To create a decoder that can also correct a phase-flip error, we need the following statement. In this chapter, we denote the Hadamard gate over m qubits as H^m.

Claim 3.2.1. $H^m |\mathbf{GHZ}^{\pm}\rangle = \sum_{x \cdot \mathbf{1} =_2 \pm} |x\rangle$

Proof.

$$H^m \left|\mathbf{GHZ}^{\pm}\right\rangle = H^m \left|0^m\right\rangle \pm H^m \left|1^m\right\rangle = \sum_{x \in \mathbb{F}_2^m} |x\rangle \pm \sum_{x \in \mathbb{F}_2^m} (-1)^{x \cdot \mathbf{1}} |x\rangle$$

$$= \sum_{x \in \mathbb{F}_2^m} \left(1 \pm (-1)^{x \cdot \mathbf{1}}\right) |x\rangle = \sum_{x \cdot \mathbf{1} =_2 \pm} |x\rangle$$

$\square$

Now it is clear how to correct a phase flip. One can apply the Hadamard transform and compute the parity of each triple. By the assumption that only a single phase flip may occur, either all the triples have the same parity or the faulted one has an opposite parity and needs to be corrected. Thus, we obtain an $[[9, 1, 3]]$ quantum error correction code. Asymptotically, this is an $[[m^2, 1, m]]$ code.

3.3 CSS Codes.

The Shor code is a specific case of the more general CSS (Calderbank-Shor-Steane) code [CS96]. A family composed by two binary codes C_X, C_Z such that $C_Z^{\perp} \subset C_X$.

Definition 3.3.1 (CSS Code). *Let C_X, C_Z classical linear codes such that $C_Z^{\perp} \subset C_X$ define the $Q(C_X, C_Z)$ to be all the code words with following structure:*

$$|\mathbf{x}\rangle := |x + C_Z^{\perp}\rangle = \frac{1}{\sqrt{C_Z^{\perp}}} \sum_{z \in C_Z^{\perp}} |x + z\rangle$$

Clearly, the codewords are all the codewords in C_X which don't belong to $C_Z^{\perp}$ and therefore the dimension of the quantum code is $\dim Q(C_X, C_Z) = \dim C_X - \dim C_{Z^{\perp}} = \dim C_X + \dim C_Z - n$. Yet, it's not stems immediately how one can correct faults. Next, we are going to repeat the decoding process of the Shor code in the general setting of CSS codes.

Lemma 3.3.1. *Let C_X, C_Z classical codes such $Q(C_X, C_Z)$ is a CSS code. Let d_X be the minimal weigh of codeword in C_X which is not in $C_Z^{\perp}$, and define by the same way d_Z to be the minimal weight of codeword in C_Z which doesn't belong to $C_X^{\perp}$. Then the distance of $Q(C_X, C_Z)$ equals to $\min d_X, d_Z$. Moreover there is a decoder which correct any fault with weight at most $d/2$.*

Proof. First let us prove the following claim:

Claim 3.3.1. *Denote by $H^{\otimes n}$ the Hadamard gate over n qubits. Then for any code C it holds that:* $H^n \left|C^{\perp}\right\rangle = |C\rangle$

Proof.

$$H^n \left|C^{\perp}\right\rangle = \frac{1}{\sqrt{|C_Z^{\perp}|}} \sum_{z \in C_Z^{\perp}} H^n |z\rangle = \frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{|C_Z^{\perp}|}} \sum_{z \in C_Z^{\perp}} \sum_{y \in \mathbb{F}_2^n} (-1)^{\langle z, y \rangle} |y\rangle$$

$$= \frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{|C_Z^{\perp}|}} \sum_{z \in C_Z^{\perp}} \left(\sum_{y \in C_Z} |y\rangle + \text{ other terms}\right)$$

(3.1)

Since the columns of matrix H_Z form a basis for the complementary space $C_Z^\perp$, and due to the dimensional theorem and the equivalence between the row rank and column rank of a matrix, we can deduce that $\dim \operatorname{rank} H_Z^\top + \dim \ker H_Z = n$, which implies that $|C_Z^\perp||C_Z| = 2^n$. Thus the norm of

$$\frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} \sum_{y \in C_Z} |y\rangle = \sqrt{\frac{|C_Z^\perp|}{2^n}} \sum_{y \in C_Z} |y\rangle \tag{3.2}$$

equals 1 and the summation over the vectors $y \notin C_Z$ in the inner closure in equation 3.1 must cancel. So we left only with a uniform superposition over the codewords in C_Z, Or in other words $H^n |C^\perp\rangle = |C\rangle$. □

Claim 3.3.1 states that, when considering CSS codes, pauli X operators can be seen as pauli Z operators in the rotated frame. That it,

$$H^n X^f |C_Z^\perp\rangle = \overbrace{H^n X^f H^n}^{Z^f} H^n |C_Z^\perp\rangle = Z^f |C\rangle$$

That insight hints a description to decoder for the quantum code. If one knows how to correct errors for each of the classical code C_X, C_Z than he can start by correct the bit flips in using the decoder of C_X, rotate the state by applying the Hadamard transform and then correct, what was before the transformation phase flips and now a bit flips by using the decoder of C_Z. Then in the end applying the Hadamard transform again for backing to the initial computation space. Indeed that decoder correct $\min\{d(C_X)/2, d(C_Z)/2\}$ errors. Yet more work is needed to show that this decoder also correct $d(Q(C_X, C_Z))/2$ errors.

Let us assume the existences of decoders of the classical codes C_X and C_Z, Denoted $D_X : \mathbb{F}_2^n \to C_x$ and D_Z. In particular for any $\xi \in \mathbb{F}_2^n$, $D_X(\xi) = \arg\min_{x \in C_X} |x + \xi|$.

Claim 3.3.2. *For any $x_0 \in C_X$ and $z_1 \neq z_2 \in C_z^\perp$, D_X correct $|x + z_1 + f\rangle, |x + z_2 + f\rangle$ into two different words in C_X.*

Proof. Suppose not, namely there exists $y \in C_X$ such that D_X correct $|x + z_1 + f\rangle, |x + z_2 + f\rangle$ into $|y\rangle$. Then we have that for both $i \in \{1, 2\}$ it holds that $d(x + z_i + f, y) \leq d(C_Z^\perp/2)$ and therefore $d(x + z_1 + f, x + z_2 + f) \leq d(C_Z^\perp)$. But

$$d(x + z_1 + f, x + z_2 + f) = |x + z_1 + f + x + z_2 + f|$$
$$= |z_1 + z_2| = d(z_1, z_2)$$

contradiction for the assumption that $z_1, z_2 \in C_Z^\perp$. □

We are ready to show step by step the decoding process. Let $P = X^f Z^e$ be an error such that $e, f < d/2$ act on the state $|x\rangle$. Denote by H_X, H_Z the parity check matrices of C_X, C_Z. Using the commute relation $[X^f, Z^e] = (-1)^{\langle e, f \rangle}$ we have that:

$$|\mathbf{x}\rangle \mapsto^P \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^f Z^e |x + z\rangle = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} (-1)^{\langle e, f \rangle} Z^e |x + z + f\rangle$$

Now the decoder computes and stores the syndrome relative to the bits code using the parity check matrix H_X. And apply the inverse gate.

$$\mapsto^{H_X} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} (-1)^{\langle e,f \rangle} Z^e |x+z+f\rangle |H_X(x+z+f)\rangle$$

$$= \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} (-1)^{\langle e,f \rangle} Z^e |x+z+f\rangle |H_X f\rangle$$

$$\mapsto^{X^f} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} Z^e |x+z\rangle |H_X f\rangle$$

Then rotating into the phases base:

$$\mapsto^{H^{\otimes n}} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^e H^{\otimes n} |x+z\rangle$$

$$= \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^e H^{\otimes n} X^x |z\rangle = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^e Z^x H^{\otimes n} |z\rangle$$

$$= \frac{1}{\sqrt{|C_Z^\perp|}} X^e Z^x H^{\otimes n} \sum_{z \in C_Z^\perp} |z\rangle = \frac{1}{\sqrt{|C_Z|}} X^e Z^x \sum_{z \in C_Z} |z\rangle$$

$$= \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x,e \rangle} Z^x \sum_{z \in C_Z} X^e |z\rangle = \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x,e \rangle} Z^x \sum_{z \in C_Z} |z+e\rangle$$

So now we have back into the begging. Only now the phase flips are playing the role of bit flips relative to the code C_Z.

$$= \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x,e \rangle} Z^x \sum_{z \in C_Z} |z+e\rangle |H_Z e\rangle$$

$$\mapsto \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x,e \rangle} Z^x \sum_{z \in C_Z} |z\rangle |H_Z e\rangle \mapsto^{H^n} \frac{1}{\sqrt{|C_Z^\perp|}} X^x \sum_{z \in C_Z^\perp} |z\rangle$$

$$= \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} |z+x\rangle = |\mathbf{x}\rangle$$

$\square$

There is still one big difference between the classic repetition code and the Shor code. While each parity check of the Shor code examines a square root number of qubits, any check of the repetition code touches no more than a constant number of qubits; that is, any check just tests if any two adjacent bits are equal. That brings us to ask whether the Shors code is really the quantum analogy for the repetition code?

For getting an hint before formally presenting a quantum LDPC code, let's take another look on the general structure of the CSS codes. The decoding procedure the proof above teach us an additional point about CSS code, the task of finding a good code quantum code, could be reduce for finding a two classic binary linear codes which their parity check matrices ortogonal to each

other. Furthermore, if one is willing to has an qLDPC code, then H_X and H_Z can't be parity check matrices of good classical code as any column of $H_z^\top$ is a codeword of C_X.

$$C_Z^\perp \subset C_X \Rightarrow H_X H_Z^\top = 0$$

And by being an LDPC code, the rows wights of H_Z is bounded by constant. Therefore there is a codeword $\in C_X$ which is also a row of H_Z that has a constant weight.

3.4 qLDPC Codes.

As exactly as in the classic case, qLDPC codes are codes in which any check act non trivially on at most a constant number of qubits, It was proved that using a good Quantum LDPC code one can achieve a fault tolerance threshold theorem at the cost of only constant overhead[1] [Got14]. We are now about to embark on a detailed review of the first quantum LDPC code [Den+02].

Recall that one way to present a code is by define the parity check matrix, Consider the $l \times l$ Tours, namely the Cayley graph of the group product $\mathbb{Z}_l \times \mathbb{Z}_l$. Associate any coordinate (bit/qubit) with an edge on the Tours. And consider the following two restrictions:

1. Each vertex requires form its local view, the bits lay on its supported edges, To has an even party. We will refer to this type of check as *cross check*.

2. Similarly, each face requires the same from its supported edges, but computes the parity in a different (specific) base. That it, the face first rotates the qubits by applying the Hadamard transform on them, and then computes their XOR. Finally, the qubits are rotated back to the computation base. We shall refer to this type of check as *face check*.

Figure 3.1: On the left is the Toric Graph. On the right are cross and face checks.

---

[1]under the assumption of having an efficient decoder.

Claim 3.4.1. *The $l \times l$ Toric code is a CSS code, with dimension 2 and distance $\Theta(l)$.*

Proof. Consider a pair of cross and a face checks. If they are not intersect (share edges) then, obviously, they commute. So suppose they share an edge. Now there are a finite number of cases (4) in which a cross check can intersect a face, and for all of them we have they must to intersect in an exactly two edges. Therefore the crosses checks commute with all the faces checks.

Now, denote by C_X and C_Z the linear codes defined by the crosses and faces checks. Observes that C_X contain all the subgraph of the tours which have only evens degree, namely all the loops. The following claim will be used to show that all the low weight loops (squeezable loops) are in $C_Z^{\perp}$.

Claim 3.4.2. *Let c be an assignment of ones on a unite square, namely a closed loop at length 4, then $c \in C_Z^{\perp}$.*

Proof. Denote by c' a codeword of C_Z, therefore the parity sum induced by c' on any square equals zero, Particularly the induced parity on the square supporting c is zero. But that parity is exactly $c \cdot c'$. As it true for any $c' \in C_Z$ we obtain that $c \in C_Z^{\perp}$. $\square$

Claim 3.4.3 (Veblen's theorem). *The set of even subgraph is a linear space spanned by simple cycles (vertices degree equals exactly 2).*

Proof. Let $P \subset E$ be an even subgraph then it must to have a simple cycle denote it by P', now notice that P/P' is also an even subgraph. To see that consider a vertex v, As P' is simple cycle, substituting P' is either not effect the degree of v in P/P' or it decrees v degree by exactly 2 so $d_{P/P'}(v) = d_P(v) - 2$. In both cases $d_{P/P'}(v)$ is even. By repeating recursively until $P/P' = \{\emptyset\}$ we get a decomposition of P into a sum of simple cycles. $\square$

Claim 3.4.4. *Associate with any vertex of the Torus a coordinate in $\mathbb{Z}_l \times \mathbb{Z}_l$. Consider a simple cycle P subset of the Torus. Denote P by the vertices composing it $v_0 v_1 .. v_k$ arranged in order. Consider a vertex $v_i \in P$ such that $\{v_{i-1}, v_i\}, \{v_i, v_{i+1}\}$ are in the same direction and denote $v = (x, y) \in \mathbb{Z}_l \times \mathbb{Z}_l$. Then there exists a vertex $u \in P$, $u \neq v_{i-1} v_{i+1}$ that shares one of the coordinates of v. Put differently, there exists z such that either $u = (z, y)$ or $u = (x, z)$.*
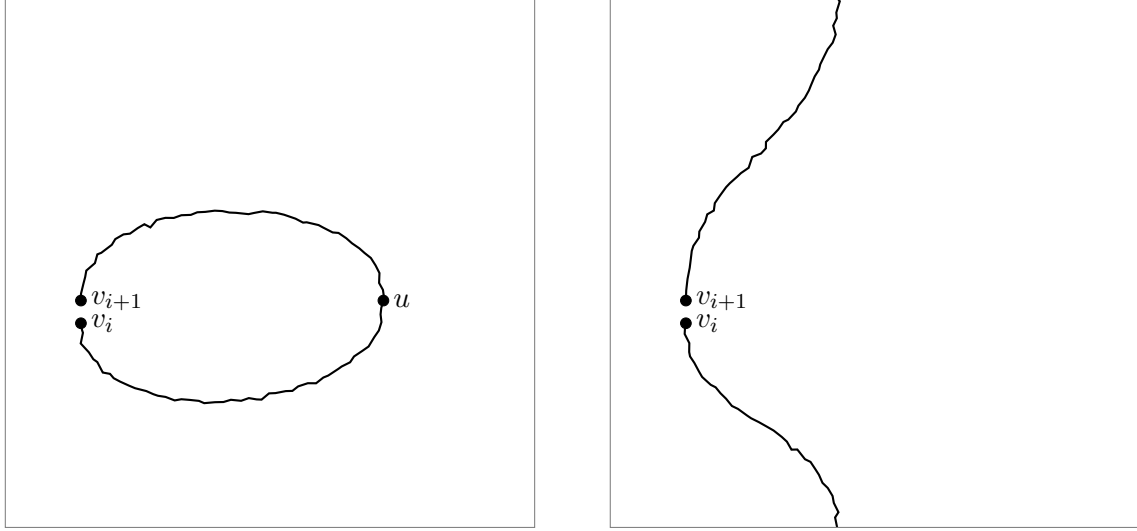
Proof. Assume without loss of generality that $v_{i-1} = (x, y-1)$ and $v_{i+1} = (x, y+1)$. Now denote by f the projection on the second coordinate, $f(a, b) = b$. and observe that for any j the distance $f(v_{j+1}) - (v_j)$ satisfies:

$$f(v_{j+1}) - f(v_j) = \begin{cases} \pm(l-1) & v_j = (\cdot, l-1), v_{j+1} = (\cdot, 0) \\ \in \{\pm 1, 0\} & \text{else} \end{cases}$$

So if there is no $u \neq v_i$ such that $f(u) = f(v_i) = f(v_{i-1}) + 1$ then there must to be an edge $\{v_j, v_{j+1}\}$, $v_j = (\cdot, l-1)$, $v_{j+1} = (\cdot, 0)$ such P pass trough it. So

$$f(v_{i-1}) \leq f(v+1) + |f(v+2) - f(v+1)| + .. + -(l-1)$$
$$\leq f(v+1) - |P| - (l-1)$$

But $|P| \leq l$. $\square$

Definition 3.4.1. *Horizontal and Vertical diameters. Let P be defined again as exactly as in claim 3.4.4. We will say that the horizontal diameter of P is:*

$$\max_{v,u\in P} \min\left\{|v_x - u_x|, |v_x + l + u_x|\right\}$$

Similarly, we define the vertical diametr to be:

$$\max_{v,u\in P} \min\left\{|v_y - u_y|, |v_y + l + u_y|\right\}$$

Claim 3.4.5. *If P is a non empty simple cycle with vertical and horizontal diameters d_1, d_2, then it can either be a square or can be decomposed into two simple cycles P_1, P_2 with either the vertical diameter of both of them being strictly less than d_1 and their horizontal diameters being at most d_2, or the horizontal diameter of both of them being strictly less than d_2 and their vertical diameters being at most d_1.*

Proof. If P is not a square, it must have either a vertical or horizontal diameter greater than 1. We will assume, without loss of generality, that the vertical diameter is greater than 1. Let $v_0 v_1 .. v_k$ be the vertices that compose P, and let v_i, v_j be the vertices at maximal distance. We will pick the first vertex $v_{i'}$ from the left of v_i such that $v_{i',y} \neq v_{i,y}$. By claim 3.4.4, there must be at least one vertex v_l in P such that $v_{i,y} = v_{l,y}$ and is the closest in horizontal distance to v_i. We will denote by $P^{\uparrow}$ the path $v_i, v_{i+1}..v_{l-1}, v_l$ and by $P^{\downarrow}$ the path $v_l, v_{l+1}..v_{i-1}, v_i$. We will also denote by L the straight line $v_i, (v_{i,x} + 1, v_{i,y}), (v_{i,x} + 2, v_{i,y}), ..v_l$.

Observes that the vertical diameter will be lower by exactly 1 than the vertical diameter of P, and the horizontal distance between any point on L to other point on P will be lower than the horizontal distance of between the end of L (namely $\{v_i, v_l\}$), to the same point. So if $P^{\uparrow}\cup L, P^{\downarrow}\cup L$ are simple cycles then we get the desire. Suppose that $P^{\uparrow}\cup L$ is not a simple cycle. This means that the path $v_i, v_{i+1}..v_{l-1}, v_l.. (v_{i,x} + 2, v_{i,y}), (v_{i,x} + 1, v_{i,y}) v_i$ must contain an inner loop. This loop cannot be supported on $P^{\uparrow}$ alone, because $P^{\uparrow} \subset P$ so the inner loop would have to be contained also in P, which is a contradiction. Therefore, the inner loop is also supported on L, so it follows that there is a vertex in L with degree > 2 in $P^{\uparrow}\cup L$, namely a vetex $u \neq v_i, v_l$ such that $u_y = v_{i,y}$, and also $u \in P$. But by the construction of L, $|u_x - v_{i,x}| < |v_{l,x} - v_{i,x}|$, which is a contradiction to the fact that v_l was chosen to minimize the horizontal distance. Thus, we have our claim. $\qquad\square$

We have now established that $c \in C_X$ with $|c| < l$ can be decomposed into simple cycles, each of which has a weight less than l. Applying claim 3.4.5 recursively, we can further decompose each of these cycles into a sum of unit squares. However, claim 3.4.2 states that unit squares are in $C_Z^\perp$, so $c \in C^\perp$.

$\square$

3.5   Quantum Expander Codes.

As similar to the classical case, the next natural question to ask is whether there are codes with positive rates. The quantum expanders were the first quantum LDPC codes to achieve a square-root distance and positive rate [TZ14; LTZ15]. The leading insight was the idea that the Toric code could be represented as a variant product of the repetition code. For example, consider the cross restriction in Figure fig. 3.1; that restriction can be obtained by gluing two vertices of two different cycle graphs.

Definition 3.5.1. *For any two matrices A, B, with the same number of rows, denote by $[A, B]$ the matrix obtained by attach B next to A from right. Let $H_1, H_2 \in \mathbb{F}_2^{n \times r}$ be the parity check matrices. Define the bit and the phase parity checks matrices to be:*

$$H_X = \begin{bmatrix} H_1 \otimes I_r & I_n \otimes H_2^\top \end{bmatrix}$$
$$H_Z = \begin{bmatrix} I_r \otimes H_2 & H_1^\top \otimes I_n \end{bmatrix}$$

The matrices are orthogonal to each other as $H_X H_Z^\top = H_1 \otimes H_2^\top + H_1 \otimes H_2^\top = 0$ and therefore the pair define a valid CSS code. We will call to that code the Hyperproduct and denote it by $Q(H_1 \times H_2)$.

Obliviously, if $H_1, H_2, H_1^\top, H_2^\top$ are parity checks matrices of an LDPC codes, so are H_X, H_Z as their maximal row weight is at most two times larger.

Example 3.5.1. *The Toric code could be thought as the Hyperproduct of the repetition code with himself. The parity check matrices of the codes are given follow. The left 3×3 matrix corresponds to the repetition code while the right 18×9 corresponds to the vertices check of the Toric code.*

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Claim 3.5.1. *Let $A, B \in \mathbb{M}_{n \times r_1}, \mathbb{M}_{n \times r_2}$ then $\dim \ker[A, B]$ is $\dim \ker A + n$.*

Proof. The proof is omitted.

$\square$

Claim 3.5.2. *Let k_1, k_2 be the demission of the codes with the full rank parity check matrices H_1, H_2. Then the dimension of the Hyperproduct code is $\geq k_1 k_2$.*

Proof. We will find the dimensions of each of the classical codes defined by $\ker H_X$ and $\ker H_Z$. Notice that the length of the $H \otimes I_n$ equals $n \times n = n^2$, And assuming the fullness of the ranks, the length of $I_{r_1} \otimes H_2^\top$ is $r_1 \cdot r_2$. Thus the length of $\ker H_X$ is $n^2 + r_1 r_2$ Now, recall that for any matrix A it holds that $\dim \ker (A \otimes I_l) = l \cdot \dim \ker A$. Therefore using claim 3.5.1 we obtain that the dimension of $\ker H_X$ is $k_1 n + r_1 r_2$

By the same arguments we have that $\dim C_Z = k_2 n + r_1 r_2$ Thus the dimension of the quantum code is:

$$\dim Q\,(C_X, C_Z) = \dim C_X + \dim C_Z - \left(n^2 + (n - k_1)(n - k_2)\right)$$
$$= (k_1 + k_2)\,n + 2(n - k_1)(n - k_2) - \left(n^2 + (n - k_1) \cdot (n - k_2)\right)$$
$$= k_1 k_2$$

$\square$

Remark 3.5.1. *Let H_1' be a parity check matrix obtained by puncturing columns from H_1, denote by k_1' the dimension of that code. Then the Hyperproduct $Q(H_1' \times H_2)$ is a CSS code with dimension $k_1' k_2$. Moreover if the number of columns left after the puncturing is less the distance of $\ker H_1$ then it must to holds that $k_1' = 0 \Rightarrow \dim Q(H_1' \times H_2) = 0$. Otherwise, one can take a non trivial codeword of $\ker H_1'$ and extending it to a valid codeword of $\ker H_1$ by set any punctured coordinate of it to zero. The yielded codeword has weight less than d which is contradiction.*

Claim 3.5.3. *Denote by d the minimal distance of $\ker H_1$. Any codeword x of $C_X = \ker H_X$ with edge at most d belongs to $C_Z^\perp$.*

Proof. Define by H_1' the matrix which obtained by puncturing from H_1 the columns associated with the coordinates e_i such that subspace corresponding to $e_i \otimes I_n$ doesn't support x. Denote the by S the set of the reaming coordinates. For example if $H_1 \otimes I_n = I_2 \otimes I_2$ and $x = [1, 0, 0, 0]$ then H_1' is the unit matrix $[1, 0]^\top$ obtained by puncturing the second column of $H_1 = I_2$, and $S = \{e_1\}$.

As $|x| < d$ we have that $|S| < d$, Namely H_1' supported on less than d coordinates and therefore $\dim Q\,(H_1' \times H_2) = k_1' k_2 = 0$. Thus, by the fact that for any CSS code $\dim C = \dim C_X - \dim C_Z^\perp$ it follows that $\dim \ker H_X' = \dim \ker H_Z'^\perp \Rightarrow \ker H_X' = \ker H_Z'^\perp$. Denote by x' the restriction of x to the columns of H_X' and clearly, by the definition of the construction, x' is a codeword of $\ker H_X'$. Thus x' is also codeword of $\ker H_Z'^\perp$ and by the same argument, x is also a codeword of $\ker H_Z'^\perp$. $\square$

Immediately from claim 3.5.3 we obtain that existences of quantum LDPC codes with positive rate and $\Theta\,(\sqrt{n})$ distance by taking the Hyperproduct of two classical expender codes.

Theorem 3.5.1. *There exists an infinity family of QLDPC codes with positive rate and $\Theta(\sqrt{n})$ distance.*

Chapter 4

Good qLDPC Codes and LTC.

The existence of good quantum LDPC codes and locally testable codes (LTCs) was considered an open problem for roughly two decades. Although they seemed to be related only by containing the word "code" in their names, they were proven to exist by the same construction. They first appeared in [Din+22] as good locally testable codes and not long after that in [PK21], in which they also extended and derived the result to obtain the quantum code. We emphasize that even though they developed the same codes, their proofs are not similar at all. Here, we follow the [LZ22] work, which simplifies the original proof and does not rely on any concept more complicated than what we have already seen in the previous chapters. They also coined the term "Quantum Tanner Codes" referring to the fact that C_X and C_Z are classical Tanner codes. Yet, the proof we present is not exactly the same, as we use a small code that requires satisfying a stronger assumption (the w-robustness property 4.1.1) relative to the original work. The reason why they had to use a weaker assumption is because the existence of codes satisfying the stronger one was proven a year later [KP22]. Relying on the stronger assumption allows us to simplify the proof even more and get rid of another requirement that the small code has to satisfy (The p-resistance to puncturing 4.1.2).

4.1   Quantum Tanner Codes.

Recall our insight that for a pair of LDPC codes to define a good CSS code, they must both be poor codes in the sense that they must have a constant distance. Therefore, we understand that any codeword in C_X with small weight belongs to $C_Z^\perp$. To prove this, we will construct a proof such that if $x \in C_X$ and $|x|$ is small, then there is a small codeword $z \in C_Z^\perp$ such that $|x + z| < |x|$; by repeating this process recursively, it follows that $x \in C_Z^\perp$. To formulate this theorem, we will need to define more definitions.

The next two definitions are concerned with the properties of the small code that will be set on the edges. Using them, one can characterize cases in which a local view can be reduced by subtracting a codeword from the dual code.

Definition 4.1.1 (*w-Robustness*). *Let C_A and C_B be codes of length Δ with minimum distance $\delta_0\Delta$. $C = \left(C_A^\perp \otimes C_B^\perp\right)^\perp$ will be said to be w-robust if for any codeword $c \in C$ of weight less than w, it follows that c can be decomposed into a sum of $c = t + s$ such that $t \in C_A \otimes \mathbb{F}^B$ and $s \in \mathbb{F}^A \otimes C_B$, where s and t are each supported on at most $\frac{w}{\delta_0\Delta}$ rows and columns. For convenience, we will denote by B' (A') the rows (columns) supporting t (s) and use the notation $t \in C_A \otimes \mathbb{F}^{B'}$.*
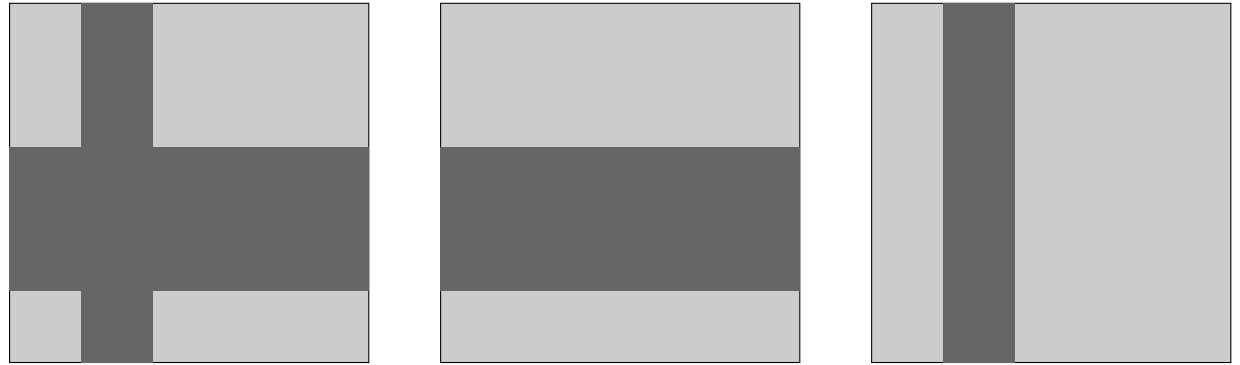
$$c \in C_A \otimes C_B \quad = \quad t \in C_A \otimes \mathbb{F}^B \quad + \quad s \in \mathbb{F}^A \otimes C_B$$

Figure 4.1: w-Robustness, Any low-weight codeword of the dual tensor code c can be decomposed into a sum $t + s$, where t is a collection of rows, each of which is a codeword in C_A, and similarly s is a collection of columns, each of which is a codeword of C_B.

The definition we gave for w-Robustness is identical to the one stated by Zemor and Leverrier, but we also included the decomposition property in the definition. We refer readers to the appendix section in [LZ22] for an existence proof of w-robustness codes for $w = \Delta^{3/2-\varepsilon}, \varepsilon > 0$, through random construction. We note that the random construction also yields the Gilbert-Varshamov bound. Though, we need a w-robustness codes for $\Delta^{3/2+\varepsilon}$ where ε is positive. For achiving that we use the theorem proven in [KP22]:

Theorem 4.1.1. *Fix $\rho_A, \rho_B \in (0,1)$, and let κ be:*

$$\kappa = \frac{1}{2} \min \left(\frac{1}{4} H_2^{-1} \left(\frac{\rho_A}{8} \right) H_2^{-1} \left(\frac{\rho_B}{8} \right), H_2^{-1} \left(\frac{\rho_A \rho_B}{8} \right) \right)$$

where H_2^{-1} denotes the inverse of the binary entropy function. Let C_A, C_B be Δ-length and $\rho_A \Delta, \rho_B \Delta$ codimension codes sampled uniformly at random. Then, with high probability as $\Delta \to \infty$, for any codeword of their dual tensor code $c \in \left(C_A^\perp \otimes C_B^\perp \right)^\perp$, there exists a decomposition of c into a sum of $c = t + s$ such that $t \in C_A \otimes \mathbb{F}^B$ and $s \in \mathbb{F}^A \otimes C_B$, where s and t are each supported on at most $\frac{c}{\kappa \Delta}$ rows and columns. We call such codes κ product expanding.

Note that the fact that sampling succeeds with high probability implies that, with high probability, the codes that are sampled have a good distance, as well as their duals. By denoting $\delta \leftarrow \min\{\kappa, \delta\}$, we can say that, for any rate ρ and large enough Δ, there exists $\delta > 0$ such that $\left(C_A^\perp \otimes C_B^\perp \right)^\perp$ is $\Delta^{3/2+\varepsilon}$-robust for $\varepsilon < \frac{1}{2}$ and $C_A, C_B, C_A^\perp, C_B^\perp$ have rate and relative distance of at least ρ and δ, respectively.

Definition 4.1.2 (p-Resistance to Puncturing.)**.** *Let p, w be integers. We will say that the dual tensor code $C_A \otimes \mathbb{F} + \mathbb{F} \otimes C_B$ is w-robust with p-resistance to puncturing, if the code obtained by removing (puncturing) a subset of at most p rows and columns is w-robust.*

Our proof does not utilize p-resistance to puncturing, yet it is a fundamental component in [LZ22]. Therefore, we will later indicate where and how precisely the p-resistance is being used. Now, we will define exactly what the code is.

Definition 4.1.3 (Quantum Tanner Code.). *Let Γ be a group at size n. And let A, B be a two generator set of Γ such that if $a \in A$ (B) then also $a^{-1} \in A$ (B^{-1}) and that for any $g \in \Gamma, a \in A, b \in B$ it holds that $g \neq agb$. Define the left-right Cayley complex to be the graph $G = (\Gamma, E)$ obtained by taking the union of the two Cayley graphs generated by A and B. So the vertices pair u, v are set on a square diagonal only if there are $a \in A$ and $b \in B$ such that $u = avb$. We can assume that G is a bipartite graph (otherwise just take $\Gamma' = \Gamma \times \mathbb{Z}_2$ and define the product to be $a(u, \pm) = (au, \mp)$).*

Now divide the graph into positive and negative vertices according to their coloring V_- and V_+. And define the positive graph to be $G^+ = (V_+, E)$ and by $G^- = (V_-, E)$ the negative graph, where E denotes the squares, put differently there is an edge between v and u in G^+ if both vertices are positive and they are laid on the ends of a square's diagonal.

The quantum Tanner code is a CSS code, such that C_X is defined to be the classical Tanner code $\mathcal{T}\left(G^+, \left(C_A^\perp \otimes C_B^\perp\right)^\perp\right)$ and C_Z is defined as $\mathcal{T}\left(G^-, (C_A \otimes C_B)^\perp\right)$. Note that in contrast to the classical Tanner code, in the quantum case it will be more convenient to think of codewords as assignments set on the squares and not on the edges.

Note that any check of C_X can be thought of as multiplying one of the matrices in $C_A^\perp \otimes C_B^\perp$ by the local view of some positive vertex. Similarly, the checks of C_Z are obtained by multiplying the matrices in $C_A \otimes C_B$ in the local view of the negative vertices. As any two sibling positive and negative vertices share either a row or a column, it is easy to see that the checks commute. Therefore, by definition, $H_X H_Z^\top = 0$, which implies that the code is a CSS code.

Now we can state formally the theorem:

Theorem 4.1.2. *For $\varepsilon \in \left(0, \frac{1}{2}\right)$, $\delta_0 > 0$, large enough Δ, and small codes C_A, C_B with distance at least $\delta_0 \Delta$ there exist constant $\zeta > 0$ such if the dual tensor code of C_A, C_B is $\Delta^{1\frac{1}{2}} + \varepsilon$-robust then there exists an infinite family of square complexes for which the Tanner code $\mathcal{T}\left(G^+, \left(C_A^\perp \otimes C_B^\perp\right)^\perp\right)$ defined by the complexes and the dual tensor code such that for any codeword c with weigh less than $\zeta \cdot \frac{1}{4} n \Delta^2$ there exist a negative vertex in $v \in V^-$ and code word in $y \in C_A \otimes C_B$ supported only on the squers adjoint to v such that $|c + y| < |c|$.*

Observe that $y \in C_Z^\perp \subset C_X$, so $c + y \in C_X$ and $|c + y| < |c| < \zeta \cdot n \frac{1}{4} \Delta^2$. Therefore, by theorem 4.1.2, there is another $y_1 \in C_Z^\perp$ such that $|c + y + y_1| < |c + y| < |c|$. Repeating this process enough times yields a series of $y, y_1, y_2, \ldots, y_l$, all of them in $C_Z^\perp$, such that:

$$|c + y + \cdots + y_l| = 0 \Rightarrow c = y + y_1 + y_2 + \cdots + y_l \Rightarrow c \in C_Z^\perp$$

Claim 4.1.1. *The distance of the dual tensor code is at least $\delta_0 \Delta$.*

Proof. By the robustness property, any codeword of the dual tensor code with a weight less than $\delta_0 \Delta$ is supported on at most one row. Let c be such a codeword and denote by i the number of the non-trivial row. Fix a $c' \in C_A^\perp$ such that the ith coordinate of c' is non-zero and consider the multiplication of c with the codewords of $C_A^\perp \otimes C_B^\perp$ of the following form:

$$J = \left\{c' \otimes c_b : c_b \in C_B^\perp\right\}$$

Figure 4.2: Local environment of a square complex.

So the ith row of any $x \in J$ is a codeword of $C_B^\perp$ and in total, collecting all the ith rows of codewords in J sums up to all the code words in $C_B^\perp$. On the other hand, $c \cdot x = 0$ for all $x \in J$; that is, $c \cdot x = c_i \cdot c_b = 0$. Thus we obtain that $c_i \in C_B$ and therefore $|c_i| \geq \delta_0 \Delta \Rightarrow |c| \geq \delta_0 \Delta$, which is a contradiction. $\qquad\square$

Definition 4.1.4. *Let S and S_- denote the positive and negative vertices that support the codeword $c \in C_X$, respectively. Furthermore, let S_e and S_n denote the exceptional and normal vertices, respectively, where the weight of the local view for any vertex in S_e is greater than $\Delta^{3/2+\varepsilon}$, and S_n is the complementary set of vertices. An edge in G will be said to be heavy if it supports more than $\delta_0 \Delta - \Delta^{\frac{1}{2}+\varepsilon}/\delta_0$ squares in G. Let $T \subset S_-$ denote the negative vertices connected to S_n by at least one heavy edge. Additionally, let $T_s \subset T$ denote the vertices in S_- that are surrounded by only normal vertices. Finally, for any pair of vertex subsets A, B such that $A \subset V_+$ and $B \subset V_-$, let $d_{B \to A}$ denote the average number of heavy edges leaving B and going to A.*

Claim 4.1.2. *for any $\varepsilon \in (0,1)$ and large enough Δ it holds that $|S| \leq \Delta^\varepsilon |S_-|$*

Proof. Suppose not, namely that $|S| > \Delta^\varepsilon|S_-|$, then $|x|/|S_-| > \Delta^\varepsilon|x|/|S| > \Delta^\varepsilon \cdot \delta_0\Delta$ But:

$$\frac{|x|}{|S_-|} = \frac{\Theta\left(E(S_-, S_-)\right)}{|S_-|} \leq \Theta(\Delta^2)\frac{|S_-|}{n} + \Theta(\Delta) \to_{n\to\infty} \Theta(\Delta)$$

$\square$

Claim 4.1.3. *At least $1 - \Delta^{-\frac{\varepsilon}{4}}$ portion of the negative vertices adjoin to only normal vertices.*

Proof. Suppose through contradiction that for Δ^{-p} portion of the negative vertices $v_- \in V_-$ have at least one (Δ^γ) sibling in S_e. Therefore $\Delta^{-p}|S_-| \leq \Delta|S_e|$ combining with claim 4.1.2 it follows that $|S| \leq \Delta^{1+\varepsilon+p}|S_e|$:

$$\Delta^{3/2+\varepsilon} \leq \frac{E(S, S_e)}{|S_e|} = \Theta\left(\Delta^2\right)\frac{|S|}{n} + \Theta\left(\Delta\right)\sqrt{\frac{|S|}{|S_e|}}$$
$$\leq \Theta(\Delta^2)\frac{|S|}{n} + \Theta(\Delta)\Theta\left(\Delta^{\frac{1+\varepsilon+p}{2}}\right)$$

Thus we obtain contradiction for any $p < \varepsilon/2$. In particular for $p = \varepsilon/4$ we obtain that at least $1 - \Delta^{-\varepsilon/4}$ portion of the negative vertices are surrounded by only normal vertices. $\square$

Claim 4.1.4. *Let x be a codeword of $\left(C_A^\perp \otimes C_B^\perp\right)^\perp$ and $\xi < w$ such that $d(x, \mathbb{F}^A \otimes C_B) + d(x, C_A \otimes \mathbb{F}^B) \leq \xi$. Then $d(x, C_A \otimes C_B) < 3\xi$.*

Proof. Denote by R the closest codeword of $C_A \otimes \mathbb{F}^B$ to x. Similarly, denote by C the closest codeword of $\mathbb{F}^A \otimes C_B$ to x. Notice that $C + R \in \left(C_A^\perp \otimes C_B^\perp\right)^\perp$. In addition, the weight of $C + R$ is bounded by:

$$|R + C| = |x + (x + R) + x + (x + C)|$$
$$\leq |(x + R)| + |(x + C)| \leq d\left(x, C_A \otimes \mathbb{F}^B\right) + d\left(x, \mathbb{F}^A \otimes C_B\right)$$
$$\leq w$$

Therefore, by the robustness property, there are $r \in C_A \otimes \mathbb{F}^B$ and $c \in \mathbb{F}^A \otimes C_B$ such that $R + C = r + c$. And r, c are supported on at most $|R + C|/\delta_0\Delta$ rows and columns. (Here r and c play the role of s, t in definition 4.1.1.)

Now observe that on one hand $C + c = R + r$, and on the other hand $C + c \in \mathbb{F}^A \otimes C_B$ and $R + r \in C_A \otimes \mathbb{F}^B$. Therefore, $C + c \in C_A \otimes \mathbb{F}^B \cap \mathbb{F}^A \otimes C_B$. Namely, $C + c \in C_A \otimes C_B$. Thus we have:

$$d\left(x, C_A \otimes C_B\right) \leq d\left(x, C\right) + d\left(C, C_A \otimes C_B\right)$$
$$\leq \xi + |c|$$

And in the same way we obtain also that $d\left(x, C_A \otimes C_B\right) \leq \xi + |r|$. Since c, r are supported on at most $|R+C|/\delta_0\Delta$ rows and columns, the weight of the string obtained by joining a single row of r with c grows by at least $\delta_0\Delta - |R + C|/\delta_0\Delta > 0$. Therefore, $|c| < |c + r| = |R + C|$. Thus, in total, $d\left(x, C_A \otimes C_B\right) \leq 3\xi$. $\square$

Claim 4.1.5. *Suppose that $v \in T_s$, Namely v is surounded by only normal vertices. Then:*

$$d\left(c_v, C_A \otimes C_B\right) < \Theta\left(\Delta^{3/2+\varepsilon}\right)$$

Proof. By being surrounded only by normal vertices any row in the local view of v is code-word of C_A plus at most $\Delta^{3/2+\varepsilon}/\Delta = \Delta^{\frac{1}{2}+\varepsilon}$ faults. So correcting the rows require flipping at most $\Delta \cdot \Delta^{\frac{1}{2}+\varepsilon}$ bits in total. Thus $d\left(c_v, C_A \otimes \mathbb{F}^B\right) < \Delta^{3/2+\varepsilon}$. In same way we obtain that $d\left(c_v, \mathbb{F}^A \otimes C_B\right) < \Delta^{3/2+\varepsilon}$. Noitce that, in parituclar, $d(c_v, \left(C_A^\perp \otimes C_B^\perp\right)^\perp) \le \Delta^{3/2+\varepsilon}$.

Denote by y the closest codeword of $\left(C_A^\perp \otimes C_B^\perp\right)^\perp$ to c_v. And observes that the distance between y to either $C_A \otimes \mathbb{F}^B$ or $\mathbb{F}^A \otimes C_B$ is at most $2 \cdot \Delta^{3/2+\varepsilon}$. To see it consider the decoding:

$$y \to x \to C_A \otimes \mathbb{F}^B$$

Therefore form claim 4.1.4 it follows that $d\left(y, C_A \otimes C_B\right) < 3\xi$, So $d\left(x, C_A \otimes C_B\right) < \Delta^{3/2+\varepsilon} + 3\xi$. $\qquad\square$

Claim 4.1.6 (The Technical Lemma)**.** *Let $A \subset S$ and $B \subset S_-$ subsets of the positive and the negetive vertices support x, and $\alpha \le \Delta^2, \beta \le \Delta$ their minimal degrees in $G_\square, G$ induced by x. Assume the following conditions hold:*

1. $\beta = \frac{\delta}{4\sqrt{\Delta}}\alpha + \Theta\left(\Delta\right)$

2. B *defined to be all the vertices connected to $\bar{A}$ by at least one heavy edge.*

3. *Any vertex in $\bar{A}$ has at least one heavy edge.*

Then: $d_{B \to \bar{A}} = \Omega\left(\Delta\right)$.

Proof. By the given $|S| \le \frac{2|x|}{\delta_0 \Delta} \le \zeta\frac{2n\Delta}{\delta}$ we have that $|S|/n \le \zeta \cdot \frac{2\Delta}{\delta}$. Then by the Mixing Expander Lemma we have that:

$$\alpha|A| \le |E(A, S)| \le \frac{\Delta^2}{n}|A||S| + 4\Delta\sqrt{|A||S|} \le |A| \cdot \zeta\frac{2\Delta^3}{\delta} + 4\Delta\sqrt{|A||S|}$$

$$\Rightarrow \sqrt{|A|}\left(\alpha - \zeta\frac{2\Delta^3}{\delta}\right) \le 4\Delta\sqrt{|S|}$$

$$\Rightarrow |A| \le \left(\alpha - \zeta\frac{2\Delta^3}{\delta}\right)^{-2} \cdot 16\Delta^2|S|$$

And by repetting the same calculation but consider B in the G graph we obtain:

$$\Rightarrow |B| \le \left(\beta - \zeta\frac{4 \cdot 2\Delta^2}{\delta}\right)^{-2} \cdot 16\Delta|S|$$

$$\Rightarrow |B| \le \left(\frac{\delta}{\sqrt{\Delta}}\alpha - \zeta\frac{4 \cdot 2\Delta^2}{\delta}\right)^{-2} \cdot 16\Delta|S|$$

$$= \frac{\Delta}{\delta^2}\left(\alpha - 4 \cdot 2\zeta\frac{\Delta^{2\frac{1}{2}}}{\delta^2}\right)^{-2} \cdot 16\Delta|S|$$

And for large enugh Δ the above is bounded by:

$$\left(\alpha - 4 \cdot 2\zeta \frac{\Delta^{2\frac{1}{2}}}{\delta^2}\right) \geq \left(\alpha - \zeta\frac{2\Delta^3}{\delta}\right) \Rightarrow |B| \leq \frac{1}{\delta^2}\left(\alpha - \zeta\frac{2\Delta^3}{\delta}\right)^{-2} \cdot 16\Delta^2|S|$$

Now, choose ζ such $\left(\alpha - \zeta\frac{2\Delta^3}{\delta}\right) \geq 16^{\frac{1}{2}} \cdot 100\Delta^{1\frac{1}{2}}$ yields that: $|A| \leq 10^{-4}\Delta^{-1}|S| \Rightarrow |\bar{A}| \geq \left(1 - 10^{-4}\Delta^{-1}\right)|S|$, And $|B| \leq 10^{-4}\frac{|S|}{16\delta_0^2\Delta}$. Conditions (2) and (3) garunte that any vertex in $\bar{A}$ is connected to at least on vertex of B. And therfore, B covers $\bar{A}$, that is, $d_{B\to\bar{A}} \cdot |B| \geq \bar{A}$, Hence:

$$d_{B\to\bar{A}} \geq \frac{|\bar{A}|}{|B|} \geq \left(1 - 10^{-4}\Delta^{-1}\right)10^4 \cdot \delta_0^2\Delta = \Theta\left(\Delta\right)$$

$\square$

Claim 4.1.7. *S_e and T satisfies the requrirments of claim 4.1.6 with $A = S_e$, $B = T$, $\alpha = \Delta^{3/2+\varepsilon}$ and $\beta = \delta_0\Delta + \Delta^{\frac{1}{2}+\varepsilon}$. That is, the average of havey edges form T to S_n is $\Theta\left(\Delta\right)$.*

Proof. Conditions (1) and (2) holds by definition of S_e, T for values $\alpha = \Delta^{3/2+\varepsilon}$ and $\beta = \delta_0\Delta - \Delta^{\frac{1}{2}+\varepsilon}/\delta_0$. It left to show that any normal vertex has at least on heaviy edge. By claim 4.1.1 any normal vertex v has weight at least $\delta_0\Delta$, yet by robustness there are $t, s \in C_A \otimes \mathbb{F}^B, \mathbb{F}^B \otimes C_B$ such that $t + s = c_v$. Assume that that any row of c_v has weight less than $\delta_0\Delta - \Delta^{\frac{1}{2}+\varepsilon}/\delta_0$. Pick an arbitery row, and denote it by τ, Now observes that by the fact that c_v has support on at most $\Delta^{\frac{1}{2}+\varepsilon}/\delta_0$ columns, then τ is at distance at most $\Delta^{\frac{1}{2}+\varepsilon}/\delta_0$ from C_A. But, by assumption, $|\tau| < \delta_0\Delta - \Delta^{\frac{1}{2}+\varepsilon}/\delta_0$ and therefore the closet codeword to τ in C_A has weight less than $\delta_0\Delta$ in contradiction to the fact that the distance of C_A is at least $\delta_0\Delta$.

Conditions (1) and (2) hold by definition of S_e, T for values $\alpha = \Delta^{3/2+\varepsilon}$ and $\beta = \delta_0\Delta - \Delta^{\frac{1}{2}+\varepsilon}/\delta_0$. It remains to show that any normal vertex has at least one heavy edge. By robustness there are $t, s \in C_A \otimes \mathbb{F}^B, \mathbb{F}^B \otimes C_B$ such that $t + s = c_v$. Assume that any row of c_v has weight less than $\delta_0\Delta - \Delta^{\frac{1}{2}+\varepsilon}/\delta_0$. Pick an arbitrary row, and denote it by τ. Now observe that by the fact that c_v has support on at most $\Delta^{\frac{1}{2}+\varepsilon}/\delta_0$ columns, then τ is at a distance of at most $\Delta^{\frac{1}{2}+\varepsilon}/\delta_0$ from C_A. But, by assumption, $|\tau| < \delta_0\Delta - \Delta^{\frac{1}{2}+\varepsilon}/\delta_0$ and therefore the closest codeword to τ in C_A has weight less than $\delta_0\Delta$, in contradiction to the fact that the distance of C_A is at least $\delta_0\Delta$. $\square$

Claim 4.1.8. *There is a normal-surounded vertex in $v \in T_s$ in weight at least $\Theta\left(\Delta^2\right)$.*

Proof. We know from claim 4.1.7 that $d_{T\to S_n}$ is linear in Δ. Morever claim 4.1.3 tell us that most of the vertices in S_- are surrounded only by normal vertices. Denote by $\mathbf{E}\left[d(v)|v \in T_s\right]$ the expected degree of heavy edges connected to vertex in T_s. Using the conditional expection formula we get:

$$\begin{aligned}
d_{T\to S_n} &= \mathbf{E}\left[d(v)|v \in T_s\right]\mathbf{Pr}\left[v \in T_s\right] + \mathbf{E}\left[d\left(v\right)|v \in T/T_s\right]\mathbf{Pr}\left[v \in T/T_s\right]\\
&\leq \mathbf{E}\left[d(v)|v \in T_s\right]\mathbf{Pr}\left[v \in T_s\right] + \mathbf{E}\left[d\left(v\right)|v \in T/T_s\right]\mathbf{Pr}\left[v \in S_-/T_s\right]\\
&\leq \mathbf{E}\left[d(v)|v \in T_s\right] \cdot 1 + \Delta \cdot \Delta^{-\varepsilon/4}\\
&\Rightarrow \mathbf{E}\left[d(v)|v \in T_s\right] \geq d_{T\to S_n} - \Delta^{\frac{3}{4}\varepsilon} = \Theta\left(\Delta\right)
\end{aligned}$$

Therefore there is at least a single vertex in T_s connected to $\Theta(\Delta)$ havey edges. Combining the fact that edge is heavy edge if there are at least $\delta_0\Delta - \Delta^{\frac{1}{2}+\varepsilon}$ non trival bits on it's squares we get the desired. $\qquad\square$

Remark 4.1.1. *For small codes which are robust for $w < \Delta^{3/2}$, as in the original proof of [LZ22], claim 4.1.3 no longer holds. However, assuming the dual tensor code is also p-resistant to puncturing 4.1.2, one can still prove claim 4.1.8.*

We are about to finish the proof of the theorem. Combining claim 4.1.8 and claim 4.1.5 we obtain the the existnes of a negative vertex which is both at distance $\Theta\left(\Delta^{3/2+\varepsilon}\right)$ form $C_A \otimes C_B$ and weight at least $\Theta\left(\Delta^2\right)$. Denote by $v \in T_s$ that vertex, by c_v it's local view and by $y \in C_A \otimes C_B$ the closest codeword to c_v. Subtructing y from c_v yilds:

$$|c_v + y| = d\left(c_v, y\right) = \Theta\left(\Delta^{3/2+\varepsilon}\right) < |c_v|$$
$$\Rightarrow |c + y| < |c|$$

4.2 LTC.

As exactly as in the polynomial code case, we will prove that a code is locally testable by presenting a decoder that can correct small errors and reject errors greater than a linear threshold. But before that, let us define the LTC code.

Definition 4.2.1. *Consider the Tanner code defined on the square complex and the negative vertices as presented above, but instead of taking the dual tensor as the local code, we take the product code: $\mathcal{T}\left(G^+, C_A \otimes C_B\right)$.*

Now, let us define the disagreement code, which we can conceptually think of as the strings obtained by an attempt to decode by local correction. That is, any vertex chooses the closest codeword to its local view and the summation of the suggestions is the disagreement. Notice that if the initial assignment was a valid codeword then each of the vertices suggest the codeword it sees on its local view. Hence, for any edge will suggest exactly the same bit and the summation on the edge will be equal to zero. Because it is true for all the edges, the total string obtained will be the zero codeword. Nevertheless, any assignment on an edge which equals 1 points to a disagreement between the vertices at the edge's ends.

Definition 4.2.2 (The Disagreement Code). *Given a Tanner code $C = \mathcal{T}(G, C_0)$, define the code $C_\oplus$ to contain all the words equal to the formal summation $\sum_{v \in V(G)} c_v$ when c_v is an assignment of a codeword $c_v \in C_0$ on the edges of the vertex $v \in V(G)$. We call to such code the **disagreement code** of C, as edges are set to 1 only if their connected vertices contribute to the summation codewords that are different on the corresponding bit to that edge. In addition, we will call to any contribute c_v, the **suggestion** of v. And notice that by linearity, each vertex suggests, at most, a single suggestion.*

Finally, given a bits assessment $x \in \mathbb{F}_2^E$ over the edges of G, we will denote by $x^\oplus \in C_\oplus$ the codeword which obtained by summing up suggestions set such each vertex suggests the closet codeword to his local view. Namely, for each $v \in V$ define:

$$c_v \leftarrow \arg_{\tilde{c} \in C_0} \min d(x|_v, \tilde{c}) \ \ \forall v \in V$$
$$x^\oplus \leftarrow \sum_{v \in V} c_v$$

We will think about $x^\oplus$ as the disagreement between the vertices over x.

Definition 4.2.3. Let $C = \mathcal{T}(G, C_0)$. We say that $x \in C_\oplus$ is **reducible** if there exists a vertex v and a small codeword c_v, for which, adding the assignment of c_v over the v's edges to x decreases the weight. Namely, $|x + c_v| < |x|$. If $x \in C_\oplus$ is not a reducible codeword then we say that x is **irreducible** .

Lemma 4.2.1 (Linearity Of The Disagreement). *Consider the code $C = \mathcal{T}(G, C_0)$. Let $x \in \mathbb{F}_2^E$ then for any $y \in C$ it holds that:*

$$(x + y)^\oplus = (x)^\oplus$$

Proof. Having that $y \in C$ followes $y|_v \in C_0$ and therefore

$$\arg_{\tilde c \in C_0} \min d(z, \tilde c) = y|_v + \arg_{\tilde c \in C_0} \min d(z, \tilde c + y|_v)$$

Hence the suggestion made by vertrx v is:

$$
\begin{aligned}
c_v &\leftarrow \arg_{\tilde c \in C_0} \min d((x + y)|_v, \tilde c) \\
&\leftarrow y|_v + \arg_{\tilde c \in C_0} \min d((x + y)|_v, \tilde c + y|_v) \\
&\leftarrow y|_v + \arg_{\tilde c \in C_0} \min d(x|_v, \tilde c)
\end{aligned}
$$

It follows that:

$$
\begin{aligned}
(x + y)^\oplus &= \sum_{v \in V} c_v = \sum_{v \in V} y|_v + \sum_{v \in V} \arg_{\tilde c \in C_0} \min d(x|_v, \tilde c) \\
&= y^\oplus + x^\oplus = x^\oplus
\end{aligned}
$$

When the last transition follows immediately by the fact that $y \in C$ and therefore any pair of connected vertices contribute the same value for their associated edge $\qquad\square$

Furthermore, in the case of the LTC code definition 4.2.1, the disagreement code is simply C_X (or C_Z). This is because any vertex contributes to the local view of the positive vertices codewords in either $C_A \otimes \mathbb{F}^B$ or $\mathbb{F}^A \otimes C_B$.

4.3 Decoding and Testing

For completeness, we show exactly how Theorem 1 implies testability. The following section repeats Leiverar's and Zemor's proof [LZ22]. Consider a binary string x that is not a codeword. The main idea is the observation that the number of bits fillped by (any) decoder, while decoding x, bounds the distance $d(x, C)$ from above. In addition, the number of positive checks in the first iteration is exactly the number of violated restrictions.

Definition 4.3.1. *Let $L = \{L_i\}_0^{2|E|}$ be a series of $2|E|$. Such that for each vertex $v \in V$ $\sum_{e = \{u, v\}} L_{e_v} \in C_0$. We will call L a Potential list and refer to the e_v'the element of L as a suggestion made by the vertex $v \in V$ for the edge $e \in E$. Sometimes we will use the notation L_v to denote all the L's coordinates of the form $L_{e_v} \forall e \in Support(v)$. Define the Force of L to be the following sum $F(L) = \sum_{e = \{v, u\} \in E} (L_{e_v} + L_{e_u})$ and notice that $F(L) \in C_\oplus$. And define the state $S(L) \subset \mathbb{F}_2^{|E|}$ of L as the vector obtained by choosing an arbitrary value from $\{L_{e_v}, L_{e_u}\}$ for each edge $e \in E$.*

Claim 4.3.1. *Let L be the Potential list. If $F(L) = 0$ then $S(L) \in C$.*

Proof. Denote by $\phi(e) \subset \{L_{e_v}, L_{e_u}\}$ the value which was chosen to $e = \{v, u\} \in E$. By $F(L) = 0$, it follows that $L_{e_v} + L_{e_u} = 0 \Rightarrow L_{e_v} = L_{e_u} = \phi(e)$ for any $e \in E$. Hence for every $v \in V$ we have that $S(L)|_v = \sum_{u \sim v} \phi(\{v, u\}) = \sum_{u \sim v} L_{e_v} \in C_0 \Rightarrow S(L) \in C$ $\qquad\square$

The decoding goes as follows. First, each vertex suggests the closet C_0's codeword to his local view. Those suggestions define a Potential list, denote it by L, then if $F(L) < \tau$, by Theorem 1, one could find a suggestion of vertex v and a codeword c_v such that updating the value of $L_v \leftarrow L_v + c_v$ yields a Potential list with lower force. Therefore repeating the process till the force vanishes, obtain a Potential list in which its state is a codeword.

Definition 4.3.2. *Let $\tau > 0, f : \mathbb{N} \to \mathbb{R}^+$, and consider a Tanner Code $C = \mathcal{T}(G, C_0)$. Let us Define the following decoder and denote it by $\mathcal{D}$.*

> **Data:** $x \in \mathbb{F}_2^n$
> **Result:** $\arg\min \{y \in C : |y + x|\}$ if $d(y, C) < \tau$ and False otherwise.
> 1 $L \leftarrow \text{Array}\{\}$
> 2 **for** $v \in V$ **do**
> 3 $\quad c'_v \leftarrow \arg\min \{y \in C_0 : |y + x|_v|\}$
> 4 $\quad L_v \leftarrow c'_v$
> 5 **end**
> 6 $z \leftarrow \sum_{v \in V} c'_v$
> 7 **if** $|z| < \tau \frac{n}{f(n)}$ **then**
> 8 $\quad$ **while** $|z| > 0$ **do**
> 9 $\quad\quad$ find v and $c \in C_0$ such that $|z + c_v| < |z|$
> 10 $\quad\quad z \leftarrow z + c_v$
> 11 $\quad\quad L_v \leftarrow L_v + c_v$
> 12 $\quad$ **end**
> 13 **else**
> 14 $\quad$ reject.
> 15 **end**
> 16 **return** $S(L)$

Algorithm 1: Decoding

Theorem 4.3.1. *Consider a Tanner Code $C = [n, n\rho, n\delta]$ and the disagreement code $C_\oplus$ defined by it. Suppose that for every codeword $z \in C_\oplus$ in $C_\oplus$ such that $|z| < \tau' n / f(n)$, there exists another codeword $y \in C_\oplus$ such that $|y| < |z|$, set $\tau \leftarrow \frac{\tau'}{6\Delta} \delta$ then,*

1. *$\mathcal{D}$ corrects any error at a weight less than $\tau n / f(n)$.*

2. *C is $f(n)$ testable code.*

Proof. So it is clear from the claim claim 4.3.1 above that if the condition at line (6) is satisfied, then $\mathcal{D}$ will converge into some codeword in C. Hence, to complete the first section, it left to show that

$\mathcal{D}$ returns the closest codeword. Denote by e the error, and by simple counting arguments; we have that $\mathcal{D}$ flips at most:

$$d_{\mathcal{D}}(x, C) \le 2|e|\Delta + \tau \frac{n}{f(n)} \Delta$$

bits. Hence, by the assumption,

$$d_{\mathcal{D}}(x, C) \le 3\Delta\tau \frac{n}{f(n)} \le 3\Delta\tau\delta n < \frac{1}{2}\delta n$$

Therefore the code word returned by $\mathcal{D}$ must be the closet. Otherwise, it contradicts the fact that the relative distance of the code is δ. To obtain the correctness of the second section, we will separate when the conditional at the line (5) holds and not. And prove that the testability inequality holds in both cases. Let $x \in \mathbb{F}_2^n$ and consider the running of $\mathcal{D}$ over x. Assume the first case, in which the conditional at line (5) is satisfied. In that case, $\mathcal{D}$ decodes x into its closest codeword in C. Therefore:

$$d(x, C) \le d_D(x, C) \le m\xi(x)\Delta + |z|\Delta$$
$$\le m\xi(x)\Delta + m\xi(x)\Delta^2$$
$$\frac{d(x, C)}{n} \le \kappa_1 \xi(x)$$

Now, consider the other case in which: $|z| \ge \tau \frac{n}{f(n)}$.

$$\frac{d(x, C)}{n} \le 1 \le \frac{|z|}{\tau n} f(n) \le \frac{m}{n} \frac{1}{\tau} \Delta\xi(x) f(n)$$
$$\le \kappa_2 \xi(x) f(n)$$

Picking $\kappa \leftarrow \max\{\kappa_1, \kappa_2\}$ proves $f(n)$-testability   □

Chapter 5

Further Research.

Both good quantum LDPC and good LTC had been open problems for more than decades, so the fact that they have been solved in one stroke raises two interesting questions. The first one is whether the construction that obtains each of them alone can also yield good quantum LTC? What is the exact challenge that prevents the inventors from proceeding into quantum testability?

The second question is whether one of the codes can be obtained without giving the other. Either positive or negative answers to these questions will shed light on our understanding of what quantumness is.

In this chapter, we present our attempts to provide answers. The chapter is divided into two parts. In the first, we demonstrate a variation of the classical Tanner code that defines many equations. We believe that this variation contains enough structure to guarantee degeneration of equations similarly to what occurs in the polynomial code. In the second chapter, we show that the polynomial code is not w-robust. If it were not the case, then one could hope to obtain quantum testability by considering a Tanner code in which any vertex restricts its local view to be a codeword of a quantum code.

5.1 Local Majority $=^?$ Local Testability.

We begin by demonstrating that selecting C_0, the small code in the Tanner code, to have a large distance, which can also be considered as adding numerous restrictions, yields testability. However, the amount one will have to enlarge the distance to cannot be achieved with a rate greater than $\frac{1}{2}$ by the Singleton bound.

5.1.1 Almost LTC With Zero Rate

Theorem 5.1.1 (LTC Zero Rate)**.** *There exist a constant $\alpha > 0$ and an infinte family of Tanner Codes $C = \mathcal{T}(G, C_0)$ such that any ireducable [4.2.3] codeword x of a coresponding disagreement code $x \in C_\oplus$ at length n, weight at least αn.*

Proof. By induction over the number of vertices $V' \subset V$, which suggest a nontrivial codeword to x. Base, assume that a single vertex $v \in V$ suggests a nontrivial codeword $c_v \in C_0$. Then it's clear that $x = c_v$. And therefore, we have that $|x + c_v| = 0 < |x|$.

Assume the correctness of the argument for every codeword defined by at most m nontrivial suggestions made by $V' \subset V$. And consider the graph (V', E') induced by them. If the graph has more than a single connectivity component, then any of them is also a codeword of $C_\oplus$ but composed of at most $m - 1$ nontrivial suggestions. Therefore, by the assumption, we could find a vertex v and a proper small codeword $c_v \in C_0$, such that the addition of the suggestion will decrease the weight of the codeword defined on that component and therefore decrease the total weight of x.

So, we can assume that the vertices in V' compose a single connectivity component. Let be $x|_v \in \mathbb{F}_2^\Delta$ the bits of x on the indices corresponding to v's edges. For any $S \subset E$, define $w_S(x)$ as the weight that x induces over S. Sometimes we will refer to $w_S(x)$ as the **flux** induced by x over S.

The genreal idea of the proof is to show that if the distance of the small code is large ($\geq \frac{2}{3}$) and x is ireducable [4.2.3] codeword then there exist an indepandent subset of vertices $U \subset V'$, at linear size, that induce a significant flux over E/E'. If U has linear size than also x has a linear size, And if not, Then we will show that no serious interface has been occurred. claim 5.1.1 and claim 5.1.2 state that if one is willing to hide an ireducable [4.2.3] error then he has to touch at least a linar number of verties. claim 5.1.3 and claim 5.1.5 quanitify the flux that induced by such errors.

Claim 5.1.1. *For any $v \in V'$ and corresponded suggestion c_v it holds that:* $w_{E'}(c_v) \geq \frac{1}{2}\delta_0\Delta$.

Proof. Notice that any edge of E connected only to a single vertex in V' equals the corresponding bit in the original suggestion made by c_v. Hence for every $v \in V'$, it holds that:

$$w_{E/E'}(x|_v) = w_{E/E'}(c_v) \Rightarrow w_{E/E'}(x|_v) \leq |x \cap c_v|$$

Now consider the weight of $x + c_v$, By the assumption that x is ireducable code word of $c_\oplus$ we have that:

$$|x + c_v| = |x| + |c_v| - 2|x \cap c_v| > |x|$$
$$\Rightarrow |x \cap c_v| < \frac{1}{2}|c_v|$$
$$w_{E'}(c_v) = |c_v| - w_{E/E'}(c_v) = |c_v| - w_{E/E'}(x|_v)$$
$$\geq |c_v| - |x \cap c_v| \geq \frac{1}{2}|c_v| = \frac{1}{2}\delta_0\Delta$$

$\square$

Consider an arbitrary vertex $r \in V'$, and consider the DAG obtained by the BFS walk over the subgraph (V', E') starting at r. Denote this directed tree by T.

Claim 5.1.2. *The size of T is at least:*

$$|T| \geq \left(\frac{1}{4}\delta_0 - \frac{\lambda}{\Delta}\right)n$$

Proof. By claim 5.1.1 any $v \in T$ the degree of v is at least $\frac{1}{2}\delta_0\Delta$ we have that: $E(T,T) \geq \frac{1}{2} \cdot$

$\frac{1}{2}\delta_0\Delta|T|$. Combine the Mixining Expander Lemma we obtain:

$$\frac{1}{4}\delta_0\Delta|T| \leq \frac{\Delta}{n}|T|^2 + \lambda|T|$$
$$\Rightarrow \left(\frac{\Delta}{n}|T| + \lambda - \frac{1}{4}\delta_0\Delta\right)|T| \geq 0$$
$$\Rightarrow |T| \geq \left(\frac{1}{4}\delta_0 - \frac{\lambda}{\Delta}\right)n$$

$\square$

Claim 5.1.3. *Suppose that G is an expander graph with a second eigenvalue λ, then For any layer U there exist a layer U' such that:*

(1) $|U'| \geq |U|$

(2) $w_{E/E'}(x|_{U'}) \geq \Delta|U'|\left(\delta_0 - \frac{2}{3} - \frac{2\lambda}{\Delta}\right)$

Proof. Consider layer U and denote by U_{-1} and U_{+1} the preceding and the following layers to U in T. It follows from the expander mixing lemma that:

$$w_{E/E'}(x|_U) \geq \delta_0\Delta|U| - w\left(E(U_{-1}\bigcup U_{+1}, U)\right) \geq$$
$$\delta_0\Delta|U| - E(U_{-1}\bigcup U_{+1}, U)$$
$$\delta_0\Delta|U| - \Delta\frac{|U||U_{-1}|}{n} - \Delta\frac{|U||U_{+1}|}{n}$$
$$- \lambda\sqrt{|U||U_{-1}|} - \lambda\sqrt{|U||U_{+1}|}$$

Claim 5.1.4. *We can assume that $|U| \geq |U_{-1}|, |U_{+1}|$.*

Proof. Suppose that $|U_{+1}| > |U|$, so we could choose U to be U_{+1}. Continuing stepping deeper till we have that $|U| > |U_{+1}|, |U_{-1}|$. Simiraly, if $|U| > |U_{+1}|$ but $|U_{-1}| > |U|$, the we could take steps upward by replacing U_{-1} with U. At the end of the process, we will be left with U at a size greater than the initial layer and $|U| > |U_{+1}|, |U_{-1}|$ $\square$

Using claim 5.1.4, we have that $(|U_{+1}| + |U_{-1}|)/n < \frac{2}{3}$ and therefore:

$$w_{E/E'}(x|_U) \geq \left(\delta_0 - \frac{2}{3} - \frac{2\lambda}{\Delta}\right)\Delta|U|$$

$\square$

That immediately yields the following: let $U_{\max} = \arg\max_{U \text{ layer in } T}|U|$ then:

$$|x| \geq w_{E/E'}(x|_{U_{\max}}) \geq \left(\delta_0 - \frac{2}{3} - \frac{2\lambda}{\Delta}\right)\Delta|U_{\max}|$$

Claim 5.1.5. *Consider again the maximal layer $U_{\max}$ then:*

$$w_{E/E'}(x) \geq \left(\delta_0 - \frac{|U_{\max}|}{n} - \frac{\lambda}{\Delta}\right)\Delta|T|$$

Proof. Similarly to above, now we will bound the flux that all the nodes in T induce over E/E'. Denote by $U_0, U_1..U_m$ the layers of T ordered corresponded to their height, thus we obtain:

$$w_{E/E'}(x) \geq \delta_0 \Delta |T| - \sum_{i \in [m]} w\left(E\left(U_i, U_{i+1}\right)\right)$$

$$\geq \delta_0 \Delta |T| - \sum_{i \in [m]} E\left(U_i, U_{i+1}\right)$$

$$\geq \delta_0 \Delta |T| - \sum_{i \in [m]} \frac{\Delta}{n}|U_i||U_{i+1}| + \lambda \sqrt{|U_i||U_{i+1}|}$$

$$\geq \delta_0 \Delta |T| - \sum_{i \in [m]} \frac{\Delta}{n}|U_i||U_{i+1}| + \lambda \frac{|U_i| + |U_{i+1}|}{2}$$

$$\geq \delta_0 \Delta |T| - \frac{\Delta}{n}|T||U_{\max}| - \lambda |T|$$

$$\geq \left(\delta_0 - \frac{|U_{\max}|}{n} - \frac{\lambda}{\Delta}\right)\Delta |T|$$

$\square$

Claim 5.1.6. *Consider $G = (V, E)$ a Δ-ramunjan graph and let U be a subset of V such that $|U| \geq \frac{1}{9}n$ then, there is must to be at least one vertex in U such the number of closed loops pass through it, is less than $\sqrt{\Delta} \cdot n$.*

Claim 5.1.7. *Alternate proof of fulx inequality, which dosn't assume that there is no interference inside the layers. $w\left(E\left(U, U\right)\right) > 0$.*

Proof. Separeate into the following cases, First assume that $|U_{\max}|/n > \frac{1}{3}$ then we have that the total interference with $U_{\max}$ layers is at most:

$$\frac{\Delta |U_{\max}|\left(n - |U_{\max}|\right)}{n} + \lambda \sqrt{|U_{\max}|n} \leq \left(1 - \frac{|U_{\max}|}{n} + \sqrt{3}\frac{\lambda}{\Delta}\right)\Delta |U_{\max}|$$

$$\leq \left(\frac{2}{3} + \sqrt{3}\frac{\lambda}{\Delta}\right)\Delta |U_{\max}|$$

And therefore we have that the flux induced by $U_{\max}$ is at least:

$$\left(\delta_0 \Delta - \frac{2}{3} + \sqrt{3}\frac{\lambda}{\Delta}\right)\Delta |U_{\max}|$$

So it lefts to consider the case in which for every layer it holds that $|U_{\max}| \leq \frac{1}{3}n$. At that case we count the fulx induced by the whole three T which is what exactly we have prove in **??** minus the inner interference at the tree, That it we need only to subtract $\sum \frac{\Delta|U_i|^2}{n} + \lambda|U_i| \leq \left(\frac{|U_{\max}|}{n} + \lambda/\Delta\right)|T|$ So we obtained that in that case:

$$w_{E/E'}(x) \geq \left(\delta_0 - 2\frac{|U_{\max}|}{n} - 2\lambda/\Delta\right)\Delta |T| \geq \left(\delta_0 - \frac{2}{3} - 2\frac{\lambda}{\Delta}\right)\Delta |T|$$

$\square$

Proof of Theorem 1. Consider the size of the maxiaml layer $|U_{\max}|$ and sepearte to the following two cases. First, consider the case that $|U_{\max}| \geq \alpha n$ in that case it follows immedily by claim 5.1.3 that if $\delta_0 > \frac{2}{3} - \frac{2\lambda}{\Delta}$ there exists $\alpha' > 0$ such that:

$$|x| \geq \left(\delta_0 - \frac{2}{3} - \frac{2}{\lambda}\Delta\right)\Delta|U_{\max}| \geq \alpha' n$$

So, it is lefts to consider the second case in which $|U_{\max}| < \alpha n$ in that case, we have from claim 5.1.5 inequality that:

$$|x| \geq w_{E/E'}(x) \geq \left(\delta_0 - \frac{|U_{\max}|}{n} - \frac{\lambda}{\Delta}\right)\Delta|T|$$
$$\geq \left(\delta_0 - \alpha - \frac{\lambda}{\Delta}\right)\Delta|T|$$

Setting $\alpha \geq \frac{2}{3}$ we complete the proof $\qquad\square$

Unfortunately, Singleton bound does not allow both $\delta_0 > \frac{2}{3}$ and $\rho_0 \geq \frac{1}{2}$, so in total, we have proven the existence of an LDPC code which is good in terms of testability and distance yet has a zero rate. In the following subsection, we present an attempt to overcome this problem by considering a variant of Tanner code, in which every vertex checks only a $\frac{2}{3}$ fraction of the edges in its support. By doing this, we obtain a good LDPC code, in which the flux on the trivial vertices is proportional to the relation in claim 5.1.3. However, the disagreement derived form a valid codeword is not the zero string anymore. We leave figuring out if there exists an alternative tester as an open problem.

5.1.2 Overcoming The Vanishing Rate.

Consider the following code; instead of associating each edge with a pair of checks, let's define the vertices to be the checks of small codes over $q \in [0, 1]$ fraction of their edges. That is, now each vertex defines only $(1 - \rho_0)q\Delta$ restrictions. Hence, the rate of the code is at least:

$$\rho\frac{1}{2}\Delta n \geq \frac{1}{2}\Delta n - (1 - \rho_0)q\Delta n$$
$$\Rightarrow \rho \geq \left(2\rho_0 + \left(\frac{1}{q} - 2\right)\right)q$$
$$\rho_0 \geq 1 - \frac{1}{2q}$$

for example, if $q = 2/3$, then for having constant rate, it is enough to ensure that $\rho_0 \geq 1 - \frac{3}{4} = \frac{1}{4}$.

Intuition For Testability. Before expand the construction let's us justify why one should even expects that removing constraints preserves testability. Assume that is guaranteed that the lower bound of the flux on the trivial vertices remains up to multiplication by the fraction factor q, or put it differently, one could just stick q in every inequality without lose correctness, Then:

$$w_{E/E'}(x|_U) \geq \delta_0 q\Delta|U| - qw\left(E(U_{-1}\bigcup U_{+1}, U)\right)$$
$$\Rightarrow |x| \geq \left(\delta_0 - \frac{2}{3} - \frac{2\lambda}{\Delta}\right)q\Delta|U_{\max}|$$

As you can see, ireducable [4.2.3] words of the disagreement have a linear weight, despite that the original code has non-vanish rate.

Yet, we still need more to prove a linear distance. By repeating the proof of the Singleton Bound (Ref. 2.3), it follows that the small code $\tilde{C}_0$ obtained by ignoring arbitrary $(q - \frac{1}{2})\Delta$ coordinates yields a code with distance:

$$\left(\delta_0 q - \left(q - \frac{1}{2}\right)\right)\Delta$$

Assuming we can engineer an expander family such that the graphs obtained by removing half of the edges connected to each vertex are also expanders, and in addition, each edge is checked by both vertices on its support with respect to $\tilde{C}_0$, then a good Tanner Code can be defined on the restricted graphs. Any string that satisfies the original checks will also have a linear weight. To achieve this property, we will restrict ourselves to a particular family of Cayley Graphs.

Theorem 5.1.2. *There exist a constant $\alpha > 0$ and infinite family of codes which satisfies theorem 5.1.1 and also good.*

Definition 5.1.1 (Testability Tanner Code)**.** *Let $q > \frac{1}{2}$ and let J be a generator set for group Γ such that $|J| = \Delta$, $q|\Delta$, J closed for inverse, and there exist subset of J, denote it by, J' such that J' is a generator set of Γ and $|J'| = \frac{1}{2}\Delta$. Let C_0 be a code with parameters $C_0 = q\Delta[1, \rho_0, \delta_0]$. For any vertex associate a subset $\bar{J}_v \subset J/J'$ at size:*

$$|\bar{J}_v| = \left(q - \frac{1}{2}\right)\Delta \Rightarrow |\bar{J}_v \cup J'| = q\Delta$$

Define the code $\mathcal{T}(J, q, C_0)$ to be the subspace such that any vertex's local view over the edges defined by $\bar{J}_v \cup J'$ is a codeword of C_0. In addition, let's associate a code $\tilde{C}_v$ obtained for any vertex by ignoring the bits supported on the $\bar{J}_v$ coordinates. Notice that code defined by requiring that the local view of any vertex v of $\text{Cayley}(\Gamma, J')$ is a codeword of $\tilde{C}_v$ is a Tanner code. Denote it by $\tilde{\mathcal{T}}(J, q, C_0)$.

Claim 5.1.8. *Let J be defined as above such that both $\text{Cayley}(\Gamma, J)$, $\text{Cayley}(\Gamma, J')$ are expanders with algebraic expansion greater than λ and C_0 with the parameters $\rho_0 > 1 - \frac{1}{2q}$ and $\delta_0 q - \left(q - \frac{1}{2}\right) > 2\lambda/\Delta$. Then the code $\mathcal{T}(J, q, C_0)$ is a good code.*

Proof. We have already proved that the code has a positive rate, so it is left to show a constant relative distance.

Consider a codeword x and denote by x' the restriction of x to $Cayley(\Gamma, J')$ which is a codeword of $\tilde{C} = \tilde{\mathcal{T}}(J, q, C_0)$. But $\tilde{C}$ is a Tanner Code such that any vertex sees at least $\tilde{\delta}_0\Delta :=$ $\left(\delta_0 - \left(q - \frac{1}{2}\right)\right)\Delta$ nontrivial bits. Denote by S the vertices subset supports x', and by $E(S, S)$ the edges from S to itself, and by using the fact that $Cayley(\Gamma, J')$ is an expander with second eigenvalue at most λ we have that:

$$\frac{|x'|}{|S|} \geq \tilde{\delta}_0\Delta \Rightarrow |S| \geq \left(\tilde{\delta} - \frac{2\lambda}{\Delta}\right)\Delta n$$

By the assumption that $\tilde{\delta} > 2\lambda/\Delta$ we have that S must has liner size, and therefore $|x'|$ also must to be linear in n. Finally as $x' \subset x$ we obtain the correctness of the claim. $\square$

Claim 5.1.9 (Existence of such *Cayley's*). *Let S be a generator set such that* $\text{Cayley}(\Gamma, S)$ *has a second largest eigenvalue greater then* λ, *And consider an arbitrary group element* $g \in \Gamma$ *and denote by* S_g *the set* gSg^{-1}. *Then the second eigenvalue of the graph obtained by* $(\Gamma, S) \cup (\Gamma, S)$ *is at most* 2λ.

Proof. Denote by G, G' the *Cayley* graphs corresponding to S, S_g, for convenient we will use the notation of $\sum_{v \sim_G u}$ to denote a summation over all the neighbors of v in the graph G. Let $A_{G'}$ be the adjacency matrix of G'. Recall that G' is a Δ regular graph, and therefore the uniform distribution $\mathbf{1}$ is the eigenstate with the maximal eigenvalue, and the second eigenvalue is given by the min-max principle:

$$\max_{f \perp \mathbf{1}} \frac{f^\top A_{G'} f}{f^\top f} = \max_{f \perp \mathbf{1}} \sum_v \sum_{u \sim_{G'} v} \frac{f(u) f(v)}{f^\top f}$$

$$= \max_{f \perp \mathbf{1}} \sum_v \sum_{\tau \in S} \frac{f\left(g\tau g^{-1} v\right) f(v)}{f^\top f}$$

$$= \max_{f \perp \mathbf{1}} \sum_{gv} \sum_{\tau \in S} \frac{f\left(g\tau g^{-1} gv\right) f(gv)}{f^\top f}$$

$$= \max_{f \perp \mathbf{1}} \sum_{gv} \sum_{\tau \in S} \frac{f(g\tau v) f(gv)}{f^\top f}$$

$$= \max_{f \perp \mathbf{1}} \sum_{gv} \sum_{u \sim_G v} \frac{f(gu) f(gv)}{f^\top f}$$

As for any function $f : V \to \mathbb{R}$ one could define a function $f' : E \to \mathbb{R}$ such that $f'(v) = f(v)$ and f' preserves the norm:

$$f'^\top f' = \sum_{v \in V} f'(v) f'(v) = \sum_{v \in V} f^\top(vg) f(vg) = f^\top f$$

$$\Rightarrow \max_{f \perp \mathbf{1}} \frac{f^\top A_{G'} f}{f^\top f} = \max_{f \perp \mathbf{1}} \sum_{gv} \sum_{u \sim_G v} \frac{f(gu) f(vg)}{f^\top f}$$

By the Interlacing Theorem, [Hae95] the second eigenvalue of any subgraph of G' is less than the λ', In particular, the eigenvalue of the graph obtained by taking the edges that are associated with elements of the S_g / S.

Denote that subgraph by $G'_{/S}$. Because $S_g / S \cap S = \emptyset$, we have that the edges sets of G, G' are disjointness sets. Hence the adjacency matrix of the graphs union equals the sum of their adjacency matrices. So in total, we obtain that:

$$\lambda' = \max_{f \perp \mathbf{1}} \frac{f^\top \left(A_G + A_{G'_{/S}}\right) f}{f^\top f}$$

$$\leq \max_{f \perp \mathbf{1}} \frac{f^\top A_G f}{f^\top f} + \max_{f \perp \mathbf{1}} \frac{f^\top A_{G'_{/S}} f}{f^\top f}$$

$$\leq \lambda + \lambda = 2\lambda$$

$\square$

Claim 5.1.10. *If Δ is a constant greater than two, and G is a λ-algebraic expander with girth at length $\Omega(\log n)$, then there exists a $g \in \Gamma$ such that $S_g \cap S = \emptyset$.*

Proof. As $\Delta > 2$ there must be at least two different elements $s_1, s_2 \in S$ such that $s_1 \neq s_2, s_2^{-1}$. Pick $g = s_1 s_2$. Now assume through contradiction that there exists a pair $s, r \in S$ such that $gsg^{-1} = r \Rightarrow gs = rg$ and notice that the fact that $s_1 \neq s_2^{-1}$ guarantees that both terms are a product of 3 element group. Therefore either that there is a 6-length cycle in the graph, Or that there is element-wise equivalence, namely $s_1 = r, s_2 = s_1, s = s_2$. The first case contradict the lower bound on the expander girth, which is at least $\Omega(\log_\Delta(n))$, while the other stand in contradiction to the fact that $s_1 \neq s_2$ $\qquad \square$

Remark 5.1.1 (Regarding Quantum Codes.). *Notice that any complex designed to hold CSS qLDPC codes must have constant length cycles. Otherwise, the distance of C_x will not be constant, and therefore the condition $H_x H_z^\top = 0$ could be satisfied only if H_z is not a constant row-weight matrix, Put differently C_z is not an LDPC code. Consequently, any trial to generalize the construction for obtaining quantum codes must not rely on claim 5.1.10.*

Remark 5.1.2 (Note On Random Construction.). *One might wondering if using Cayley is necessary. We conjecture that there is a constant $c > 0$ such that sampling pair of $(1 + c)\frac{1}{2}\Delta$ regular random graphs, and than take the anti-symmetry union of them might also obtain a good expander such that each of the residue part also has good expansion with heigh probability.*

Claim 5.1.11. *Consider the graph G and the code C as defined in 5.1.1 and let S, T be a pair of disjointness vertices subsets. And let x_S and x_T codewords of the $C_\oplus$ such that x_S suggested only by vertices in S, and in similar manner x_T suggested only by T's vertices. Then the flux of S over T is at most:*

$$w_T(x_S) \leq E_{G'}(S, T) \leq \frac{1}{2}\Delta\frac{|S||T|}{n} + \lambda\sqrt{|S||T|}$$

Proof. The only edges that can interfere are the edge defined by J', Namely the edges which belong to $Cayley(\Gamma, J')$. Therefore it's enough to use the mixing expander lemme on the $\frac{1}{2}\Delta$-regular graph. $\qquad \square$

Proof of theorem 5.1.2. Notice that $\frac{1}{2} < \frac{2}{3} = q$, Thus repeating the proof of theorem 5.1.1 yields that:

$$\begin{aligned}
w_{E/E'}(x|_U) &\geq \delta_0 q\Delta|U| - w\left(E(U_{-1}\bigcup U_{+1}, U)\right) \\
&\geq \delta_0 q\Delta|U| - \frac{\Delta}{2}\frac{|U|(|U_{-1}| + |U_{+1}|)}{n} - \lambda\sqrt{|U|(|U_{-1}| + |U_{+1}|)} \\
&\geq \left(\delta_0 - \frac{2}{3} - \frac{1}{q}\frac{\lambda}{\Delta}\right)q\Delta|U|
\end{aligned}$$

When the last inequality follows from the fact that the proof of claim 5.1.4 does not rely on graph structure arguments. The same arguments also lead to an analogous inequality for claim 5.1.5. Choosing J such that $Cayley(\Gamma, J)$ is ramnujan provide that $2\lambda/q\Delta$ scale as $\Theta(\Delta^{-1/2})$. That close the case in which there is a linear size layer of nontrivial suggestions. $\qquad \square$

Open problem 5.1.1. *Is there a decoder/tester for 5.1.1 that can take advantage of the high flux induced by the disagreement on the trivial vertices?*

5.2 The Polynomial-Code Is Not w-Robust.

One idea for constructing is to use the polynomial code instead of C_0. This follows from the fact that if one picks a degree strictly greater than $\Delta/2$, then $C_0^\perp \subset C_0$ and therefore one could choose C_z to be the same code defined on the negative vertices of the graph. Here we prove that the dual-tensor code, in that case, is not w-robust, meaning that any such construction should be considered another way of proving the Reduction Lemma.

Claim 5.2.1. *Let C_0 be the $[\Delta, d, \Delta - d]$ polynomial code. Then any code word in $\left(C_0^\perp \otimes C_0^\perp\right)^\perp$ is a polynomial in $F[x, y]$ at degree at most $\Delta + d$*

Proof. Consider base element $C_0 \otimes \mathbb{F}$, denote it by $c = g_i \otimes e_j$. And notice that c has representation in $F[x, y]$ of $\prod_{y' \neq j} (y - y') g_i(x)$. By the fact that $g_i(x) \in C_0$ we have that degree of c is at most $\Delta + \delta$. Hence any element in the subspace of $C_0 \otimes \mathbb{F}$ is a polynomial at degree at most $\Delta + d$. $\square$

Claim 5.2.2. *The dual-tensor polynomial code is not $\Delta^{1+\varepsilon}$-robust, for any $\varepsilon > 0$.*

Proof. Consider the following polynomial

$$P(x, y) = \prod_{i \neq \Delta - 1} (x + iy) = \prod_{i \neq 1} (x - iy)$$

The degree of any monomial is at most $\Delta - 1$, Thus it clear that $P \in \left(C_0^\perp \otimes C_0^\perp\right)^\perp$. And by the fact that for any $x \neq y$ there exists $i \neq 1$ such that $x = iy$ we have that $P(x, y) = 0$. Hence the weight of P is at most $|\{(x, y) : x = y\}| = \Delta$. Yet, for any $x = y$ it follows:

$$P(x, x) = \prod_{i \neq \Delta - 1} (x + ix) = x^{\Delta - 1} \prod_{i \neq \Delta - 1} (1 + i) = (\Delta - 1)! =_\Delta -1 \neq_\Delta 0$$

Put it differently, the diagonal of the matrix, and only it, has only non-zero values, therefore the P cannot be decompised to a sum of $s + t$ such that $s \in C_0 \otimes \mathbb{F}$, $t \in \mathbb{F} \otimes C_0$ and s (t) is supported at most of Δ^ε rows (columns). But, P is a codeword of the dual tensor code at weight less than $\Delta^{1+\varepsilon}$. So, the dual-tensor polynomial code is not a w-robust code, for $w = \Delta^{1+\varepsilon}$. Put it differently, the diagonal of the matrix, and only it, has non-zero values, so the P cannot be decomposed into a sum of $s + t$ such that $s \in C_0 \otimes \mathbb{F}$, $t \in \mathbb{F} \otimes C_0$ and s (t) is supported on at most Δ^ε rows (columns). However, P is a codeword of the dual tensor code at a weight less than $\Delta^{1+\varepsilon}$. Therefore, the dual-tensor polynomial code is not a w-robust code, for $w = \Delta^{1+\varepsilon}, \varepsilon > 0$. $\square$

Open problem 5.2.1. *Is there a quantum CSS code that is also $\Delta^{1\frac{1}{2}} + \varepsilon$-robust, for some $\varepsilon > 0$? And if so, can it be set on the square complex such that all the checks still commute?*

Bibliography

[Ham50]   R. W. Hamming. "Error detecting and error correcting codes". In: *The Bell System Technical Journal* 29.2 (1950), pp. 147–160. DOI: `10.1002/j.1538-7305.1950.tb00463.x`.

[RS60]   Irving S. Reed and Gustave Solomon. "Polynomial Codes Over Certain Finite Fields". In: *Journal of The Society for Industrial and Applied Mathematics* 8 (1960), pp. 300–304.

[Tan81]   R. Tanner. "A recursive approach to low complexity codes". In: *IEEE Transactions on Information Theory* 27.5 (1981), pp. 533–547. DOI: `10.1109/TIT.1981.1056404`.

[Hae95]   Willem H. Haemers. "Interlacing eigenvalues and graphs". In: *Linear Algebra and its Applications* 226-228 (1995). Honoring J.J.Seidel, pp. 593–616. ISSN: 0024-3795. DOI: `https://doi.org/10.1016/0024-3795(95)00199-2`. URL: `https://www.sciencedirect.com/science/article/pii/0024379595001992`.

[Sho95]   Peter W. Shor. "Scheme for reducing decoherence in quantum computer memory". In: *Phys. Rev. A* 52 (4 Oct. 1995), R2493–R2496. DOI: `10.1103/PhysRevA.52.R2493`. URL: `https://link.aps.org/doi/10.1103/PhysRevA.52.R2493`.

[CS96]   A. R. Calderbank and Peter W. Shor. "Good quantum error-correcting codes exist". In: *Physical Review A* 54.2 (Aug. 1996), pp. 1098–1105. DOI: `10.1103/physreva.54.1098`. URL: `https://doi.org/10.1103%2Fphysreva.54.1098`.

[Gro96]   Lov K. Grover. *A fast quantum mechanical algorithm for database search*. 1996. arXiv: `quant-ph/9605043 [quant-ph]`.

[SS96]   M. Sipser and D.A. Spielman. "Expander codes". In: *IEEE Transactions on Information Theory* 42.6 (1996), pp. 1710–1722. DOI: `10.1109/18.556667`.

[Sho97]   Peter W. Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. DOI: `10.1137/s0097539795293172`. URL: `https://doi.org/10.1137%2Fs0097539795293172`.

[AB99]   Dorit Aharonov and Michael Ben-Or. *Fault-Tolerant Quantum Computation With Constant Error Rate*. 1999. arXiv: `quant-ph/9906129 [quant-ph]`.

[AK99]   Ashish Ahuja and Sanjiv Kapoor. *A Quantum Algorithm for finding the Maximum*. 1999. arXiv: `quant-ph/9911082 [quant-ph]`.

[Den+02] Eric Dennis et al. "Topological quantum memory". In: *Journal of Mathematical Physics* 43.9 (Sept. 2002), pp. 4452–4505. DOI: `10.1063/1.1499754`. URL: `https://doi.org/10.1063%2F1.1499754`.

[HLW06]   Shlomo Hoory, Nathan Linial, and Avi Wigderson. "Expander graphs and their applications". In: *Bulletin of the American Mathematical Society* 43.4 (2006), pp. 439–561.

[Got14]   Daniel Gottesman. *Fault-Tolerant Quantum Computation with Constant Overhead*. 2014. arXiv: 1310.2984 [quant-ph].

[TZ14]   Jean-Pierre Tillich and Gilles Zemor. "Quantum LDPC Codes With Positive Rate and Minimum Distance Proportional to the Square Root of the Blocklength". In: *IEEE Transactions on Information Theory* 60.2 (Feb. 2014), pp. 1193–1202. DOI: 10.1109/tit.2013.2292061. URL: https://doi.org/10.1109%2Ftit.2013.2292061.

[LTZ15]   Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zemor. "Quantum Expander Codes". In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, Oct. 2015. DOI: 10.1109/focs.2015.55. URL: https://doi.org/10.1109%2Ffocs.2015.55.

[BGW19]   Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation". In: *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 351–371. ISBN: 9781450372664. URL: https://doi.org/10.1145/3335741.3335756.

[PK21]   Pavel Panteleev and Gleb Kalachev. *Asymptotically Good Quantum and Locally Testable Classical LDPC Codes*. 2021. DOI: 10.48550/ARXIV.2111.03654. URL: https://arxiv.org/abs/2111.03654.

[AF22]   "Maximum distance separable (MDS) code". In: *The Error Correction Zoo*. Ed. by Victor V. Albert and Philippe Faist. 2022. URL: https://errorcorrectionzoo.org/c/mds.

[Din+22] Irit Dinur et al. *Good Locally Testable Codes*. 2022. DOI: 10.48550/ARXIV.2207.11929. URL: https://arxiv.org/abs/2207.11929.

[KP22]   Gleb Kalachev and Pavel Panteleev. *Two-sided Robustly Testable Codes*. 2022. arXiv: 2206.09973 [cs.IT].

[LZ22]   Anthony Leverrier and Gilles Zémor. *Quantum Tanner codes*. 2022. arXiv: 2202.13641 [quant-ph].