

Toric code

小泉勇樹

2023 年 11 月 20 日

0 目次

1	量子誤り訂正への導入	1
1.1	Stabilizer 群による量子誤り訂正	2
1.2	余談：誤り訂正符号の記述は誤り訂正だけで十分か？	3
2	Toric code as a Quantum Error Correction Code	4
2.1	Toric code 模型	4
2.2	V_α, P_β の可換性と完全性	5
2.3	基底状態あるいは符号空間	5
2.4	エラー訂正	7
2.5	異なるグラフ上の Toric code	8
2.6	\mathbb{Z}_d (Modular) toric code	8
2.7	符号空間	9
2.8	余談：トーラス上の toric code は量子メモリとして現実的か？	9

1 量子誤り訂正への導入

Toric code は Kitaev によって提唱された模型であり、量子計算機理論から物性理論において幅広い分野において研究されている。量子計算機分野との関連を見るために、まず量子誤り訂正符号について少し紹介していく。

そもそも量子計算機分野において、量子誤り訂正が何故必要なのか？ という問いについて解説しておく、特定の問題について古典計算機よりも量子計算機における計算アルゴリズムによる解法が、計算量的に有利であることが知られている [1], [2]。ただ、現在の量子計算機は環境によるノイズにより脆弱であるため、正しい計算出力結果をほとんど出すことができない。そのため、量子計算機上において環境から雑音に強い robust な量子状態、量子誤り訂正符号を作ることが必要となる。

資料 [3] においては、古典的な three-bit code および Shor code による導入がされているが、ここでは簡単に一般論を語ることにする。

1.1 Stabilizer 群による量子誤り訂正

ほとんどの量子誤り訂正符号 (Shor code から Toric code まで) は Pauli 群とその自由部分群によってほとんど記述することができる。そのため、まず以下を定義しておく。

定義 1.1: Pauli 群

Pauli 群 \mathcal{P}_n は $\{I, Z, X, Y\} \subset \text{End}(\mathbb{C}^2)$ によって生成される, $\text{End}((\mathbb{C}^2)^{\otimes n})$ の元を集めた群である:

$$\mathcal{P}_n := \{c \times \bigotimes_{i=1}^n U_i \mid c \in \{\pm 1, \pm i\}, U_i \in \{I, X, Y, Z\}\}. \quad (1.1)$$

定義 1.2: Stabilizer 群と Stabilizer 符号

Stabilizer 群 \mathcal{S} とは Pauli 群の自明でない部分可換群である。さらにこの \mathcal{S} によって与えられる Stabilizer 符号 $\mathcal{Q}(\mathcal{S})$ は次のように定義される:

$$\mathcal{Q}(\mathcal{S}) := \text{span}\{|\psi\rangle \in (\mathbb{C}^2)^{\otimes n} \mid \forall M \in \mathcal{S}, M|\psi\rangle = |\psi\rangle\}. \quad (1.2)$$

なぜ符号とは何かを説明しておく。まず、量子計算を行うために、量子情報 $|\psi\rangle \in (\mathbb{C}^2)^{\otimes k}$, $(k \leq n)$ の情報を雑音から保護するのが目的であった。このときに、量子情報の保存に必要な量子ビット数 k よりも多くの量子ビットを使うことによって、情報に冗長性を持たせることができる。つまり、 $(\mathbb{C}^2)^{\otimes k}$ の基底を $(\mathbb{C}^2)^{\otimes n}$ の元に対応させることで、冗長性を持った $(\mathbb{C}^2)^{\otimes k} \simeq \mathcal{Q}(\mathcal{S}) \subset (\mathbb{C}^2)^{\otimes n}$ を構成できる。このような情報を encode する先の部分ベクトル空間を符号空間と呼ぶ。このような情報の encode を定めるルールづけを符号というわけだが、基本的に符号空間を指して符号と呼ぶことが多々ある。

ただ量子誤り訂正を考える上で、冗長性を持たせただけではダメで、雑音によるノイズの影響を除去することを考えたい。最初に簡単なノイズエラーとして X エラーを考える。ここでエラーを検知するにはどうすれば良いかを考える。前提として量子計算機上においては、 $|\psi\rangle \in \mathcal{Q}(\mathcal{S})$ という量子状態に対して $M \in \mathcal{S}$ について、 $|\psi\rangle$ を壊すことなく M の固有値を得ることが可能である。^{*1}。すると符号空間に含まれる量子状態は Stabilizer 演算子に関する固有値は必ず 1 となるはずである。ここで、1 量子ビットに対する X エラーによって $|\psi\rangle \mapsto X_i |\psi\rangle$ という状態になったとしよう^{*2}。このとき、もし $MX_i = -X_iM$ であれば、 $M(X_i |\psi\rangle) = -X_iM |\psi\rangle = -(X_i |\psi\rangle)$ であることから、 $X_i |\psi\rangle$ は M の -1 固有状態であることがわかる。したがって、エラーの起こった状態に対して M に関する測定をすれば、それと非可換な X_i に関するエラーを検知できたことになる。ここで Stabilizer 群に含まれる演算子に関して、測定を行うことをシンドローム測定^{*3}という。

もちろん、生じる全てのエラーを検知することはできないのでできる限り幅広いエラーの検知を考える上で、重みという概念の定義する:

^{*1} 量子状態 $|\psi\rangle$ が壊れるというのは、例えば $|\psi\rangle \propto |0\rangle + |1\rangle$ に対応する状態は射影測定 $\{|0\rangle\langle 0|, |1\rangle\langle 1|\}$ によって容易に異なる状態に変化してしまうことを指す。つまり、射影測定後の状態は $|\psi\rangle$ ではなくなる。

^{*2} 以下、 X_i のように Pauli 演算子の下つき添え字は、 i 番目の量子ビットに作用する演算子であることと約束する。

^{*3} 元々は古典誤り訂正符号からの専門用語。パリティチェックの文脈においてエラーが生じているかを 0, 1 の出力結果から判断するわけだが、この 0, 1 の組み合わせをエラーシンドロームというところから来ている。

定義 1.3: Pauli エラーの重み

$P = c \otimes_{i=1}^N U_i \in \mathcal{P}_N$ の形でかけるエラーを以下 Pauli エラーと呼称することにする．ここでこのエラーの重み $\text{wt}(P)$ とは

$$\text{wt}(P) = |\{U_i \mid U_i \neq I\}| \quad (1.3)$$

である．

ここで Pauli エラーについてのみ考えてきたが、実は次の定理によって Pauli エラーのみ考えれば幅広いエラー検知ができることが可能とする：

定理 1.1: 量子誤り訂正符号に関する線型性

N qubits で構成された量子誤り訂正符号 Q が演算子 A, B でかけるエラーが検知（訂正）できると仮定する．このとき、 Q は A, B の線型結合^aで表されるエラーも検知（訂正）可能である．

^a 本題ではないため量子通信路の概念をすっ飛ばしてエラーを表現しているが、量子系に対する操作は全て量子通信路、つまり CPTP 写像として書くことができ、CPTP 写像の線型結合という意味で線型結合と言っている．

これより、砕いた言い方をすれば Pauli 群の元で複素係数 c を無視したものを集めれば、これは $\text{End}(\mathcal{C}^{\otimes n})$ の基底をなすので任意のエラーは $\text{End}(\mathcal{C}^{\otimes n})$ の元であることを踏まえれば、どのようなエラーであっても Pauli エラーの線型結合に分解できる．

そのため、基本的に量子ビットに対して局所的なエラーが独立に生じる場合を考えて、 $\text{wt}(P) \leq t < N$ である P のエラーを検知できるように符号を構成するのが、量子誤り訂正符号の基本的な考え方となる．

1.2 余談：誤り訂正符号の記述は誤り訂正だけで十分か？

余談ではあるが、資料や初歩的な教科書で述べている量子誤り訂正符号については、エラー検知および訂正のみしか語られておらず、実際には符号上での量子ビットに対する任意の操作、つまりユニバーサルな論理操作などが必要となるが、大抵の場合は記述されることは少ない．資料においては単純な X, Z 操作およびエラー検知しか載っていないことに留意せよ．興味があれば [4], [5] などを見ると良い．

そういう意味で、Shor コードについてはエラー検知、訂正の仕方を学ぶ上では題材としてはわかりやすいが、量子計算のための符号としては要素が足りていない^{*4}．1990 年台に、デコヒーレンスから量子情報を保存できるメモリの構成が可能としたという意味でインパクトのある符号であるが、量子誤り訂正において重要な結果である threshold theorem について繋がらないことが難しいところである．何故 Threshold theorem について特筆すべきかは、その結果を見れば明らかである [4]．

定理 1.2: Threshold Theorem

局所的で独立な Pauli エラーが p の確率で発生するエラーモデルにおいて、重み t の Pauli エラーを訂正できる $[[n, 1, 2t + 1]]$ 量子誤り訂正符号 Q を考える．この時、 Q についてユニバーサルな論理

^{*4} 私の調べた限りだと、ユニバーサルな量子計算を行うための論理操作セットの構成がされていない．

ゲートセットおよび論理 $|0\rangle$ 状態生成, Z 測定, 誤り訂正を行う量子回路が構成できると仮定する. この時, n 個の量子ビットを使って Q を構成するわけだが, 新しくできた論理量子ビット n 個を用いて新しい論理量子ビットを構成することを考える^a.

L 回このような操作を繰り返すことによって構成できる, $[[n^L, 1, (2t+1)^L]]$ 符号についてある閾値 p_T が存在して, 元の物理ビットのエラー値が $p < p_T$ ならば, 任意の正数 ε に対して, ある L が存在して $[[n^L, 1, (2t+1)^L]]$ 符号による論理量子回路の出力結果の確率分布は, エラーの全くない計算結果の確率分布との差を ε 以下にできる.

^a このような論理量子ビットの構成の仕方を concatenated code と呼ぶ

ここで p_T は Q の論理操作の構成方法にのみ依存するため, エラー率 p_T 未満の量子ビットを十分用意できれば, エラー耐性のついた量子計算すなわち Fault Tolerant Quantum Computation (FTQC) を実行できることを示している.^{*5}

2 Toric code as a Quantum Error Correction Code

Toric code は簡潔に述べれば Kitaev によって提唱されたトーラス上のスピン模型であり, 厳密に解くことができる模型である [6]. 量子情報では基底状態 (これが前に述べた符号空間の元に対応するわけだが) に対して外部磁場の摂動に対して robust な論理量子ビットができ^{*6}, かつ局所的なシンドローム測定により誤り訂正が可能という点で画期的である. 次の章でも見るように物性物理の文脈においては, 「トポロジカル秩序相」の最も基本的な例となっている.

2.1 Toric code 模型

まず, トーラス上のスピン配位を考えるために $N_x \times N_y$ 正方格子の辺にスピンの位置している状況を考える. ここで正方格子について, 周期的境界条件をつければトーラス上に $2 \times N_x N_y$ ^{*7} のスピンの位置することがわかる. これより, 全系の Hilbert 空間の次元は $N := 2^{N_x N_y}$ として, 2^N となる. 以下, このような Hilbert 空間を \mathcal{H} と書き, これに作用する演算子集合を $\mathcal{L}(\mathcal{H})$ と書く.

このようなスピン系に対して, Toric code の Hamiltonian は次で定義する:

定義 2.1: Torus 上の Toric code Hamiltonian

$$H = - \sum_{\alpha} V_{\alpha} - \sum_{\beta} P_{\beta}. \quad (2.1)$$

ここで V_{α} とは vertex 演算子と呼ばれ

$$V_{\alpha} := \prod_{i \in \partial \alpha} \sigma_z^{(i)} \quad (2.2)$$

^{*5} もちろんエラーモデルが正しく, 物理量子ビットが長時間動かせるという仮定のもとであるが.

^{*6} 誤り訂正符号の文脈で言えば, 符号距離が大きい

^{*7} 2 についているのは, 頂点の数は $N_x N_y$ であるが最近接頂点との辺にスピンの位置するので $4N_x N_y / 2$ と求められるからである.

とかける．ここで $\partial\alpha$ とは α に近接する辺のスピンを考える^a．また， P_β は Plaquette 演算子であり，

$$P_\beta := \prod_{j \in \partial\beta} \sigma_x^{(j)}. \quad (2.3)$$

である．

^a ホモロジーでの境界準同型を取ることに一致する意味で

Pauli 演算子の性質から

$$V_\alpha^2 = 1, \quad P_\beta^2 = 1 \quad (2.4)$$

であることは明らかである．

2.2 V_α, P_β の可換性と完全性

当然であるが σ_z, σ_x は自分自身と可換であるから，

$$[V_\alpha, V_{\alpha'}] = 0, \quad [P_\beta, P_{\beta'}] = 0. \quad (2.5)$$

それに加えて V_α, P_β についても実は可換である．これは，考えている α, β が離れているすなわち $\partial\alpha \cap \partial\beta = \emptyset$ であれば明らかである．それに加えて， β の頂点が α であった場合でも V_α, P_β で考えているスピン配位 j は $j \in \partial\alpha \setminus \partial\beta$ (or $\partial\beta \setminus \partial\alpha$) あるいは $j \in \partial\alpha \cap \partial\beta$ のいずれかである．ここで $j \in \partial\alpha \cap \partial\beta$ になるものが非可換性を生み出す可能性があるが，必ず $j \in \partial\alpha \cap \partial\beta$ なる j は必ず 2 つ現れるため，この場合でも V_α, P_β は可換である．従って，

$$[V_\alpha, P_\beta] = 0. \quad (2.6)$$

ここで V_α, P_β は Pauli 演算子の積によって構成されるため固有値は ± 1 である．この時，ある α について V_α の固有値が異なる二つの状態 $|\psi\rangle, |\psi'\rangle$ は直交するので， V_α の固有値によって Hilbert 空間が分割されていくわけである．ここで α, β の総数は $2N_x N_y$ であるから，もし V_α, P_β の固有値の組みで Hilbert 空間の元が一意に定まるのなら， V_α, P_β はこのようなトーラス上の Hilbert 空間を完全に記述できることになる．

ただ，この命題は偽である．というのも，式 (2.4) によって固有値の自由度が制限されるからである．従って， α, β に関して固有値の自由度はそれぞれ $N_x N_y - 1$ となることから，固有値の選び方は $2^{2(N_x N_y - 1)}$ となる．しかし，この残された自由度が強固な符号空間を作る上で重要なものとなる．

2.3 基底状態あるいは符号空間

ここまで，Toric code Hamiltonian の演算子に関する性質について見てきたが，実は $\{V_\alpha, P_\beta\}_{\alpha \in A, \beta \in B}$ ^{*8} は Pauli 群 $\mathcal{P}_{2N_x N_y}$ の自明でない部分可換群 $\mathcal{S}_{\text{toric}}$ になっている．つまり，これによって stabilizer 符号 $\mathcal{Q}_{\text{toric}} := \mathcal{Q}(\mathcal{S}_{\text{toric}})$ を作れると同時に，この $|\psi\rangle \in \mathcal{Q}_{\text{toric}}$ は全ての stabilizer 演算子 V_α, P_β の固有値が 1 となるように定めていることから式 (2.1) の基底状態となっていることがわかる．これが，量子誤り訂正符号 Toric code である．

^{*8} A, B はそれぞれトーラス上の全ての頂点，矩形を集めた集合

では、実際 $|\psi\rangle$ がどのような形をしているかを考えた時に、次の命題が成立する：

命題 2.1: Toric code の表現

$|\psi\rangle \in \mathcal{Q}_{\text{Toric}}$ は \uparrow のスピンを辿った時にできる曲線を l として、それとスピン配位を同一視して l で書いた時、次の形でかける。

$$|\psi\rangle \propto \sum_{[l]} |l\rangle \quad (2.7)$$

ここで $H_1(\text{Torus})$ は考えているトーラス上をチェーン複体とみなした時の、ホモロジー群とし、 $[l]$ はその意味での同値類を取ったもの。平易に言えば、ループのみを有するスピン配位を持つ $|l\rangle$ から B_p をかけていくと得られるスピン配位を全て集めたものと考えて良い。

証明 $|\psi\rangle$ が式 (2.7) の形でかければ、 $|\psi\rangle \in \mathcal{Q}_{\text{Toric}}$ であることを示す。 $V_\alpha |\psi\rangle = |\psi\rangle$ であることは、 $|l\rangle$ に含まれる \uparrow のスピン配位を辿ってできる曲線は閉曲線であることから、明らかである。さらに $P_\beta |l\rangle = |l'\rangle$ であるが、 $l' \in [l]$ であることから、 $P_\beta |\psi\rangle = |\psi\rangle$ である。 ■

ここで $[l]$ 、すなわちあるループから出発して別のループに遷移する時を考えるときに以下の4種類のループから始めれば $\mathcal{Q}_{\text{toric}}$ に含まれるスピン配位を全て得ることができる^{*9}。さらに、 $\mathcal{Q}_{\text{toric}}$ の次元は4であることから、このホモロジー類のサイクルの選び方の総数と一致している。そのため、あるサイクル l から別のサイクル l' に写す演算子を考えれば、Toric code 内である論理状態から別の論理状態へと写す論理操作 X, Z を定義できることになる。そして、それは torus 上の非自明なループを作る次の操作に一致する：

$$X := \prod_{i \in l} \sigma_x^{(i)} \quad (2.8)$$

$$Z := \prod_{i \in l'} \sigma_z^{(i)} \quad (2.9)$$

ここで l, l' はそれぞれトーラス上の元の格子上の非自明なループ、双対格子上の非自明なループを表す。ホモロジー群の言葉で言えば、非自明な 1-cycle, 1-cocycle である。これより、ループのない状態を論理量子ビットの状態として $|0, 0\rangle = |\psi_{ee}\rangle$ ^{*10} として、トーラスの穴の周りを一周するループを l_0 、トーラスの穴を一周するループを l_1 とすることで、論理状態として

$$|1, 0\rangle = |\psi_{oe}\rangle = X_0 |0, 0\rangle := \prod_{i \in l_0} \sigma_x^{(i)} |0, 0\rangle, \quad (2.10)$$

$$|0, 1\rangle = |\psi_{eo}\rangle = X_1 |0, 0\rangle := \prod_{i \in l_1} \sigma_x^{(i)} |0, 0\rangle, \quad (2.11)$$

とすることに定義する。このように定めた X_0, X_1 に対して論理操作の Z_0, Z_1 は

$$X_i Z_j = (-1)^{(i+j)+1} Z_j X_i \quad (i, j \in \{0, 1\}) \quad (2.12)$$

という関係式が成り立って欲しいので、

$$Z_0 = \prod_{j \in l'_1} \sigma_z^{(j)}, \quad (2.13)$$

$$Z_1 = \prod_{j \in l'_0} \sigma_z^{(j)} \quad (2.14)$$

^{*9} これは $H(T) = \mathbb{Z} \times \mathbb{Z}$ であることに対応する。

^{*10} 資料の記法と、論理量子ビットとして自然な記法を同時に書いている。どちらの記法を使っても良い

と選べば論理 Pauli 操作に求められる代数関係が自然と得られる。ただし、 l'_1 は双対格子でトーラスの穴を一周するループ、 l'_0 はトーラスの穴の周りを一周するループである。これにより、 $\mathcal{Q}_{\text{toric}}$ の基底を得ることができたので、 $\forall |\psi\rangle \in \mathcal{Q}_{\text{toric}}$ は次のように張られる：

$$|\psi\rangle = c_{00}|0,0\rangle + c_{10}|1,0\rangle + c_{01}|0,1\rangle + c_{11}|1,1\rangle \quad (\forall c_{ij} \in \mathbb{C}). \quad (2.15)$$

ただし $\sum |c_{ij}|^2 = 1$ である。

このように考えていくことで、 V_α, P_β で生成される演算子集合 $\mathcal{S}_{\text{toric}}$ はそれぞれ 2-boundaries, 0-coboundaries であることを意味し、 $\mathcal{L}(\mathcal{Q}_{\text{toric}})$ は 1-homologies あるいは 1-cohomologies に対応することがわかる。また、 A_s, B_p と可換な演算子集合^{*11} は非自明なループに限らない閉曲線 l, l' について

$$\prod_{i \in l} \sigma_x^{(i)}, \quad \prod_{j \in l'} \sigma_z^{(j)} \quad (2.16)$$

という形をとる演算子によって生成される集合であり、これが 1-cycle, 1-cocycle に対応する。

2.4 エラー訂正

Toric code の論理状態については上で見た通りなので、エラー訂正について考えていく。まず、どこかの辺 i において σ_x^i エラーが生じたと仮定する。これについて、 $i \in \partial\alpha$ なる α について $\sigma_x^{(i)} V_\alpha = -V_\alpha \sigma_\alpha^{(i)}$ であるから、 V_α に関して測定を行うことで Fig.25.9 のように、エラーが生じている辺の二つの頂点での測定が -1 を返すため、それに接している辺でエラーが生じていることがわかる。

二つ以上のエラーが生じている場合でも簡単である。Fig.25.10(左) のように隣接する二つの辺でエラーが生じた時には、シンδροーム測定においてエラーを返すのはちょうど黄色で記された頂点のみであり、エラーを示した頂点二つを辿ってできる曲線について σ_x^i を作用させていけばエラー訂正が完了となる。三つ以上のエラーが生じた場合でも、同様である。ただし、シンδροーム値から適切なエラー訂正操作を推定する手続きは、格子上における最小重み完全マッチング (Minimum-weight perfect matching, MWPM) 問題に帰着される [6]。

特筆すべきことは、もしエラーが自明なループに沿って生じた場合はエラーは検知できない、どちらかといえば何も影響を与えないことである。これは P_β によって生成される演算子と等価になることから、符号空間の元に対して何も変化を与えないので明らかである。

では、この σ_x がどのような検知できないエラーをもたらすのか、それは符号空間の元を別の符号空間の元に移してしまう時である。それはちょうど Fig.25.11 のように、トーラス上の格子における非自明なループに沿ってエラーが生じた場合である。これは X_0, X_1 と区別がつかず、エラー検知ができないのにエラーを生じさせる。例えば、図のようなエラーが生じた時には X_0 を作用させた時と同じなので

$$c_{00}|0,0\rangle + c_{10}|1,0\rangle + c_{01}|0,1\rangle + c_{11}|1,1\rangle \mapsto c_{00}|\textcolor{red}{1},0\rangle + c_{10}|\textcolor{red}{0},0\rangle + c_{01}|\textcolor{red}{1},1\rangle + c_{11}|\textcolor{red}{0},1\rangle \quad (2.17)$$

となってしまう。

同様に $\sigma_z^{(j)}$ に関するエラーも上の議論を双対格子上的の曲線について考え、 V_α, P_β を逆にして読み替えばほとんど同じ議論となる^{*12}。唯一異なる点とすれば非自明なループによって Z_0 によって、量子状態は次の

^{*11} Stabilizer 群に対して、これは Centralizer あるいは中心化群と呼ばれる

^{*12} X, Z 基底が等価であることを考えれば、ほとんど自明なことであるが。

ように攪乱を受けるということである：

$$c_{00}|0,0\rangle + c_{10}|1,0\rangle + c_{01}|0,1\rangle + c_{11}|11\rangle \mapsto c_{00}|0,0\rangle - c_{10}|1,0\rangle + c_{01}|0,1\rangle - c_{11}|11\rangle \quad (2.18)$$

これを防ぐには, torus 自体を大きくして非自明なループにそってエラーが生じる可能性をほとんどゼロにすれば良い. このような考え方によって, toric code は重み $\min(N_x, N_y)$ の Pauli エラーに対しては対処できない場合が生じることがわかる. これより, Toric code とは $[[2N_xN_y, 2, 2\min(N_x, N_y) + 1]]$ 量子誤り訂正符号であることがわかる.

2.5 異なるグラフ上の Toric code

Toric code の場合独立な vertex, plaquette 演算子の数は $N_xN_y - 1, N_xN_y - 1$ というふう考えたわけだが, 一般のグラフ (V, E, F) について, 同じように全ての E についてスピンを配置し V, F の境界準同型によって E の元をとってそれに関する vertex 演算子, plaquette 演算子を考える. 同様の議論によって, このグラフにおける Hilbert 空間の基底状態の次元は次のように与えられる：

$$2^{E-(V-1+F-1)} = 4^{E-(F+V)} = 4^g \quad (2.19)$$

2.6 \mathbb{Z}_d (Modular) toric code

これまでの, 誤り訂正符号が qubit, すなわち符号空間の最小単位が \mathbb{C}^2 になるように構成を進めてきたが, 以下では Gottesmann によって提唱された d -次元量子系による量子計算, すなわち qudit による誤り耐性付き量子計算 [7] を toric code によって構成するためにはどのような模型を用意すれば良いかを説明する. qudit 以外にも \mathbb{Z}_d ゲージ理論にも繋がる話なので, 模型として知っておいて損はないと思われる.

Gottesman は, \mathbb{C}^d の基底 $\{|i\rangle\}_{i \in \{0, \dots, d-1\}}$ に対して次の性質を満たすものを qudit-Pauli 演算子として定義した [7]:

$$X_d|j\rangle = |j+1\rangle, \quad Z_d|j\rangle = \omega^j|j\rangle \quad (\text{where, } X_dZ_d = \omega^{-1}Z_dX_d) \quad (2.20)$$

これによって生成される高次元 Pauli 群, それに付随する Stabilizer 群を考えることによって d 次元量子誤り訂正符号を作ることができる^{*13}.

このような議論を toric code に適用するために, グラフ (V, E, F) に対して E に d 次元系, qudit, あるいは d 自由度を持つ矢印^{*14}をおく. ただ注意しなければならないのは, E には向きがついていることに留意せよ.

ここで全系の qudit 配位が $|s\rangle = \otimes |s_i\rangle$ であるとして, V_α に対応するような演算子として電流吸い込みを表す演算子 Q_α を次のように定義する：

$$Q_\alpha|s\rangle := \left(\sum_{i \in \partial\alpha} s_i \mod d \right) |s\rangle \quad (2.21)$$

ここで $\partial\alpha$ は向き付けも考慮した境界をとっている. ここで全系の条件として次の条件を課す：

$$\sum_\alpha Q_\alpha = 0, \quad \text{or} \quad \prod_\alpha e^{i2\pi Q_\alpha/N} = 1. \quad (2.22)$$

^{*13} 基本的に d が素数である場合を考えるとが多い. なぜだかは知らないが.

^{*14} 色んな表現をしたがどれでも良い. 想像しやすいのを撰べば良い.

また、 P_β に対応するような演算子として次を定義する：

$$A_\beta |s\rangle := \bigotimes_{(\mp 1) \cdot j \in \partial \beta} |s_j \pm 1\rangle \otimes (\dots) \quad (\text{複合同順}). \quad (2.23)$$

すなわち、 A_β は矩形 β の向きづけられた（つまり反時計周りに）境界をとった時に、その境界の向きが元の向きづけられた辺と同じ方向ならその辺に位置する qudit の準位を下げ、逆方向なら準位を上げることを意味する。ここで、 A_β についても次のような条件が課されている：

$$\prod_\beta A_\beta = 1 \quad (2.24)$$

これはトーラス上での格子なら、一つの辺は二つの矩形によって共有されることから、その二つの矩形に関する A_β による qudit の準位は変わらないことから明らかである。

2.7 符号空間

Q_α, A_β は可換であることを見る。二次元上のグラフを考えているので、二つの辺 $i, j \in \partial \alpha \cap \partial \beta \neq \emptyset$ が存在する場合のみを考えれば十分である。仮に i, j が α に対する向きが同じ $c = \pm 1$ である場合は、 Q_α, A_β の $|s_i\rangle |s_j\rangle$ に対する作用は

$$Q_\alpha A_\beta |s_i\rangle |s_j\rangle = Q_\alpha |s_i \pm 1\rangle |s_j \mp 1\rangle = c(s_i + s_j \bmod d) |s_i \pm 1\rangle |s_j \mp 1\rangle, \quad (2.25)$$

$$A_\beta Q_\alpha |s_i\rangle |s_j\rangle = A_\beta c(s_i + s_j \bmod d) |s_i\rangle |s_j\rangle = c(s_i + s_j \bmod d) |s_i \pm 1\rangle |s_j \mp 1\rangle. \quad (2.26)$$

より、この時は可換。次に i を α に対する向きが出ていく方向の辺として、 j をその逆向きの辺とした場合、

$$Q_\alpha A_\beta |s_i\rangle |s_j\rangle = Q_\alpha |s_i \pm 1\rangle |s_j \pm 1\rangle = (s_i - s_j \bmod d) |s_i \pm 1\rangle |s_j \pm 1\rangle, \quad (2.27)$$

$$A_\beta Q_\alpha |s_i\rangle |s_j\rangle = A_\beta (s_i - s_j \bmod d) |s_i\rangle |s_j\rangle = (s_i - s_j \bmod d) |s_i \pm 1\rangle |s_j \mp 1\rangle. \quad (2.28)$$

この時も可換。以上より、 Q_α, A_β は可換である。

これより Toric code と同様に $e^{i2\pi Q_\alpha/d}, A_\beta$ を stabilizer 群として考えれば stabilizer 符号を構成することができる。この場合でも、符号空間の元 $|\psi\rangle$ に対して課される条件は $\forall \alpha \in V, e^{i2\pi Q_\alpha/d} |\psi\rangle = |\psi\rangle$ かつ $\forall \beta, A_\beta |\psi\rangle = |\psi\rangle$ である。

エラーの修正の仕方について \mathbb{Z}_2 toric code とほとんど同じなので、省略する。シンδροーム測定によって得られた $Q_\alpha = \pm m$ を欠陥として、その欠陥を対消滅させようとして欠陥を移動させるように演算子を作用させていけば良い。この考え方は次の章である準粒子励起に繋がる考え方である。

2.8 余談：トーラス上の toric code は量子メモリとして現実的か？

この節で、Kitaev の提唱した量子メモリとしてトーラス上の toric code を見てきたが、ここではユニバーサルゲートセットについては触れなかったが効率的なユニバーサルセットが存在し、誤り訂正のデコーダも MWPM に帰着できるという FTQC 可能な量子誤り訂正符号として革新的であるはあるものの、「理論家の夢は実験家の悪夢」とはいうように、どのような方式の量子ビットであってもトーラス上に配置し、局所的とはいえシンδροーム測定を行い、量子計算を行うために、量子制御を行うことはまさに実現を考える上では悪夢以外の何物でもないだろう。そのため、toric code から派生した符号で現実的に実現可能な量子誤り訂正符号

として、表面符号というものを考えている。興味があれば、表面符号の理論に関する文献としては [8], 実験の実装として [9] をみよ。

■ 2 参考文献

- [1] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134, 1994. <https://ieeexplore.ieee.org/document/365700>.
- [2] A Yu Kitaev. Quantum measurements and the abelian stabilizer problem. arXiv preprint quant-ph/9511026, 1995. <https://arxiv.org/abs/quant-ph/9511026>.
- [3] Steven H Simon. Topological quantum: Lecture notes and proto-book. Unpublished prototype.[online] Available at: <http://www-thphys.physics.ox.ac.uk/people/SteveSimon>, Vol. 26, p. 35, 2020.
- [4] Daniel Gottesman. An introduction to quantum error correction and fault-tolerant quantum computation. In Quantum information science and its contributions to mathematics, Proceedings of Symposia in Applied Mathematics, Vol. 68, pp. 13–58, 2010. <https://arxiv.org/abs/0904.2557>.
- [5] Michael A Nielsen and Isaac L Chuang. Quantum computation and quantum information. Cambridge university press, 2002.
- [6] A Yu Kitaev. Fault-tolerant quantum computation by anyons. Annals of physics, Vol. 303, No. 1, pp. 2–30, 2003. <https://www.sciencedirect.com/science/article/pii/S0003491602000180>.
- [7] Daniel Gottesman. Fault-tolerant quantum computation with higher-dimensional systems. In NASA International Conference on Quantum Computing and Quantum Communications, pp. 302–313. Springer, 1998. <https://www.sciencedirect.com/science/article/pii/S0960077998002185?via%3Dihub>.
- [8] Austin G Fowler, Matteo Mariantoni, John M Martinis, and Andrew N Cleland. Surface codes: Towards practical large-scale quantum computation. Physical Review A, Vol. 86, No. 3, p. 032324, 2012. <https://journals.aps.org/pr/abstract/10.1103/PhysRevA.86.032324>.
- [9] Suppressing quantum errors by scaling a surface code logical qubit. Nature, Vol. 614, No. 7949, pp. 676–681, 2023. <https://www.nature.com/articles/s41586-022-05434-1>.