From stabiliser codes to topological quantum computing

based on a joint project w. Ulrich Krähmer

01.06.2026

arXiv:2506.09249

Sebastian Halbig
Sebastian.Halbig@uni-marburg.de

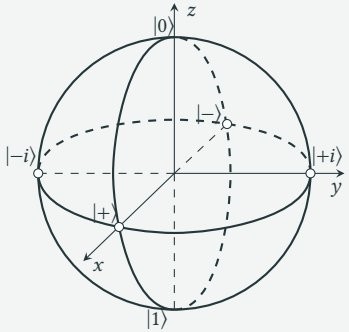
University of Marburg

Why quantum computing

Classical: Bits and Boolean logic.

Quantum: Qubits (or more general qudits) + unitary transformations and measurements.

Quantum states: States of a single qubit:



Classical states: Only north and south pole.

Why quantum computing?

This gives us in theory greater computational power[1].

Quantum advantage

Quantum computing can (conjecturally) solve certain problems much faster than classical computers.

Examples:

- Search: Grover's algorithm
- Cryptography: Shor's algorithm
- Rep. theory: Computing plethysms

Computing plethysms

[1] It is an open problem whether $P \subsetneq BQP$.

What is the issue?

Observation: All these algorithms are written with idealised *logical* qubits in mind.
Main issue: To get logical qubits we need error-correction.

This is hard!

Theorem (No-cloning theorem)

There is no unitary map $U\colon\ \mathbb{C}^2 \otimes \mathbb{C}^2 \longrightarrow \mathbb{C}^2 \otimes \mathbb{C}^2$ *satisfying*

$$U(a \otimes b) = a \otimes a \qquad \text{for all } a, b \in \mathbb{C}^2.$$

The stabiliser formalism: Error-correction based on the representation theory of finite groups.

Multi-qubit systems

We model a system with $n \in \mathbb{N}$ qubits by the tensor product $(\mathbb{C}^2)^{\otimes n}$.

Write the standard basis of $\mathbb{C}^2$ as

$$|0\rangle \stackrel{\text{def}}{=} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \in \mathbb{C}^2, \qquad |1\rangle \stackrel{\text{def}}{=} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \in \mathbb{C}^2.$$

We extend this to the 4-dimensional space $\mathbb{C}^2 \otimes \mathbb{C}^2$ and chose an orthonormal basis which we write as

$$|00\rangle, \qquad |01\rangle, \qquad |10\rangle, \qquad |11\rangle.$$

More generally, $(\mathbb{C}^2)^{\otimes n}$ has the *computational* basis

$$(|s\rangle)_{s \in \{0,1\}^m}.$$

Pauli matrices

Pauli matrices:

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Pauli matrices act on qubits. For example $\sigma_x|0\rangle = |1\rangle$.

Pauli group: G_1 is the group generated by the Pauli matrices. Think: All possible finite products of Pauli matrices.

Pauli n-group: Let $1 \leq j \leq n$ and $\sigma \in G_1$ and consider the matrix $M = I_2^{\otimes j-1} \otimes \sigma \otimes_2^{n-j}$. Then

$$M|s_1 \dots s_{j-1} s_j s_{j+1} \dots s_n\rangle = |s_1 \dots s_{j-1} \sigma(s_j) s_{j+1} \dots s_n\rangle.$$

The *Pauli n-group* G_n is the group generated by all matrices $I_2^{\otimes j-1} \otimes \sigma \otimes I_2^{\otimes n-j}$, where $\sigma \in G_1$, $1 \leq j \leq n$.

Stabiliser codes

Definition

Let $\mathrm{Prot}(S)$ be a 2^k-dimensional subspace of $(\mathbb{C}^2)^{\otimes n}$ such that the stabiliser group

$$G_n \supset S \stackrel{\text{def}}{=} \{s \in G_n \mid s(v) = v \text{ for all } v \in \mathrm{Prot}(S)\}$$

is abelian. Then, we call $\mathrm{Prot}(S)$ an $[n, k]$ *stabiliser code*.

Idea: $\mathrm{Prot}(S)$ encodes the logical qubits.

G_n spans the vector space $\mathrm{Mat}_{2^n}(\mathbb{C})$. $\rightsquigarrow$ study potential "errors" $E \in G_n$:

trivial: E acts trivially on $\mathrm{Prot}(S) \implies E \in S$.

correctable: E does anticommute with some $s \in S$.

logical: $E \in Z(S) \setminus S$: E acts non-trivially on S.
Here: $Z(S) = \{g \in G_n \mid gs = sg\}$.

States and errors

Definition

Let $v \in (\mathbb{C}^2)^{\otimes n}$ be non-zero. The corresponding state is the set

$$[|v\rangle] \stackrel{\text{def}}{=} \{\lambda|v\rangle \mid \lambda \in \mathbb{C}\}.$$

Equivalent: A state is the orthogonal projection into the subspace spanned by $|v\rangle$:

$$\rho_{|v\rangle} \stackrel{\text{def}}{=} |v\rangle\langle v| \colon\ (\mathbb{C}^2)^{\otimes n} \longrightarrow (\mathbb{C}^2)^{\otimes n}, \qquad |w\rangle \longmapsto \langle v \,|\, w\rangle_{\text{std}} \cdot |v\rangle.$$

Changing a state: Modelled by a *unitary matrix* $E \in U(2^n) \subset \mathrm{Mat}_{2^n}(\mathbb{C})$ (so $E^\dagger E = I = EE^\dagger$):

$$[|v\rangle] \longmapsto [E|v\rangle] \quad \leftrightsquigarrow \quad \rho_{|v\rangle} \longmapsto E\rho_{|v\rangle}E^\dagger$$

Correcting an error: syndrome measurement

Let $E \in U(2^n)$ be some error and $s \in S$. We decompose

$$E = E_C + E_A, \qquad \text{with } E_C = \tfrac{1}{2}(E + sE) \text{ and } A = \tfrac{1}{2}(E - sE).$$

Fact: All Pauli matrices have order 2 and for all $a, b \in G_n$ we have $ab = \pm ba$. Therefore, we have $s^2 = \mathrm{id}$.

Consequence: $sE_c = E_c s$ and $sE_A = -E_A s$.

Correcting an error: syndrome measurement

Let ρ be our initial state and set $\tau_0 = E\rho E^\dagger$.
Reminder: $|0\rangle = \left(\begin{smallmatrix}1\\0\end{smallmatrix}\right)$ and $|1\rangle = \left(\begin{smallmatrix}0\\1\end{smallmatrix}\right)$.

- We add an *ancilla qubit* to get $\tau_1 = |0\rangle\langle 0| \otimes \tau_0 = \left(\begin{smallmatrix}\tau_0 & 0\\ 0 & 0\end{smallmatrix}\right)$.
- Applying the Hadamard gate $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix}1 & 1\\ 1 & -1\end{smallmatrix}\right) \otimes \mathrm{id}$ yields $\tau_2 = H\tau_1 H^\dagger = \frac{1}{2}\left(\begin{smallmatrix}\tau_0 & \tau_0\\ \tau_0 & \tau_0\end{smallmatrix}\right)$
- We apply the controlled-s-gate $U_s = |0\rangle\langle 0| \otimes \mathrm{id} + |1\rangle\langle 1| \otimes s = \left(\begin{smallmatrix}I & 0\\ 0 & s\end{smallmatrix}\right)$ and get

$$\tau_3 = U_s \tau_2 U_s^\dagger = \frac{1}{2}\begin{pmatrix}\tau_0 & \tau_0 s^\dagger\\ s\tau_0 & s\tau_0 s^\dagger\end{pmatrix}$$

Correcting an error: syndrome measurement

- We applying the Hadamard $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right) \otimes \mathrm{id}$ to $\tau_3 = U_s \tau_2 U_s^\dagger = \frac{1}{2}\left(\begin{smallmatrix} \tau_0 & \tau_0 s^\dagger \\ s\tau_0 & s\tau_0 s^\dagger \end{smallmatrix}\right)$:

$$\begin{aligned}\tau_4 = H\tau_3 H^\dagger &= \frac{1}{4}\begin{pmatrix} \tau_0 + \tau_0 s^\dagger + s\tau_0 + s\tau_0 s^\dagger & \dots \\ \dots & \tau_0 - \tau_0 s^\dagger - s\tau_0 + s\tau_0 s^\dagger \end{pmatrix} \\ &= \frac{1}{4}\begin{pmatrix} (\mathrm{id}+s)\tau_0(\mathrm{id}+s)^\dagger & \dots \\ \dots & (\mathrm{id}-s)\tau_0(\mathrm{id}-s)^\dagger \end{pmatrix} = \begin{pmatrix} E_C \rho E_C^\dagger & \dots \\ \dots & E_A \rho E_A^\dagger \end{pmatrix}.\end{aligned}$$

Correcting an error: syndrome measurement

- Measuring the ancilla qubit leads to the probabilities

$$|0\rangle : \quad \mathrm{tr}(E_C \rho E_C^\dagger), \qquad |1\rangle : \quad \mathrm{tr}(E_A \rho E_A^\dagger).$$

- New state:

$$\text{if } |0\rangle : \frac{E_C \rho E_C^\dagger}{\mathrm{tr}(E_C \rho E_C^\dagger)}, \qquad \text{if } |1\rangle : \frac{E_A \rho E_A^\dagger}{\mathrm{tr}(E_A \rho E_A^\dagger)}.$$

Correcting an error: syndrome measurement

Upshot

We repeat the syndrome measurement for a generating set $s_1, \dots, s_{n-k}$ of S. The *syndrome* of the resulting error $E' \in \mathrm{U}(2^n)$ is $(m_1, \dots, m_{n-k}) \in \mathbb{F}_2^{n-k}$ such that

$$s_i E' s_i = (-1)^{m_i} E', \qquad 1 \leq i \leq n-k.$$

Observation: $r, t \in G_n$ such that $s_i r s_i = (-1)^{m_i} r$ and $s_i t s_i = (-1)^{m_i} t$ for all $1 \leq i \leq n-k$. Set $z = r^{-1}t$. Then $rz = t$ and

$$s_i z s_i = s_i r^{-1} s_i s_i t s_i = (-1)^{2m_i} r^{-1} t = r^{-1} t = z.$$

Therefore $z \in Z(S)$.

Correcting an error

For simplicity: We assume $E' = r \in G_n$.

Error-correction procedure

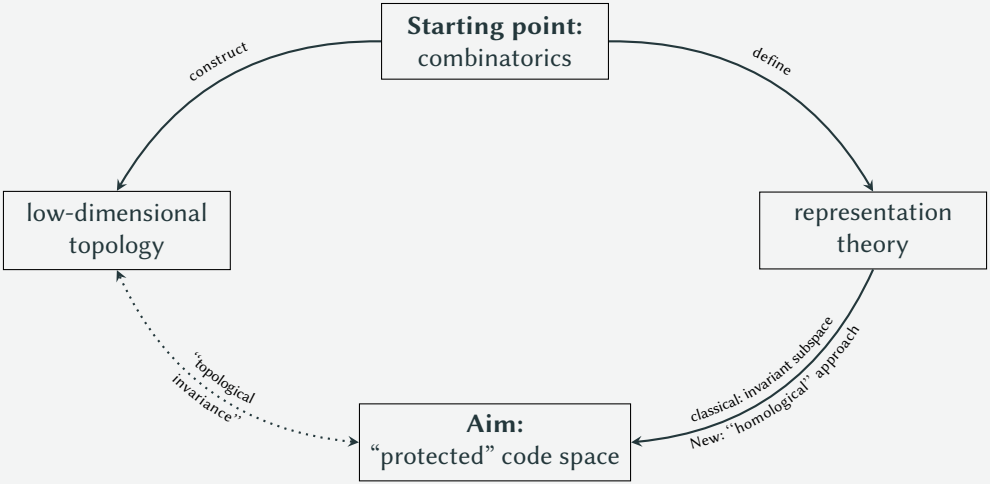
- **initial state:** ρ,
- **state after error:** $E\rho E^\dagger$,
- **state after syndrome measurement:** $E'\rho E'^\dagger$.
- **error-correction:** Chose $t \in G_n$ such that t has the same syndrome as E'. Then $E' = tz$ for some $z \in Z(S)$ and

$$t^\dagger E'\rho E'^\dagger t = z^\dagger E'^\dagger E'\rho E' E'^\dagger z = z^\dagger \rho z.$$

- Classical error correction strategies allow us to maximise the likelihood that $z \in S$ and so $z^\dagger \rho z = \rho$.

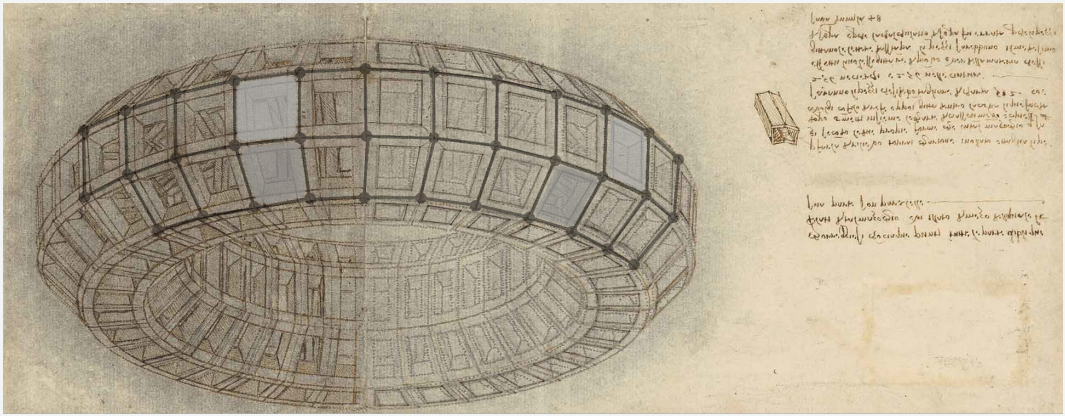
Kitaev's toric code as a stabiliser code.

The big picture



The toric code: the Hilbert space

Fix $n \in \mathbb{N}$ and consider a $n \times n$ lattice embedded in the torus $\mathbb{T}$:



A “mazzocchio” drawn by L. Da Vinci. Overlays highlight vertices, edges, and faces.

Toric code: the Hilbert space

Goal: Build a quantum spin model where particles are associated to the edges of the graph.

Step 1: Define the *extended Hilbert space* $\mathbb{M}_n = \otimes_{i,j=1}^{n} \mathbb{C}^2_{\text{vert}} \otimes \mathbb{C}^2_{\text{hor}}$.

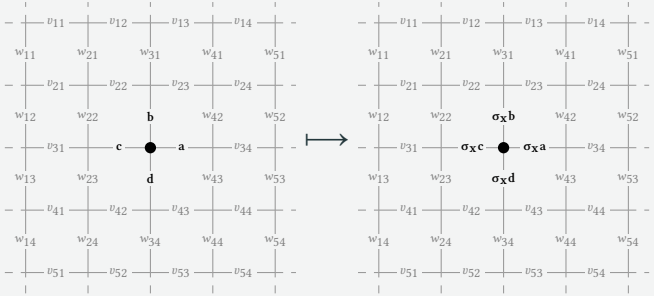
(We associate to each vertex of the graph one *vert*ical edge going "up" and one *hor*izontal edge going to the right.)

Toric code: electric interaction

Step 2a: (Electric) nearest neighbour interaction modelled using the vertices V.

Set $\sigma_x = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. Given a *vertex* (point in the lattice) v define

$$A_v \colon \mathbb{M}_n \longrightarrow \mathbb{M}_n.$$

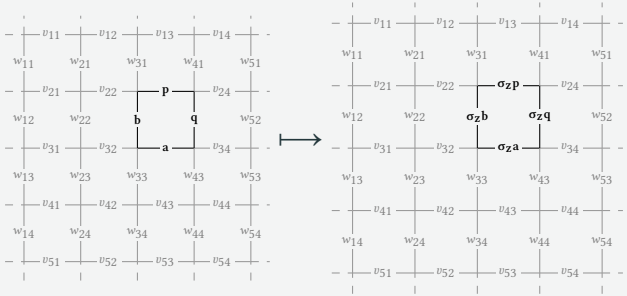


Toric code: magnetic interaction

Step 2b: (Magnetic) nearest neighbour interaction modelled using the faces F.

Set $\sigma_z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$. Given a *face* (square of the lattice) f define

$$B_f \colon \mathbb{M}_n \longrightarrow \mathbb{M}_n$$



Toric code: the Hamiltonian

Step 3: Define the Hamiltonian

$$H = \sum_{v \in V} 1 - A_v + \sum_{f \in F} 1 - B_f.$$

It ground state is the code space (memory) of Kitaev's quantum computer:

$$\mathrm{Prot}_n \overset{\text{def}}{=} \ker H = \{m \in \mathbb{M}_k \mid A_v(m) = m = B_f(m)\ \forall v \in V, f \in F\}.$$

Since all vertex and face operators commute, this is a stabiliser code.

Theorem (Kitaev'97)

For every $n \in \mathbb{N}$*, we have* $\mathrm{Prot}_n \cong \mathrm{span}_{\mathbb{C}}\{\mathrm{Hom}_{\mathrm{Grp}}(\pi_1(\mathbb{T}), \mathbb{Z}_2)\}$ *and so the toric code* $\mathrm{Prot}_n \subset (\mathbb{C}^2)^{\otimes 2n^2}$ *is a* $[2n^2, 2]$ *stabiliser code.*

From the toric code to a homology theory of Hopf algebras.

High-level overview

Up to "bookkeeping"[2] most aspects of the toric code can be vastly generalised:

- Instead of the torus, consider any closed orientable surface Σ.
- Instead of $\mathbb{C}\mathbb{Z}_2$: any Hopf algebra H with invertible antipode over a field $\Bbbk$.
- The extended Hilbert space becomes a Yetter–Drinfeld module $\mathbb{M}_E$ over $H^{\otimes V}$, where $V = \{0\text{-cells}\}$ and $E = \{1\text{-cells}\}$ of a *chosen(!)* CW-decomposition of Σ.

Theorem (Buerschaper, Mombelli, Christandl, Aguado '13)

Let H be semisimple, cosemisimple and $0 \neq \Lambda \in D(H)$ an integral. If we set

$$\mathrm{Prot} \stackrel{\text{def}}{=} \mathbb{M}_E^{\mathrm{inv}} = \Lambda^{\otimes V} \triangleright \mathbb{M}_E,$$

Then $\dim(\mathrm{Prot})$ *does not depend on the choice of CW-decomposition of* Σ.

[2] The bookkeeping is highly non-trivial though.

The non-semisimple case

Observation: $\mathbb{M}_E^{\mathrm{inv}}$ is not a suitable protected space in the non-semisimple case. Upon closer inspection:

$$\mathrm{Prot} = \{m \in \mathbb{M}_E \mid m_{[-1]} \otimes m_{[0]} = 1 \otimes m\} \cap \{m \in \mathbb{M}_E \mid h \triangleright m = \varepsilon(h)m \ \forall h \in H^{\otimes V}\}.$$

Note: If H is semisimple, we have $\mathrm{Hom}_{H^{\otimes V}}(\Bbbk, \mathbb{M}_E) \cong \Bbbk \otimes_{H^{\otimes V}} \mathbb{M}_E$.
This is *not* true if H is non-semisimple.

Definition (Gugenheim '62, Hofstetter '94, Canepeel–Raianu '95)

Given a Yetter–Drinfeld module $(M, \delta, \triangleright)$ over a Hopf algebra H, a group-like $a \in H$ and a character $\zeta\colon H \longrightarrow \Bbbk$. The *bitensor product* of $\Bbbk_\zeta^a$ and M is

$$\mathrm{Bit}_H(\Bbbk_\zeta^a, M) \stackrel{\text{def}}{=} \frac{X}{X \cap Y}, \qquad X = \{m \in M \mid \delta(m) = a \otimes m\},\ Y = \ker\zeta \triangleright M.$$

The non-semisimple Kitaev model

Theorem (Krähmer–H '25)

Let H be any Hopf algebra with invertible antipode and set

$$\mathrm{Prot} \stackrel{\text{def}}{=} \mathrm{Bit}_{\tilde{H}^{\otimes V}}(\Bbbk^{a}_{\zeta}, \mathbb{M}_E) \qquad (\tilde{H} \text{ some refinement of } H \ \&\ a, \zeta \text{ related to } S \text{ of } \tilde{H}).$$

Then, $\dim(\mathrm{Prot})$ *does not depend on the choice of CW-decomposition of Σ.*

Question: Does this return the usual protected space in the semisimple-cosemisimple setting?

What does the bitensor product compute?

Let H be finite-dimensional. The distinguished group-like $g \in H$ is the unique group-like element such that for any left integral $\lambda \in H^*$ and $\alpha \in H^*$, we have $\lambda\alpha = \alpha(g)\lambda$.

Theorem (Krähmer–H '25)

Let H be finite-dimensional and $g \in H$ the distinguished group-like. The following are equivalent

- *The functors* $\mathrm{Bit}_H(\Bbbk^g_\varepsilon, -)$ *and* $\mathrm{Hom}_{D(H)}({}^1_\varepsilon\Bbbk, -)$ *are naturally isomorphic,*
- *H is semisimple and $g = 1$.*

Kaplansky's conjecture and its consequences

Kaplansky's 5th conjecture '75:

If H is semisimple, then $S^2 = \mathrm{id}$.

Observation: If true, Kaplansky's conjecture implies that the distinguished group-like of any semisimple Hopf algebra is central.

Conversely, we may ask:

Question

Let g be the distinguished group-like of a semisimple Hopf algebra H.

- Is g central?
- Is $g = 1$?

Thank you!