Journal Club

Julien Bréhier

November 15, 2023

UNIVERSITÄT BONN

Table of contents

1. No-teleportation Theorem

1. No-teleportation Theorem
2. No-cloning Theorem

1. No-teleportation Theorem
2. No-cloning Theorem
3. No-deleting Theorem

1. No-teleportation Theorem
2. No-cloning Theorem
3. No-deleting Theorem
4. No-broadcast Theorem

1. No-teleportation Theorem
2. No-cloning Theorem
3. No-deleting Theorem
4. No-broadcast Theorem
5. No-hiding Theorem

1. No-teleportation Theorem
2. No-cloning Theorem
3. No-deleting Theorem
4. No-broadcast Theorem
5. No-hiding Theorem

A positive note !

The Gottesman-Knill Theorem : ‘ ... <u>can</u> ...’

The Pauli Matrices :

$$\sigma_x = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ \sigma_y = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ \sigma_z = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The Pauli Matrices :

$$\sigma_x = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_z = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The Pauli Group

$$\mathcal{P}_n = \{ \pm 1, \pm i \} \times \{ X^{a_1} Z^{b_1} \otimes ... \otimes X^{a_n} Z^{b_n} \}$$

The normalizer of the Pauli Group

$$\mathcal{C}_n = \{ c \in \mathcal{U}(2^n) \mid c\mathcal{P}_n c^\dagger = \mathcal{P}_n \}$$

The normalizer of the Pauli Group

$$\mathcal{C}_n = \{ c \in \mathcal{U}(2^n) \mid c\mathcal{P}_n c^\dagger = \mathcal{P}_n \}$$

This group can be generated by:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \; C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \; P = \sqrt{Z} = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{2}} \end{pmatrix}$$

What does a Clifford map a Pauli to ?

$$HZ \longleftrightarrow XH$$
$$PX \longleftrightarrow YP$$
$$PZ \longleftrightarrow ZP$$
$$C(X_1 \otimes \mathbb{1}_2) \rightarrow (X_1 \otimes X_2)C$$
$$C(\mathbb{1}_1 \otimes X_2) \rightarrow (\mathbb{1}_1 \otimes X_2)C$$
$$C(Z_1 \otimes \mathbb{1}_2) \rightarrow (Z_1 \otimes \mathbb{1}_2)C$$
$$C(\mathbb{1}_1 \otimes Z_2) \rightarrow (Z_1 \otimes Z_2)C$$

Stabilizer / Stabilized sub-space

$$S_{|\psi\rangle} = \{ s \in \mathcal{P}_n \mid s|\psi\rangle = +|\psi\rangle \}$$

Stabilizer / Stabilized sub-space

$$S_{|\psi\rangle} = \{\, s \in \mathcal{P}_n \mid s\,|\psi\rangle = +\,|\psi\rangle \,\}$$

$$V_S = \{\ |\psi\rangle \in \mathbb{C}^{2n} \mid \forall s \in S,\ s\,|\psi\rangle = |\psi\rangle \,\}$$

Stabilizer / Stabilized sub-space

$$S_{|\psi\rangle} = \{\, s \in \mathcal{P}_n \mid s\,|\psi\rangle = +\,|\psi\rangle \,\}$$

$$V_S = \{\; |\,\psi\rangle \in \mathbb{C}^{2n} \mid \forall s \in S,\; s\,|\psi\rangle = |\psi\rangle \,\}$$

Conservation of the stabilizer group

$$c\,|\psi\rangle = c\;s\,|\psi\rangle = c\;s\;c^{\dagger}c\,|\psi\rangle = (c\;s\;c^{\dagger})c\,|\psi\rangle$$

Stabilizer / Stabilized sub-space

$$S_{|\psi\rangle} = \{\, s \in \mathcal{P}_n \mid s\,|\psi\rangle = +\,|\psi\rangle \,\}$$

$$V_S = \{\ |\ \psi\rangle \in \mathbb{C}^{2n} \mid \forall s \in S,\ s\,|\psi\rangle = |\psi\rangle \,\}$$

Conservation of the stabilizer group

$$c\,|\psi\rangle = c\ s\,|\psi\rangle = c\ s\ c^{\dagger}c\,|\psi\rangle = (c\ s\ c^{\dagger})c\,|\psi\rangle$$

$$|\psi'\rangle = s'\,|\psi'\rangle$$

Stabilizer / Stabilized sub-space

$$S_{|\psi\rangle} = \{\, s \in \mathcal{P}_n \mid s\,|\psi\rangle = +\,|\psi\rangle \,\}$$

$$V_S = \{\ |\psi\rangle \in \mathbb{C}^{2n} \mid \forall s \in S,\ s\,|\psi\rangle = |\psi\rangle \,\}$$

Conservation of the stabilizer group

$$c\,|\psi\rangle = c\;s\,|\psi\rangle = c\;s\;c^{\dagger}c\,|\psi\rangle = (c\;s\;c^{\dagger})c\,|\psi\rangle$$

$$|\psi'\rangle = s'\,|\psi'\rangle$$

Simply :

$$s \rightarrow c\;s\;c^{\dagger}$$

$$s_1\ s_2 \rightarrow c\ (s_1\ s_2)\ c^{\dagger} = c\ s_1\ c^{\dagger}\ c\ s_2\ c^{\dagger} = (c\ s_1\ c^{\dagger}\ )(c\ s_2\ c^{\dagger})$$

Generating basis of a group

If $G = \mathrm{Span}\left\{ K \right\}$ is a finite group over V, $|K| \leq \log_2(|G|)$

Generating basis of a group

If $G = \mathrm{Span}\{K\}$ is a finite group over V, $|K| \leq \log_2(|G|)$
This yields : $|K| \leq n$

Generating basis of a group

If $G = \mathrm{Span}\,\{\, K \,\}$ is a finite group over V, $|K| \leq \log_2(|G|)$
This yields : $|K| \leq n$

Stabilized subspace

Theorem : If the stabilizer group S has s independent and commuting generators in $\mathcal{P}_n \setminus \{\, -(\mathbb{1}^{\otimes n}) \,\}$, the stabilizied subspace V_S has size $|V_S| = 2^{n-s}$

Storage

From the way we wrote the Pauli group $\mathcal{P}_n = \{ \pm 1 \} \times \{ X^{a_1} Z^{b_1} \otimes ... \otimes X^{a_n} Z^{b_n} \}$, storing a Pauli string is $2n + 1$ bits of informations.

Storage

From the way we wrote the Pauli group $\mathcal{P}_n = \{\pm 1\} \times \{X^{a_1} Z^{b_1} \otimes \ldots \otimes X^{a_n} Z^{b_n}\}$, storing a Pauli string is $2n + 1$ bits of informations.

Operations

Updating S at every step is $O(n)$ operations.

Storage

From the way we wrote the Pauli group $\mathcal{P}_n = \{ \pm 1 \} \times \{ X^{a_1} Z^{b_1} \otimes \ldots \otimes X^{a_n} Z^{b_n} \}$, storing a Pauli string is $2n + 1$ bits of informations.

Operations

Updating S at every step is $O(n)$ operations.

Measurements

Doing a measurement is $O(n^3)$

G.-K. Theorem

A unitary evolution including only:

- Initialization in the measurement basis ($O(1)$)
- Operations in the Clifford group* ($O(n)$)
- Measurement of Pauli operators ($O\left(n^3\right)$)

can be simulated efficiently (ie. in polynomial time) on a classical computer.

G.-K. Theorem

A unitary evolution including only:

- Initialization in the measurement basis ($O(1)$)
- Operations in the Clifford group* ($O(n)$)
- Measurement of Pauli operators ($O(n^3)$)

can be simulated efficiently (ie. in polynomial time) on a classical computer.

What for ?

Quantum teleportation, GHZ experiment, superdense coding, QEC protocols, ...

The Gottesman-Knill Theorem

Why is measurements $O(n^3)$?

Updating the stabilizer group

When measuring a quantity $q \in \mathcal{P}_n$, we get a result ± 1 and thus $\pm q$ becomes a stabilizer.

Updating the stabilizer group

When measuring a quantity $q \in \mathcal{P}_n$, we get a result ± 1 and thus $\pm q$ becomes a stabilizer.
2 cases :

Updating the stabilizer group

When measuring a quantity $q \in \mathcal{P}_n$, we get a result ± 1 and thus $\pm q$ becomes a stabilizer.
2 cases :

- q commutes with S.
  Find the decomposition of q in the basis of stabilizers (matrix inversion) to find deterministically the value of the measurement.

Updating the stabilizer group

When measuring a quantity $q \in \mathcal{P}_n$, we get a result ± 1 and thus $\pm q$ becomes a stabilizer.
2 cases :

- q commutes with S.
  Find the decomposition of q in the basis of stabilizers (matrix inversion) to find deterministically the value of the measurement.
- q anti-commutes with at least one element in S.
  Choose an anti-commuting stabilizer s_1, multiply all the others by s_1 and replace s_1 with q. Flip a coin to get the measurement.

Price to pay

$s\,(2n+1) \to 2s\,(2n+1)$ bits, $O\left(n^3\right) \to O\left(n^2\right)$ complexity

How ?

Add s “destabilizers”, generating the entire $\mathcal{P}_n$ such that :

$$\left[D_i, D_j\right] = 0, \forall i, j$$

$$\{\, S_i, D_i \,\} = 0 \forall i$$

$$\left[D_i, S_j\right] = 0, \forall i \neq j$$

Quick example

See the board

Quick example

See the board

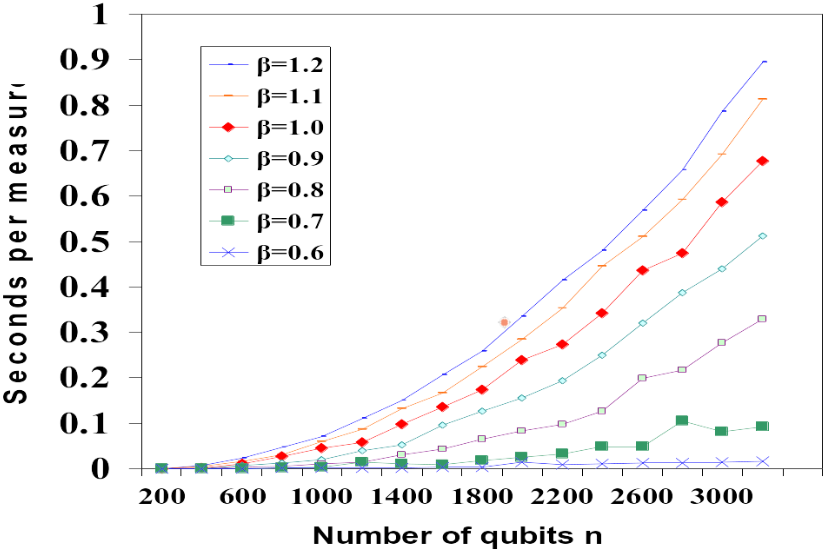
Use of D_i

We have to solve $\sum_s c_h S_h = \pm Z_a$.
However:

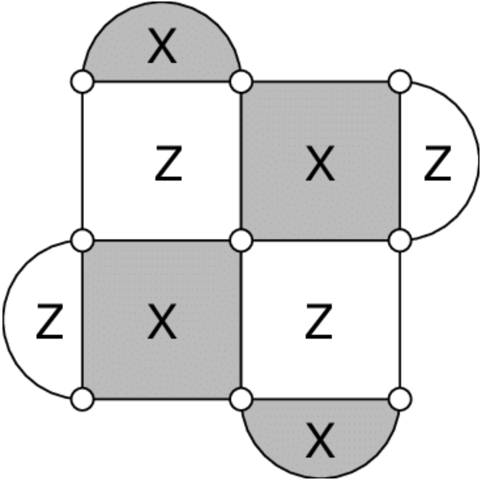
$$c_i \equiv \sum_s c_h \left(D_i \square S_h \right) \equiv D_i \square Z_a \pmod 2$$

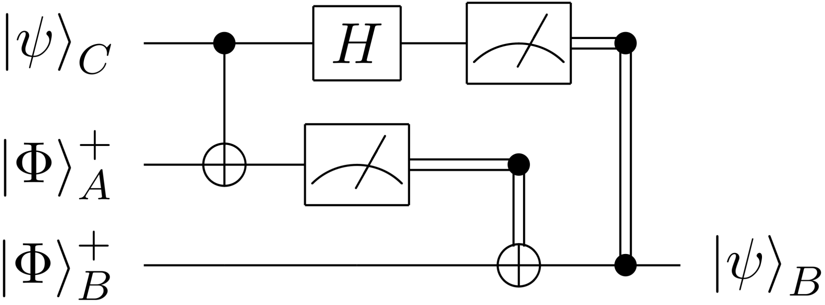
CHP simulation

Perfomance run by S. Aaronson, 256 MB ram, Pentium III 650MHz



Examples
Surface codes




$|\psi\rangle_C$
$|\Phi\rangle_A^+$
$|\Phi\rangle_B^+$
H
$|\psi\rangle_B$

A stronger formalism

Graph states : https://arxiv.org/abs/quant-ph/0504117

Small non-Clifford noise

Exponentially bad but manageable in a small amount.

References

D. Gottesman's Thesis : https://arxiv.org/abs/quant-ph/9705052

D. Gottesman's paper on the potency of Clifford simulation : https://arxiv.org/abs/quant-ph/9807006

D. Gottesman's and S. Aaronson's paper : https://arxiv.org/abs/quant-ph/0406196

S. Aaronson's website : https://www.scottaaronson.com/chp/

Introductory lecture notes on stabilizer formalism and clifford simulation : https://quantum.phys.cmu.edu/groupth/talk30Jan2009.pdf

stim, a Pyhton fast stabilizer circuit simulator : https://github.com/quantumlib/Stim

Thank you for your attention

Quantum Error Correction Sonnet, Daniel Gottesman

We cannot clone, perforce; instead, we split
Coherence to protect it from that wrong
That would destroy our valued quantum bit
And make our computation take too long.
Correct a flip and phase - that will suffice.
If in our code another error's bred,
We simply measure it, then God plays dice,
Collapsing it to X or Y or Zed.
We start with noisy seven, nine, or five
And end with perfect one. To better spot
Those flaws we must avoid, we first must strive
To find which ones commute and which do not.
With group and eigenstate, we've learned to fix
Your quantum errors with our quantum tricks.