Quantum error correction

SQMS/GGI Summer School on Quantum Simulation of Field Theories

July 28, 2022

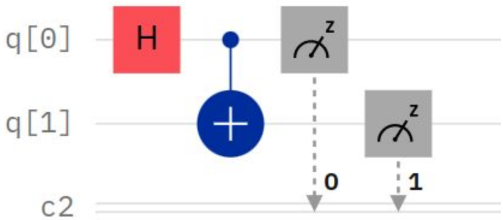
Ryan LaRose

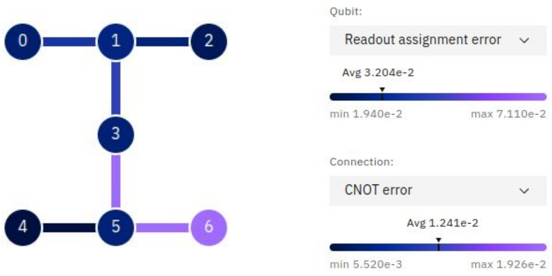
Motivation

- We want to do something with a quantum computer, like simulate field theories.
- But quantum computers are inherently noisy devices.

Motivation

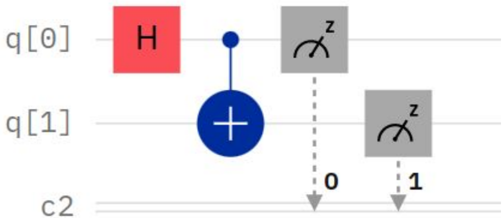
https://quantum-computing.ibm.com/

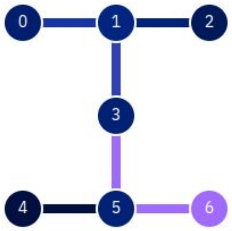


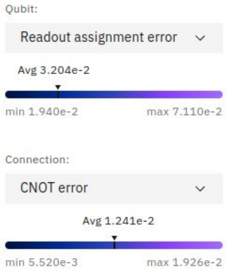


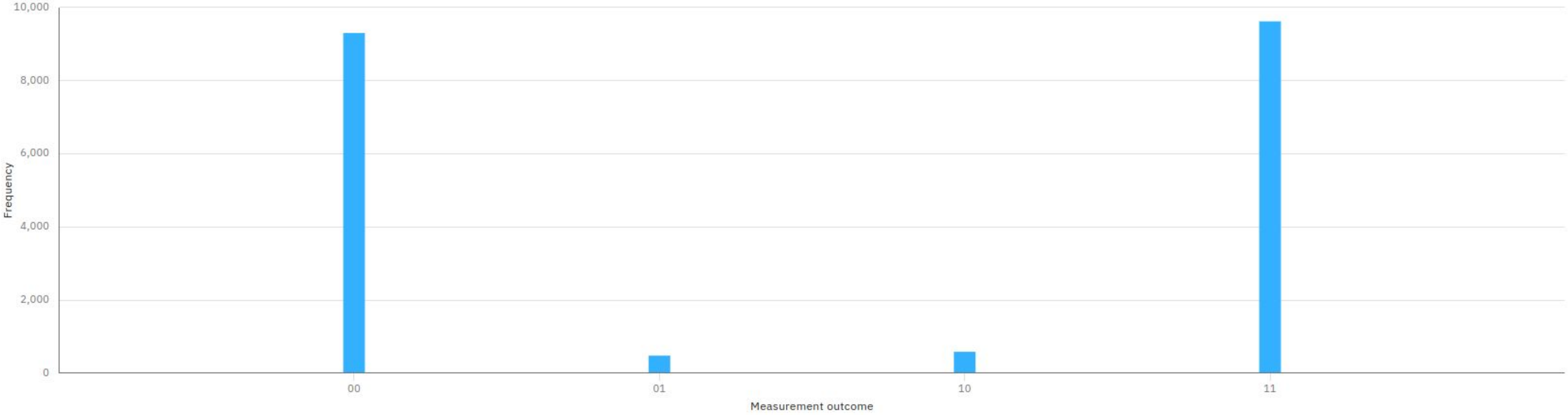
Motivation

https://quantum-computing.ibm.com/









Motivation

- We want to do something with a quantum computer, like simulate field theories.
- But quantum computers are inherently noisy devices.

- So, we have to do something to deal with the noise / errors.

Objectives

1. Define the key elements and principles of quantum error correction.
2. Introduce the stabilizer formalism through the repetition code.
3. Link to recent literature on experimental QEC.

Exercises in red

Recap of the gate model of quantum computing

- Qubits
  - Complex vectors

$$|0\rangle := [1,\ 0]^T \quad |1\rangle := [0,\ 1]^T$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$\alpha, \beta \in \mathbb{C} \qquad |\alpha|^2 + |\beta|^2 = 1$$

Recap of the gate model of quantum computing

- Qubits
  - Complex vectors

$$|0\rangle := [1,\ 0]^T \quad |1\rangle := [0,\ 1]^T$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$\alpha, \beta \in \mathbb{C} \quad |\alpha|^2 + |\beta|^2 = 1$$

- Gates
  - Unitary operators, e.g.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Recap of the gate model of quantum computing

- Qubits
  - Complex vectors

$$|0\rangle := [1,\, 0]^T \qquad |1\rangle := [0,\, 1]^T$$

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$\alpha, \beta \in \mathbb{C} \qquad |\alpha|^2 + |\beta|^2 = 1$$

- Gates
  - Unitary operators, e.g.

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

- Measurements
  - Set of operators $\{M_i\}$ such that $\sum_i M_i^\dagger M_i = I$
  - Probability of outcome *i* is $p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle$
  - State after obtaining outcome *i* is $M_i|\psi\rangle/\sqrt{p(i)}$

A bit of history

- Shor (1994): “Check out this poly-time algorithm for factoring.”

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

Abstract

A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally

A bit of history

- Community (1994): “Cool! But there’s no way this could ever be done in practice. Quantum states are very fragile.”

Maintaining coherence in quantum computers

W. G. Unruh*
Canadian Institute for Advanced Research, Cosmology Program, Department of Physics, University of British Columbia, Vancouver, Canada V6T 1Z1

(Received 10 June 1994)

The effects of the inevitable coupling to external degrees of freedom of a quantum computer are examined. It is found that for quantum calculations (in which the maintenance of coherence over a large number of states is important), not only must the coupling be small, but the time taken in the quantum calculation must be less than the thermal time scale $\hbar/k_B T$. For longer times the condition on the strength of the coupling to the external world becomes much more stringent.

A bit of history

- Shor (1995): “Check out this quantum error correcting code.”

Scheme for reducing decoherence in quantum computer memory

Peter W. Shor*
AT&T Bell Laboratories, Room 2D-149, 600 Mountain Avenue, Murray Hill, New Jersey 07974
(Received 17 May 1995)

Recently, it was realized that use of the properties of quantum mechanics might speed up certain computations dramatically. Interest has since been growing in the area of quantum computation. One of the main difficulties of quantum computation is that decoherence destroys the information in a superposition of states contained in a quantum computer, thus making long computations impossible. It is shown how to reduce the effects of decoherence for information stored in quantum memory, assuming that the decoherence process acts independently on each of the bits stored in memory. This involves the use of a quantum analog of error-correcting codes.

A bit of history

- Shor (1996): Fault-tolerant quantum computation.

Fault-Tolerant Quantum Computation

Peter W. Shor
AT&T Research

Abstract

It has recently been realized that use of the properties of quantum mechanics might speed up certain computations dramatically. Interest in quantum computation has since been growing. One of the main difficulties in realizing quantum computation is that decoherence tends to destroy the information in a superposition of states in a quantum computer, making long computations impossible. A further difficulty is that inaccuracies in quantum state transformations throughout the computation accumulate, rendering long computations unreliable. However, these obstacles may not be as formidable as originally believed. For any quantum

Errors in quantum computers

- Classically, bits can flip. (Or be erased.)
  - i.e., 0 -> 1 and 1 -> 0 with some probability p.

Errors in quantum computers

- Classically, bits can flip. (Or be erased.)
  - i.e., 0 -> 1 and 1 -> 0 with some probability p.
- Qubits have a larger state space, therefore more things can go wrong.
  - Any operation which can be considered a gate can also be considered an error.
  - Example: Pauli errors

Errors in quantum computers

- Classically, bits can flip. (Or be erased.)
  - i.e., 0 -> 1 and 1 -> 0 with some probability p.
- Qubits have a larger state space, therefore more things can go wrong.
  - Any operation which can be considered a gate can also be considered an error.
  - Example: Pauli errors

$$X|0\rangle = |1\rangle$$
$$X|1\rangle = |0\rangle$$

Bit flip

$$Z|0\rangle = |0\rangle$$
$$Z|1\rangle = -|1\rangle$$

Phase flip

$$Y|0\rangle = i|1\rangle = iXZ|0\rangle$$
$$Y|1\rangle = -i|0\rangle = iXZ|1\rangle$$

Bit & phase flip

The most important fact about QEC

- Errors are continuous (analog). How can we hope to correct these?

The most important fact about QEC

- Errors are continuous (analog). How can we hope to correct these?
- Suppose some error *E* introduces a relative phase $E|\psi\rangle = \alpha|0\rangle + e^{i\delta}\beta|1\rangle$
- The angle delta could be (in principle) infinitesimal.

The most important fact about QEC

- Errors are continuous (analog). How can we hope to correct these?
- Suppose some error *E* introduces a relative phase $E|\psi\rangle = \alpha|0\rangle + e^{i\delta}\beta|1\rangle$
- The angle delta could be (in principle) infinitesimal.

- Any error can be written as (discrete) Pauli errors with continuous coeffs
  - This is because Paulis (+ identity) span $\mathbb{C}^{2\times 2}$

$$E|\psi\rangle = (e_0 I + e_1 X + e_2 Y + e_3 Z)|\psi\rangle \quad \forall E, |\psi\rangle$$

The most important fact about QEC

- Errors are continuous (analog). How can we hope to correct these?
- Suppose some error *E* introduces a relative phase $E|\psi\rangle = \alpha|0\rangle + e^{i\delta}\beta|1\rangle$
- The angle delta could be (in principle) infinitesimal.

- Any error can be written as (discrete) Pauli errors with continuous coeffs
  - This is because Paulis (+ identity) span $\mathbb{C}^{2\times 2}$

$$E|\psi\rangle = (e_0 I + e_1 X + e_2 Y + e_3 Z)|\psi\rangle \qquad \forall E, |\psi\rangle$$

- But coefficients e_*i* could still be (in principle) infinitesimal.
  - Is it possible to deal with this?

The most important fact about QEC

- **Measurement maps continuous errors to discrete errors.**

The most important fact about QEC

- **Measurement maps continuous errors to discrete errors.**
- Suppose we measure the error state using operators $\{M_i\}$

$$E|\psi\rangle = (e_0 I + e_1 X + e_2 Y + e_3 Z)|\psi\rangle$$

- Then, with probability *p(i)* we get

$$M_i E|\psi\rangle / \sqrt{p(i)}$$

The most important fact about QEC

- **Measurement maps continuous errors to discrete errors.**
- Suppose we measure the error state using operators $\{M_i\}$

$$E|\psi\rangle = (e_0 I + e_1 X + e_2 Y + e_3 Z)|\psi\rangle$$

- Then, with probability *p(i)* we get

$$M_i E|\psi\rangle / \sqrt{p(i)}$$

- *This collapses the superposition and makes the continuous coefficient an irrelevant global phase*

The most important fact about QEC

- **Measurement maps continuous errors to discrete errors.**
- Suppose we measure the error state using operators $\{M_i\}$

$$E|\psi\rangle = (e_0 I + e_1 X + e_2 Y + e_3 Z)|\psi\rangle$$

- Then, with probability *p(i)* we get

$$M_i E|\psi\rangle / \sqrt{p(i)}$$

- *This collapses the superposition and makes the continuous coefficient an irrelevant global phase*
  - For example, we could choose M_i such that $E|\psi\rangle \mapsto \eta_i \sigma_i |\psi\rangle$
  - The $\sigma_i \in \{I, X, Y, Z\}$ is now a discrete error which can be corrected.
  - The $\eta_i \in \mathbb{C}$ is continuous **but is a global phase, so doesn't matter**.

Classical error correction : The repetition code

- A key concept in error correction is adding redundancy.
- For example, given a bit, we can make three copies of it:
  - 0 -> 000
  - 1 -> 111

Classical error correction : The repetition code

- A key concept in error correction is adding redundancy.
- For example, given a bit, we can make three copies of it:
  - 0 -> 000
  - 1 -> 111
- This is known as the (classical) **repetition code**.
- The idea is very simple: If an error occurs on one bit only, we can correct it by looking at the other two bits and taking a majority vote.

Classical error correction : The repetition code

- A key concept in error correction is adding redundancy.
- For example, given a bit, we can make three copies of it:
    - 0 -> 000
    - 1 -> 111
- This is known as the (classical) **repetition code**.
- The idea is very simple: If an error occurs on one bit only, we can correct it by looking at the other two bits and taking a majority vote.

- Suppose each bit flips independently with probability p. For which p is the repetition code beneficial?

Analyzing the repetition code

- Suppose each bit flips independently with probability *p*. For which *p* is the repetition code beneficial?
    - The probability of an error without the encoding is p.
    - With the encoding, the probability of an error is prob(> 1 bit flips) which is

$$p_e := 3p^2(1-p) + p^3 = 3p^2 - 2p^3$$

    - By setting $p_e < p$ we find that the repetition code is better provided that

$$p < 1/2$$

QEC: Subtle point about adding redundancy

- Given the classical repetition code, we might try to do the same with qubits, i.e. map

$$|\psi\rangle \mapsto |\psi\rangle|\psi\rangle|\psi\rangle$$

- This is not possible in general, as expressed by the "no cloning theorem"

Aside: Remark about no cloning

- Note in the previous proof the only properties we used were tensor products and linearity.
- In this respect no cloning is also a classical theorem.
- Specifically: No linear *stochastic* map (not necessarily unitary map) can clone arbitrary classical probability distributions in tensor product.
  - See http://info.phys.unm.edu//~crosson/Phys572/QI-572-L9.pdf for more. (The proof is the same, but there is a longer, interesting discussion.)

QEC: Can we add any redundancy?

- From no cloning we cannot make copies of our state as in the classical repetition code. Can we copy anything?
- Claim: We can "copy basis information" in the following sense:

$$\alpha|0\rangle + \beta|1\rangle \mapsto \alpha|000\rangle + \beta|111\rangle$$

QEC: Can we add any redundancy?

- From no cloning we cannot make copies of our state as in the classical repetition code. Can we copy anything?
- Claim: We can "copy basis information" in the following sense:

$$\alpha|0\rangle + \beta|1\rangle \mapsto \alpha|000\rangle + \beta|111\rangle$$

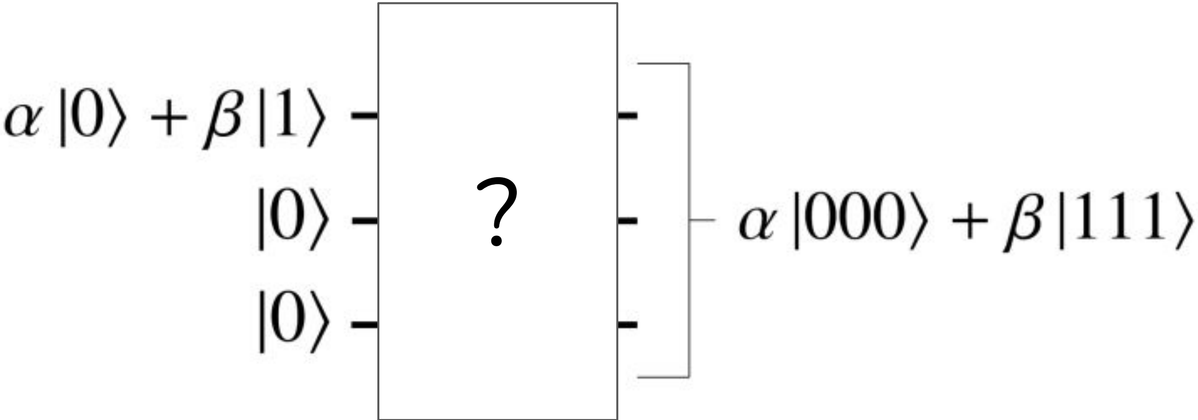
- *Note: |0> and |1> are orthogonal, so this can be viewed as the exception to no-cloning*
- How can this be done?

QEC: Can we add any redundancy?

- From no cloning we cannot make copies of our state as in the classical repetition code. Can we copy anything?
- Claim: We can “copy basis information” in the following sense:

$$\alpha|0\rangle + \beta|1\rangle \mapsto \alpha|000\rangle + \beta|111\rangle$$

- *Note: |0> and |1> are orthogonal, so this can be viewed as the exception to no-cloning*
- How can this be done?

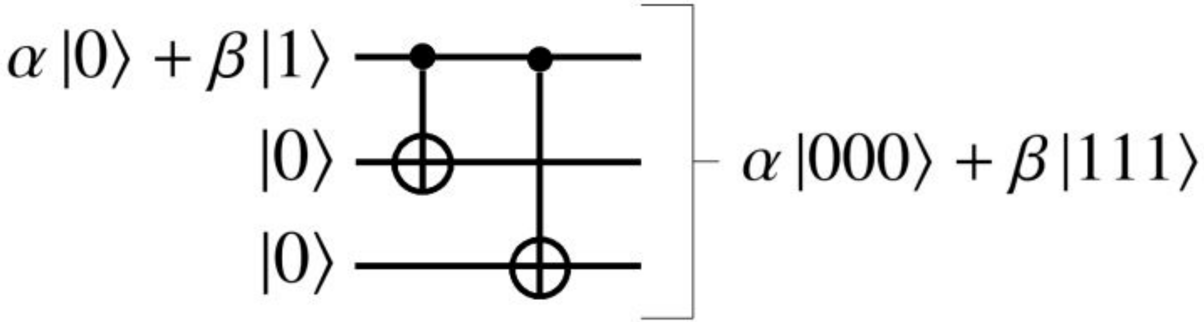


QEC: Can we add any redundancy?

- From no cloning we cannot make copies of our state as in the classical repetition code. Can we copy anything?
- Claim: We can “copy basis information” in the following sense:

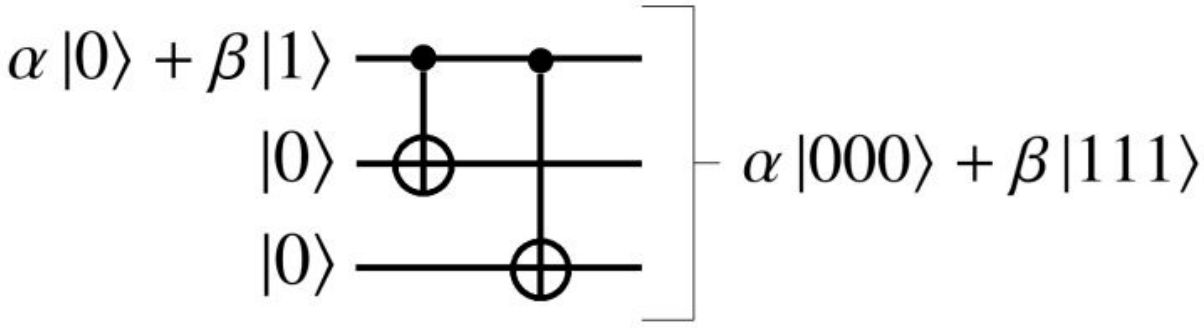
$$\alpha|0\rangle + \beta|1\rangle \mapsto \alpha|000\rangle + \beta|111\rangle$$

- *Note: |0> and |1> are orthogonal, so this can be viewed as the exception to no-cloning*
- How can this be done?



QEC: Can we add any redundancy?

- Note that this encoding circuit entangles the “input” qubit with two other qubits.



- Since errors in quantum computers are due to (for the most part) qubits entangling with their environment, we can understand a quote from John Preskill:
- *“We have learned that it is possible to fight entanglement with entanglement.”*

Repetition code for bit flip errors

- The encoding a|0> + b|1> -> a |000> + b|111> gives us redundancy. Now what?
- We need to check which errors (if any) occured in the encoded state.

Repetition code for bit flip errors

- The encoding a|0> + b|1> -> a |000> + b|111> gives us redundancy. Now what?
- We need to check which errors (if any) occured in the encoded state.
- We do this by (projective) measurements. What projections should we apply to find out what happened?
- There are four things that can happen:

Repetition code for bit flip errors

- The encoding a|0> + b|1> -> a |000> + b|111> gives us redundancy. Now what?
- We need to check which errors (if any) occured in the encoded state.
- We do this by (projective) measurements. What projections should we apply to find out what happened?
- There are four things that can happen:

No qubit was flipped.	$P_0 = \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$

Repetition code for bit flip errors

- The encoding a|0> + b|1> -> a |000> + b|111> gives us redundancy. Now what?
- We need to check which errors (if any) occured in the encoded state.
- We do this by (projective) measurements. What projections should we apply to find out what happened?
- There are four things that can happen:

No qubit was flipped.	$P_0 = \|000\rangle \langle 000\| + \|111\rangle \langle 111\|$
The first qubit was flipped.	

Repetition code for bit flip errors

- The encoding a|0> + b|1> -> a |000> + b|111> gives us redundancy. Now what?
- We need to check which errors (if any) occured in the encoded state.
- We do this by (projective) measurements. What projections should we apply to find out what happened?
- There are four things that can happen:

No qubit was flipped.	$P_0 = \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$
The first qubit was flipped.	$P_1 = \|100\rangle\langle 100\| + \|011\rangle\langle 011\|$

Repetition code for bit flip errors

- The encoding a|0> + b|1> -> a |000> + b|111> gives us redundancy. Now what?
- We need to check which errors (if any) occured in the encoded state.
- We do this by (projective) measurements. What projections should we apply to find out what happened?
- There are four things that can happen:

No qubit was flipped.	$P_0 = \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$
The first qubit was flipped.	$P_1 = \|100\rangle\langle 100\| + \|011\rangle\langle 011\|$
The second qubit was flipped.	

Repetition code for bit flip errors

- The encoding a|0> + b|1> -> a |000> + b|111> gives us redundancy. Now what?
- We need to check which errors (if any) occured in the encoded state.
- We do this by (projective) measurements. What projections should we apply to find out what happened?
- There are four things that can happen:

No qubit was flipped.	$P_0 = \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$
The first qubit was flipped.	$P_1 = \|100\rangle\langle 100\| + \|011\rangle\langle 011\|$
The second qubit was flipped.	$P_2 = \|010\rangle\langle 010\| + \|101\rangle\langle 101\|$

Repetition code for bit flip errors

- The encoding a|0> + b|1> -> a |000> + b|111> gives us redundancy. Now what?
- We need to check which errors (if any) occured in the encoded state.
- We do this by (projective) measurements. What projections should we apply to find out what happened?
- There are four things that can happen:

No qubit was flipped.	$P_0 = \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$
The first qubit was flipped.	$P_1 = \|100\rangle\langle 100\| + \|011\rangle\langle 011\|$
The second qubit was flipped.	$P_2 = \|010\rangle\langle 010\| + \|101\rangle\langle 101\|$
The third qubit was flipped.	

Repetition code for bit flip errors

- The encoding a|0> + b|1> -> a |000> + b|111> gives us redundancy. Now what?
- We need to check which errors (if any) occured in the encoded state.
- We do this by (projective) measurements. What projections should we apply to find out what happened?
- There are four things that can happen:

No qubit was flipped.	$P_0 = \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$
The first qubit was flipped.	$P_1 = \|100\rangle\langle 100\| + \|011\rangle\langle 011\|$
The second qubit was flipped.	$P_2 = \|010\rangle\langle 010\| + \|101\rangle\langle 101\|$
The third qubit was flipped.	$P_3 = \|001\rangle\langle 001\| + \|110\rangle\langle 110\|$

Turning the table

- By measuring these operators, we learn what errors (if any) occurred.
- Since we know which error occurred, we can correct it.

Syndrome measurement	Meaning	Correction operator
$P_0 = \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$	No qubit was flipped.	I
$P_1 = \|100\rangle\langle 100\| + \|011\rangle\langle 011\|$	The first qubit was flipped.	X_0
$P_2 = \|010\rangle\langle 010\| + \|101\rangle\langle 101\|$	The second qubit was flipped.	X_1
$P_3 = \|001\rangle\langle 001\| + \|110\rangle\langle 110\|$	The third qubit was flipped.	X_2

Repetition code for phase flip errors

- We can now correct bit flip (X) errors. Can we modify this for phase (Z) errors?

Repetition code for phase flip errors

- We can now correct bit flip (X) errors. Can we modify this for phase (Z) errors?
  - These are related by a change of basis

$$HXH = Z$$

  - Thus we can do the encoding:
    - |0> -> (|0> + |1>)(|0> + |1>)(|0> + |1>)
    - |1> -> (|0> - |1>)(|0> - |1>)(|0> - |1>)
    - And update the syndrome measurements in a similar way.
      - Q: What should they be?

Repetition code for phase flip errors

- We can now correct bit flip (X) errors. Can we modify this for phase (Z) errors?
    - These are related by a change of basis

$$HXH = Z$$

    - Thus we can do the encoding:
        - |0> -> (|0> + |1>)(|0> + |1>)(|0> + |1>)
        - |1> -> (|0> - |1>)(|0> - |1>)(|0> - |1>)
        - And update the syndrome measurements in a similar way.
            - Q: What should they be?

What about both bit flip and phase flip errors?

Shor's 9-qubit code

- This is formed by **concatenating** the bit flip and phase flip codes.
  - Concatenation is an important, often used concept in error correction.
  - The idea is simply to combine the two codes.
- Step 1: Apply bit flip code to physical qubit.
- Step 2: Apply phase flip code to the *logical qubit*.

Shor's 9-qubit code

- This is formed by **concatenating** the bit flip and phase flip codes.
  - Concatenation is an important, often used concept in error correction.
  - The idea is simply to combine the two codes.
- Step 1: Apply bit flip code to physical qubit.
- Step 2: Apply phase flip code to the *logical qubit*.

$$|0\rangle \mapsto |000\rangle \mapsto (|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) =: |\bar{0}\rangle$$

Shor's 9-qubit code

- This is formed by **concatenating** the bit flip and phase flip codes.
  - Concatenation is an important, often used concept in error correction.
  - The idea is simply to combine the two codes.
- Step 1: Apply bit flip code to physical qubit.
- Step 2: Apply phase flip code to the *logical qubit*.

$$|0\rangle \mapsto |000\rangle \mapsto (|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) =: |\bar{0}\rangle$$

$$|1\rangle \mapsto |111\rangle \mapsto (|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) =: |\bar{1}\rangle$$

Shor's 9-qubit code

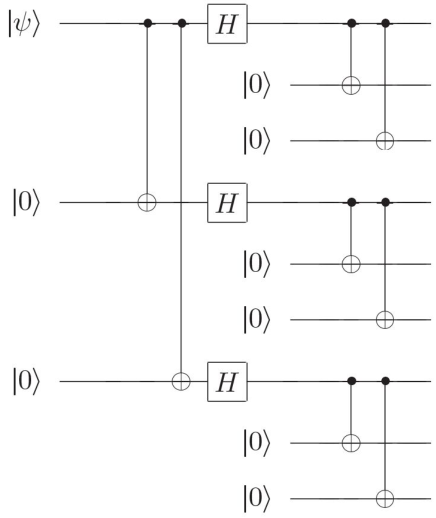
- This is formed by **concatenating** the bit flip and phase flip codes.
  - Concatenation is an important, often used concept in error correction.
  - The idea is simply to combine the two codes.
- Step 1: Apply bit flip code to physical qubit.
- Step 2: Apply phase flip code to the *logical qubit*.

$$|0\rangle \mapsto |000\rangle \mapsto (|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) =: |\bar{0}\rangle$$

$$|1\rangle \mapsto |111\rangle \mapsto (|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) =: |\bar{1}\rangle$$

$$|\bar{\psi}\rangle = \alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$$

Shor’s 9-qubit code



- This is formed by **concatenating** the bit flip and phase flip codes.
  - Concatenation is an important, often used concept in error correction.
  - The idea is simply to combine the two codes.
- Step 1: Apply bit flip code to physical qubit.
- Step 2: Apply phase flip code to the *logical qubit*.

$$|0\rangle \mapsto |000\rangle \mapsto (|000\rangle + |111\rangle)(|000\rangle + |111\rangle)(|000\rangle + |111\rangle) =: |\bar{0}\rangle$$

$$|1\rangle \mapsto |111\rangle \mapsto (|000\rangle - |111\rangle)(|000\rangle - |111\rangle)(|000\rangle - |111\rangle) =: |\bar{1}\rangle$$

$$|\bar{\psi}\rangle = \alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$$

Note 1: Error correction vs. fault tolerance

- **Error correction**:
  - Theory in which some components do not have errors (by assumption)
  - E.g., state preparation is perfect, errors occur only during gates
  - This is “easier” than fault tolerance (simplifying assumptions)

- **Fault tolerance**:
  - Theory in which all components have errors and errors are not allowed to propagate.
  - State preparation, gates, measurements, …
  - This is “harder” than error correction (no simplifying assumptions)

Note 2: Redundancy vs. partitioning

Blue = good basis vector (*codeword*)

Red = bad basis vector (*error state*)

- |000>
- |001>
- |010>
- |011>
- |100>
- |101>
- |110>
- |111>

From projections to stabilizers

- Remember the four projectors for the bit-flip code?

Syndrome measurement	Meaning	Correction operator
$P_0 = \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$	No qubit was flipped.	I
$P_1 = \|100\rangle\langle 100\| + \|011\rangle\langle 011\|$	The first qubit was flipped.	X_0
$P_2 = \|010\rangle\langle 010\| + \|101\rangle\langle 101\|$	The second qubit was flipped.	X_1
$P_3 = \|001\rangle\langle 001\| + \|110\rangle\langle 110\|$	The third qubit was flipped.	X_2

From projections to stabilizers

- Remember the four projectors for the bit-flip code?

Syndrome measurement	Meaning	Correction operator
$P_0 = \|000\rangle\langle 000\| + \|111\rangle\langle 111\|$	No qubit was flipped.	I
$P_1 = \|100\rangle\langle 100\| + \|011\rangle\langle 011\|$	The first qubit was flipped.	X_0
$P_2 = \|010\rangle\langle 010\| + \|101\rangle\langle 101\|$	The second qubit was flipped.	X_1
$P_3 = \|001\rangle\langle 001\| + \|110\rangle\langle 110\|$	The third qubit was flipped.	X_2

- **There's a more succinct way to determine which errors occurred**.

From projections to stabilizers

- Consider measuring the operator $Z_1 Z_2 \equiv ZZI$

From projections to stabilizers

- Consider measuring the operator $Z_1 Z_2 \equiv ZZI$

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1|$$

From projections to stabilizers

- Consider measuring the operator $Z_1 Z_2 \equiv ZZI$

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1|$$

$$ZZ = (|0\rangle\langle 0| - |1\rangle\langle 1|)\,(|0\rangle\langle 0| - |1\rangle\langle 1|)$$

From projections to stabilizers

- Consider measuring the operator $Z_1 Z_2 \equiv ZZI$

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1|$$

$$ZZ = (|0\rangle\langle 0| - |1\rangle\langle 1|)(|0\rangle\langle 0| - |1\rangle\langle 1|)$$

$$ZZ = \underbrace{(|00\rangle\langle 00| + |11\rangle\langle 11|)}_{\text{+1 eigenspace. Bits are the same.}} - \underbrace{(|01\rangle\langle 01| + |10\rangle\langle 10|)}_{\text{-1 eigenspace. Bits are different.}}$$

From projections to stabilizers

- Just as Z1 Z2 asks if the first two bits are the same/different, Z2 Z3 asks if the second two bits are the same/different.
- Q: Given this information, can you determine which of the three bits flipped?

From projections to stabilizers

- Just as Z1 Z2 asks if the first two bits are the same/different, Z2 Z3 asks if the second two bits are the same/different.
- Q: Given this information, can you determine which of the three bits flipped?
  - Example: Bits 1 and 2 are the same, bits 2 and 3 are different.

From projections to stabilizers

- Just as Z1 Z2 asks if the first two bits are the same/different, Z2 Z3 asks if the second two bits are the same/different.
- Q: Given this information, can you determine which of the three bits flipped?
  - Example: Bits 1 and 2 are the same, bits 2 and 3 are different.
    - Bit 3 flipped.

From projections to stabilizers

- Just as Z1 Z2 asks if the first two bits are the same/different, Z2 Z3 asks if the second two bits are the same/different.
- Q: Given this information, can you determine which of the three bits flipped?
  - Example: Bits 1 and 2 are the same, bits 2 and 3 are different.
    - Bit 3 flipped.
  - Example: Bits 1 and 2 are different, bits 2 and 3 are the same.

From projections to stabilizers

- Just as Z1 Z2 asks if the first two bits are the same/different, Z2 Z3 asks if the second two bits are the same/different.
- Q: Given this information, can you determine which of the three bits flipped?
  - Example: Bits 1 and 2 are the same, bits 2 and 3 are different.
    - Bit 3 flipped.
  - Example: Bits 1 and 2 are different, bits 2 and 3 are the same.
    - Bit 1 flipped.

From projections to stabilizers

- Just as Z1 Z2 asks if the first two bits are the same/different, Z2 Z3 asks if the second two bits are the same/different.
- Q: Given this information, can you determine which of the three bits flipped?
  - Example: Bits 1 and 2 are the same, bits 2 and 3 are different.
    - Bit 3 flipped.
  - Example: Bits 1 and 2 are different, bits 2 and 3 are the same.
    - Bit 1 flipped.
  - Example: Bits 1 and 2 are different, bits 2 and 3 are different.

From projections to stabilizers

- Just as Z1 Z2 asks if the first two bits are the same/different, Z2 Z3 asks if the second two bits are the same/different.
- Q: Given this information, can you determine which of the three bits flipped?
  - Example: Bits 1 and 2 are the same, bits 2 and 3 are different.
    - Bit 3 flipped.
  - Example: Bits 1 and 2 are different, bits 2 and 3 are the same.
    - Bit 1 flipped.
  - Example: Bits 1 and 2 are different, bits 2 and 3 are different.
    - Bit 2 flipped.

From projections to stabilizers

- Just as Z1 Z2 asks if the first two bits are the same/different, Z2 Z3 asks if the second two bits are the same/different.
- Q: Given this information, can you determine which of the three bits flipped?
  - Example: Bits 1 and 2 are the same, bits 2 and 3 are different.
    - Bit 3 flipped.
  - Example: Bits 1 and 2 are different, bits 2 and 3 are the same.
    - Bit 1 flipped.
  - Example: Bits 1 and 2 are different, bits 2 and 3 are different.
    - Bit 2 flipped.
  - Example: Bits 1 and 2 are the same, bits 2 and 3 are the same.

From projections to stabilizers

- Just as Z1 Z2 asks if the first two bits are the same/different, Z2 Z3 asks if the second two bits are the same/different.
- Q: Given this information, can you determine which of the three bits flipped?
    - Example: Bits 1 and 2 are the same, bits 2 and 3 are different.
        - Bit 3 flipped.
    - Example: Bits 1 and 2 are different, bits 2 and 3 are the same.
        - Bit 1 flipped.
    - Example: Bits 1 and 2 are different, bits 2 and 3 are different.
        - Bit 2 flipped.
    - Example: Bits 1 and 2 are the same, bits 2 and 3 are the same.
        - No bit flipped.

From projections to stabilizers

- Just as Z1 Z2 asks if the first two bits are the same/different, Z2 Z3 asks if the second two bits are the same/different.
- Q: Given this information, can you determine which of the three bits flipped?
    - Example: Bits 1 and 2 are the same, bits 2 and 3 are different.
        - Bit 3 flipped.
    - Example: Bits 1 and 2 are different, bits 2 and 3 are the same.
        - Bit 1 flipped.
    - Example: Bits 1 and 2 are different, bits 2 and 3 are different.
        - Bit 2 flipped.
    - Example: Bits 1 and 2 are the same, bits 2 and 3 are the same.
        - No bit flipped.
- **This was exactly our table from before!**

From projections to stabilizers

- Just as Z1 Z2 asks if the first two bits are the same/different, Z2 Z3 asks if the second two bits are the same/different.
- Q: Given this information, can you determine which of the three bits flipped?
  - Example: Bits 1 and 2 are the same, bits 2 and 3 are different.
    - Bit 3 flipped.
  - Example: Bits 1 and 2 are different, bits 2 and 3 are the same.
    - Bit 1 flipped.
  - Example: Bits 1 and 2 are different, bits 2 and 3 are different.
    - Bit 2 flipped.
  - Example: Bits 1 and 2 are the same, bits 2 and 3 are the same.
    - No bit flipped.
- **This was exactly our table from before!**

Q: Could we do the same with Z1 Z2 and Z1 Z3?

From projections to stabilizers

Syndrome $(\langle Z_1 Z_2 \rangle, \langle Z_2 Z_3 \rangle)$	Meaning	Correction operator
(1, 1)	No qubit was flipped.	I
(-1, 1)	The first qubit was flipped.	X_0
(-1, -1)	The second qubit was flipped.	X_1
(1, -1)	The third qubit was flipped.	X_2

The operators Z1 Z2 and Z2 Z3 are known as **stabilizer elements**.

Stabilizer elements? Elements of what?

The operators Z1 Z2 and Z2 Z3 are known as **stabilizer elements**.

+1 eigenstates of Z1 Z2: **|000>**, |001>, |110>, and **|111>**

+1 eigenstates of Z2 Z3: **|000>**, |100>, |011>, and **|111>**

These are *almost* the codewords of the bit-flip code, **|000>** and **|111>**.

Stabilizer elements? Elements of what?

The operators Z1 Z2 and Z2 Z3 are known as **stabilizer elements**.

+1 eigenstates of Z1 Z2: **|000>**, |001>, |110>, and **|111>**

+1 eigenstates of Z2 Z3: **|000>**, |100>, |011>, and **|111>**

These are *almost* the codewords of the bit-flip code, **|000>** and **|111>**.

Preview:

- They are elements of the **stabilizer group S** = {I, Z1 Z2, Z2 Z3, Z1 Z3}.

Stabilizer elements? Elements of what?

The operators Z1 Z2 and Z2 Z3 are known as **stabilizer elements**.

+1 eigenstates of Z1 Z2: **|000>**, |001>, |110>, and **|111>**

+1 eigenstates of Z2 Z3: **|000>**, |100>, |011>, and **|111>**

These are *almost* the codewords of the bit-flip code, **|000>** and **|111>**.

Preview:

- They are elements of the **stabilizer group S** = {I, Z1 Z2, Z2 Z3, Z1 Z3}.
    - This group is **generated** by Z1 Z2 and Z2 Z3, i.e. S = <Z1 Z2, Z2 Z3>.

Stabilizer elements? Elements of what?

The operators Z1 Z2 and Z2 Z3 are known as **stabilizer elements**.

+1 eigenstates of Z1 Z2: **|000>**, |001>, |110>, and **|111>**

+1 eigenstates of Z2 Z3: **|000>**, |100>, |011>, and **|111>**

These are *almost* the codewords of the bit-flip code, **|000>** and **|111>**.

Preview:

- They are elements of the **stabilizer group S** = {I, Z1 Z2, Z2 Z3, Z1 Z3}.
    - This group is **generated** by Z1 Z2 and Z2 Z3, i.e. S = <Z1 Z2, Z2 Z3>.
- This is a subgroup of P3 (the Pauli group on 3 qubits).

Stabilizer elements? Elements of what?

The operators Z1 Z2 and Z2 Z3 are known as **stabilizer elements**.

+1 eigenstates of Z1 Z2: **|000>**, |001>, |110>, and **|111>**

+1 eigenstates of Z2 Z3: **|000>**, |100>, |011>, and **|111>**

These are *almost* the codewords of the bit-flip code, **|000>** and **|111>**.

Preview:

- They are elements of the **stabilizer group S** = {I, Z1 Z2, Z2 Z3, Z1 Z3}.
    - This group is **generated** by Z1 Z2 and Z2 Z3, i.e. S = <Z1 Z2, Z2 Z3>.
- This is a subgroup of P3 (the Pauli group on 3 qubits).
- The subspace of P3 **stabilized** by S is spanned by |000> and |111>.

Stabilizer elements? Elements of what?

The operators Z1 Z2 and Z2 Z3 are known as **stabilizer elements**.

+1 eigenstates of Z1 Z2: **|000>**, |001>, |110>, and **|111>**

+1 eigenstates of Z2 Z3: **|000>**, |100>, |011>, and **|111>**

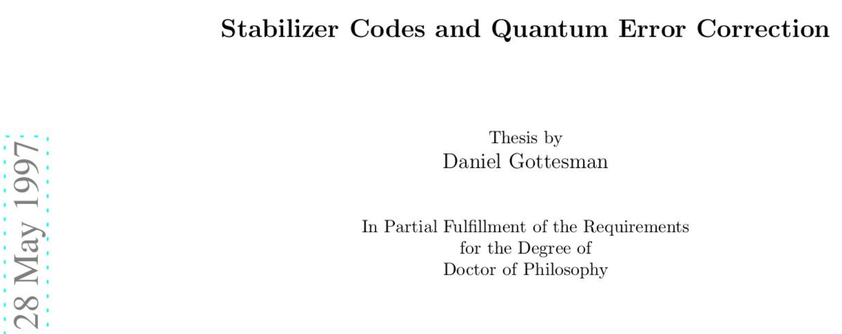
These are *almost* the codewords of the bit-flip code, **|000>** and **|111>**.

Preview:

- They are elements of the **stabilizer group S** = {I, Z1 Z2, Z2 Z3, Z1 Z3}.
  - This group is **generated** by Z1 Z2 and Z2 Z3, i.e. S = <Z1 Z2, Z2 Z3>.
- This is a subgroup of P3 (the Pauli group on 3 qubits).
- The subspace of P3 **stabilized** by S is spanned by |000> and |111>.
  - These are the codewords for the bit-flip code.

Why the stabilizer formalism?

- Describing codewords themselves is cumbersome with more complicated codes.
    - Stabilizers offer a more succinct representation.
    - Namely, via the generator representation of a group.
- Very convenient abstraction that allows for generalization.
    - Many codes can be described in the stabilizer formalism.
    - Pick a stabilizer and you have your very own code!
- First introduced by Gottesman in his 1996 PhD thesis.

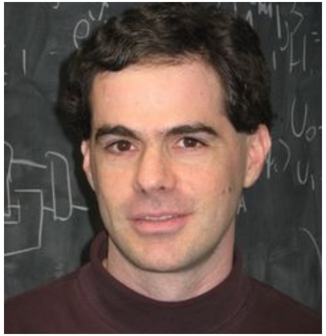


Stabilizer Codes and Quantum Error Correction

Thesis by
Daniel Gottesman

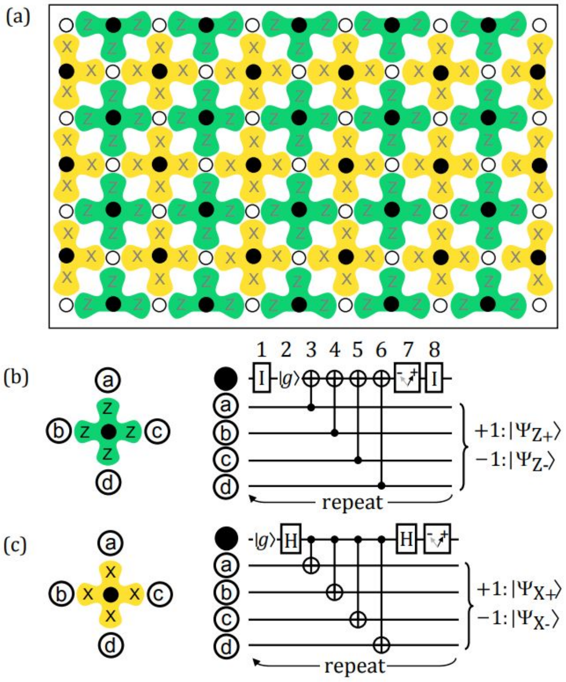
In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

28 May 1997



Current state of affairs

The surface code is a current top candidate.



Eigenvalue	$\hat{Z}_a\hat{Z}_b\hat{Z}_c\hat{Z}_d$	$\hat{X}_a\hat{X}_b\hat{X}_c\hat{X}_d$
+1	$\|gggg\rangle$	$\|++++\rangle$
	$\|ggee\rangle$	$\|++--\rangle$
	$\|geeg\rangle$	$\|+--+\rangle$
	$\|eegg\rangle$	$\|--++\rangle$
	$\|egge\rangle$	$\|-++-\rangle$
	$\|gege\rangle$	$\|+-+-\rangle$
	$\|egeg\rangle$	$\|-+-+\rangle$
	$\|eeee\rangle$	$\|----\rangle$
−1	$\|ggge\rangle$	$\|+++-\rangle$
	$\|ggeg\rangle$	$\|++-+\rangle$
	$\|gegg\rangle$	$\|+-++\rangle$
	$\|eggg\rangle$	$\|-+++\rangle$
	$\|geee\rangle$	$\|+---\rangle$
	$\|egee\rangle$	$\|-+--\rangle$
	$\|eege\rangle$	$\|--+-\rangle$
	$\|eeeg\rangle$	$\|---+\rangle$

Current state of affairs

Four important experimental QEC works:

Fault-Tolerant Operation of a Quantum Error-Correction Code

Laird Egan[1,†], Dripto M. Debroy[2], Crystal Noel[1], Andrew Risinger[1], Daiwei Zhu[1], Debopriyo Biswas[1], Michael Newman[3,*], Muyuan Li[5], Kenneth R. Brown[2,3,4,5], Marko Cetina[1,2], and Christopher Monroe[1]

Article | Open Access | Published: 14 July 2021

Exponential suppression of bit or phase errors with cyclic error correction

Google Quantum AI

Nature **595**, 383–387 (2021) | Cite this article

Article | Published: 25 May 2022

Realizing repeated quantum error correction in a distance-three surface code

Sebastian Krinner, Nathan Lacroix, Ants Remm, Agustin Di Paolo, Elie Genois, Catherine Leroux, Christoph Hellings, Stefania Lazar, Francois Swiadek, Johannes Herrmann, Graham J. Norris, Christian Kraglund Andersen, Markus Müller, Alexandre Blais, Christopher Eichler & Andreas Wallraff

Nature **605**, 669–674 (2022) | Cite this article

Suppressing quantum errors by scaling a surface code logical qubit

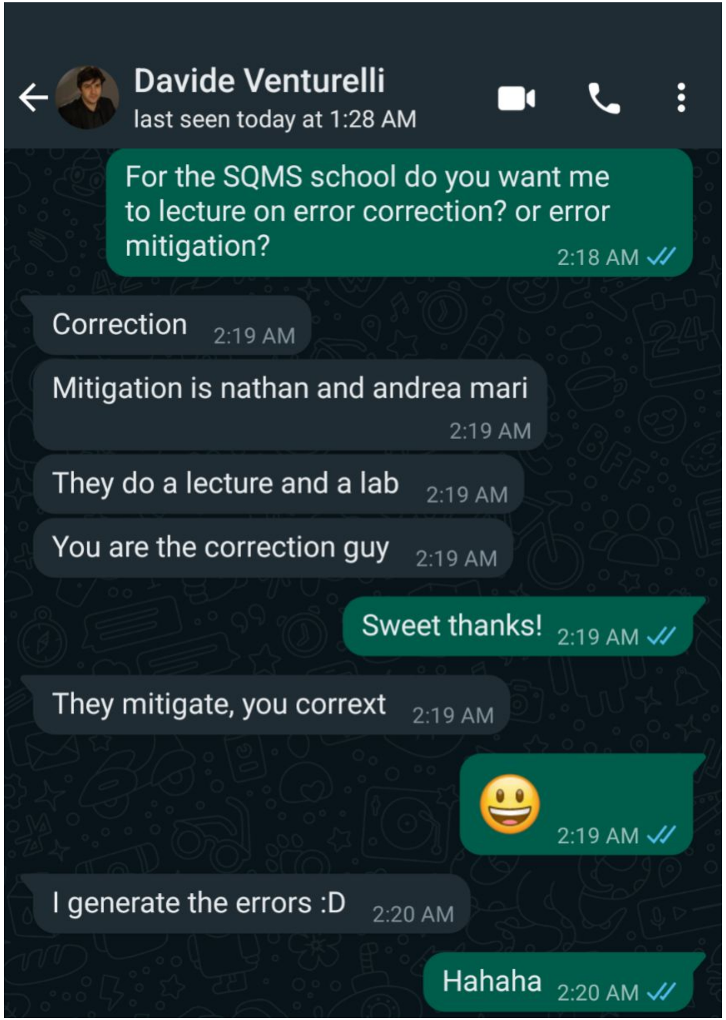
Google Quantum AI*

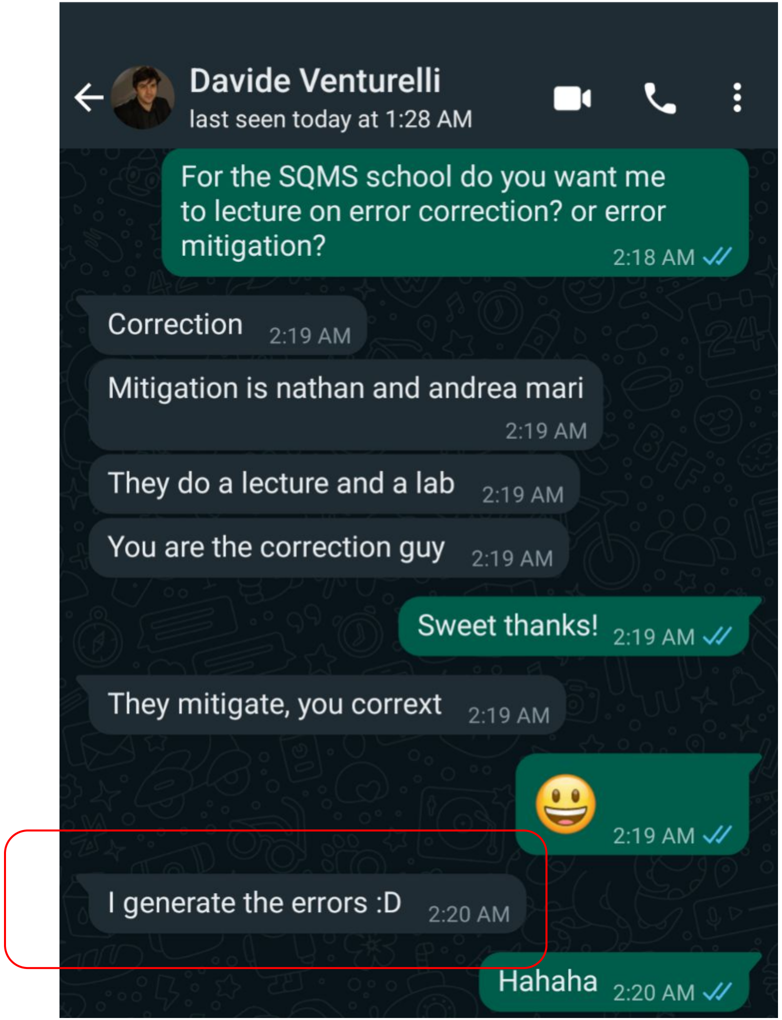
(Dated: July 21, 2022)

Objectives review

1. Define the key elements and principles of quantum error correction.
2. Introduce the stabilizer formalism through the repetition code.
3. Link to recent literature on experimental QEC.

Extras


Davide Venturelli
last seen today at 1:28 AM
For the SQMS school do you want me to lecture on error correction? or error mitigation?
2:18 AM
Correction
2:19 AM
Mitigation is nathan and andrea mari
2:19 AM
They do a lecture and a lab
2:19 AM
You are the correction guy
2:19 AM
Sweet thanks!
2:19 AM
They mitigate, you corrext
2:19 AM
2:19 AM
I generate the errors :D
2:20 AM
Hahaha
2:20 AM


Davide Venturelli
last seen today at 1:28 AM
For the SQMS school do you want me to lecture on error correction? or error mitigation? 2:18 AM
Correction 2:19 AM
Mitigation is nathan and andrea mari 2:19 AM
They do a lecture and a lab 2:19 AM
You are the correction guy 2:19 AM
Sweet thanks! 2:19 AM
They mitigate, you corrext 2:19 AM
2:19 AM
I generate the errors :D 2:20 AM
Hahaha 2:20 AM

Emphasis on Pauli errors

- To emphasize some points in the previous slide(s):
- **We can only consider Pauli errors in QEC without loss of generality.**
- **Further, we can only consider bit flip and phase flip errors WLOG.**

Emphasis on Pauli errors

- To emphasize some points in the previous slide(s):
- **We can only consider Pauli errors in QEC without loss of generality.**
- **Further, we can only consider bit flip and phase flip errors WLOG.**
  - Paulis + identity span $\mathbb{C}^{2\times 2}$
  - Y = i XZ and global phase doesn't matter
  - (Identity is not an error!)

Proof of no cloning

- Suppose there exists a U such that $$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$$

Proof of no cloning

- Suppose there exists a U such that
- If this is for arbitrary states, then

$$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$$
$$U|\phi\rangle|0\rangle = |\phi\rangle|\phi\rangle$$

Proof of no cloning

- Suppose there exists a U such that
- If this is for arbitrary states, then
- Then, by definition,

$$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$$

$$U|\phi\rangle|0\rangle = |\phi\rangle|\phi\rangle$$

$$U(|\psi\rangle + |\phi\rangle)|0\rangle = (|\psi\rangle + |\phi\rangle)(|\psi\rangle + |\phi\rangle)$$

Proof of no cloning

- Suppose there exists a U such that
- If this is for arbitrary states, then
- Then, by definition,

$$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$$
$$U|\phi\rangle|0\rangle = |\phi\rangle|\phi\rangle$$

$$\begin{aligned} U(|\psi\rangle + |\phi\rangle)|0\rangle &= (|\psi\rangle + |\phi\rangle)(|\psi\rangle + |\phi\rangle) \\ &= |\psi\rangle|\psi\rangle + |\psi\rangle|\phi\rangle + |\phi\rangle|\psi\rangle + |\phi\rangle|\phi\rangle \end{aligned}$$

Proof of no cloning

- Suppose there exists a U such that
- If this is for arbitrary states, then
- Then, by definition,

$$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$$
$$U|\phi\rangle|0\rangle = |\phi\rangle|\phi\rangle$$

$$\begin{aligned} U(|\psi\rangle + |\phi\rangle)|0\rangle &= (|\psi\rangle + |\phi\rangle)(|\psi\rangle + |\phi\rangle) \\ &= |\psi\rangle|\psi\rangle + |\psi\rangle|\phi\rangle + |\phi\rangle|\psi\rangle + |\phi\rangle|\phi\rangle \end{aligned}$$

- However, by linearity,

$$U(|\psi\rangle + |\phi\rangle)|0\rangle = U|\psi\rangle|0\rangle + U|\phi\rangle|0\rangle = |\psi\rangle|\psi\rangle + |\phi\rangle|\phi\rangle$$

Proof of no cloning

- Suppose there exists a U such that
- If this is for arbitrary states, then
- Then, by definition,

$$U|\psi\rangle|0\rangle = |\psi\rangle|\psi\rangle$$
$$U|\phi\rangle|0\rangle = |\phi\rangle|\phi\rangle$$

$$\begin{aligned} U(|\psi\rangle + |\phi\rangle)|0\rangle &= (|\psi\rangle + |\phi\rangle)(|\psi\rangle + |\phi\rangle) \\ &= |\psi\rangle|\psi\rangle + |\psi\rangle|\phi\rangle + |\phi\rangle|\psi\rangle + |\phi\rangle|\phi\rangle \end{aligned}$$

- However, by linearity,

$$U(|\psi\rangle + |\phi\rangle)|0\rangle = U|\psi\rangle|0\rangle + U|\phi\rangle|0\rangle = |\psi\rangle|\psi\rangle + |\phi\rangle|\phi\rangle$$

- By taking the inner product of these equations, we can see there can only exist such a *U* if the states |psi> and |phi> are orthogonal

Understanding the projectors: More detail

- Measurements
  - Set of operators $\{M_i\}$ such that $\sum_i M_i^\dagger M_i = I$
  - Probability of outcome *i* is $p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle$
  - State after obtaining outcome *i* is $M_i|\psi\rangle/\sqrt{p(i)}$

- The encoded state (logical qubit) is $|\bar{\psi}\rangle := \alpha|000\rangle + \beta|111\rangle$
- Suppose no qubit was flipped. (Case 1 out of 4).

Understanding the projectors: More detail

- Measurements
  - Set of operators $\{M_i\}$ such that $\sum_i M_i^\dagger M_i = I$
  - Probability of outcome *i* is $p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle$
  - State after obtaining outcome *i* is $M_i|\psi\rangle/\sqrt{p(i)}$

- The encoded state (logical qubit) is $|\bar{\psi}\rangle := \alpha|000\rangle + \beta|111\rangle$
- Suppose no qubit was flipped. (Case 1 out of 4).
- Then:

$$p(0) = \langle\bar{\psi}|P_0|\bar{\psi}\rangle = \langle\bar{\psi}|\left(|000\rangle\langle000| + |111\rangle\langle111|\right)|\bar{\psi}\rangle = 1$$

Understanding the projectors: More detail

- Measurements
  - Set of operators $\{M_i\}$ such that $\sum_i M_i^\dagger M_i = I$
  - Probability of outcome *i* is $p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle$
  - State after obtaining outcome *i* is $M_i|\psi\rangle/\sqrt{p(i)}$

- The encoded state (logical qubit) is $|\bar{\psi}\rangle := \alpha|000\rangle + \beta|111\rangle$
- Suppose no qubit was flipped. (Case 1 out of 4).
- Then:

$$p(0) = \langle\bar{\psi}|P_0|\bar{\psi}\rangle = \langle\bar{\psi}|\left(|000\rangle\langle000| + |111\rangle\langle111|\right)|\bar{\psi}\rangle = 1$$

$$p(1) = \langle\bar{\psi}|P_1|\bar{\psi}\rangle = \langle\bar{\psi}|\left(|100\rangle\langle100| + |011\rangle\langle011|\right)|\bar{\psi}\rangle = 0$$

Understanding the projectors: More detail

- Measurements
  - Set of operators $\{M_i\}$ such that $\sum_i M_i^\dagger M_i = I$
  - Probability of outcome *i* is $p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle$
  - State after obtaining outcome *i* is $M_i|\psi\rangle/\sqrt{p(i)}$

- The encoded state (logical qubit) is $|\bar{\psi}\rangle := \alpha|000\rangle + \beta|111\rangle$
- Suppose no qubit was flipped. (Case 1 out of 4).
- Then:

$$p(0) = \langle\bar{\psi}|P_0|\bar{\psi}\rangle = \langle\bar{\psi}|\left(|000\rangle\langle 000| + |111\rangle\langle 111|\right)|\bar{\psi}\rangle = 1$$

$$p(1) = \langle\bar{\psi}|P_1|\bar{\psi}\rangle = \langle\bar{\psi}|\left(|100\rangle\langle 100| + |011\rangle\langle 011|\right)|\bar{\psi}\rangle = 0$$

$$p(2) = p(3) = 0$$

Understanding the projectors: More detail

- Measurements
  - Set of operators $\{M_i\}$ such that $\sum_i M_i^\dagger M_i = I$
  - Probability of outcome *i* is $p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle$
  - State after obtaining outcome *i* is $M_i|\psi\rangle/\sqrt{p(i)}$

- The encoded state (logical qubit) is $|\bar{\psi}\rangle := \alpha|000\rangle + \beta|111\rangle$
- Suppose no qubit was flipped. (Case 1 out of 4).
- Then:

$$p(0) = \langle\bar{\psi}|P_0|\bar{\psi}\rangle = \langle\bar{\psi}|\left(|000\rangle\langle000| + |111\rangle\langle111|\right)|\bar{\psi}\rangle = 1$$

$$p(1) = \langle\bar{\psi}|P_1|\bar{\psi}\rangle = \langle\bar{\psi}|\left(|100\rangle\langle100| + |011\rangle\langle011|\right)|\bar{\psi}\rangle = 0$$

$$p(2) = p(3) = 0$$

State after measuring P_0 is

$$P_0|\bar{\psi}\rangle/\sqrt{p(0)} = |\bar{\psi}\rangle$$

Understanding the projectors: More detail

- Measurements
  - Set of operators $\{M_i\}$ such that $\sum_i M_i^\dagger M_i = I$
  - Probability of outcome *i* is $p(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle$
  - State after obtaining outcome *i* is $M_i|\psi\rangle/\sqrt{p(i)}$

- The encoded state (logical qubit) is $|\bar{\psi}\rangle := \alpha|000\rangle + \beta|111\rangle$
- Suppose no qubit was flipped. (Case 1 out of 4).
- Then:

$$p(0) = \langle\bar{\psi}|P_0|\bar{\psi}\rangle = \langle\bar{\psi}|\left(|000\rangle\langle 000| + |111\rangle\langle 111|\right)|\bar{\psi}\rangle = 1$$

$$p(1) = \langle\bar{\psi}|P_1|\bar{\psi}\rangle = \langle\bar{\psi}|\left(|100\rangle\langle 100| + |011\rangle\langle 011|\right)|\bar{\psi}\rangle = 0$$

$$p(2) = p(3) = 0$$

Do this with the other 3 projectors!

State after measuring P_0 is

$$P_0|\bar{\psi}\rangle/\sqrt{p(0)} = |\bar{\psi}\rangle$$