Correcting coherent errors with the surface code

Sergey Bravyi (IBM)
Matthias Englbrecht (Munich)
Robert Koenig (Munich)
Nolan Peard (MIT, Munich)

QIP 2018, Delft
How to preserve quantum information for a long time?

Build better qubits
How to preserve quantum information for a long time?

Build better qubits

Quantum error correction

- Redundant encoding of quantum states
- Diagnose errors by syndrome measurements
- Syndrome-dependent recovery operations
How to preserve quantum information for a long time?

**Build better qubits**

![Qubit diagram]

**Quantum error correction**

- Redundant encoding of quantum states
- Diagnose errors by syndrome measurements
- Syndrome-dependent recovery operations

Classical simulation of quantum error correction circuits with toy noise models provides insights into how well a given quantum code can perform in practice.
How to preserve quantum information for a long time?

**Build better qubits**

**Quantum error correction**

- Redundant encoding of quantum states
- Diagnose errors by syndrome measurements
- Syndrome-dependent recovery operations

Classical simulation of quantum error correction circuits with toy noise models provides insights into how well a given quantum code can perform in practice.

**This talk:** efficient algorithms for a classical simulation of large-scale QEC circuits
Coherent vs Pauli noise models

Pauli noise: models random errors caused by unwanted interactions with the environment

\[ N_i(\rho) = (1 - \epsilon) \rho + \epsilon_x X \rho X + \epsilon_y Y \rho Y + \epsilon_z Z \rho Z \]

Coherent noise: models systematic errors caused by imprecision in the classical control

\[ N_i(\rho) = U \rho U^\dagger \quad U \in SU(2) \]
Stabilizer-type quantum codes: Clifford encoding/decoding circuits.
Stabilizer-type quantum codes: Clifford encoding/decoding circuits.

Pauli noise is easy to simulate numerically (use Gottesman-Knill)
Stabilizer-type quantum codes: Clifford encoding/decoding circuits.

Pauli noise is easy to simulate numerically (use Gottesman-Knill)

Coherent noise is described by Clifford+SU(2) circuits.
Brute-force simulation requires exponential time.
Special case: surface codes
Special case: **surface codes**

Encodes one logical qubit into \( n = d^2 \) physical qubits with distance \( d \)

\[
\begin{align*}
\text{d = 3} & \quad \text{d = 5} \\
\end{align*}
\]

\[= \text{qubit}\]

Wen, PRL (2003)
Bombin and Martin-Delgado, PRA (2007)
Special case: **surface codes**

Encodes one logical qubit into \( n = d^2 \) physical qubits with distance \( d \)

\[
\begin{align*}
\text{\( d = 3 \)} & \quad \text{Z X} \\
\text{\( d = 5 \)} & \quad \text{ZZZZZ XXXX}
\end{align*}
\]

Wen, PRL (2003)  
Bombin and Martin-Delgado, PRA (2007)
Special case: surface codes

Encodes one logical qubit into \( n = d^2 \) physical qubits with distance \( d \)

Logical (encoded) states are defined as +1 eigenvectors of all stabilizers.
Special case: **surface codes**

Encodes one logical qubit into $n = d^2$ physical qubits with distance $d$

$$X^L = X \otimes d$$
$$Z^L = Z \otimes d$$

logical Pauli operators

= qubit

stabilizers (parity checks):

ZZZZZ

XXXXX

ZZ

XX
Special case: **surface codes**

High error threshold (above 1%) for the Pauli noise.
Syndrome readout requires only nearest-neighbor gates on a grid.
Fowler et al, PRA (2009)

One of the most attractive candidates for an experimental realization

DiCarlo et al. (TU Delft)
Nature Communications 2015
Physical Review Applied 2017

Takita, Corcoles, et al. (IBM)
Nature Communications 2015, PRL 2016

Special case: **surface codes**

High error threshold (above 1%) for the Pauli noise.
Syndrome readout requires only nearest-neighbor gates on a grid.
*Fowler et al, PRA (2009)*

One of the most attractive candidates for an experimental realization

**What about coherent noise?**
Our results

- Large-scale simulation of the surface codes subject to coherent errors such as systematic Z-rotations. Simulated systems with up to 2400 qubits.

- Efficient and exact simulation algorithm. Runtime: $O(d^4)$

- Estimates of the error threshold and the effective logical channel.

\[
\begin{align*}
\text{d = 3} & \quad \text{d = 5}
\end{align*}
\]
Result 1: simulating storage of logical states with coherent errors $U = \exp(i\theta Z)$
Result 1: simulating storage of logical states with coherent errors $U = \exp(i\theta Z)$
Result 1: simulating storage of logical states with coherent errors $U = \exp(i\theta Z)$

- **Input**: distance $d$, angle $\theta$, initial state $\psi^L$
- **Output**: syndrome $s$, final logical state $\phi^L(s)$
Result 1: simulating storage of logical states with coherent errors $U = \exp(i\theta Z)$

- **Input:** distance $d$, angle $\theta$, initial state $\psi^L$
- **Output:** syndrome $s$, final logical state $\phi^L(s)$

The Z-rotation angle can be qubit-dependent.
Result 2: simulating logical state preparation with coherent SU(2) errors
Result 2: simulating logical state preparation with coherent SU(2) errors

Surface code enables preparation of logical-X (or Z) basis states by initializing each physical qubit in the X-basis, measuring the syndrome, and applying a Pauli correction:
Result 2: simulating logical state preparation with coherent SU(2) errors

We simulated a noisy version of this protocol with **errors in the initial state preparation**: 

\[
\begin{align*}
|\psi\rangle \otimes n & \quad \text{measure syndrome } s \\
\text{product state} & \quad \rightarrow \\
\text{apply Pauli correction } C(s) & \quad \rightarrow \\
|\phi^L(s)\rangle & \quad \text{final logical state}
\end{align*}
\]
Result 2: simulating logical state preparation with coherent SU(2) errors

We simulated a noisy version of this protocol with \textit{errors in the initial state preparation}:

\[ |\psi\rangle \otimes n \xrightarrow{\text{measure syndrome } s} \xrightarrow{\text{apply Pauli correction } C(s)} |\phi^L(s)\rangle \]

\textbf{Input}: distance \( d \), initial state \( |\psi\rangle \in \mathbb{C}^2 \)

\textbf{Output}: syndrome \( s \), final state \( \phi^L(s) \)

\textbf{Runtime}: \( O(d^4) \).
Our algorithms rely on a mapping of the surface code to a system of Majorana fermions
Wen (2003)
Kitaev (2005)
Terhal, Hassler, DiVincenzo (2012)
Previous work on simulation of QEC with coherent errors

- Surface code. Simulation by tensor network algorithms (PEPS). Runtime is exponential in the code distance $d$. Works for any single-qubit noise channels 

- Repetition code (1D surface code). Analytic solution.
  Greenbaum and Dutton, Quant. Sci. Technol. 2018

- Repetition code with the circuit-based noise model (noisy encoding/decoding). Simulation by mapping to dynamics of free fermions.
  Suzuki, Fujii, and Koashi, PRL 2017
Outline

- Majorana representation of the surface code
- Sketch of the simulation algorithm
- Numerical results
Majorana fermions

$n$ qubits = $2n$ Majorana modes

\[ c_1 = X_1 \quad c_3 = Z_1 X_2 \quad \ldots \quad c_{2n-1} = Z_1 \cdots Z_{n-1} X_n \]
\[ c_2 = Y_1 \quad c_4 = Z_1 Y_2 \quad \ldots \quad c_{2n} = Z_1 \cdots Z_{n-1} Y_n \]

Commutation rules:

\[ c_p c_q = -c_q c_p \quad \text{if} \quad p \neq q \]
\[ c_p^2 = I \]

Products of Majorana operators $c_p$ form an operator basis of $n$ qubits
Suppose $|\phi\rangle$ is a normalized $n$-qubit state. Define a covariance matrix

$$M_{p,q} = \begin{cases} 
\langle \phi | i c_p c_q | \phi \rangle & \text{if } p \neq q \\
0 & \text{if } p = q 
\end{cases}$$
Suppose $|\phi\rangle$ is a normalized $n$-qubit state. Define a covariance matrix

$$M_{p,q} = \begin{cases} 
\langle \phi | i c_p c_q | \phi \rangle & \text{if } p \neq q \\
0 & \text{if } p = q
\end{cases}$$

We say that $|\phi\rangle$ is a **Gaussian state** if it obeys Wick’s theorem:

$$-\langle \phi | c_p c_q c_r c_s | \phi \rangle = M_{p,q} M_{r,s} - M_{p,r} M_{q,s} + M_{p,s} M_{q,r}$$

(and similar formulas for the higher-order correlators)
Suppose $|\phi\rangle$ is a normalized $n$-qubit state. Define a covariance matrix

$$M_{p,q} = \begin{cases} 
\langle \phi | i c_p c_q | \phi \rangle & \text{if } p \neq q \\
0 & \text{if } p = q
\end{cases}$$

We say that $|\phi\rangle$ is a **Gaussian state** if it obeys Wick’s theorem:

$$-\langle \phi | c_p c_q c_r c_s | \phi \rangle = M_{p,q} M_{r,s} - M_{p,r} M_{q,s} + M_{p,s} M_{q,r}$$

(and similar formulas for the higher-order correlators)

A Gaussian state is fully specified by its covariance matrix $M$. This requires only $O(n^2)$ real parameters.
Fermionic Linear Optics (Knill 2001, DiVincenzo and Terhal 2002)

• Operations that map Gaussian states to Gaussian states.
• Can be efficiently simulated by keeping track of the covariance matrix
**Fermionic Linear Optics** (Knill 2001, DiVincenzo and Terhal 2002)

- Operations that map Gaussian states to Gaussian states.
- Can be efficiently simulated by keeping track of the covariance matrix.

<table>
<thead>
<tr>
<th>FLO operation</th>
<th>Simulation cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>two-mode rotation</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>( \exp(\theta c_p c_q) )</td>
<td></td>
</tr>
</tbody>
</table>
**Fermionic Linear Optics** (Knill 2001, DiVincenzo and Terhal 2002)

- Operations that map Gaussian states to Gaussian states.
- Can be efficiently simulated by keeping track of the covariance matrix.

<table>
<thead>
<tr>
<th>FLO operation</th>
<th>Simulation cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>two-mode rotation ( \exp (\theta c_p c_q) )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>two-mode parity projector ( \frac{1}{2} (I + ic_p c_q) )</td>
<td>( O(n^2) )</td>
</tr>
</tbody>
</table>
**Fermionic Linear Optics** (Knill 2001, DiVincenzo and Terhal 2002)

- Operations that map Gaussian states to Gaussian states.
- Can be efficiently simulated by keeping track of the covariance matrix

<table>
<thead>
<tr>
<th>FLO operation</th>
<th>Simulation cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>two-mode rotation ( \exp(\theta c_p c_q) )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>two-mode parity projector ( \frac{1}{2}(I + ic_p c_q) )</td>
<td>( O(n^2) )</td>
</tr>
<tr>
<td>two-mode initialization ( ic_p c_q</td>
<td>0\rangle =</td>
</tr>
</tbody>
</table>
Majorana representation of the surface code  Wen (2003)

$4n$ Majorana fermions $c_1, \ldots, c_{4n}$  (blue dots)
Majorana representation of the surface code  Wen (2003)

$4n$ Majorana fermions $c_1, \ldots, c_{4n}$ (blue dots)

Edge operators: $\begin{align*}
e & \quad A_e = ic_p c_q \\
p & \quad q
\end{align*}$

Majorana commutation rules imply $A_e A_{e'} = A_{e'} A_e$
Majorana representation of the surface code  Wen (2003)

4n Majorana fermions $c_1, \ldots, c_{4n}$  (blue dots)

Edge operators:

$$A_e = i c_p c_q$$

Majorana commutation rules imply

$$A_e A_{e'} = A_{e'} A_e$$

Strategy for simulating syndrome measurements:
Express each plaquette stabilizer as a product of edge operators $A_e$.
Measure syndromes of the edge operators (two-mode parity measurements).
Each plaquette syndrome is a product of the edge syndromes.
Majorana C4-code:

stabilizer:

\[ S = -c_1 c_2 c_3 c_4 \]

\[ \overline{X} = i c_1 c_4 = i c_2 c_3 \]

\[ \overline{Y} = i c_2 c_4 = -i c_1 c_3 \]

\[ \overline{Z} = i c_3 c_4 = i c_1 c_2 \]
Majorana C4-code:

stabilizer: \[ S = -c_1 c_2 c_3 c_4 \]
\[ \overline{X} = i c_1 c_4 = i c_2 c_3 \]
\[ \overline{Y} = i c_2 c_4 = -i c_1 c_3 \]
\[ \overline{Z} = i c_3 c_4 = i c_1 c_2 \]

- Any logical state is Gaussian
- Any logical (non-Pauli) operator can be realized by Fermionic Linear Optics
plaquette operator
plaquette operator

plaquette operator encoded by C4 code
plaquette operator

encoded by $C_4$ code

edge operators on the boundary of $f$
plaquette operator

$B_f$

plaquette operator encoded by C4 code

edge operators on the boundary of $f$

$\overline{B}_f = A_1 A_2 A_3 A_4$
The same formula applies to X-type plaquette operators
Simulating syndrome measurements: overview

Goal: measure syndromes of all plaquette operators $B_f$ on a product state $|\psi \otimes n\rangle$. 
Simulating syndrome measurements: overview

Goal: measure syndromes of all plaquette operators $B_f$ on a product state $|\psi \otimes n\rangle$.

• Suffices to measure syndromes of C4-encoded plaquette operators $\overline{B_f}$ on the C4-encoded product state $|\overline{\psi} \otimes n\rangle$.
Simulating syndrome measurements: overview

Goal: measure syndromes of all plaquette operators $B_f$ on a product state $|\psi \otimes n\rangle$.

- Suffices to measure syndromes of C4-encoded plaquette operators $\overline{B_f}$ on the C4-encoded product state $|\overline{\psi} \otimes n\rangle$

- Suffices to measure syndromes of edge operators $A_e$ on $|\overline{\psi} \otimes n\rangle$
Simulating syndrome measurements: overview

Goal: measure syndromes of all plaquette operators $B_f$ on a product state $|\psi^\otimes n\rangle$.

- Suffices to measure syndromes of C4-encoded plaquette operators $\overline{B_f}$ on the C4-encoded product state $|\overline{\psi}^\otimes n\rangle$

- Suffices to measure syndromes of edge operators $A_e$ on $|\overline{\psi}^\otimes n\rangle$

- The state $|\overline{\psi}^\otimes n\rangle$ is Gaussian since any logical state of C4 is Gaussian.
Simulating syndrome measurements: overview

Goal: measure syndromes of all plaquette operators $B_f$ on a product state $|\psi^{\otimes n}\rangle$.

- Suffices to measure syndromes of C4-encoded plaquette operators $\overline{B_f}$ on the C4-encoded product state $|\overline{\psi}^{\otimes n}\rangle$

- Suffices to measure syndromes of edge operators $A_e$ on $|\overline{\psi}^{\otimes n}\rangle$

- The state $|\overline{\psi}^{\otimes n}\rangle$ is Gaussian since any logical state of C4 is Gaussian.

- Simulating two-mode parity measurements on a Gaussian state is easy. Runtime $O(n^2)$ per measurement. Overall runtime is $O(n^3)$. 
Runtime can be improved by measuring edge operators in a specific order:

Active modes
Runtime can be improved by measuring edge operators in a specific order:

Tensor product of two-mode states

Tensor product of four-mode states

Active modes
Runtime can be improved by measuring edge operators in a specific order:

At each time step all except for $O(\sqrt{n})$ “active” modes are in a product state. Inactive modes can be removed from the simulator. This reduces the runtime to $O(n^2)$.
How to measure final logical state:
How to measure final logical state:

Claim: once all edge operators have been measured, the reduced state of the four corner modes is the final logical state encoded by the C4-code.
How to measure final logical state:

**Claim:** once all edge operators have been measured, the reduced state of the four corner modes is the final logical state encoded by the C4-code

The logical Bloch vector is determined by expectation values of the “logical” edge operators $\bar{X} = ic_1c_4$, $\bar{Y} = ic_2c_4$, $\bar{Z} = ic_3c_4$
Simulating storage of a logical state: sketch of main ideas

\[ |\psi^L\rangle \xrightarrow{\text{bitwise Z-rotation error}} U^\otimes_n |\psi^L\rangle \xrightarrow{\text{ideal syndrome measurement}} \Pi_s U^\otimes_n |\psi^L\rangle \xrightarrow{\text{Pauli correction } C(s)} |\phi^L(s)\rangle \]

initial logical state

syndrome \( s \)

final logical state
Simulating storage of a logical state: sketch of main ideas

Encode each qubit of the initial logical state by $C_4$. This results in a non-Gaussian state.

Measure syndrome of X-stabilizers by bitwise X-measurements.

Construct a sequence of Gaussian states that results in the same measurement statistics for bitwise X-measurements.
Structure of the logical channel

We shall only consider Z-rotation errors $U = \exp(i\theta Z)$

Syndrome probabilities: $p(s) = ||\Pi_s U^\otimes n |\psi^L\rangle||^2$

- syndrome projector
- initial logical state
Structure of the logical channel

We shall only consider Z-rotation errors $U = \exp(i\theta Z)$

Syndrome probabilities: $p(s) = \|\Pi_s U \otimes n |\psi^L\rangle\|^2$

Lemma (logical rotation angle)
The syndrome probability distribution $p(s)$ does not depend on the initial logical state. The logical qubit undergoes a Z-rotation by some angle $\theta(s)$. 

syndrome projector \hspace{2cm} initial logical state
Structure of the logical channel

\[ \Phi_s^L(\rho) = e^{i\theta(s)\overline{Z}} \rho e^{-i\theta(s)\overline{Z}} \]
Logical error rate: \[ P^L = 2 \sum_s p(s) |\sin \theta(s)| \]

average diamond-norm distance between the logical channel \( \Phi_s^L \) and the identity
Logical error rate: \( P^L = 2 \sum_s p(s) | \sin \theta(s) | \)

average diamond-norm distance between the logical channel \( \Phi^L_s \) and the identity Pauli correction was computed using the min-weight matching decoder.

The decoder does not depend on \( \theta \) (all edge weights are set to one)
50,000 syndrome samples per data point

Estimated error threshold:
\[ 0.09\pi \leq \theta_0 \leq 0.11\pi \]
Estimated error threshold:

\[ 0.09\pi \leq \theta_0 \leq 0.11\pi \]

50,000 syndrome samples per data point
Pauli Twirl Approximation:
replace coherent physical noise

\[ N(\rho) = e^{i\theta Z} \rho e^{-i\theta Z} \]

by its Pauli twirled version

\[ N_{\text{twirl}}(\rho) = (1 - \epsilon)\rho + \epsilon Z\rho Z \]

\[ \epsilon = \sin^2(\theta) \]

The same error threshold!
PTA underestimates the logical error rate in the sub-threshold regime.

PTA gives a good estimate of the error threshold.
Logical Z-rotation angle: probability distribution

\[ d = 9 \]

\[ d = 25 \]

Pauli-Z error

coherent

\[ d = 9, \theta = 0.08 \cdot \pi \]

\[ d = 25, \theta = 0.08 \cdot \pi \]

coherent
How to quantify coherence of the logical-level noise?
How to quantify coherence of the logical-level noise?

**Full logical channel:**

\[ \Phi_s(\rho) = e^{i\theta(s)Z^L} \rho e^{-i\theta(s)Z^L} \]
How to quantify coherence of the logical-level noise?

**Full logical channel:**

\[
\Phi_s(\rho) = e^{i\theta(s)Z^L} \rho e^{-i\theta(s)Z^L}
\]

**Twirled logical channel:**

(random part of the full channel)

\[
\Phi_s^{\text{twirl}}(\rho) = \cos^2 \theta(s) \rho + \sin^2 \theta(s) Z^L \rho Z^L
\]
How to quantify coherence of the logical-level noise?

**Full logical channel:**
\[
\Phi_s(\rho) = e^{i\theta(s)Z^L} \rho e^{-i\theta(s)Z^L}
\]

**Twirled logical channel:**
(random part of the full channel)
\[
\Phi_{s^{twirl}}(\rho) = \cos^2 \theta(s) \rho + \sin^2 \theta(s) Z^L \rho Z^L
\]

Now we have two different logical error rates:

\[
P^L = \sum_s p(s) \|\Phi_s - \text{Id}\|_\diamond
\]
\[
P_{twirl}^L = \sum_s p(s) \|\Phi_{s^{twirl}} - \text{Id}\|_\diamond \leq P^L
\]
How to quantify coherence of the logical-level noise?

**Full logical channel:**

\[ \Phi_s(\rho) = e^{i\theta(s)Z^L} \rho e^{-i\theta(s)Z^L} \]

**Twirled logical channel:**

(random part of the full channel)

\[ \Phi_{s}^{\text{twirl}}(\rho) = \cos^2 \theta(s) \rho + \sin^2 \theta(s) Z^L \rho Z^L \]

Now we have two different logical error rates:

\[ P^L = \sum_s p(s) \| \Phi_s - \text{Id} \|_\diamond \]

\[ P^{L\text{twirl}} = \sum_s p(s) \| \Phi_{s}^{\text{twirl}} - \text{Id} \|_\diamond \leq P^L \]

We shall use the ratio \( P^L / P_{twirl}^L \) to quantify coherence of the logical noise.
Increasing the code distance makes the logical noise less coherent.

The degree of coherence of the logical channel.

The graph shows the relationship between the Z-rotation angle $\theta/\pi$ and the ratio $P_L/P_{\text{twir}}$. Different curves represent different code distances $d$, with $d = 5$, $d = 9$, $d = 13$, $d = 17$, $d = 21$, and $d = 25$. As the code distance increases, the ratio decreases, indicating that the logical channel becomes less coherent.
Average logical channel

\[
\Phi(\rho) = \sum_s p(s)e^{i\eta(s)Z}\rho e^{-i\eta(s)Z}
\]

Appropriate model if the environment has no access to the measured syndromes.

Quantify coherence of the average logical channel using the ratio of two error rates

\[
P^L = \|\Phi - \text{Id}\|_{\diamond} \quad P^L_{\text{twirl}} = \|\Phi^{\text{twirl}} - \text{Id}\|_{\diamond} \leq P^L
\]
Increasing the code distance makes the logical noise less coherent.

**Conjecture:** the average logical channel has no coherent part in the limit $d \to \infty$. 

The degree of coherence for the average logical channel.
Coherent vs Pauli noise models

Example of a uncorrectable Z-error
Coherent vs Pauli noise models

Example of a uncorrectable Z-error
Minimum weight correction
Example of a uncorrectable Z-error

Minimum weight correction

$N \sim 2^{R/2}$

$R$ = Manhattan distance between $\alpha$ and $\beta$
Coherent vs Pauli noise models

Example of a uncorrectable Z-error

Minimum weight correction

$N \sim 2^{R/2}$

$R = \text{Manhattan distance between } \alpha \text{ and } \beta$

Total probability of all uncorrectable errors connecting $\alpha$ and $\beta$ can be amplified due to a constructive interference.

**Coherent noise:**

$$P_f \sim \left| \sum_{j=1}^{N} A_j \right|^2$$

**Pauli noise:**

$$P_f \sim \sum_{j=1}^{N} |A_j|^2$$
Conclusions

Efficient simulation algorithm for error correction with Z-rotation errors Runtime $O(d^4)$. Simulated systems with up to 2400 qubits.

The observed error threshold is close to $0.1\pi$ which agrees very well with the threshold of the Pauli twirled noise model.

Pauli twirl approximation significantly underestimates the logical error rate in the sub-threshold regime.

Increasing the code distance makes the logical-level noise less coherent.