Topological quantum error correcting codes

David Poulin

Département de Physique
Université de Sherbrooke

Joint work with: H. Bombin, S. Bravyi, G. Duclos-Cianci, O. Landon-Cardinal, and B. Terhal

The University of New Mexico, Albuquerque, February 2013
Error correction is required to protect quantum information from noise affecting qubits.

This is achieved by measuring certain check operators on the system.

Quantum computers will spend most of their time performing error correction.

We want to keep error correction as simple as possible.

In a 2D geometry, checks should be local.
Error correction is required to protect quantum information from noise affecting qubits.

This is achieved by measuring certain check operators on the system.

Quantum computers will spend most of their time performing error correction.

We want to keep error correction as simple as possible.

In a 2D geometry, checks should be local.
Error correction is required to protect quantum information from noise affecting qubits.

This is achieved by measuring certain check operators on the system.

Quantum computers will spend most of their time performing error correction.

We want to keep error correction as simple as possible.

In a 2D geometry, checks should be local.
Error correction is required to protect quantum information from noise affecting qubits.

This is achieved by measuring certain check operators on the system.

Quantum computers will spend most of their time performing error correction.

We want to keep error correction as simple as possible.

In a 2D geometry, checks should be local.
Error correction is required to protect quantum information from noise affecting qubits.

This is achieved by measuring certain check operators on the system.

Quantum computers will spend most of their time performing error correction.

We want to keep error correction as simple as possible.

In a 2D geometry, checks should be local.
Outline

1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Fault tolerance
5. Other topological codes
6. 2D Fault-Tolerant Quantum Cellular Automaton
Outline

1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Fault tolerance
5. Other topological codes
6. 2D Fault-Tolerant Quantum Cellular Automaton
Kitaev’s code

Lattice

- Two-dimensional square lattice
- Periodic boundary conditions
Site operator:
\[ A_s = \prod_{i \in \mathcal{V}(s)} \sigma_x^i \]

Plaquette operator:
\[ B_p = \prod_{i \in \mathcal{V}(p)} \sigma_z^i \]

Hamiltonian:
\[ H = - \left( \sum_s A_s + \sum_p B_p \right) \]
Site operator:
\[ A_s = \prod_{i \in v(s)} \sigma_x^i \]

Plaquette operator:
\[ B_p = \prod_{i \in v(p)} \sigma_z^i \]

Hamiltonian:
\[ H = - (\sum_s A_s + \sum_p B_p) \]
Kitaev’s code

Hamiltonian

- Site operator:
  \[ A_s = \prod_{i \in \nu(s)} \sigma_x^i \]

- Plaquette operator:
  \[ B_p = \prod_{i \in \nu(p)} \sigma_z^i \]

- Hamiltonian:
  \[ H = - (\sum_s A_s + \sum_p B_p) \]
Hamiltonian

\[ H = - (\sum_s A_s + \sum_p B_p) \]

- \([A_s, A_{s'}] = [B_p, B_{p'}] = 0\]
- \([A_s, B_p] = 0\]
- The Hamiltonian is a sum of commuting terms.
  - Exactly solvable
  - Constant gap
- Ground space \(|\psi\rangle\)
  - \(A_s |\psi\rangle = + |\psi\rangle\)
  - \(B_p |\psi\rangle = + |\psi\rangle\)
Hamiltonian

- $H = -(\sum_s A_s + \sum_p B_p)$
- $[A_s, A_{s'}] = [B_p, B_{p'}] = 0$
- $[A_s, B_p] = 0$
- The Hamiltonian is a sum of commuting terms.
  - Exactly solvable
  - Constant gap
- Ground space $|\psi\rangle$
  - $A_s |\psi\rangle = +|\psi\rangle$
  - $B_p |\psi\rangle = +|\psi\rangle$
Hamiltonian

- \( H = -\left( \sum_s A_s + \sum_p B_p \right) \)
- \([A_s, A_{s'}] = [B_p, B_{p'}] = 0\)
- \([A_s, B_p] = 0\)

The Hamiltonian is a sum of commuting terms.

- Exactly solvable
- Constant gap

Ground space \(|\psi\rangle\)

- \( A_s |\psi\rangle = +|\psi\rangle \)
- \( B_p |\psi\rangle = +|\psi\rangle \)
Hamiltonian

\[ H = -\left( \sum_s A_s + \sum_p B_p \right) \]

\[ [A_s, A_{s'}] = [B_p, B_{p'}] = 0 \]

\[ [A_s, B_p] = 0 \]

The Hamiltonian is a sum of commuting terms.

- Exactly solvable
- Constant gap

Ground space \( |\psi\rangle \)

- \( A_s |\psi\rangle = +|\psi\rangle \)
- \( B_p |\psi\rangle = +|\psi\rangle \)
Hamiltonian

\[ H = - \left( \sum_s A_s + \sum_p B_p \right) \]

- \([A_s, A_{s'}] = [B_p, B_{p'}] = 0\)
- \([A_s, B_p] = 0\)
- The Hamiltonian is a sum of commuting terms.
  - Exactly solvable
  - Constant gap
- Ground space \(|\psi\rangle\)
  - \(A_s |\psi\rangle = + |\psi\rangle\)
  - \(B_p |\psi\rangle = + |\psi\rangle\)
String operators

\[ \overline{Z}_1 = \prod_{i \in \gamma_1} \sigma_z^i \]

- \([\overline{Z}_1, B_p] = 0\]
- \([\overline{Z}_1, A_s] = 0\]
- \([\overline{Z}_1, H] = 0\]

\[ [Z_1, B_p] = 0 \]
\[ [Z_1, A_s] = 0 \]
\[ [Z_1, H] = 0 \]
String operators

\[ \overline{Z}_1 = \prod_{i \in \gamma_1} \sigma_z^i \]

\[ [\overline{Z}_1, B_p] = 0 \]

\[ [\overline{Z}_1, A_s] = 0 \]

\[ [\overline{Z}_1, H] = 0 \]
Kitaev’s code

String operators

\[ Z_1 = \prod_{i \in \gamma_1} \sigma_z^i \]

\[ [\overline{Z}_1, B_p] = 0 \]

\[ [\overline{Z}_1, A_s] = 0 \]

\[ [\overline{Z}_1, H] = 0 \]

David Poulin (Sherbrooke)
String operators

- $\overline{Z}_1 = \prod_{i \in \gamma_1} \sigma_z^i$
- $[\overline{Z}_1, B_p] = 0$
- $[\overline{Z}_1, A_s] = 0$
- $[\overline{Z}_1, H] = 0$
Kitaev’s code

String operators

\[ X_1 = \prod_{i \in \gamma_1} \sigma_z^i \]

- \([X_1, B_p] = 0\)
- \([X_1, A_s] = 0\)
- \([\overline{X}_1, H] = 0\)
- \(\{X_1, \overline{Z}_1\} = 0\)
Kitaev’s code

String operators

\[ \overline{X}_1 = \prod_{i \in \gamma_1} \sigma_z^i \]

\[ [\overline{X}_1, B_p] = 0 \]

\[ [\overline{X}_1, A_s] = 0 \]

\[ [\overline{X}_1, H] = 0 \]

\[ \{\overline{X}_1, \overline{Z}_1\} = 0 \]
String operators

\[ X_1 = \prod_{i \in \gamma_1} \sigma_z^i \]
\[ [X_1, B_p] = 0 \]
\[ [X_1, A_s] = 0 \]
\[ [X_1, H] = 0 \]
\[ \{X_1, Z_1\} = 0 \]
Kitaev’s code

String operators

\[ \overline{X}_1 = \prod_{i \in \gamma_1} \sigma^i_z \]

\[ [\overline{X}_1, B_p] = 0 \]

\[ [\overline{X}_1, A_s] = 0 \]

\[ [\overline{X}_1, H] = 0 \]

\[ \{ \overline{X}_1, \overline{Z}_1 \} = 0 \]
Kitaev's code

**String operators**

- $\overline{X}_1 = \prod_{i \in \gamma_1} \sigma_z^i$
- $[\overline{X}_1, B_p] = 0$
- $[\overline{X}_1, A_s] = 0$
- $[\overline{X}_1, H] = 0$
- $\{\overline{X}_1, \overline{Z}_1\} = 0$
By reflecting around the diagonal, we obtain two new symmetry operators:

\[ \{ X_2, Z_2 \} = 0. \]
\[ \{ X_1, Z_1 \} = 0. \]
\[ [ Z_1, Z_2 ] = 0 \]
\[ [ X_1, X_2 ] = 0 \]
\[ [ X_2, Z_1 ] = 0 \]
\[ [ X_1, Z_2 ] = 0 \]
Kitaev’s code

Second set of symmetries

- By reflecting around the diagonal, we obtain two new symmetry operators

\[ \{ \overline{X}_2, \overline{Z}_2 \} = 0. \]

\[ \{ \overline{X}_1, \overline{Z}_1 \} = 0. \]

\[ [\overline{Z}_1, \overline{Z}_2] = 0 \]

\[ [\overline{X}_1, \overline{X}_2] = 0 \]

\[ [\overline{X}_2, \overline{Z}_1] = 0 \]

\[ [\overline{X}_1, \overline{Z}_2] = 0 \]

Two encoded qubits

David Poulin (Sherbrooke)
By reflecting around the diagonal, we obtain two new symmetry operators

\[ \{\overline{X}_2, \overline{Z}_2\} = 0. \]

\[ \{\overline{X}_1, \overline{Z}_1\} = 0. \]

\[ [\overline{Z}_1, \overline{Z}_2] = 0 \]

\[ [\overline{X}_1, \overline{X}_2] = 0 \]

\[ [\overline{X}_2, \overline{Z}_1] = 0 \]

\[ [\overline{X}_1, \overline{Z}_2] = 0 \]
By reflecting around the diagonal, we obtain two new symmetry operators:

\[\{X_2, Z_2\} = 0.\]
\[\{X_1, Z_1\} = 0.\]
\[[Z_1, Z_2] = 0\]
\[[X_1, X_2] = 0\]
\[[X_2, Z_1] = 0\]
\[[X_1, Z_2] = 0\]
By reflecting around the diagonal, we obtain two new symmetry operators

\[
\{\bar{X}_2, \bar{Z}_2\} = 0.
\]

\[
\{\bar{X}_1, \bar{Z}_1\} = 0.
\]

\[
[\bar{Z}_1, \bar{Z}_2] = 0
\]

\[
[\bar{X}_1, \bar{X}_2] = 0
\]

\[
[\bar{X}_2, \bar{Z}_1] = 0
\]

\[
[\bar{X}_1, \bar{Z}_2] = 0
\]
By reflecting around the diagonal, we obtain two new symmetry operators

\[ \{\bar{X}_2, \bar{Z}_2\} = 0. \]
\[ \{\bar{X}_1, \bar{Z}_1\} = 0. \]
\[ [\bar{Z}_1, \bar{Z}_2] = 0 \]
\[ [\bar{X}_1, \bar{X}_2] = 0 \]
\[ [\bar{X}_2, \bar{Z}_1] = 0 \]
\[ [\bar{X}_1, \bar{Z}_2] = 0 \]
Trivial cycles
Trivial cycles
Trivial cycles
Non-trivial cycles
Non-trivial cycles
Non-trivial cycles
Non-trivial cycles
Kitaev's code

Trivial cycles and ground space

\[ H = - \left( \sum_s A_s + \sum_p B_p \right) \]

- The $A_s$ and $B_p$ are trivial cycles.
- Trivial action on ground space
  \[ A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle \]
- $A_s$ and $B_p$ generate all trivial loops.

Trivial loops act trivially on ground space.
Kitaev’s code

Trivial cycles and ground space

- $H = - (\sum_s A_s + \sum_p B_p)$
- The $A_s$ et $B_p$ are trivial cycles
- Trivial action on ground space
  - $A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle$
- $A_s$ $B_p$ generate all trivial loops.

Trivial loops act trivially on ground space
H = −(∑_s A_s + ∑_p B_p)

The $A_s$ et $B_p$ are trivial cycles

Trivial action on ground space

$A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle$

$A_s B_p$ generate all trivial loops.

Trivial loops act trivially on ground space
Kitaev's code

Trivial cycles and ground space

- $H = -(\sum S A_s + \sum p B_p)$
- The $A_s$ et $B_p$ are trivial cycles
- Trivial action on ground space
  $A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle$
- $A_s B_p$ generate all trivial loops.

Trivial loops act trivially on ground space
Kitaev’s code

Trivial cycles and ground space

- $H = - (\sum_s A_s + \sum_p B_p)$
- The $A_s$ et $B_p$ are trivial cycles
- Trivial action on ground space
  $A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle$
- $A_s$ $B_p$ generate all trivial loops.
Kitaev’s code

Trivial cycles and ground space

- \( H = -(\sum_s A_s + \sum_p B_p) \)
- The \( A_s \) et \( B_p \) are trivial cycles
- Trivial action on ground space
  - \( A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle \)
- \( A_s \) \( B_p \) generate all trivial loops.

Trivial loops act trivially on ground space
Kitaev’s code

Trivial cycles and ground space

- \( H = -\left( \sum_s A_s + \sum_p B_p \right) \)
- The \( A_s \) et \( B_p \) are trivial cycles
- Trivial action on ground space
  \( A_s |\psi\rangle = B_p |\psi\rangle = +1 |\psi\rangle \)
- \( A_s \) \( B_p \) generate all trivial loops.

Trivial loops act trivially on ground space
\( \gamma_1 \) and \( \gamma_2 \) wrap around the torus: they are non-trivial cycles.
Kitaev’s code

Gauge choice

\[ |\psi\rangle = B_{p'} |\psi\rangle \]
\[ \overline{Z}_1 |\psi\rangle = \overline{Z}_1 B_{p'} |\psi\rangle \]
\[ \overline{Z}_1 \equiv \overline{Z}_1 B_{p'} \]
\[ \equiv \overline{Z}_1 B_{p'} B_{p''} \]
\[ \equiv \overline{Z}_1 \prod_{p} B_{p} \]
\[
|\psi\rangle = B_{p'} |\psi\rangle
\]
\[
\overline{Z}_1 |\psi\rangle = \overline{Z}_1 B_{p'} |\psi\rangle
\]
\[
\overline{Z}_1 \equiv \overline{Z}_1 B_{p'}
\]
\[
\equiv \overline{Z}_1 B_{p'} B_{p''}
\]
\[
\equiv \overline{Z}_1 \prod_{p} B_p
\]
Kitaev’s code

Gauge choice

- $|\psi\rangle = B_{p'} |\psi\rangle$
- $\overline{Z}_1 |\psi\rangle = \overline{Z}_1 B_{p'} |\psi\rangle$
- $\overline{Z}_1 \equiv \overline{Z}_1 B_{p'}$
  $\equiv \overline{Z}_1 B_{p'} B_{p''}$
  $\equiv \overline{Z}_1 \prod_{p} B_{p}$
Kitaev’s code

Gauge choice

- $|\psi\rangle = B_{p'} |\psi\rangle$
- $\overline{Z}_1 |\psi\rangle = \overline{Z}_1 B_{p'} |\psi\rangle$
- $\overline{Z}_1 \equiv \overline{Z}_1 B_{p'}$
  $\equiv \overline{Z}_1 B_{p'} B_{p''}$
  $\equiv \overline{Z}_1 \prod_p B_p$
Gauge choice

\[ |\psi\rangle = B_{p'} |\psi\rangle \]

\[ \overline{Z}_1 |\psi\rangle = \overline{Z}_1 B_{p'} |\psi\rangle \]

\[ \overline{Z}_1 \equiv \overline{Z}_1 B_{p'} \equiv \overline{Z}_1 B_{p'} B_{p''} \equiv \overline{Z}_1 \prod_p B_p \]
Kitaev’s code

Gauge choice

- \( |\psi\rangle = B_{p'} |\psi\rangle \)
- \( \overline{Z}_1 |\psi\rangle = \overline{Z}_1 B_{p'} |\psi\rangle \)
- \( \overline{Z}_1 \equiv \overline{Z}_1 B_{p'} \)
- \( \equiv \overline{Z}_1 B_{p'} B_{p''} \)
- \( \equiv \overline{Z}_1 \prod_p B_p \)
Kitaev’s code

Gauge choice

\[ |\psi\rangle = B_p' |\psi\rangle \]

\[ \bar{Z}_1|\psi\rangle = \bar{Z}_1 B_p' |\psi\rangle \]

\[ \bar{Z}_1 \equiv \bar{Z}_1 B_p' \]

\[ \equiv \bar{Z}_1 B_p' B_p'' \]

\[ \equiv \bar{Z}_1 \prod_p B_p \]
One degree of freedom associated to each non-trivial cycle.
Operator in same homological class act identically on ground space.
We encode the quantum information is those degrees of freedom:
- The information can only be modified by topologically non-trivial operators.
- Robust when $(\ell \to \infty)$... ?
One degree of freedom associated to each non-trivial cycle.
Operator in same homological class act identically on ground space.
We encode the quantum information is those degrees of freedom:
  The information can only be modified by topologically non-trivial operators.
  Robust when $(\ell \to \infty)$...?
One degree of freedom associated to each non-trivial cycle.

Operator in same homological class act identically on ground space.

We encode the quantum information is those degrees of freedom:
- The information can only be modified by topologically non-trivial operators.
- Robust when \( \ell \to \infty \) ... ?
One degree of freedom associated to each non-trivial cycle.
Operator in same homological class act identically on ground space.
We encode the quantum information is those degrees of freedom:
  The information can only be modified by topologically non-trivial operators.
  Robust when \( \ell \to \infty \)
One degree of freedom associated to each non-trivial cycle.
Operator in same homological class act identically on ground space.
We encode the quantum information is those degrees of freedom:
  - The information can only be modified by topologically non-trivial operators.
  - Robust when \((\ell \to \infty)\)...?
Consider error $E = \sigma_x^i$. 

- $\sigma_x^i$ anti-commutes with adjacent plaquettes.
- $\sigma_x^i|\psi\rangle$ is a -1 eigenstate of $B_p$ and $B_{p'}$.
- Since $H = - (\sum_s A_s + \sum_p B_p)$, $\sigma_x^i$ costs 2 energy units.
- This error has created a pair of magnetic particles.
Consider error $E = \sigma_x^i$.

$\sigma_x^i$ anti-commutes with adjacent plaquettes.

$\sigma_x^i|\psi\rangle$ is a -1 eigenstate of $B_p$ and $B_{p'}$.

Since $H = -\left(\sum_s A_s + \sum_p B_p\right)$, $\sigma_x^i$ costs 2 energy units.

This error has created a pair of magnetic particles.
Consider error $E = \sigma^i_x$.

$\sigma^i_x$ anti-commutes with adjacent plaquettes.

$\sigma^i_x |\psi\rangle$ is a -1 eigenstate of $B_p$ and $B_{p'}$.

Since $H = -(\sum_s A_s + \sum_p B_p)$, $\sigma^i_x$ costs 2 energy units.

This error has created a pair of magnetic particles.
Consider error $E = \sigma_x^i$. 

$\sigma_x^i$ anti-commutes with adjacent plaquettes. 

$\sigma_x^i |\psi\rangle$ is a -1 eigenstate of $B_p$ and $B_p'$. 

Since $H = - (\sum_s A_s + \sum_p B_p)$, 

$\sigma_x^i$ costs 2 energy units. 

This error has created a pair of magnetic particles.
Consider error $E = \sigma^i_x$.

$\sigma^i_x$ anti-commutes with adjacent plaquettes.

$\sigma^i_x |\psi\rangle$ is a -1 eigenstate of $B_p$ and $B_{p'}$.

Since $H = - (\sum_s A_s + \sum_p B_p)$, $\sigma^i_x$ costs 2 energy units.

This error has created a pair of magnetic particles.
New error occurs on neighboring qubit:

- Restores the sign of the middle plaquette
- Flips the sign of the right plaquette

No net energy cost: particle has moved
New error occurs on neighboring qubit:

- Restores the sign of the middle plaquette
- Flips the sign of the right plaquette

No net energy cost: particle has moved
New error occurs on neighboring qubit:

- Restores the sign of the middle plaquette
- Flips the sign of the right plaquette

No net energy cost: particle has moved
New error occurs on neighboring qubit:

- Restores the sign of the middle plaquette
- Flips the sign of the right plaquette

No net energy cost: particle has moved
Error chains are attached to particles, each with given energy.

- Particles can move around at no energy cost.
- Error chains can be stretched freely.
Error chains are attached to particles, each with given energy.

Particles can move around at no energy cost.

Error chains can be stretched freely.
Error chains are attached to particles, each with given energy.

Particles can move around at no energy cost.

Error chains can be stretched freely.
An error can annihilate two particles
The particle’s worldline is left behind after fusion.
Particle fusion can leave behind a worldline corresponding to a logical operation
• An error can annihilate two particles
• The particle’s worldline is left behind after fusion.
• Particle fusion can leave behind a worldline corresponding to a logical operation.
An error can annihilate two particles.
The particle’s worldline is left behind after fusion.
Particle fusion can leave behind a worldline corresponding to a logical operation.
Particle annihilation

- An error can annihilate two particles.
- The particle’s worldline is left behind after fusion.
- Particle fusion can leave behind a worldline corresponding to a logical operation.

Memory corruption
The same story holds for $\sigma_z$ errors.
These will create electrical particles located at the lattice’s vertices (plaquette of dual lattice).
Outline

1 Kitaev’s code

2 Decoding problem

3 Renormalization Group Decoder

4 Fault tolerance

5 Other topological codes

6 2D Fault-Tolerant Quantum Cellular Automaton
Decoding problem

Error syndrome & decoding

- An error produces defects (error syndrome)
  - Measure particle position, but not worldline.
  - Many worldlines consistent with defects.
  - Worldline with different homologies have different effect on ground space: MUST be distinguished.

Decoding
Infer worldline homology from particle location.

15 % Noise rate
An error produces defects (error syndrome)

Measure particle position, but not worldline.

Many worldlines consistent with defects.

Worldline with different homologies have different effect on ground space: MUST be distinguished.

Decoding

Infer worldline homology from particle location.

15 % Noise rate
Decoding problem

Error syndrome & decoding

- An error produces defects (error syndrome)
- Measure particle position, but not worldline.
- Many worldlines consistent with defects.
- Worldline with different homologies have different effect on ground space: MUST be distinguished.

Decoding

Infer worldline homology from particle location.

15 % Noise rate
An error produces defects (error syndrome)

Measure particle position, but not worldline.

Many worldlines consistent with defects.

Worldline with different homologies have different effect on ground space: MUST be distinguished.

Decoding
Infer worldline homology from particle location.
Decoding problem

Error syndrome & decoding

- An error produces defects (error syndrome)
- Measure particle position, but not worldline.
- Many worldlines consistent with defects.
- Worldline with different homologies have different effect on ground space: MUST be distinguished.

Decoding
Infer worldline homology from particle location.
An error produces defects (error syndrome)

Measure particle position, but not worldline.

Many worldlines consistent with defects.

Worldline with different homologies have different effect on ground space: MUST be distinguished.

Decoding

Infer worldline homology from particle location.

15 % Noise rate
Decoding problem

Existing methods

Energy Minimization, Dennis, Kitaev, Landahl, & Preskill ’02

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

Depolarization error model

- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
## Existing methods

### Energy Minimization, Dennis, Kitaev, Landahl, & Preskill ’02
- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

### Depolarization error model
- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
Existing methods

Energy Minimization, Dennis, Kitaev, Landahl, & Preskill ’02

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

Depolarization error model

- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
**Energy Minimization, Dennis, Kitaev, Landahl, & Preskill ’02**

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

**Depolarization error model**

- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
Decoding problem

Existing methods

Energy Minimization, Dennis, Kitaev, Landahl, & Preskill ’02

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

Depolarization error model

- Independent on every qubit.
- No error with probability $1 - p$.
- Error X, Y, or Z with probability $p/3$. 
Existing methods

Energy Minimization, Dennis, Kitaev, Landahl, & Preskill ’02

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

Depolarization error model

- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
Decoding problem

Existing methods

Energy Minimization, Dennis, Kitaev, Landahl, & Preskill ’02

- Find shortest path connecting all defects.
- Equivalent to minimizing energy of random bond Ising model.
- Edmonds’ perfect matching algorithm: $O(\ell^6)$

- It is very slow, $O(\ell^6)$, limited to lattices $\ell \approx 100$.
- It is not optimal:
  - Does not take into account the topological equivalence of errors.
  - Does not take into account correlations between magnetic and electric particles.

Depolarization error model

- Independent on every qubit.
- No error with probability $1 - p$.
- Error $X$, $Y$, or $Z$ with probability $p/3$. 
Outline

1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Fault tolerance
5. Other topological codes
6. 2D Fault-Tolerant Quantum Cellular Automaton
Scale invariance

- Original $B_p$ checks
  - Basis change (row operations on $C$)
  - Obtain scale invariant generators
  - Structure similar to a concatenated code.

Scale-invariant structure
Use Renormalization group techniques

Duclos-Cianci & DP '10
Scale invariance

- Original $B_p$ checks
- Basis change (row operations on $C$)
- Obtain scale invariant generators
- Structure similar to a concatenated code.

Scale-invariant structure
Use Renormalization group techniques

Duclos-Cianci & DP ’10
Scale invariance

- Original $B_p$ checks
- Basis change (row operations on $C$)
- Obtain scale invariant generators
- Structure similar to a concatenated code.

Scale-invariant structure
Use Renormalization group techniques

Duclos-Cianci & DP ’10
Scale invariance

- Original $B_p$ checks
- Basis change (row operations on $C$)
  - Obtain scale invariant generators
  - Structure similar to a concatenated code.

Scale-invariant structure
Use Renormalization group techniques

Duclos-Cianci & DP ’10
Renormalization Group Decoder

Scale invariance

- Original $B_p$ checks
- Basis change (row operations on $C$)
- Obtain scale invariant generators
- Structure similar to a concatenated code.

Scale-invariant structure
Use Renormalization group techniques

Duclos-Cianci & DP ’10
Original $B_p$ checks
Basis change (row operations on $C$)
Obtain scale invariant generators
Structure similar to a concatenated code.

Scale-invariant structure
Use Renormalization group techniques

Duclos-Cianci & DP ’10
Original $B_p$ checks
Basis change
(row operations on $C$)
Obtain scale invariant generators
Structure similar to a concatenated code.

Scale-invariant structure
Use Renormalization group techniques

Duclos-Cianci & DP ’10
**Renormalisation: information coarse graining**

- Break the lattice into sub-lattices (unit cells)
- View each of these unit cell as a tiny topological code encoding two qubits
- Decode this code using brute force:
  - Compute the probability associated to each logical operator $\{I, X, Y, Z\}$.  
- Glue 8 such logical qubits together to obtain a renormalized unit cell
- Some qubits overlap with two unit cells:
  - Impose consistency conditions & solve with belief propagation
Break the lattice into sub-lattices (unit cells)

View each of these unit cell as a tiny topological code encoding two qubits

Decode this code using brute force:
- Compute the probability associated to each logical operator \( \{I, X, Y, Z\}^2 \).

Glue 8 such logical qubits together to obtain a renormalized unit cell

Some qubits overlap with two unit cells:
- Impose consistency conditions & solve with belief propagation
Renormalization Group Decoder

Renormalisation: information coarse graining

- Break the lattice into sub-lattices (unit cells)
- View each of these unit cell as a tiny topological code encoding two qubits
- Decode this code using brute force:
  - Compute the probability associated to each logical operator \( \{I, X, Y, Z\}^2 \).
- Glue 8 such logical qubits together to obtain a renormalized unit cell
- Some qubits overlap with two unit cells:
  - Impose consistency conditions & solve with belief propagation
Renormalization Group Decoder

Renormalisation: information coarse graining

- Break the lattice into sub-lattices (unit cells)
- View each of these unit cell as a tiny topological code encoding two qubits
- Decode this code using brute force:
  - Compute the probability associated to each logical operator \( \{I, X, Y, Z\}^2 \).
- Glue 8 such logical qubits together to obtain a renormalized unit cell
- Some qubits overlap with two unit cells:
  - Impose consistency conditions & solve with belief propagation
Renormalisation: information coarse graining

- Break the lattice into sub-lattices (unit cells)
- View each of these unit cell as a tiny topological code encoding two qubits
- Decode this code using brute force:
  - Compute the probability associated to each logical operator $\{I, X, Y, Z\}^2$.
- Glue 8 such logical qubits together to obtain a renormalized unit cell
- Some qubits overlap with two unit cells:
  - Impose consistency conditions & solve with belief propagation
Threshold $\approx 16.5\%$, compared to $15.5\%$ for PMA.

- Exponentially faster with better performances.
Bit-flip threshold $\approx 8.2\%$, compared to $10.3\%$ for PMA.
Much faster even without parallelization ($10^6$ sites).
Illustrates flexibility.
Outline

1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Fault tolerance
5. Other topological codes
6. 2D Fault-Tolerant Quantum Cellular Automaton
So far, we have assumed that we can perfectly monitor the presence of defects.

- Measurements will themselves be noisy.
- Performing error correction with noisy instruments can kill the computation.
- Can model erroneous measurements by ghost defects appearing with probability $p$.
- Can overcome measurement errors by repeating the measurements periodically.
- Model becomes 2+1 dimensional.
So far, we have assumed that we can perfectly monitor the presence of defects.

Measurements will themselves be noisy.

- Performing error correction with noisy instruments can kill the computation.
- Can model erroneous measurements by ghost defects appearing with probability $p$.
- Can overcome measurement errors by repeating the measurements periodically.
- Model becomes 2+1 dimensional.
So far, we have assumed that we can perfectly monitor the presence of defects.

Measurements will themselves be noisy.

Performing error correction with noisy instruments can kill the computation.

- Can model erroneous measurements by ghost defects appearing with probability $p$.
- Can overcome measurement errors by repeating the measurements periodically.
- Model becomes 2+1 dimensional.
So far, we have assumed that we can perfectly monitor the presence of defects.

Measurements will themselves be noisy.

Performing error correction with noisy instruments can kill the computation.

Can model erroneous measurements by ghost defects appearing with probability $p$.

Can overcome measurement errors by repeating the measurements periodically.

Model becomes 2+1 dimensional.
So far, we have assumed that we can perfectly monitor the presence of defects.

Measurements will themselves be noisy.

Performing error correction with noisy instruments can kill the computation.

Can model erroneous measurements by ghost defects appearing with probability $p$.

Can overcome measurement errors by repeating the measurements periodically.

Model becomes 2+1 dimensional.
Noisy measurements

- So far, we have assumed that we can perfectly monitor the presence of defects.
- Measurements will themselves be noisy.
- Performing error correction with noisy instruments can kill the computation.
- Can model erroneous measurements by ghost defects appearing with probability $p$.
- Can overcome measurement errors by repeating the measurements periodically.
- Model becomes 2+1 dimensional.
Some defects stay put.
Some defects diffuse.
Some charges can fuse.
Some charges can nucleate.
Some defects are missing, and assumed to be there.
Some defects shouldn’t be there, and are ignored.
Some defects stay put.
- Some defects diffuse.
- Some charges can fuse.
- Some charges can nucleate.
- Some defects are missing, and assumed to be there.
- Some defects shouldn’t be there, and are ignored.
Some defects stay put.
Some defects diffuse.
Some charges can fuse.
Some charges can nucleate.
Some defects are missing, and assumed to be there.
Some defects shouldn’t be there, and are ignored.
2+1 world-lines

- Some defects stay put.
- Some defects diffuse.
- Some charges can fuse.
  - Some charges can nucleate.
  - Some defects are missing, and assumed to be there.
  - Some defects shouldn’t be there, and are ignored.
Some defects stay put.
Some defects diffuse.
Some charges can fuse.
Some charges can nucleate.
Some defects are missing, and assumed to be there.
Some defects shouldn’t be there, and are ignored.
Some defects stay put.
Some defects diffuse.
Some charges can fuse.
Some charges can nucleate.
Some defects are missing, and assumed to be there.
Some defects shouldn’t be there, and are ignored.
Some defects stay put.
Some defects diffuse.
Some charges can fuse.
Some charges can nucleate.
Some defects are missing, and assumed to be there.
Some defects shouldn’t be there, and are ignored.
Fault tolerant threshold of roughly 1.8 %

Comparable to the 2.9% reported by Harrington et al. using slow decoders

Duclos-Cianci & DP (unpublished)
Possible generalizations

Why?

- Achieve higher rates $k/n$.
- Achieve higher threshold $p_{th}$.
- Obtain thermal stability.
Defects in Kitaev’s code can move freely, so not thermally stable.
Defects in Kitaev’s code can move freely, so not thermally stable.
Defects in Kitaev’s code can move freely, so not thermally stable.
Defects in Kitaev’s code can move freely, so not thermally stable.
Defects in Kitaev’s code can move freely, so not thermally stable.
Why are classical computers so reliable?

- Defects in Kitaev’s code can move freely, so not thermally stable.
Why are classical computers so reliable?

- Defects in Kitaev’s code can move freely, so not thermally stable.
Why are classical computers so reliable?

Defects in Kitaev’s code can move freely, so not thermally stable.
Why are classical computers so reliable?

- Defects in Kitaev’s code can move freely, so not thermally stable.
Why are classical computers so reliable?

- Defects in Kitaev’s code can move freely, so not thermally stable.
Possible generalizations

Why?

- Achieve higher rates $k/n$.
- Achieve higher threshold $p_{th}$.
- Obtain thermal stability.

Codes with local check operators in a 2D geometry.

- Codes with different generators (other than $A_s = X^\otimes 4$ and $B_p = Z^\otimes 4$ and on different lattice).
  Pauli codes
- Codes whose generators are not Pauli operators.
  Non-Pauli codes
- Codes whose checks do not commute.
  Subsystem codes
- etc.
Other topological codes

Possible generalizations

Why?

- Achieve higher rates $k/n$.
- Achieve higher threshold $p_{th}$.
- Obtain thermal stability.

Codes with local check operators in a 2D geometry.

- Codes with different generators (other than $A_s = X^4$ and $B_p = Z^4$ and on different lattice).
  - **Pauli codes**
    - Codes whose generators are not Pauli operators.
    - Non-Pauli codes
  - Codes whose checks do not commute.
    - Subsystem codes
  - etc.
Possible generalizations

Why?

- Achieve higher rates $k/n$.
- Achieve higher threshold $p_{\text{th}}$.
- Obtain thermal stability.

Codes with local check operators in a 2D geometry.

- Codes with different generators (other than $A_s = X^\otimes 4$ and $B_p = Z^\otimes 4$ and on different lattice).
  - Pauli codes

- Codes whose generators are not Pauli operators.
  - Non-Pauli codes
    - Codes whose checks do not commute.
      - Subsystem codes
    - etc.
Other topological codes

Possible generalizations

Why?

- Achieve higher rates $k/n$.
- Achieve higher threshold $p_{th}$.
- Obtain thermal stability.

Codes with local check operators in a 2D geometry.

- Codes with different generators (other than $A_s = X^\otimes 4$ and $B_p = Z^\otimes 4$ and on different lattice).
  - Pauli codes
- Codes whose generators are not Pauli operators.
  - Non-Pauli codes
- Codes whose checks do not commute.
  - Subsystem codes
  - etc.
Possible generalizations

Why?

- Achieve higher rates $k/n$.
- Achieve higher threshold $p_{th}$.
- Obtain thermal stability.

Codes with local check operators in a 2D geometry.

- Codes with different generators (other than $A_s = X^\otimes 4$ and $B_p = Z^\otimes 4$ and on different lattice).
  - **Pauli codes**
- Codes whose generators are not Pauli operators.
  - **Non-Pauli codes**
- Codes whose checks do not commute.
  - **Subsystem codes**
- etc.
Other topological codes

Local equivalence

Bombin, Duclos-Cianci, & DP ’12

All 2D Pauli codes are locally equivalent to Kitaev’s code.

Bad news

- Cannot achieve higher rates.
- Cannot achieve thermal stability.
  - Bravyi & Terhal ’09

Good news

- Can use existing decoding algorithms (PMA or RG) with any one of them.
  - Bombin & Martin-Delgado ’07 ’08 ’09, Topological color codes
Other topological codes

Local equivalence

Bombin, Duclos-Cianci, & DP ’12

All 2D Pauli codes are locally equivalent to Kitaev’s code.

Bad news

- Cannot achieve higher rates.
- Cannot achieve thermal stability.
  - Bravyi & Terhal ’09

Good news

- Can use existing decoding algorithms (PMA or RG) with any one of them.
  - Bombin & Martin-Delgado ’07 ’08 ’09, Topological color codes
Other topological codes

Local equivalence

### Bombin, Duclos-Cianci, & DP ’12
All 2D Pauli codes are locally equivalent to Kitaev’s code.

### Bad news
- Cannot achieve higher rates.
- Cannot achieve thermal stability.
  - Bravyi & Terhal ’09

### Good news
- Can use existing decoding algorithms (PMA or RG) with any one of them.
  - Bombin & Martin-Delgado ’07 ’08 ’09, Topological color codes
Other topological codes

Subsystem codes

Motivation

- Aharonov & Eldar ’11: Topological order requires 4-qubit commuting checks.
  - Low-weight non-commuting checks possible?
  - Less error-prone

- Bombin ’10, Topological subsystem colour codes
  - Weight=2.
  - Low threshold.

- Bravyi, Duclos-Cianci, DP, Suchara
  - Weight = 3.
  - High threshold.
  - Surface with boundaries.
Other topological codes

Subsystem codes

Motivation

- Aharonov & Eldar ’11: Topological order requires 4-qubit commuting checks.
  - Low-weight non-commuting checks possible?
  - Less error-prone

- Bombin ’10, Topological subsystem colour codes
  - Weight=2.
  - Low threshold.

- Bravyi, Duclos-Cianci, DP, Suchara
  - Weight = 3.
  - High threshold.
  - Surface with boundaries.
Other topological codes

Subsystem codes

Motivation

- Aharonov & Eldar ’11: Topological order requires 4-qubit commuting checks.
  - Low-weight non-commuting checks possible?
  - Less error-prone

- Bombin ’10, Topological subsystem colour codes
  - Weight=2.
  - Low threshold.

- Bravyi, Duclos-Cianci, DP, Suchara
  - Weight = 3.
  - High threshold.
  - Surface with boundaries.
Subsystem codes

Motivation

Aharonov & Eldar '11: Topological order requires 4-qubit commuting checks.
- Low-weight non-commuting checks possible?
- Less error-prone

Bombin '10, Topological subsystem colour codes
- Weight=2.
- Low threshold.

Bravyi, Duclos-Cianci, DP, Suchara
- Weight = 3.
- High threshold.
- Surface with boundaries.
Other topological codes

Thermal stability

Not possible for . . .
- Pauli codes, Bravyi & Terhal ’09
- Non-Pauli codes with local topological order
  Landon-Cardinal & DP (PRL’13)
  - Existence of string operator
    Bravyi & Terhal ’09; Bravyi, DP, Terhal ’10; Haah & Preskill ’12

Unknown for . . .
- Subsystem codes.
- Topological spin-glasses? (Not locally topologically ordered)
Other topological codes

Thermal stability

Not possible for . . .

- Pauli codes, Bravyi & Terhal ’09
- Non-Pauli codes with local topological order
  Landon-Cardinal & DP (PRL’13)
  - Existence of string operator
    Bravyi & Terhal ’09; Bravyi, DP, Terhal ’10; Haah & Preskill ’12

Unknown for . . .

- Subsystem codes.
- Topological spin-glasses? (Not locally topologically ordered)
Other topological codes

Thermal stability

Not possible for . . .

- Pauli codes, Bravyi & Terhal ’09
- Non-Pauli codes with local topological order
  Landon-Cardinal & DP (PRL’13)
  - Existence of string operator
    Bravyi & Terhal ’09; Bravyi, DP, Terhal ’10; Haah & Preskill ’12

Unknown for . . .

- Subsystem codes.
- Topological spin-glasses? (Not locally topologically ordered)
Outline

1. Kitaev’s code
2. Decoding problem
3. Renormalization Group Decoder
4. Fault tolerance
5. Other topological codes
6. 2D Fault-Tolerant Quantum Cellular Automaton
Confinement

Current proposal

- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward

- If particles were attracting each other, they would be confined and none of this would be necessary.
- String tension in 2D?
- Errors would be thermally suppressed (keep system cool).
Confinement

Current proposal

- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward

- If particles were attracting each other, they would be confined and none of this would be necessary.
  - String tension in 2D?
  - Errors would be thermally suppressed (keep system cool).
Confinement

Current proposal
- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward
- If particles were attracting each other, they would be confined and none of this would be necessary.
- String tension in 2D?
- Errors would be thermally suppressed (keep system cool).


Confinement

Current proposal

- Make millions of measurements.
- Send data to classical processor to be analyzed
- Feed information forward

- If particles were attracting each other, they would be confined and none of this would be necessary.
- String tension in 2D?
- Errors would be thermally suppressed (keep system cool).
Simulated confinement

- Take snapshot of particle locations \((x_1, x_2, ...)\)
- Sent this data to an external computer.
- Computer simulates gravitational \(V\) field as if particles were massive.
- Determine how particles would move if gravitationally attracted.
- Move particles accordingly

Locally simulated confinement

Control unit at each site location to

1. Perform syndrome measurements (determine local mass)
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

- Take snapshot of particle locations \((x_1, x_2, \ldots)\)
- Sent this data to an external computer.
  - Computer simulates gravitational V field as if particles were massive.
  - Determine how particles would move if gravitationally attracted.
  - Move particles accordingly

Locally simulated confinement

Control unit at each site location to

1. Perform syndrome measurements (determine local mass)
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

- Take snapshot of particle locations \((x_1, x_2, \ldots)\)
- Send this data to an external computer.
- Computer simulates gravitational \(V\) field as if particles were massive.
  - Determine how particles would move if gravitationally attracted.
  - Move particles accordingly

Locally simulated confinement

Control unit at each site location to

1. Perform syndrome measurements (determine local mass)
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

- Take snapshot of particle locations \((x_1, x_2, \ldots)\)
- Sent this data to an external computer.
- Computer simulates gravitational \(V\) field as if particles were massive.
- Determine how particles would move if gravitationally attracted.
- Move particles accordingly

Locally simulated confinement

Control unit at each site location to:
1. Perform syndrome measurements (determine local mass)
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

- Take snapshot of particle locations \((x_1, x_2, \ldots)\)
- Sent this data to an external computer.
- Computer simulates gravitational \(V\) field as if particles were massive.
- Determine how particles would move if gravitationally attracted.
- Move particles accordingly

Locally simulated confinement

Control unit at each site location to
1. Perform syndrome measurements (determine local mass)
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

- Take snap shot of particle locations \((x_1, x_2, ...)\)
- Sent this data to an external computer.
- Computer simulates gravitational \(V\) field as if particles were massive.
- Determine how particles would move if gravitationally attracted.
- Move particles accordingly

Locally simulated confinement

Control unit at each site location to

1. Perform syndrome measurements (determine local mass)
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

- Take snapshot of particle locations \((x_1, x_2, \ldots)\)
- Sent this data to an external computer.
- Computer simulates gravitational \(V\) field as if particles were massive.
- Determine how particles would move if gravitationally attracted.
- Move particles accordingly

Locally simulated confinement

Control unit at each site location to

1. Perform syndrome measurements (determine local mass)
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
**Simulated confinement**

- Take snapshot of particle locations \((x_1, x_2, \ldots)\)
- Send this data to an external computer.
- Computer simulates gravitational \(V\) field as if particles were massive.
- Determine how particles would move if gravitationally attracted.
- Move particles accordingly

**Locally simulated confinement**

Control unit at each site location to

1. Perform syndrome measurements (determine local mass)
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

- Take snapshot of particle locations \((x_1, x_2, \ldots)\)
- Sent this data to an external computer.
- Computer simulates gravitational \(V\) field as if particles were massive.
- Determine how particles would move if gravitationally attracted.
- Move particles accordingly

Locally simulated confinement

Control unit at each site location to:

1. Perform syndrome measurements (determine local mass)
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Simulated confinement

- Take snapshot of particle locations \((x_1, x_2, \ldots)\)
- Sent this data to an external computer.
- Computer simulates gravitational \(V\) field as if particles were massive.
- Determine how particles would move if gravitationally attracted.
- Move particles accordingly

Locally simulated confinement

Control unit at each site location to

1. Perform syndrome measurements (determine local mass)
2. Simulate a confining potential
   - Exchange messages with neighboring control units
3. Control neighboring qubits
Control unit

- Control unit holds value of local potential $V$ and $\nabla V$
- Measures presence of defect (syndrome)
- Updates potential
  $$\nabla^2 V - \frac{\partial^2}{\partial t^2} V = -\rho$$
- Move particles according to force
  $$F = -\nabla V$$
Control unit

- Control unit holds value of local potential $V$ and $\nabla V$
- Measures presence of defect (syndrome)
- Updates potential
  $\nabla^2 V - \frac{\partial^2}{\partial t^2} V = -\rho$
- Move particles according to force $F = -\nabla V$
Control unit

- Control unit holds value of local potential $V$ and $\nabla V$
- Measures presence of defect (syndrome)
- Updates potential
  \[ \nabla^2 V - \frac{\partial^2}{\partial t^2} V = -\rho \]
- Move particles according to force $F = -\nabla V$
Control unit holds value of local potential $V$ and $\nabla V$

Measures presence of defect (syndrome)

Updates potential
\[ \nabla^2 V - \frac{\partial^2}{\partial t^2} V = -\rho \]

Move particles according to force $F = -\nabla V$
Preliminary result 1D Majorana chain (Ising)

- Assume perfect measurements
- Increase lifetime with system size
Assume perfect measurements
Gravity is a good decoder
Topological codes = local checks for 2D physical qubit layout

- Decoding problem: infer defect worldline homology from "snapshots" of their configuration.
  - RG decoding algorithm
- Local equivalence between Pauli codes
  - Universality of decoding algorithms
  - Extends limitations to other codes
- No thermal stability with local topological order
  - Search for topological spin glasses.
- Possible fault-tolerant 2D quantum cellular automaton
  - Application to non-Abelian models
Topological codes = local checks for 2D physical qubit layout

- Decoding problem: infer defect worldline homology from "snapshots" of their configuration.
  - RG decoding algorithm
- Local equivalence between Pauli codes
  - Universality of decoding algorithms
  - Extends limitations to other codes
- No thermal stability with local topological order
  - Search for topological spin glasses.
- Possible fault-tolerant 2D quantum cellular automaton
  - Application to non-Abelian models
Topological codes = local checks for 2D physical qubit layout

- Decoding problem: infer defect worldline homology from "snapshots" of their configuration.
  - RG decoding algorithm
- Local equivalence between Pauli codes
  - Universality of decoding algorithms
  - Extends limitations to other codes
- No thermal stability with local topological order
  - Search for topological spin glasses.
- Possible fault-tolerant 2D quantum cellular automaton
  - Application to non-Abelian models
Topological codes = local checks for 2D physical qubit layout

- Decoding problem: infer defect worldline homology from "snapshots" of their configuration.
  - RG decoding algorithm
- Local equivalence between Pauli codes
  - Universality of decoding algorithms
  - Extends limitations to other codes
- No thermal stability with local topological order
  - Search for topological spin glasses.
- Possible fault-tolerant 2D quantum cellular automaton
  - Application to non-Abelian models
Topological codes = local checks for 2D physical qubit layout

- Decoding problem: infer defect worldline homology from "snapshots" of their configuration.
  - RG decoding algorithm
- Local equivalence between Pauli codes
  - Universality of decoding algorithms
  - Extends limitations to other codes
- No thermal stability with local topological order
  - Search for topological spin glasses.
- Possible fault-tolerant 2D quantum cellular automaton
  - Application to non-Abelian models
Topological codes = local checks for 2D physical qubit layout

- Decoding problem: infer defect worldline homology from "snapshots" of their configuration.
  - RG decoding algorithm
- Local equivalence between Pauli codes
  - Universality of decoding algorithms
  - Extends limitations to other codes
- No thermal stability with local topological order
  - Search for topological spin glasses.
- Possible fault-tolerant 2D quantum cellular automaton
  - Application to non-Abelian models
Topological codes = local checks for 2D physical qubit layout

- Decoding problem: infer defect worldline homology from "snapshots" of their configuration.
  - RG decoding algorithm
- Local equivalence between Pauli codes
  - Universality of decoding algorithms
  - Extends limitations to other codes
- No thermal stability with local topological order
  - Search for topological spin glasses.
- Possible fault-tolerant 2D quantum cellular automaton
  - Application to non-Abelian models
Topological codes = local checks for 2D physical qubit layout

- Decoding problem: infer defect worldline homology from "snapshots" of their configuration.
  - RG decoding algorithm
- Local equivalence between Pauli codes
  - Universality of decoding algorithms
  - Extends limitations to other codes
- No thermal stability with local topological order
  - Search for topological spin glasses.
- Possible fault-tolerant 2D quantum cellular automaton
  - Application to non-Abelian models