Lecture 19

Dec 3, 2024

Today: Toric Code Correction & Hamiltonian Simulation.

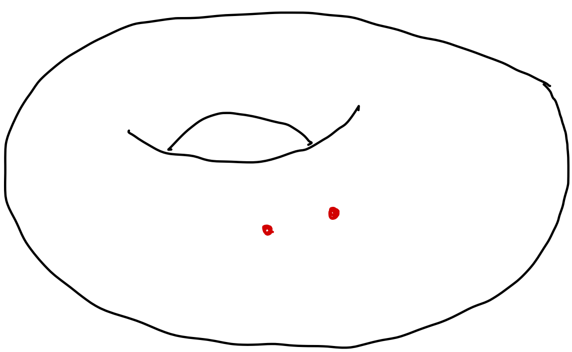
Correction: After measuring the syndrome (i.e. all the X-checks and Z-checks), we want to find the correction that takes us back to the codespace.

Classical repetition code:

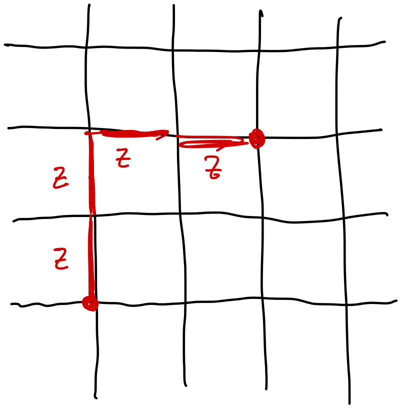
If most bits are 0, correct to 0^n.

If most bits are 1, correct to 1^n

First toric code example:



Assume syndrome measurements were +1 except at these two vertex checks.



X-checks correct Z-errors.

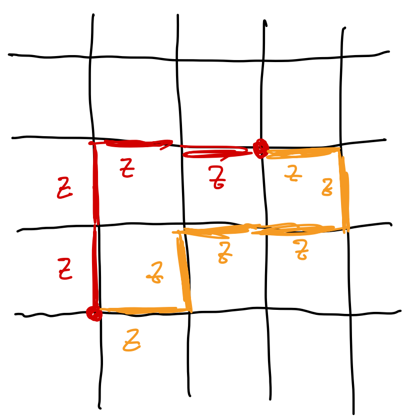
What Pauli error(s) can have this syndrome pattern?

Up to a trivial error, it must be a path connecting the two end points.

How do we know which path?

We don't. All paths are equivalent up to a trivial error.

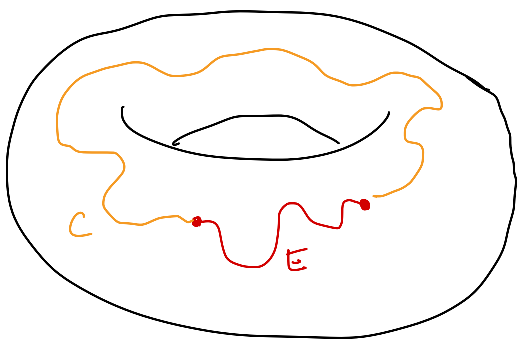
If the true error was in red, and the bad error in orange, correcting by flipping orange,



yields an overall trivial Z-flip as it creates a boundary.

So, short answer is it didn't matter, as long as Error E + Correction C don't generate a non-trivial cycle.

Ex.

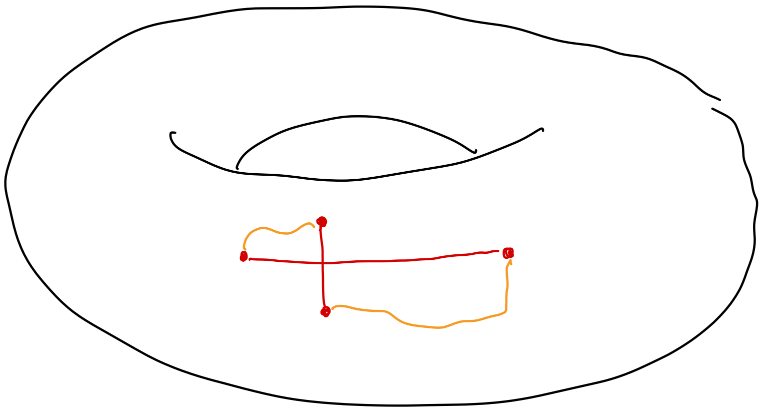


If our correction went around the torus, we would have a problem.

Correction assumes errors of size $< O(\sqrt{n})$, so we should only find corrections that are also short.

Note: computation of the correction can be done from the syndrome by a classical computer.

What if more than 2 vertices flag an error?



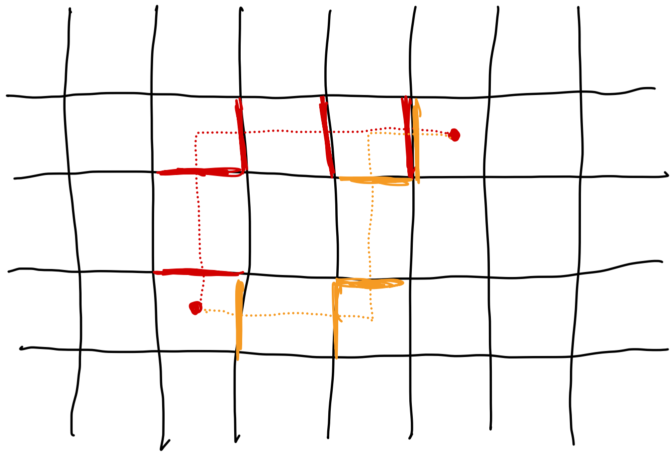
Just pair up flags in such a way that net correction length is $O(\sqrt{n})$.

If true error was red and correction orange, then error is still corrected.

Lastly, what about bit-flip errors and Z-checks flagging?

We look for co-path corrections.

These co-paths are equivalent up to co-boundary.



Hamiltonian Simulation

One of the strongest applications of q. computers is simulating q. mechanics which we believe would take exponential time on a classical computer.

Given a Hamiltonian $H = H(t)$, the evolution of a state is defined by

$$\frac{d}{dt} |\psi(t)\rangle = H(t)\, |\psi(t)\rangle.$$

When H is time-invariant,

$$|\psi(t)\rangle = e^{-iHt} |\psi(0)\rangle.$$

Matrix exponentiation: if $H = \sum_a \lambda_a |a\rangle\langle a|$ ← Spectral decomposition

then $e^{-iHt} = \sum_a e^{-i\lambda_a t} |a\rangle\langle a|$ ← unitary

Furthermore, $e^{-i(\mathcal{U}H\mathcal{U}^\dagger)t} = \mathcal{U} e^{-iHt} \mathcal{U}^\dagger$

Pf. $\mathcal{U}H\mathcal{U}^\dagger = \sum_a \lambda_a \underbrace{\mathcal{U}|a\rangle}\langle a| \mathcal{U}^\dagger$

new eigenvectors

$$e^{-i(\mathcal{U}H\mathcal{U}^\dagger)t} = \sum_a e^{-i\lambda_a t} \mathcal{U}|a\rangle\langle a|\mathcal{U}^\dagger$$

$$= \mathcal{U}\left(\sum_a e^{-i\lambda_a t} |a\rangle\langle a| \right) \mathcal{U}^\dagger . \qquad \blacksquare$$

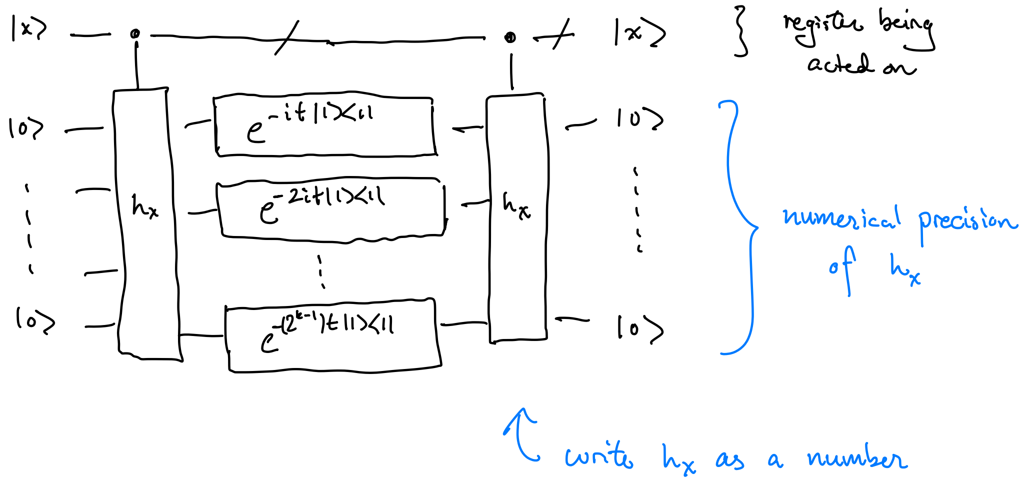
If we can diagonalize a Hamiltonian H s.t. $H = \mathcal{U} H_{diag} \mathcal{U}^\dagger$

then evolving w.r.t. H involves 2 applications of $\mathcal{U}$ + evolving by H_{diag}.

$$H_{diag} = \sum_x h_x |x\rangle\langle x| = \begin{pmatrix} h_0 & & \\ & \ddots & \\ & & h_{2^\ell-1} \end{pmatrix}$$

real.

Circuit for $e^{-iH_{diag}t}$



If $h_x = h_x^{(0)} + 2h_x^{(1)} + 4h_x^{(2)} + \dots + (2^{k-1})h_x^{(k-1)}$

binary decomposition, then we map

$$|x\rangle \mapsto e^{-ih_x^{(0)}t} \cdot e^{-2ih_x^{(1)}t} \cdot \dots \cdot e^{-(2^{k-1})h_x^{(k-1)}t} |x\rangle$$

$$= e^{-ih_x t} |x\rangle$$

When H_{diag} is local, acting on a few qubits, this is implementable.

Furthermore, local Hamiltonian terms h_j can be diagonalized easily. Therefore, we can implement the evolution of local Hamiltonian terms h_j easily.

What about the evolution of $H = \sum h_j$?

Even though each h_j is diagonalizable, the net sum may be hard to diagonalize and would end up a matrix on n-qubits

Solution:

$$e^{-i(h_1+h_2)t} = \lim_{m\to\infty}\left(e^{\frac{-ih_1 t}{m}}\, e^{\frac{-ih_2 t}{m}}\right)^m$$

Lie product formula.

$$\left\| \left(e^{\frac{-ih_1 t}{m}}\, e^{\frac{-ih_2 t}{m}}\right)^m - e^{-i(h_1+h_2)t} \right\| \leq \epsilon$$

for $m = O\left(\frac{(\nu t)^2}{\epsilon}\right)$ where $\nu = \max\{\|h_1\|, \|h_2\|\}$.

Unappealing to evolve t^2 time for t-evolution, but there are speedups to get it to $t^{1+\delta}$ for any $\delta > 0$.

general evolution of $H = \sum_{j=1}^{J} h_j$. Let $m = O\left(\frac{(\nu t)^2}{\epsilon}\right)$

where $\nu = \max\{\|h_1\|, \dots, \|h_J\|\}$ (typically 1 wlog).

① Compute circuit for $e^{-ih_j t'}$ for $t' = t/m$.

using diagonalization + diagonal evolution

② apply $\left(e^{-ih_1 t'} e^{-ih_2 t'} \cdots e^{-ih_J t'}\right)^m$.

Runtime $= \frac{Jm^2}{\epsilon}$ and can be improved to

$\frac{Jm^{1+\delta}}{\epsilon}$.

Other simulable Hamiltonians include sparse Hamiltonians

where H is mostly 0 except poly(n) entries per row.

$\exists$ efficient alg for simulating evolution in this case.