Lecture 28: Tues May 2

Today we'll see a beautiful formalism for quantum error correction that has many roles in quantum computation.

Last time we discussed the **Quantum Fault Tolerance Theorem**, which says that even if *all* qubits in a system have some rate of noise, by:

- doing a bunch of gates in parallel
- applying measurement
- discarding bad qubits and replacing them
- And doing this all hierarchically (i.e. having layers of error atop one another)

we'll *still* be able to do quantum computation, and the cost will be asymptotically reasonable

$T \rightarrow O(T \log^C T)$

This theorem set the research agenda for a lot of experimentalists, who began focusing on attempts to minimize error. Once we can decrease error past a certain threshold, we'll be able to push it arbitrarily small by repeatedly applying our error correction techniques.

The best gauge of how research in quantum computing is going is the *reliability of qubits*. Journalists often ask about things like the number of qubits, or "can you factor 15 into 3 and 5?" but more important is crossing the threshold which would allow us to get arbitrarily small error.

We're not there yet, but lots of progress is being made in two fronts:

1. Making qubits more reliable

   Initially, ε (each qubit's probability of failing at each time step) was close to 1, and the quantum state would barely hold at all. The decoherence rates of IBM's Quantum Experience, for example, wouldn't have been possible ten years ago.

   John Martinez, with Google, has been able to get ε down to 1/1000 with a small number of qubits. That's already past the threshold, but adding more qubits creates more error, so the trick is to find a way to add qubits while keeping error down.

2. Creating better error correction codes

   There are many tradeoffs here. If you used a quantum error correction code that used thousands of physical qubits for each logical qubit, you could get decoherence down to 3-5%.

   We're likely to soon see quantum error correction used to keep a logical qubit alive for longer that the physical qubits below it. People are close to figuring this out, but it's not *quite* there yet.

Stabilizer Circuits

$$\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

are circuits that can be made out of only the gates cNOT, Hadamard, and P =

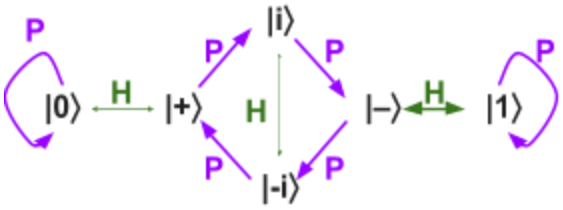
Stabilizer Sets

are sets of states that such a circuit can generate, starting from $|00\ldots0\rangle$.

These came up when we discussed quantum universal gates. It's not obvious that this definition wouldn't cover every quantum state. The Bell Pair is such a state, as are the states arising in Superdense Coding or Quantum Teleportation.

If you play around with these gates, you'll notice that tend to reach a discrete number of states, and never anything between them. You'll also notice that for an arbitrary number of qubits *n*, when these qubits form superpositions over *s* pure states, it follows that $|s| = 2^k$ for some k, and s is always a subspace $s \leq F_2^n$.

With only one qubit, you can only reach 6 states, as is shown (right).
We call these the 1-qubit **stabilizer states**.



What about with two qubits?

You'll find that the states you can reach, like $\frac{|00\rangle + i\,|11\rangle}{\sqrt{2}}$ or $\frac{|01\rangle - i\,|10\rangle}{\sqrt{2}}$, follow a specific pattern: For any non-zero α_x, α_y => $|\alpha_x| = |\alpha_y| = \frac{1}{\sqrt{|S|}}$

In other words, all basis states that occur with non-zero amplitudes have the same absolute value. Measuring any of these in the $|0\rangle,|1\rangle$ basis will either produce: $|0\rangle$ 100% of the time,
$|1\rangle$ 100% of the time,
or a 50-50 chance of producing $|0\rangle$ or $|1\rangle$.

So what gives?
Before we answer that, we need to define a few things.

A unitary U *stabilizes* a pure state $|\Psi\rangle$ if $U|\Psi\rangle = |\Psi\rangle$.

This only holds for positive eigenstates of U. Global phase matters here!
If $U|\Psi\rangle = -|\Psi\rangle$, it does not stabilize $|\Psi\rangle$.

Notice that if U and V both stabilize $|\Psi\rangle$, then any combination of them *also* stabilizes $|\Psi\rangle$. Also, the identity matrix, I, stabilizes everything.

This means that all the unitaries that stabilize $|\Psi\rangle$ form a group.

We already know that unitaries have inverses and are associative.

The next ingredient we need are the **Pauli Matrices**.

These four matrices come up a lot in quantum physics. For example, you can use them to stabilize the Bloch Sphere. They are:

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

Notice that they match up with the errors that can occur.

No error $\quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
$\mathbf{I}|1\rangle = |1\rangle$

Bit flip $\quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
$\mathbf{X}|1\rangle = |0\rangle$

Phase flip $\quad \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$
$\mathbf{Z}|1\rangle = -|1\rangle$

and Both $\quad \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -i \\ 0 \end{pmatrix}$
$\mathbf{Y}|1\rangle = -i|0\rangle$

That's not a coincidence!

The Pauli Matrices satisfy several beautiful identities.

$X^2 = Y^2 = Z^2 = I$ $\quad XY = iZ$ $\quad YX = -iZ$

$YZ = iX$ $\quad ZY = -iX$

$ZX = iY$ $\quad XZ = -iY$

If you've seen the quaternions, you may notice that they satisfy the same kinds of relations.
This is also not a coincidence! Nothing is a coincidence in math!

Also, all of them are unitary and Hermitian.
So what does each of them stabilize?

I stabilizes everything
–I stabilizes nothing Remember: global phase matters, so $-I|\Psi\rangle \neq |\Psi\rangle$.
X stabilizes $|+\rangle$
–X stabilizes $|-\rangle$
Z stabilizes $|0\rangle$
–Z stabilizes $|1\rangle$
Y stabilizes $|i\rangle$
–Y stabilizes $|-i\rangle$

So the six 1-qubit stabilizer states each correspond to a Pauli Matrix.

For a given *n*-qubit pure state $|\Psi\rangle$, we define $|\Psi\rangle$'s **stabilizer group** as:
The group of all tensor products of Pauli Matrices that stabilize $|\Psi\rangle$.
We know this is a group since being Pauli (and being a stabilizer) is closed under multiplication.
Additionally, this group is abelian.

For example, the stabilizer group of $|0\rangle$ is { I, Z } closed because $Z^2 = I$
and that of $|+\rangle$ is { I, X }

The stabilizer group of $|0\rangle \otimes |+\rangle$ will be the product of those groups
{ I⊗I, I⊗X, Z⊗I, Z⊗X } as a convention we omit the ⊗'s { II, IX, ZI, ZX}

For a slightly more interesting example, what's the stabilizer group of a Bell Pair?
We know XX is in it because $\frac{X|0\rangle \otimes X|0\rangle + X|1\rangle \otimes X|1\rangle}{\sqrt{2}} = \frac{|11\rangle + |00\rangle}{\sqrt{2}} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$.
The same argument can be made for –YY.
We can get the last element by doing component-wise multiplication: XX * –YY = –(iZ)(iZ) = ZZ
So the stabilizer group of $\frac{|00\rangle + |11\rangle}{\sqrt{2}}$ is { II, XX, –YY, ZZ }
You can likewise find the stabilizer group of $\frac{|00\rangle - |11\rangle}{\sqrt{2}}$ to be { II, –XX, YY, ZZ }

So now, here's an amazing fact:
The stabilizer states on *n* qubits are exactly the states with a stabilizer group of size 2^n.

We wont see a proof of this, only an intuition for why it's true.

So the 1-qubit stabilizer states are those with 2 elements in their stabilizer group.
The 2-qubit stabilizer states are those with 4 elements in their stabilizer group.
And so forth.

This is a completely different characterization of stabilizer states, a structural one. It tells us what invariant is being preserved without any mention of quantum mechanics.

Furthermore, you make want to know
What is the size of the generation sets?

i.e. the minimum number of elements it would take to produce all others (using multiplication)

It's n ± the tensor product of the Pauli Matrices

So to specify a stabilizer group, you only need to specify a generator state of size n, and this group uniquely determines the state.

(X X)

So for the Bell Pair, you could give (Z Z), which is enough to generate the group { II, XX, –YY, ZZ }

Now we get to a crucial point:
How many qubits are necessary to represent such a stabilizer set?

(Z I X Y)

So given (Y X Z Z) , how many bits of information is this?

(I I X X)

It's O(n), or more specifically: $2n^2 + n$

bits to represent each Pauli —^ ^ ^ number of ± signs to keep track of
|
| number of Pauli Matrices

Writing out the entire group would have otherwise taken 2^n bits. This is (one part of) why the stabilizer formalism is important.

There's an important result from 1999,

The Gottesman-Knill Theorem

which says that stabilizer circuits acting on the all-zero state, $|00\ldots0\rangle$, can be simulated classically in polynomial time.

A more cynical way of interpreting it is to say:
stabilizer sets can't get better-than-polynomial speedups.

This is done by only keeping track of generator sets, and it covers anything you might call "simulating": predicting a measurement between $|0\rangle$, $|1\rangle$ or 50-50 between them, doing a sample over the distribution of possible measurement outcomes, etc.

The one time that Professor Aaronson (being a theorist) ever wrote code that people actually went out and used, was a project in grad school for a Computer Architecture course. He made a fast simulator for stabilizer sets called CHP, letting a normal computer handle thousands of qubits (limited only by their RAM). He was only trying to pass the class, but incidentally published a paper with Gottesman for a better algorithm to implement this.

Truth be told, it had nothing to do with Computer Architecture.
He's not sure with the professor accepted it.

So for a series of qubits starting at |00…0⟩, how do we find all of its stabilizer states?

We know it contains II…I but we wont put that in the generator. It's implied.

We'll also need { ZIII…I
{ IZII…I
{ IIZI…I
:
{ IIII…Z

But this is starting to get messy.

For Gottesman-Knill, it's useful to have another representation of qubits.

Tableau Representation

which keeps track of two matrices of 1's an 0's.

The X Matrix and The Z Matrix

+ (0 0 0 0 | 1 0 0 0)
+ (0 0 0 0 | 0 1 0 0) <= Each row represents a generator state, as a sequence of Paulis
+ (0 0 0 0 | 0 0 1 0)
+ (0 0 0 0 | 0 0 0 1)
^ ^
1 if X or Y 1 if Z or Y
0 otherwise 0 otherwise

Instead of representing each Pauli in a single matrix, it is specified over two bit in separate matrices.
The above matrix represents { ZIII, IZII, IIZI, IIIZ}.

We're going to provide the rules for Tableau Representation without any formal proof that they work, but you can go through each rule and reason through why it makes sense.

We're also going to cheat a little. Keeping track of the +'s and –'s is tricky and not particularly illuminating, so we'll just ignore them. If we only want to know if measuring a qubit will give a definite answer or not (without figuring out if it's a |0⟩ or |1⟩), we can ignore the signs.

So what are the rules?

The gates available to us are cNOT, H, and P, so we need to figure out how to update the tableau for each.

- To apply H on the i^{th} qubit:
  - Swap the i^{th} column of X for the i^{th} column of Z.

This should be intuitive: Hadamard swaps the X and Z bases.

- To apply P on the i^{th} qubit:
  - Take the bitwise XOR of the i^{th} column of X *into* the i^{th} column of Z

Note that P has no effect on the tableau representation of |00…0⟩.
Coincidence? I think not.

- To apply cNOT from the i^{th} qubit to the j^{th}:
  - Take the bitwise XOR of the i^{th} column of X *into* the j^{th} column of X

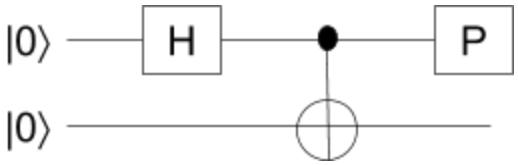
    That seems reasonable enough, *but*... remember from the homework how a cNOT from i–>j in the Hadamard basis is equivalent to a cNOT from j–>i? That means we also have to…
  - Take the bitwise XOR of the j^{th} column of Z *into* the i^{th} column of Z

These rules are enough to establish that measuring the i^{th} qubit in the $|0\rangle,|1\rangle$ basis has a determinate outcome *iff* the i^{th} outcome of the X matrix is all 0's.

Another cool fact:
The number of basis states that our state is a superposition over is 2^k, where k is the rank of the X matrix.

For the above tableau, rank(X) = 0, so it's a superposition of a single state.

Let's test this out, keeping track of the tableau for the following circuit.



We start with

(0 0 | 1 0)
(0 0 | 0 1)

After the Hadamard (swap 1st column of X and Z)

(1 0 | 0 0)
(0 0 | 0 1)

You could convert this back into Paulis by saying the current state is the one generated by (X I) [e.g. top left qubit is 1 in X, 0 in Y => top left is X]
(I Z)

That makes sense since, as we say before, these two are a generator state for $|0\rangle\otimes|+\rangle$

After the cNOT (in X: XOR 1st column into 2nd, in Z: XOR 2nd column into 1st)

(1 1 | 0 0)
(0 0 | 1 1)

This is (X X) the stabilizer generator for the Bell Pair.
(Y Y)

After the phase gate (XOR 1st column of X into 1st column of Z)

(1 1 | 1 0)
(0 0 | 1 1)

A phase gate signifies the introduction of *i*'s. This corresponds to $\frac{|00\rangle + i|11\rangle}{\sqrt{2}}$

Most quantum error correction codes are done with stabilizer circuits, making them easy to compute. As a result, the real importance of the stabilizer formalism is letting us keep track of them in a more elegant way.

For example, with Shor's 9-qubit code, we were dealing with qubits in the form $\left(\frac{|000\rangle + |111\rangle}{\sqrt{2}}\right)^{\otimes 3}$. Since you can flip any two qubit in a grouping and retain the form, we can write this state's generator as:

{ **Z Z** I I I I I I I I,

I **Z** **Z** I I I I I I,

I I I **Z** **Z** I I I I,

I I I I **Z** **Z** I I I,

I I I I I I **Z** **Z** I,

I I I I I I I **Z** **Z,**

X **X** **X** **X** **X** **X** I I I,

I I I **X** **X** **X** **X** **X** **X,**

± **X** **X** **X** **X** **X** **X** **X** **X** **X** }

The last line can have either a + or –, encoding $|\bar{0}\rangle$ or $|\bar{1}\rangle$ respectively

Now we can finally see the 5-qubit error correction code.
The state is impractical to write out explicitly, so it's usually only represented through the stabilizer formalism

{ XZZXI,
IXZZX,
XIXZZ,
ZXIXZ,
± XXXXX }