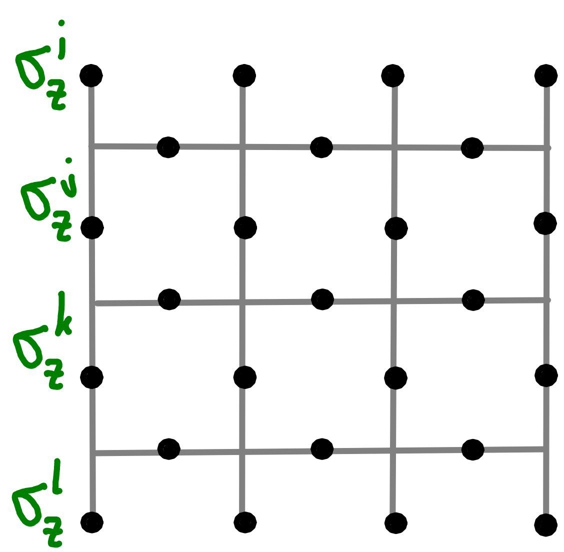
Logical Gates and Error Correction

So far we have assumed that the logical qubit is idle during error correction: it is just waiting to be used.

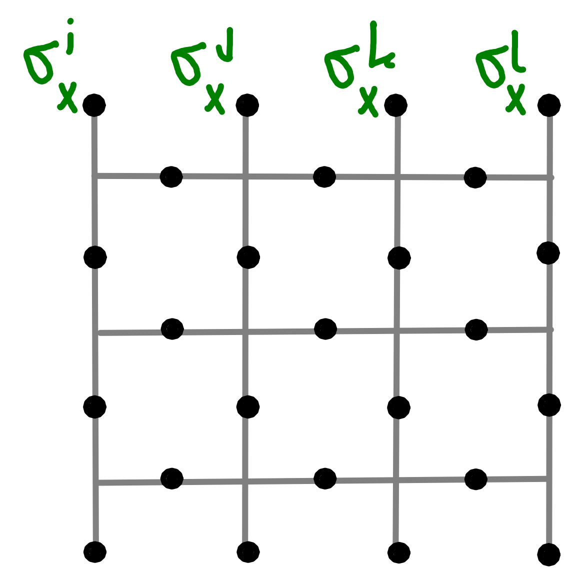
Can we also do logical operations during error correction? Can the error correction also correct imperfections in those operations?

Yes we can! Recall that logical X and Z are performed by single qubit rotations along a chain of physical qubits

$Z =$



$X =$



These could all be done in a single time step, between measurement rounds

Suppose the implementation of these is noisy, adding extra noise p' on the affected qubits

Can all noise be corrected for $p + p' < p_c$?

So far we have considered only uniform noise, with the same strength on all qubits.

In this case the noise strength is $p+p'$ on L qubits, and p on the rest. Does this matter?

After making the syndrome measurements, but before decoding, we could add in extra 'fake' noise with strength p' on the qubits not affected by the logical operator

We know how these would change the syndrome, so we change it accordingly

The decoder would then deal with a syndrome created by uniform noise of strength $p+p'$, and so be highly successful for $p+p'<p_c$.

The effects of the fake noise can then be removed from the correction operator before it is applied

Probably not the best way to decode this noise, but it proves a threshold exists

Transversal Gates

The fault tolerance of these operations comes because they act independently on each qubit of the code. Such operations are called 'transversal' and are of the form

$$U = \bigotimes_j u(j) \qquad \left[U, \prod_v \left(\frac{\mathbb{1} + A_v}{2} \right) \prod_p \left(\frac{\mathbb{1} + B_p}{2} \right) \right] = 0$$

A special kind of transversal operation is those for which the same operation is applied on every qubit

$$U = \prod_j u^j$$

These would be easier to implement, as there would be no need to perform different operations on different qubits

For example, it would be nice if $X = \prod_j \sigma_j^x$

Then a X could be performed by simply applying an x field to the whole code for a while

This is possible for some surface codes

For the planar code, the possible transversal gates are

Z (unitary and measurement)
X (unitary and measurement)
CNOT between logical qubits in two codes
Hadamard (see later)

For each, the transversal property means that their errors will be corrected by the standard error correction

These can do a lot, but are not sufficient to do any possible operation

They are all operations belonging to the Clifford group (see Autumn course)

If we want to do universal quantum computation we need more possible operations. This can be done, but needs more complex error correction methods (Magic state distillation)

Transversal Hadamard

Often we look at the effects unitaries have on the states of an orthonormal basis
(Schrödinger picture)

$U|j\rangle = |\varphi_j\rangle \quad \therefore \quad U = \sum_j |\varphi_j\rangle\langle j|$

z.B. $X|0\rangle = |1\rangle, \quad X|1\rangle = |0\rangle \quad \therefore \quad X = |0\rangle\langle 1| + |1\rangle\langle 0|$

We can also describe then by the effect they have on a complete basis of operators, like the Pauli matrices
(Heisenberg picture)

$U\sigma_x U^\dagger = \tilde{\sigma}_x, \quad U\sigma_z U^\dagger = \tilde{\sigma}_z, \quad U\sigma_y U^\dagger = i\, U\sigma_x U^\dagger U \sigma_z U^\dagger$
$= i\, \tilde{\sigma}_x \tilde{\sigma}_y$

This is the easiest way for us to consider the effects of the Hadamard gate

$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(\sigma_x + \sigma_z)$

$H\sigma_x H^\dagger = \sigma_z$

$H\sigma_z H^\dagger = \sigma_x$

Let's consider the application of this to all physical qubits

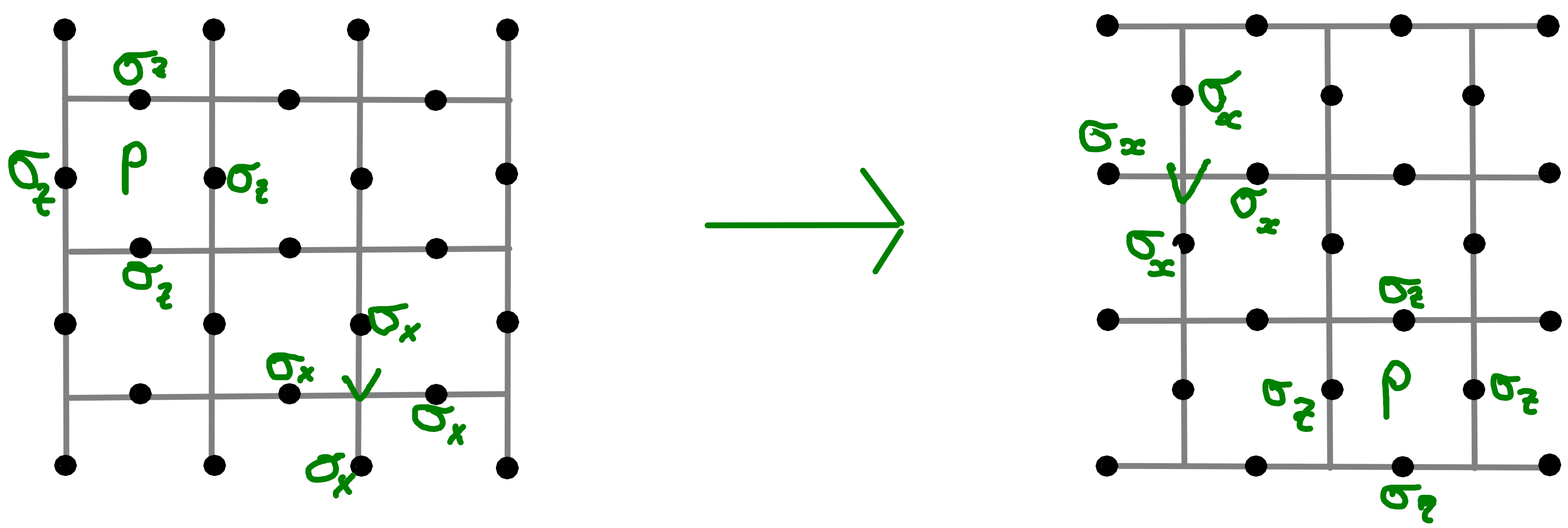
$$U = \prod_j H^j$$

Note that this operation doesn't preserve the stabilizer space, but we'll ignore that for now

If $|\psi\rangle$ is a state of the stabilizer space, what are the properties of $U|\psi\rangle$?

$$A_v|\psi\rangle = |\psi\rangle \quad \therefore \quad UA_vU^\dagger (U|\psi\rangle) = U|\psi\rangle \quad , \text{etc}$$

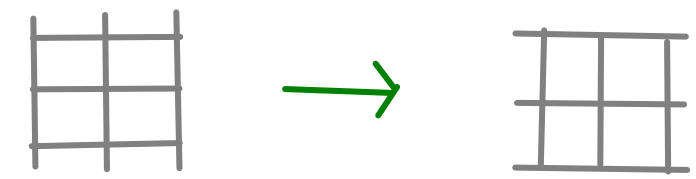
The state $U|\psi\rangle$ is not stabilized by A_v and B_p , but $UA_vU^\dagger$ and $UB_pU^\dagger$



Lattice on the right is redrawn to keep z's around vertices and x's around plaquettes

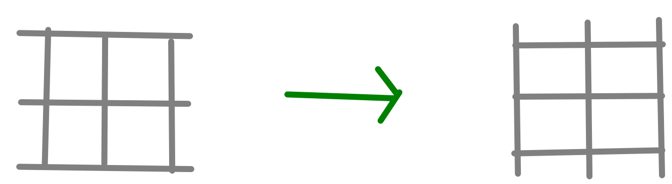
So U has the effect

$A_v \leftrightarrow B_p$, e anyons $\leftrightarrow$ m anyons ,



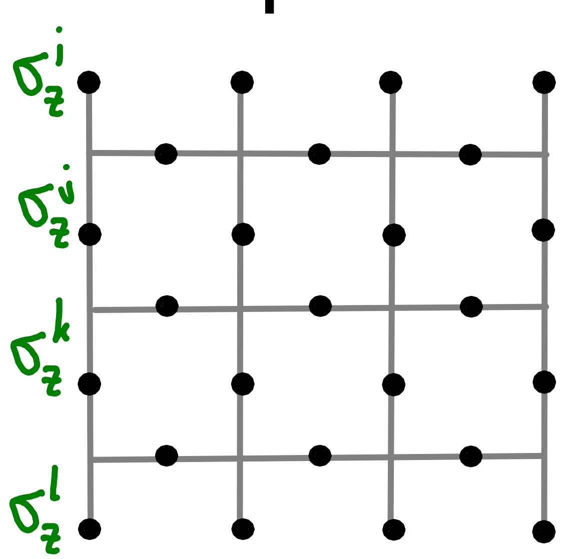
It changes the code to one for which plaquette and vertex stabilizers are swapped, and rotates it by 90 degrees. No wonder it doesn't preserve the stabilizer space!

To get an operation that does preserve the stabilizer space, we just rotate by 90 degrees again

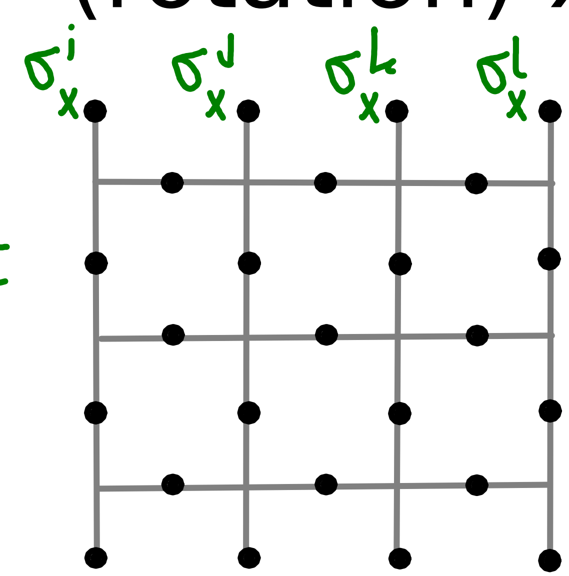


The total operation U' = (rotation) x U has the effect

U'



$U'^\dagger =$



, $\left[U, \prod_v \left(\frac{\mathbb{1}+A_v}{2}\right) \prod_p \left(\frac{\mathbb{1}+B_p}{2}\right)\right] = 0$

And so is a logical Hadamard