~~Add channels in the build\_mpo\_sequence function. Make sure you sum them!~~ marca_de_verificación_blanca

~~Index thing in qiskit. (Getting indices in which gates act in circuits)~~ marca_de_verificación_blanca

~~Come up with some tests~~ marca_de_verificación_blanca

~~Get expectation values for all subcircuits and compare with the full circuit run in Qiskit~~ marca_de_verificación_blanca

~~Fix bug Y, X gates in Hamiltonian~~ marca_de_verificación_blanca

~~Fix poor observable definition~~ Good for now

~~Make code cuter~~

~~Generalize so that it works for knitting of multiple gates~~

Now that we have the framework it would be interesting to see what circuits we can simulate. For this I would again take a Trotter circuit and try to understand

* How big can you make the mpo circuit to still run it efficiently with good accuracy (you can compare with the qiskit MPS simulator if you want)
* Try some small bond dimensions to see how the accuracy changes with the bond dimension
* If you fix the Hamiltonian, how many Trotter steps can you run? Both just with MPO and but also with cutting a gate in every step

The code works but it doesn’t seem to benefit much from using tensor networks. I think the problem is using MPO’s from the start. We should try to avoid MPO’s as much as possible and then use them at the end to introduce the measurements.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media With these settings:

 With MPO simulation

I think it’s probably time for some literature review. How to run quantum circuits efficiently, trying to find if someone has looked at mid-circuit measurements. Another way to do it is to hard-code the measurement and work always with MPS.