Institute for Theoretical Physics
University of Cologne

Prof. Dr. S. Trebst
B. Han, Q. Preiss, M. Yutushui

Quantum Computational Physics

Exercise Sheet 3

Summer Term 2026

Due date: Monday, 18.05.2025 **Discussion**: Tuesday, 19.05.2025

Website: thp.uni-koeln.de/trebst/Lectures/2026-QuantCompPhys.shtml

Exercise 5: Coherence fading into the fog

In the previous exercise sheet (ex. 4) we made use of a small circuit that allowed us to perform a (joint) parity measurement on two qubits (recall Fig. 1 on sheet 2). Today we want to study a much more general circuit which not only allows us to do projective parity measurements (like the ones we used in ex. 4), but also to perform *weak measurements* of the parity. This is achieved by introducing a parameter t (tuning the 'entangling time') in the circuit that, crucially, allows us to tune the measurement strength. In particular, it allows us to go continuously from a projective measurement at $t = \frac{\pi}{4}$ over a weak measurement between 0 and $\frac{\pi}{4}$ to no measurement at all (for $t = 0$).

But let us start with the basics. In Fig. 1 you see a circuit that allows us to perform a joint ZZ parity measurement on two qubits. This circuit does essentially the same as the one we used in ex. 4, but it uses rotational ZZ gates

$$R_{\mathrm{ZZ}}(\theta) = \exp\left(- i\frac{\theta}{2}\mathrm{ZZ}\right) \tag{1}$$

to entangle the system qubits with the ancilla qubit instead of using CNOT gates like in ex. 4. This makes the whole circuit somewhat more complicated, as we also need to correct a relative phase by applying a Z gate based on the measurement outcome of the ancilla qubit. Additionally the measurement outcome of the ancilla qubit is inverted compared to the circuit we used in ex. 4, such that a measurement outcome of +1 corresponds to an odd parity and a measurement outcome of −1 to an even parity. However, besides these differences, the circuit does essentially the same thing as the one we used in ex. 4 – it (projectively) measures the parity of the two system qubits.

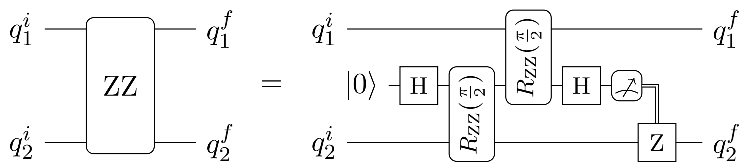

Figure 1 – *Circuit for a ZZ parity measurement.* The $R_{\mathrm{ZZ}}(\theta)$ gate is defined as $R_{\mathrm{ZZ}}(\theta) = \exp(-i\frac{\theta}{2}\mathrm{ZZ})$. Measuring the ancilla qubit will effectively measure the parity of the two qubits, such that a measurement outcome of +1 (ancilla in state $|0\rangle$) corresponds to an odd parity and a measurement outcome of −1 (ancilla in state $|1\rangle$) to an even parity (exactly opposite to the circuit we used in exercise 4). Note that this circuit picks up a global phase, which is however not observable in any experiment / quantum computation.

Now let us use this circuit structure to create a superposition of the two basis states $|00\rangle$ and $|11\rangle$. This is done by preparing the system qubits in the state $|{+}{+}\rangle$ and entangling them with an ancilla qubit (also in the $|+\rangle$ state) using R_{ZZ} gates. Measuring the ancilla qubit in the X basis will now collapse the two system qubits into the state

$$|\psi\rangle = \frac{1}{\sqrt{2}} \cdot \begin{cases} |01\rangle + |10\rangle & \text{if } s = +1, \\ i \cdot (-|00\rangle + |11\rangle) & \text{if } s = -1, \end{cases} \tag{2}$$

where s is the measurement outcome of the ancilla qubit. Using the measurement outcome to control a Pauli-X gate, we can shape the state of the system qubits into the desired superposition of $|00\rangle$ and $|11\rangle$, namely the state

$$|\psi\rangle = \frac{1}{\sqrt{2}} \cdot \begin{cases} |00\rangle + |11\rangle & \text{if } s = +1, \\ i \cdot (-|00\rangle + |11\rangle) & \text{if } s = -1. \end{cases} \tag{3}$$

You can see a graphical representation of this circuit in Fig. 2. Note that the state of the system is not the same for both measurement outcomes of the ancilla qubit (this is because we completely dropped the Z gate from Fig. 1). However, in this exercise we only care about the state being a superposition of $|00\rangle$ and $|11\rangle$, which is the case for both measurement outcomes.

a) Implement the circuit from Fig. 2 in Qiskit. You can use

python
```python
from math import pi
circuit.rzz(pi/2, first_qubit, second_qubit)
```

to apply an $R_{ZZ}(\frac{\pi}{2})$ gate to the `first_qubit` and `second_qubit`. You can use

python
```python
with circuit.if_test((classical_bit, classical_value)):
        circuit.x(system_qubit)
```

to apply a Pauli-X gate to the `system_qubit` if the `classical_bit` is equal to `classical_value`.
Use the `Sampler` to sample the basis states of the system qubits in a simulator and plot the distribution as a histogram (feel free to run the same job on a real quantum computer if you want to!). If done correctly, you should only see basis states with the system qubits in the state $|00\rangle$ or $|11\rangle$ (for the quantum computer you will have noise of course!).

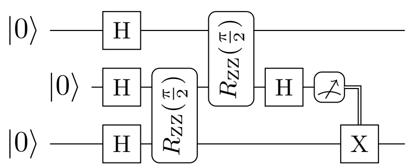

Figure 2 – Short circuit to prepare a two qubit state as a superposition of $|00\rangle$ and $|11\rangle$ using a ZZ parity measurement. The $R_{ZZ}(\theta)$ gate is defined as $R_{ZZ}(\theta) = \exp(-i\frac{\theta}{2}ZZ)$. Measuring the ancilla qubit in the $|1\rangle$ state, will collapse the two system qubits into the state $\frac{i}{\sqrt{2}}(-|00\rangle + |11\rangle)$. Measuring the ancilla qubit in the $|0\rangle$ state however, will collapse the two system qubits into the state $\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$. Using the measurement outcome to control a Pauli-X gate, we can - in runtime - correct the state to be a superposition of $|00\rangle$ and $|11\rangle$ as well, namely the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

Awesome! Now we managed to do a projective parity measurement on two qubits and to prepare a superposition of the basis states $|00\rangle$ and $|11\rangle$. However, we could have done this in the last exercise already (using CNOT gates). So let's now move on to the more interesting part of this exercise – the weak parity measurement. To do this, we need to modify the circuit from Fig. 2 slightly. Instead of using two $R_{ZZ}(\frac{\pi}{2})$ gates, we will now vary the angle in one of these gates to be $\theta = 2t$, where t is a parameter that allows us to tune the measurement strength. A value of $t = \frac{\pi}{4}$ will result in a projective parity measurement, while tuning t away from this value will effectively weaken the measurement. In the limit of $t = 0$, the measurement will have no effect on the parity of the qubits at all, as the $R_{ZZ}(0)$ gate is the identity and therefore does not entangle the system qubit with the ancilla qubit. In Fig. 3 you can see the modified circuit.

b) Implement the circuit from Fig. 3 in Qiskit and use the `Sampler` to sample the basis states of the system qubits in a simulator. Play around with different values of t and plot the distribution as a histogram. What do you observe?

Now we want to make use of the `Estimator` primitive to calculate the expectation value of the ZZ parity operator. You can do this by first defining your operator using the following code:

```python
from qiskit.quantum_info import SparsePauliOp
obs_label = "ZIZ"
obs = SparsePauliOp(obs_label)
```

Note that the middle qubit in this case is the ancilla qubit. We are only interested in the parity of the two system qubits, which is why we use the operator ZIZ.

To define the `Estimator` you can make use of the following code block:

```python
from qiskit_aer import AerSimulator
from qiskit_ibm_runtime import EstimatorV2 as Estimator

aer_simulator = AerSimulator(method="statevector")
simulation_estimator = Estimator(aer_simulator)
```

This set's the simulation method to `"statevector"`, which is exact and works fine for a small number of qubits.

Now we still have to transpile the circuit and map the observable to the circuit structure. If you are unsure how to do that you can look it up in ex. 2c) (sheet 1).

c) Calculate the expectation value of ZZ for different values between $t = 0$ and $t = \frac{\pi}{4}$ and plot them as a function of t. You can use the kwarg `precision` in the `run` method of the `Estimator` to tune the precision of your simulation.

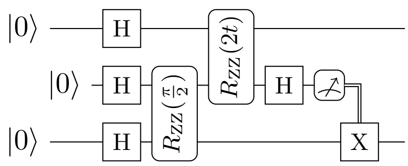

Figure 3 – Same circuit as in Fig. 2, but with a $R_{ZZ}(2t)$ gate instead of a $R_{ZZ}(\frac{\pi}{2})$ gate. This allows us to tune the measurement strenght of the ZZ parity measurement. A value of $t = \frac{\pi}{4}$ results in a projective measurement. Tuning t away from this value will effectively weaken the measurement, such that the state of the system qubits will be less affected by the measurement outcome.

Exercise 6: Preparing something spooky

Our famous friend Einstein found long-range entanglement pretty spooky and called it "spooky action at a distance" (1947 in a letter to Max Born). And, fortunately, we learned in our lectures a perfect example – let us take a closer look at the toric code!

We also know how to run circuits in Qiskit either by simulation or on the real Quantum computer. If you forgot, you can always look it up again in our earlier sheets.

In the lecture we saw that we can create the toric code using measurements. Now we want to apply this to gain a deeper understanding. Let us start with the simplest form: One plaquette.

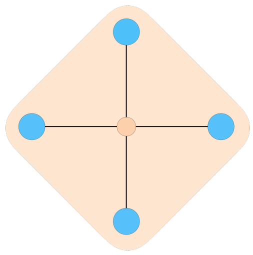

Figure 4 – One four-body measurement for the toric code

On this star we want to measure Z components of all system qubits (blue) at the same time, i.e. in a joint measurement. This can be created in Qiskit rather easy, without caring too much about geometry:

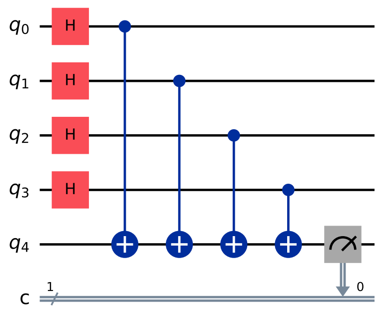

Figure 5 – Circuit for one four-body measurement using CNOT gates

a) Implement this circuit and sample it a few times (as simulation), additionally measuring all qubits in the end (in Z-direction).

b) Now try post-selecting the results i) for when you have measured the ancilla spin as up and ii) for when you have measured it as down. Compare with the equations from the lecture. Do you see the difference? Which state do we want to get?

c) Now measure the qubits in X-direction instead of Z-direction in the end. (*Hint*: If you use the Hadamard gate right before measuring, that effectively rotates the measurement direction.)

Looks much simpler, right? But careful, would we be able to tell the difference from this without knowing the measurement result of the Ancilla?

Next we can go a bit larger. We could go to quite large toric codes that way, however to keep clear and easy to visualize, we will just go one step larger and implement the toric code with two measurements in every row and column ($d = 2$). This gives us 8 system qubits and 4 ancilla qubits.

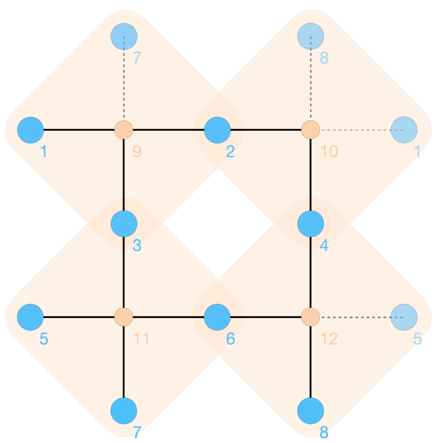

Figure 6 – Toric code by measurement with $d = 2$, meaning two 4-body measurements in every row and column

d) Implement the $d = 2$ toric code in Qiskit and sample it. Take care to also introduce the periodic boundary conditions that the traditional toric code has. We also learned that - without any additional actions - we only properly prepare the toric code when the ancilla measurement outcome has been $+1$. Thus we can directly post-select for the configurations, where all ancilla measurements yielded $+1$. (You can use the `postselect` method that we have learned about on sheet 2)

Do all the terms match what we would expect from the loop gas description? (Even number of + and - measurement results on the system qubits when measuring in the Z-direction.)

e) Sample and post-select the $d = 1$ toric code (a and b) and the $d = 2$ toric code (d) with the same number of samples. Then sum all of the post-selected configurations up and compare the two numbers. Do we get the same amount of valid toric code configurations (configurations with positive ancilla measurement outcomes everywhere)? Is post-selection a valid option for preparing larger toric codes?

f) Try to sample the $d = 1$ toric code circuit (from a and b) on an IBM-chip and see how well you can prepare the toric code this way. Does this scheme/lattice fit to the architecture of the IBM chips or do we need an adaption if we wanted to prepare larger toric codes on these chips?