Institute for Theoretical Physics
University of Cologne

Prof. Dr. S. Trebst
Q. Preiss, F. Eckstein, M. Pütz

# Quantum Computational Physics

## Exercise Sheet 4

Winter Term 2024/25

**Due date**: Tuesday, 26.11.2024 **Discussion**: Tuesday, 19.11.2024

**Website**: thp.uni-koeln.de/trebst/Lectures/2024-QuantCompPhys.shtml

Error correction on classical computers is well-established and active in almost every electronic device we use today. These devices are called *fault tolerant*, meaning that they can operate correctly even if some of their components fail. Designing fault tolerant quantum computers is a major challenge which the field of quantum error correction (QEC) aims to solve. Due to quantum mechanical principles like the *no-cloning theorem*, QEC is much more challenging than classical error correction. In this exercise sheet, we will discuss some approaches to quantum error correction and their limitations. If you are interested in learning more about QEC, you can find a great introduction here (https://arxiv.org/abs/2304.08678).

This sheet is purely analytical (besides some plots) and does not require any programming. But don't worry, we will get back to coding in the next exercise sheet!

**No-cloning Theorem**

The *no-cloning theorem* states that it is impossible to create an exact copy of an arbitrary unknown quantum state. The following proof is based on the invariance of the inner product under unitary operations. Assume that there is a unitary operator $U$ that can clone an arbitrary quantum state $|\psi\rangle$, i.e.

$$U(|\psi\rangle \otimes |0\rangle) = |\psi\rangle \otimes |\psi\rangle\,, \tag{1}$$

where $|0\rangle$ is a blank state. Then the following holds for any two states $|\psi\rangle$ and $|\phi\rangle$:

$$\langle U(\psi \otimes 0) | U(\phi \otimes 0)\rangle = \langle \psi|\phi\rangle \langle \psi|\phi\rangle \tag{2}$$

$$\langle U(\psi \otimes 0) | U(\phi \otimes 0)\rangle = \langle \psi|\phi\rangle \langle 0|0\rangle\,. \tag{3}$$

The first equality follows directly from the definition of $U$, while the second equality follows from the invariance of the inner product under unitary operations. The two equations are not compatible for general states $|\psi\rangle$ and $|\phi\rangle$. Thus the no-cloning theorem holds.

## Exercise 7: Repetiti$^{\otimes n}$ Code

The quantum *repetition code* is an important example of a quantum error-correcting code, as it is arguably the most simple code possible. It is the quantum analogue of the classical repetition code, which encodes a single logical bit $\texttt{b}_\text{L}$ into $n$ physical bits $\texttt{b}_1\texttt{b}_2\ldots\texttt{b}_n$ by repeating it $n$ times. In the correction process, the majority vote is taken to determine the value of the logical bit.

In the quantum case however, preparing $n$ qubits in the same state as our logical qubit is not possible due to the *no-cloning theorem*. Instead, we encode the logical qubit into an $n$-qubit cat state

$$\alpha |0\rangle_{\mathrm{L}} + \beta |1\rangle_{\mathrm{L}} \rightarrow \alpha |0\rangle^{\otimes n} + \beta |1\rangle^{\otimes n} . \tag{4}$$

**a)** Let's say we have a logical qubit in the state $|\psi\rangle_{\mathrm{L}} = \alpha |0\rangle_{\mathrm{L}} + \beta |1\rangle_{\mathrm{L}}$. Draw a circuit that encodes this logical qubit into an $n$-qubit cat state, i.e. $\alpha |0\rangle^{\otimes n} + \beta |1\rangle^{\otimes n}$.

**b)** Now we know how to encode our logical qubit into $n$ physical qubits. As we discussed in the lecture, we can perform parity checks on these physical qubits to detect domain walls in our cat state *without* destroying the state itself. Based on these domain walls, we can perform a correction operation to recover the logical qubit. What is the condition for the correction operation to be successful? Does it make sense to encode a logical qubit into an even number of physical qubits?

**c)** The probabilty for a bit flip error to occur on a single physical qubit is $p$. What is the probability for a logical bit flip error $p_{\mathrm{L}}$ (i.e. the probablity that our correction operation is not successful)? Plot the probablity for a logical bit flip error $p_{\mathrm{L}}$ as a function of $p$ for different numbers of physical qubits. What do you observe?
*Hint:* The probability for a bit flip error on a single physical qubit is $p$, the probability for no error is $1 - p$. What's the probablity for $k$ errors to occur on $n$ physical qubits?

**d)** The here discussed repetition code can correct bit flip errors. Can you think of a way to design a code that can correct phase flip errors? Is there a repetition code that can correct both bit flip and phase flip errors?

**e)** Based on what you have learned about the repetition code so far, would you consider it a good code for practical applications? Why or why not?

## Exercise 8: $[[\mathbf{n}, \mathbf{k}, \mathbf{d}]]$ - notation

In the context of quantum error correction, different codes are often classified by three parameters written in the so called $[[n, k, d]]$-notation. The idea here is the following: $n$ is the number of physical qubits, $k$ is the number of logical qubits, and $d$ is the code distance. One interpretation of the code distance is that a quantum error correction code with code distance $d$ can successfully correct error strings that are shorter than $d/2$. The repetition code we discussed in the previous exercise is an $[[n, 1, n-1]]$ code.

**a)** What values/relations of $n$, $k$, and $d$ are desirably for a quantum error correction code?

In the lecture we learned how to determine how many logical qubits are encoded in a given code, by counting the number of physical system qubits $n$ and the number of independent stabalizers $m$. The number of logical qubits $k$ is then given by $k = n - m$.

**b)** Derive the number of logical qubits $k$ encoded in the toric code with code distance $d = 3$ (Fig. 1) and $d = 5$ (Fig. 2) by explicitly counting the number of physical system qubits $n$ and independent stabalizers $m$.
*Hint:* Feel free to draw the stabilizers into the figures.

**c)** Derive the number of logical qubits $k$ encoded in the rotated surface code with code distance $d = 3$ (Fig. 3) and $d = 5$ (Fig. 4) by explicitly counting the number of physical system qubits $n$ and independent stabalizers $m$.

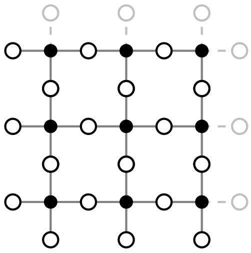

Figure 1 − Visualization of a toric code with code distance $d = 3$.

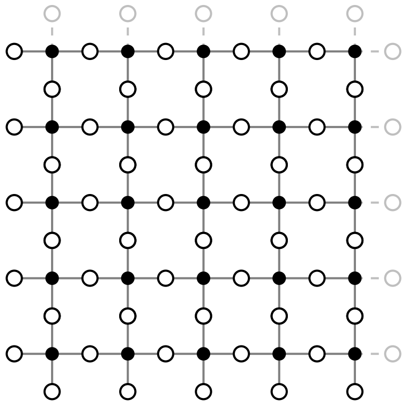

Figure 2 − Visualization of a toric code with code distance $d = 5$.

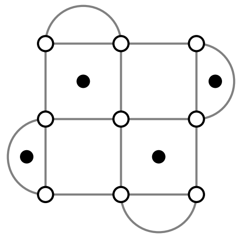

Figure 3 − Visualization of a rotated surface code with code distance $d = 3$.

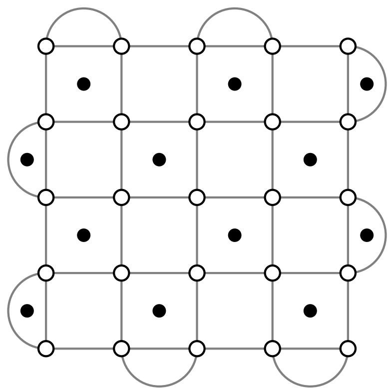

Figure 4 − Visualization of a rotated surface code with code distance $d = 5$.

## Exercise 9: Steane code 🎨

What is the Steane code?

The Steane code is a minimal stabilizer code that uses only seven physical qubits to encode one logical qubit. It is also a representative of a so-called color code (for reasons that will become apparent below). The seven physical qubits (white circles) are placed in the following way:

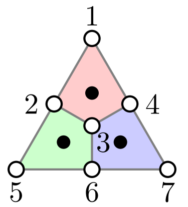


Figure 5 – Visualization of the Steane code

Similar to the toric code we have certain stabilizer measurements that always have to yield +1 to remain in the code space, where we obtain our logical qubit. These stabilizers are marked by the black dots in the middle of quadrangle. Note that each black dot has one X-stabilizer, where X is applied to all qubits of the stabilizer and one Z-stabilizer. Thus the code overall has six stabilizer:

$$\begin{array}{ll} X_1X_2X_3X_4 & Z_1Z_2Z_3Z_4 \\ X_2X_3X_5X_6 & Z_2Z_3Z_5Z_6 \\ X_3X_4X_6X_7 & Z_3Z_4Z_6Z_7 \end{array}$$

Now, why is this code interesting? As mentioned we can encode a logical qubit in this code, with seven we do not need that many and we can do decent error correction already.

For the logical operator of the logical qubits (we want to be able to manipulate the logical qubit after all) we can make different choices. A simple one is to just choose X applied to all as logical X and Z applied to all phyiscial qubits as logical Z:

$$X_L = X_1X_2X_3X_4X_5X_6X_7$$
$$Z_L = Z_1\, Z_2\, Z_3\, Z_4\, Z_5\, Z_6\, Z_7$$

Let's examine a little bit closer how that error correction works.

**a)** What happens if we by random chance apply i) one X operator on qubit 5 ($X_5$)? ii) one Z operator on qubit 2 ($Z_2$)? Can we see in the stabilizers that something has happened?

**b)** If we see something for both $X$ and $Z$ is that different from the repitition code?

**c)** Let's imagine we have the Steane code and want to keep it stable. Now we do not know to which qubit some error has been applied. We only see the stabilizers. Assuming only one qubit has been hit by noise, can we always determine where and what kind it was? Try it: What happened when all other stabilizers are positive and only the following negative?

$$i) \quad \begin{array}{l} X_1X_2X_3X_4 = -1 \\ X_3X_4X_6X_7 = -1 \end{array}$$

$$
\begin{aligned}
ii)\quad Z_1Z_2Z_3Z_4 &= -1 \\
Z_2Z_3Z_5Z_6 &= -1 \\
Z_3Z_4Z_6Z_7 &= -1
\end{aligned}
$$

$$
\begin{aligned}
iii)\quad X_3X_4X_6X_7 &= -1 \\
Z_3Z_4Z_6Z_7 &= -1
\end{aligned}
$$

When we have determined noise it is easy to correct for that operator by just applying it on the physical qubit an additional time.

**d)** Now this was assuming that only one qubit is hit by noise. What happens if two qubits have an error? What happens for example for X errors on 2 and 4 ($X_2X_4$)? If you expect one qubit noise is, will your correction be right? Or will the result of one of the logical operators change?

**e)** Indeed events with two errors pose a problem for the Steane code. So there should not be two errors within the distance of one seven physical qubit triangle. Can you write down the [[n,k,d]] notation of the Steane code?