Instruction Material

Simulating quantum many-body dynamics on a current digital quantum computer

Day 1

Practical Training
Condensed Matter Theory
Technical University Munich
Physics Department

Responsible: Prof. Dr. Michael Knap

Available Dates (Day 1/Day 2):
January 21/22, 2021
May 20/21, 2021

1 Preliminary for the Practical Training

Before you participate in the practical training, there are a few things that might get you prepared

1. Some minimal knowledge about quantum computation will be helpful, If you haven't learnt any quantum computation before, don't worry. A comprehensive reference will be the first chapter of *Quantum Computation and Quantum Information* by Nielsen and Chuang.

2. It would be very helpful if you briefly read through the content for the next day in advance, especially where some new theoretical concepts are introduced. Please also feel free to prepare questions if there are any. During the practical, you will be completing the exercises with help from the instructors.

3. Before we start coding in Cirq, we will give instruction and some time for you to install Cirq. But if you can't wait, you can install the Cirq easily following the instruction here.

4. Please create a **Google account**, which is needed to work with Google Colab (a platform for interactive python notebooks).

5. The practical training consists of Day 1 (basics) and Day 2 (application), you are expected to hand in a written report for **both** Day 1 and Day 2, respectively. You can also choose to only participate on Day 1.

2 Quantum Computation with Superconducting Qubits

Quantum computers are platforms (highly controlled quantum systems) where computation can be performed following the principles of quantum mechanics. Unlike classical computers, quantum computers harness the quantum mechanical properties such as superposition and entanglement. For this reason, quantum computers are expected to be more powerful than the classical computers.
Enormous progress has been made over the last decades in building a universal and practical quantum computer. Nowadays, many quantum platforms are available, such as cold atoms in optical lattices, superconducting qubits and trapped ions, supporting roughly 10$\sim$100 qubits for universal computation. These quantum platforms are *Noisy Intermediate-Scale Quantum* (NISQ) computers. Namely, they are not fault-tolerant and suffer from noise due to interaction with the environment. Among these platforms, superconducting qubits is, for its flexibility and scalability, one of the most popular candidates to build a large-scale quantum computer. In 2019, the milestone, so-called *quantum supremacy*, was claimed to be first demonstrated on a 53-qubit superconducting quantum processor (Sycamore) in Google [**?**].

In this practical training, you will learn about how to write codes that can be run directly on a superconducting quantum computer in Google! Now you might wonder, **what is the superconducting quantum computer?** The superconducting quantum computer is based on an architecture called the *circuit quantum electrodynamics* (cQED). The superconducting qubits are some circuits consisting of inductors, capacitors and Josephson junctions built from superconducting materials. The circuitry isolates two energy levels of the system protected by superconducting energy gap. Within this architecture, the single qubit gate is applied by coupling the qubit with some microwave pulses, and the two-qubit gates can be applied by resonating the frequency of the two qubits, or activating the inter-qubit interaction by microwave sources. By choosing an optimized set of parameters for these operations, the superconducting qubits are equipped with a set of supported (universal) gates.

Compared to other quantum computers, superconducting qubits have the advantages of being scalable and flexible in terms of design and control. The fabrication of superconducting qubits are based on existing chip-making and semiconductor technologies, which are already rather mature. The design of each superconducting qubit can also be further improved and accommodate different needs by changing the underlying circuitry. Superconducting quantum computer supports fast gate operation (such as the Transmon qubits used in Sycamore) in tens of nanoseconds, whereas the trapped ions, for example (depending on specific system) can take tens or hundreds of microseconds. This renders fast quantum computation. But superconducting qubits also come with a short coherence time (e.g. tens of microseconds for Transmon qubits, compared to the coherent time of seconds in ion trap). This strongly limits the number of operations we can do on the superconducting qubits.

It is still an active area of research, and not yet clear, as to which types of quantum platforms might serve the best as a practical quantum computer. Superconducting qubit is surely one of the leading candidates. There is a whole branch of area of research about superconducting qubits and it is rapidly expanding. Interested students can refer to, e.g. [**?**, **?**] for a survey of state-of-art superconducting qubits and links to other references.

3 Error-mitigation on NISQ Devices

The computation performed on the current, and possibly all the near-term quantum computers will be noisy, i.e. the results of the computation are subject to various types of errors, such as the decoherence errors, the gate errors, the readout errors etc. Therefore, error mitigation is an important part of quantum computing on NISQ devices. In general, understanding these errors is a very complicated task and is an area of intense research. The errors depend on the machine and architecture used, and many of them are a result of complicated quantum interaction.

On NISQ devices, we will not be able to perform full error-correction. Apart

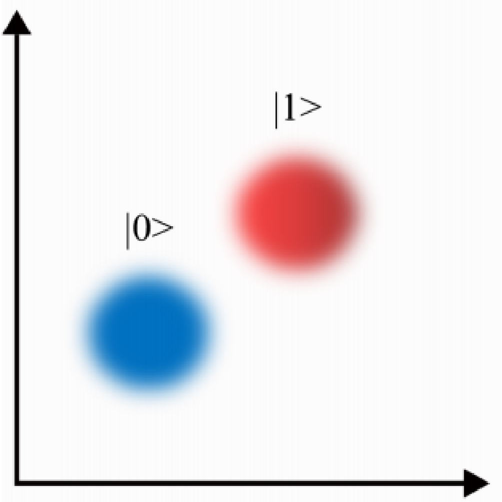

Figure 1: *Measured signals from some experiments. The state is identified by the clustering. If the two clusters have overlap, an incorrect readout can happen. Figure taken from [?].*

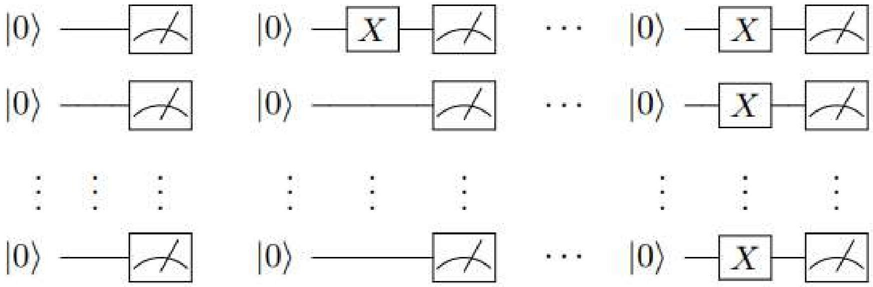

Figure 2: *The calibration circuits. Figure taken from [?]*

from the hardware improvement, we can instead mitigate the errors by some effective error modeling, and post-processing the data based on some calibration experiments. In this practical, we will implement a simple, but effective protocol to mitigate the readout errors, called the readout matrix unfolding [?].
The readout errors is an important class of errors on NISQ device. They are the errors that occur when measuring the outcomes of a computation. For example, the measurement process itself can cause transition between $|0\rangle$ and $|1\rangle$, either because it takes longer than the coherence time of the states, or some leakage of (e.g. in superconducting qubits) microwave pulses. It is also possible to simply read the wrong state if the cluster of the measured signals for the two states have overlap (Fig. ??).
The technique we will be using consists of two steps, the calibration and the unfolding. The calibration is a series of experiments performed on the device before the calculation.
Calibration: Suppose there are N qubits in the devices. Starting from the

product states $|00...0\rangle$ (which is usually the initial state on quantum computer), we can prepare 2^N (computational) basis states. For each prepared basis state $|b\rangle$, we perform measurement on all the qubits (see Fig. **??**). This is then repeated m times (which, e.g. can be the same as the number of measurements performed for readout the computation result). Therefore we perform in total $m2^N$ experiments. We can represent the data in a matrix form by

$$P_{ab} = \Pr\left(\text{Measured bitstring } a | \text{ Prepared bitstring } b\right) \tag{3.1}$$

we call this the *readout matrix* for the calibration, it contain the empirical probability distribution for each bitstring readout.
Unfolding: The unfolding step is to use P_{ab} to mitigate the errors in the measured probabilities for each strings. Suppose we have the vector of measured probabilities for each bitstring v'_a, and the error-free probabilities are v_a. The assumption we make is they are related by (as least for small error rates)

$$v'_a = \sum_b P_{ab} v_b \tag{3.2}$$

which is just a matrix equation. Therefore, the simplest approach to get v_a from v'_a is doing a matrix inversion of P, and this will be what we do in Day 2 practical using Cirq.

> ***Exercise 1:*** Show that if the measured v'_a normalizes to unity $\sum_a |v'_a| = 1$, the inverted v_a preserves the norm, i.e. $\sum_a |v_a| = 1$.

However, the simple matrix inversion may fail for a couple of reasons. The main issue is the inverted v_a can become unphysical, it has negative entries despite it should be a probability. The issue could be avoided by using more advanced unfolding techniques, such as solving the linear system Eq. (**??**) with constraints. Interested students can refer to [**?**] for full discussion. We just need to keep in mind that we should always check whether the inverted results are physical when you implement the method in the practical.

4 Cirq Basics

Cirq is an open-source python library developed by Google for simulating and running quantum circuits. In this practical, you will get to learn how to code in Cirq, and use it to simulate interesting quantum many-body system. Cirq is also directly implementable on the real quantum computer on Google. So, you might get a chance to run your quantum many-body simulation on a real superconducting quantum computer !
We will introduce the Cirq basics using the python notebook. You could also install Cirq on your laptop if you prefer. You should make sure you have python 3.5 or later, you can then install Cirq with pip (`python -m pip install cirq`).

You can find detailed instruction in here.
Use what you have just learnt to complete the following exercises (you can find even more textbook example problems solved using Cirq in the documentation page here).

4.1 Quantum Teleportation

Quantum teleportation is a technique used to transmit (teleport) a quantum state from Alice to Bob, given they pre-share an entangled state. Let us consider teleporting a single-qubit state. Suppose Alice and Bob pre-share an EPR pair $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Now Alice has the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ that she wants to teleport to Bob by classical communication. The system now has the wavefunction

$$\begin{aligned}
&\frac{1}{\sqrt{2}}|\psi\rangle_M(|0\rangle_A|0\rangle_B + |1\rangle_A|1\rangle_B) \\
&= |\Phi^+\rangle_{MA}(\alpha|0\rangle_B + \beta|1\rangle_B) + |\Phi^-\rangle_{MA}(\alpha|0\rangle_B - \beta|1\rangle_B) \\
&\quad + |\Psi^+\rangle_{MA}(\alpha|1\rangle_B + \beta|0\rangle_B) + |\Psi^-\rangle_{MA}(\alpha|1\rangle_B - \beta|0\rangle_B) \qquad (4.1)
\end{aligned}$$

where M refers to the message qubit Alice holds. A, B refers to Alice and Bob. And we define the four EPR pairs

$$\begin{aligned}
|\Phi^+\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\
|\Phi^-\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\
|\Psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\
|\Psi^-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \qquad (4.2)
\end{aligned}$$

They form a complete basis. If Alice performs a measurement on EPR pairs, Bob's qubit will collapse to one of the four state in (**??**). Alice then sends her measurement results encoded in 2 classical bits and Bob can recover the message state $|\psi\rangle$ by performing a unitary on his qubit accordingly.

> ***Exercise 2:*** Use CNOT, H and X gates to come up with the quantum circuits to prepare the four EPR pairs in Eq. (**??**). Suppose you are now given an unknown EPR state, how would you measure it?
>
> ***Exercise 3:*** As discussed above, depending on the measurement outputs from Alice, a unitary is performed on Bob's qubit to decode the message, show that this unitary can be expressed as one CNOT and one CZ (if a qubit is 1 then apply Pauli Z to the target qubit, otherwise do nothing).

> ***Exercise 4:*** Can you use Cirq to simulate (using `simulate()`) the quantum teleportation above? Use three qubits, q0 (message) and q1 belonging to Alice and q2 to Bob. Test your simulation by feeding a random one-qubit state to q0.

4.2 Rabi Oscillation

Rabi oscillation is one of the basic principles behind single-qubit manipulation in quantum computing. It refers to the oscillation between $|0\rangle$ and $|1\rangle$ states in a two-level system under periodic driving field. By adjusting the duration of the applied field (lasers or microwave pulses), one is able to perform desired single-qubit operation.

Suppose we have a two-level system (qubit), we apply a longitudinal field and a time-dependent transverse field $\mathbf{B} = B_z\hat{z} - B_x\cos(\omega t)\hat{x}$. The system has the Hamiltonian

$$\hat{H} = -\frac{\omega_0}{2}\hat{\sigma}_z + \omega_1\cos(\omega t)\hat{\sigma}_x \tag{4.3}$$

where $\omega_0 = \gamma B_z$ and $\omega_1 = \gamma B_x$ with γ being the gyromagnetic ratio. This system can be solved using the so-called *rotating wave approximation* (may sound familiar for some people?). The dynamics of a state $|\psi\rangle = \alpha(t)|0\rangle + \beta(t)|1\rangle$, initialized with $\alpha(0) = 1$, is given by

$$\begin{aligned} e^{-i\omega t/2}\alpha(t) &= \cos\left(\frac{\Omega t}{2}\right) - \frac{i\Delta}{\Omega}\sin\left(\frac{\Omega t}{2}\right) \\ e^{i\omega t/2}\beta(t) &= -\frac{i\omega_1}{\Omega}\sin\left(\frac{\Omega t}{2}\right) \end{aligned} \tag{4.4}$$

where $\Delta = \omega - \omega_0$ and $\Omega = \sqrt{\omega_1^2 + \Delta^2}$. Therefore, by choosing different t, some single-qubit operation could be performed. Next, we would like to use Cirq to time-evolve our qubit and actually observe the dynamics in Eq. (**??**)!

> ***Exercise 5:*** Compute the time-dependent population for state $|0\rangle$ and $|1\rangle$ from Eq. (**??**).

> ***Exercise 6:*** Define a sequence of single-qubit Cirq gate implementation `Ut()` for the time-evolution operator $\hat{U}(t)$, which is approximated by the time-dependent Trotterization
>
> $$\begin{aligned} \hat{U}(t) = \mathcal{T}\exp\left(-i\int_0^t \hat{H}(t)dt\right) &\approx \prod_{n=0}^{N-1}\exp\Big(-i\hat{H}(n\delta t)\delta t\Big) \\ &\approx \prod_{n=0}^{N-1}\exp\left(\frac{i\omega_0\delta t}{2}\hat{\sigma}_z\right)\exp(-i\omega_1\delta t\cos(\omega n\delta t)\hat{\sigma}_x) \end{aligned}$$

where $\delta t = t/N$ with N total time steps.
Using the simulator in Cirq, ***measure*** the population of state $|1\rangle$ for:
(a) different t. You may use the following parameters, $w_1 = 2, w_0 = 25, w = 25.5$ for $t \in [0, 4]$. And fix the time step to be $\delta t = 0.05$.
(b) different detunings Δ (Vary $\omega \in [10, 40]$, at $t = \pi/\omega_1$)
Plot your measurement results against the solution from Exercise 5.
(c) Try also $\omega = \omega_0 = \omega_1 = 2$. Compare to the analytic solution and explain your findings.

The Trotterization you used above is in fact one of the fundamental concepts in digital quantum computing for doing tasks such as adiabatic quantum computation [**?**] and simulation of quantum many-body system. In Day 2, you will get to explore the later in more details.
When you are actually running your simulation on a quantum computer, as we mentioned above, the results will be subject to various noise. Apart from the readout errors (which we will discuss on Day 2), the interaction with the environment (decoherence) is another major source of errors. This types of error become more prominent when your computation takes longer time. Let us simulate this effect in our Rabi oscillation experiment.
Consider the so-called generalized amplitude damping quantum channel (GAD)

$$\Lambda(\rho) = \sum_{i=0}^{3} E_i \rho E_i^\dagger \tag{4.5}$$

where ρ is the density matrix of our system and the Kraus operators E_i are

$$E_0 = \sqrt{p}\begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{bmatrix}, \quad E_2 = \sqrt{1-p}\begin{bmatrix} \sqrt{1-\gamma} & 0 \\ 0 & 1 \end{bmatrix}$$
$$E_1 = \sqrt{p}\begin{bmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{bmatrix}, \quad E_3 = \sqrt{1-p}\begin{bmatrix} 0 & 0 \\ \sqrt{\gamma} & 0 \end{bmatrix}$$

This quantum channel gives an effective description of noise due to both spontaneous emission and finite temperature thermalization, where $1-p \in [0, 1]$ quantifies the probability for thermal excitation and $\gamma \in [0, 1]$ is the probability for losing energy to the environment (e.g. spontaneous emission). Let us try to simulate such noise!

Exercise 7: Use the noise simulator `cirq.sample()` to simulate the time evolution in Exercise 6 in noisy environment. The GAD can be implemented by specifying

```
noise = cirq.ConstantQubitNoiseModel(
        cirq.GeneralizedAmplitudeDampingChannel(p,gamma))
```

in `cirq.sample()`. Cirq will automatically implement a GAD at the start of each circuit moment. You may use the values $p = 0.9$ and $\gamma =$

0.02, plot your noisy result against what you obtained from Exercise 6.
Try different noise parameters and time span, you should observe a decay time scale for $|1\rangle$ signal (sometimes called T_1). The results get worse when the operation takes longer, which is typical in the currently available (and near-term) machines.

In this practical we will not attempt to mitigate such errors, in fact error-mitigation for such type is in general very subtle for more than a few qubits. It is however important to understand how the these errors can affect your measurement result if your program is run on an actual device!

4.3 Single-qubit Tomography

By measuring the qubits in the computational basis, we only have access to the probability amplitude for each qubit state. To get the full information of the state, we need *quantum state tomography*. The basic idea is simple, any single-qubit density matrix can be written as a linear combination of Pauli matrices

$$\rho = \sum_{i=0}^{3} c_i \hat{\sigma}_i \tag{4.6}$$

where c_i are some real coefficients. $\hat{\sigma}_i$ are Pauli matrices with $\hat{\sigma}_0 = \mathbb{I}$, which form an orthogonal basis under the Hilbert Schmidt inner product $\langle A, B\rangle = Tr(A^\dagger B)$. To extract all the coefficients c_i from the measurement, we simply need to measure the system in all the orthogonal basis

$$c_i = \frac{1}{2}\langle\hat{\sigma}_i\rangle = \frac{1}{2}Tr(\hat{\sigma}_i\rho), \quad i = 1, 2, 3 \tag{4.7}$$

We do not need $i = 0$ because physical density matrix must have unit trace, i.e. $c_0 = 1/2$.

Exercise 8: When you do a measurement on a quantum computer, you will only measure $|0\rangle$ or $|1\rangle$ (eigenstates of Z or $\mathbb{I}$), and hence their probabilities. However this information is not enough to get $\langle\hat{\sigma}_x\rangle$ and $\langle\hat{\sigma}_y\rangle$. We need to instead measure $|0\rangle$ and $|1\rangle$ in different basis. Verify the following identities

$$X = HZH, \quad Y = SHZHS^\dagger \tag{4.8}$$

You can look up the definition of the phase gate S in Cirq.
Use Eq. (**??**) to construct Cirq circuits to measure $\langle\hat{\sigma}_x\rangle$, $\langle\hat{\sigma}_y\rangle$ and $\langle\hat{\sigma}_z\rangle$.

Exercise 9: Try the single-qubit tomograph method in Eq. (**??**), simulate the measurement of all the coefficients c_i from finite sampling, reconstruct the single-qubit density matrix for $(|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2} =$

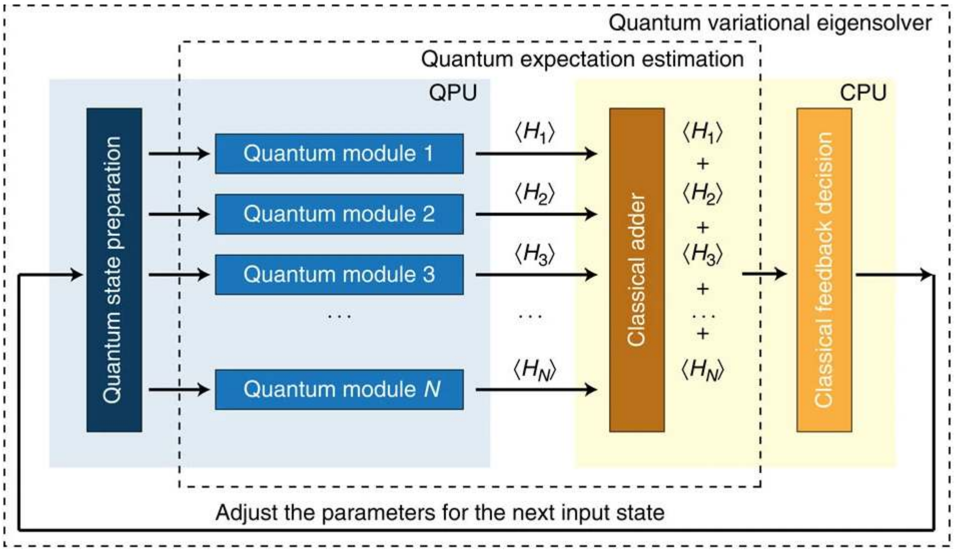
Quantum variational eigensolver
Quantum expectation estimation
QPU
CPU
Quantum state preparation
Quantum module 1
Quantum module 2
Quantum module 3
...
Quantum module N
$\langle H_1 \rangle$
$\langle H_2 \rangle$
$\langle H_3 \rangle$
...
$\langle H_N \rangle$
Classical adder
$\langle H_1 \rangle + \langle H_2 \rangle + \langle H_3 \rangle + \ldots + \langle H_N \rangle$
Classical feedback decision
Adjust the parameters for the next input state

Figure 3: *Schematics of VQE. Here H_i are the terms that sum to the total Hamiltonian H, e.g. some Pauli strings. Figure taken from [**?**]*

> $TH\,|0\rangle$ (T is a non-Clifford gate!). How do you extract the qubit state from the measured density matrix? Benchmark your result with the Cirq built-in function `cirq.experiments.state_tomography()`.
>
> ***Exercise 10:*** Use the same set of parameters as in Exercise 6, reconstruct the the oscillating qubit state by tomography (either your custom one or Cirq built-in) for $t \in [0, 2]$ and compare your result with Eq. (**??**). (Note that the states are equivalent up to an overall phase, you can set β to be real and compare α in both cases).

The idea of state tomography indeed also generalizes to many-qubit states, which we will look into in more detail on day 2.

4.4 Variational Quantum Eigensolver (VQE)

A useful application of near-term quantum computers in studying quantum many-body system is the so-called variational quantum eigensolver (VQE) [**?**, **?**]. One of the standard techniques to approximate the ground state of a local Hamiltonian $\hat{H}$ is the variational method, namely you prepare a trial state $|\psi(\theta)\rangle$ parametrized by a set of parameters $\{\theta\}$, then we search for the state that minimizes the energy $\langle\psi(\theta)|\hat{H}|\psi(\theta)\rangle$.
However, in the classical computer, the evaluation of $\langle\psi(\theta)|\hat{H}|\psi(\theta)\rangle$ for a given set of parameters $\{\theta\}$ is exponentially hard for the general case of N qubits. The idea of VQE is to leave this difficulty to the quantum computer. Because one can directly measure the energy expectation $\langle\psi(\theta)|\hat{H}|\psi(\theta)\rangle$ by measuring expectation

of a list of Pauli strings which sum to $\hat{H}$! Hence VQE is a quantum-classical hybrid algorithm.
In digital quantum computer, a simple VQE looks like

1. Define a parametrized circuit ansatz that prepares the trial state $|\psi(\theta)\rangle$.
2. Run your favourite classical algorithm to find the minimum of $\langle\psi(\theta)|\hat{H}|\psi(\theta)\rangle$, but replace the evaluation step by measurement on the quantum computer.

Let us simulate this for a simple two qubit example using Cirq. Consider a two-qubit cluster state Hamiltonian

$$\hat{H} = -\hat{\sigma}_x^1\hat{\sigma}_z^2 - \hat{\sigma}_z^1\hat{\sigma}_x^2 \tag{4.9}$$

where $1, 2$ refers to the first and second qubit. Notice that the two terms appearing in the Hamiltonian commute with each other. Complete the following exercises.

Exercise 11: Let us consider the trial state of the form

$$|\psi(a,b)\rangle = \exp\bigl(ia(\hat{\sigma}_z^1 + \hat{\sigma}_z^2)\bigr)\exp\bigl(ib(\hat{\sigma}_x^1\hat{\sigma}_z^2 + \hat{\sigma}_z^1\hat{\sigma}_x^2)\bigr)|00\rangle \tag{4.10}$$

where a, b are some real parameters. Construct a parametrized circuit for this ansatz.

Exercise 12: In general we need to run some classical algorithm (e.g. gradient descent) to find the minimum energy state. This can in fact be done in Cirq together with the machine learning framework TensorFlow Quantum (Google).
But here for the simplicity, just run a 20×20 grid search over the parameters space $a, b \in [0, \pi]$. , For each choice of parameters, measure the energy expectation $\langle\psi((a,b))|\hat{H}|\psi((a,b))\rangle$. Plot a heatmap (`imshow()`) for $\langle\psi(a,b))|\hat{H}|\psi(a,b))\rangle$ and find the best trial state. What is the approximated ground state energy?

Exercise 13: Verify that the state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|1-\rangle + |0+\rangle) \tag{4.11}$$

is the exact ground state of $\hat{H}$, where $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. Compare your result from Exercise 13 to this exact solution (you can use `dirac_notation()` to print your VQE state).

Exercise 14*: A nice thing about VQE is it is to some extent insensitive to noise! To see this, repeat Exercise 12 but with noise added (you may use the noise model from Exercise 7 and try a larger noise with $p = 0.8$ and $\gamma = 0.1$). What is your VQE final state? How did the noise change your energy landscape?

Exercise 15: Use the wavefunction simulator `simulate()` to compute a landscape for the bipartition entanglement entropy over the parameter grid, how does it compare to your energy landscape?

Instruction Material

Simulating quantum many-body dynamics on a current digital quantum computer

Day 2

Practical Training
Condensed Matter Theory
Technical University Munich
Physics Department

Responsible: Prof. Dr. Michael Knap

Available Dates (Day 1/Day 2):
January 21/22, 2021
May 20/21, 2021

1 Summary

In today's part of the training we apply the concepts introduced yesterday to a relevant many-body system using Google's Cirq library. The goal of this session is to study the phenomenon of Dynamical Quantum Phase Transitions (DQPTs) in a transverse field Ising model. Your task will be to convert the closed system quantum dynamics into a corresponding quantum circuit, and extract relevant observables such as the magnetization, the Loschmidt echo, and the entanglement dynamics from it. Along the way, we put an emphasis on the required measurement effort and the comparison with the exact solutions.

After the practical training day, please prepare a report in which you answer the questions posed throughout the instructions below, including a (brief) introduction to the topic. You can provide important plots in the main body of your report and attach the associated code in appropriately labelled appendices.

2 Prerequisites

1. Some minimal knowledge about quantum computation will be helpful, If you haven't learnt any quantum computation before, don't worry. A comprehensive reference will be the first chapter of *Quantum Computation and Quantum Information* by Nielsen and Chuang.

2. Before we start coding in Cirq, we will give instruction and some time for you to install Cirq. But if you can't wait, you can install the Cirq easily following the instruction here.

3. Please create a **Google account**, which is needed to work with Google Colab (a platform for interactive python notebooks). We provide a python notebook where you can find useful functions for completing the exercises.

4. The practical training consists of Day 1 (basics) and Day 2 (application), you are expected to hand in a written report for **both** Day 1 and Day 2, respectively. You can also choose to only participate on Day 1.

5. Please read through the following introduction on dynamical quantum phase transitions (section 3) and the transverse field Ising model (section 4) **ahead** of the practical session. You can follow the instructions even without answering the questions directly.

3 Dynamical Quantum Phase Transitions

In today's example we investigate the phenomenon of 'dynamical quantum phase transitions' (DQPTs). We provide a short introduction on the most important concepts based on the review of Ref [1]. DQPTs extend the notion of phase transitions to the realm of non-equilibrium physics. Let us recall that for the more conventional equilibrium phase transitions, the central object is the partition function

$$Z = \mathrm{Tr}\ e^{-\beta \hat{H}(\alpha)}, \tag{3.1}$$

for an inverse temperature $\beta = 1/T$ and a Hamiltonian $\hat{H}(\alpha)$ that depends on a set of parameters α (e.g. the strength of an external magnetic field). The partition function Z provides an implicit definition of a free energy density $f(\beta, \alpha)$ via

$$Z = e^{-\beta N f(\beta,\alpha)}, \tag{3.2}$$

which depends on both β and the collection of parameters α. N denotes the size of the system. Equilibrium phase transitions then occur at points where the free energy density $f(\beta, \alpha)$ becomes *non-analytic* as a function of either β or one of the parameters α. The former are temperature-driven phase transitions at some critical temperature T_c. The latter, if occuring at zero-temperature, are called quantum phase transitions.

In the context of DQPTs we now instead consider the real time evolution of a closed quantum many-body system starting from the ground state $|\psi_0\rangle$ of some Hamiltonian $\hat{H}_0$. At time $t = 0$, the Hamiltonian is abruptly changed from $\hat{H}_0$ to $\hat{H}$, a protocol which is called a *quantum quench*. Formally, the time evolution is then solved by

$$|\psi_0(t)\rangle = e^{-i\hat{H}t}\ |\psi_0\rangle\,. \tag{3.3}$$

A formal similarity to the above partition functions is achieved by computing the return amplitude to the original ground state $|\psi_0\rangle$,

$$\mathcal{G}(t) = \langle\psi_0|\psi_0(t)\rangle = \langle\psi_0|e^{-i\hat{H}t}|\psi_0\rangle\,, \tag{3.4}$$

which is called the *Loschmidt amplitude*. The associated return probability

$$\mathcal{L}(t) = |\mathcal{G}(t)|^2 = \left|\langle\psi_0|\psi_0(t)\rangle\right|^2 \tag{3.5}$$

is usually referred to as the *Loschmidt echo*. In analogy to the above equilibrium partition functions, the Loschmidt amplitude takes on a particular dependence on system size N (for large values N), such that we can write

$$\mathcal{G}(t) = e^{-N\,g(t)}, \tag{3.6}$$

with an associated (complex) rate function $g(t)$. Similarly, the Loschmidt echo is written as

$$\mathcal{L} = e^{-N\,\lambda(t)}. \tag{3.7}$$

We will refer to the rate function $\lambda(t)$, which can alternatively be defined as $\lambda(t) = -\lim_{N\to\infty} \frac{1}{N} \log \mathcal{L}(t)$ in the thermodynamic limit, as the *Loschmidt rate*.

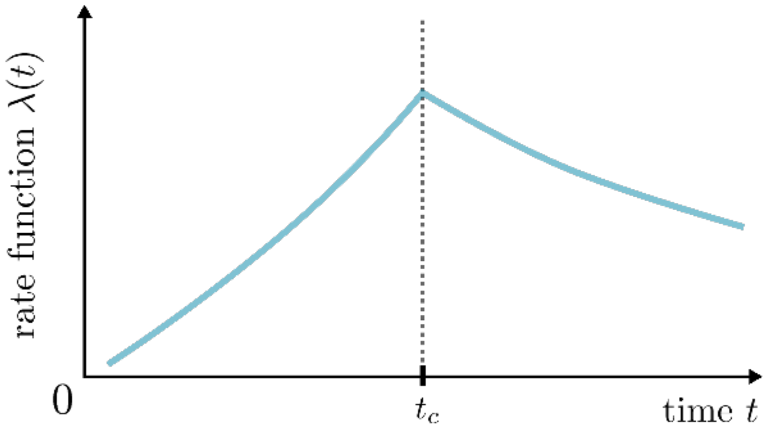

Figure 1: ***Dynamical Phase Transition.*** *(Figure 1 from Ref. [1]) At critical times t_c, the Loschmidt rate $\lambda(t)$ experiences non-analytic kinks.*

> ***Exercise 1:*** Show that $\lambda(t)$ and $g(t)$ are related via $\lambda(t) = 2\,\mathrm{Re}[g(t)]$. The exponential dependence of $\mathcal{G}(t)$ and $\mathcal{L}(t)$ on system size N in Eq. (3.6) and Eq. (3.7) is only in general valid if the quench changes the energy density, i.e. inserts an extensive amount of energy into the system. Can you come up with an intuitive reason for this (no calculations necessary)?

Equipped with these definitions, we can now identify DQPTs as points where the rates $\lambda(t)$, $g(t)$ become non-analytic as a function of time. DQPTs occur at *critical times* t_c where the rate $\lambda(t_c)$ typically shows a kink-like behavior in the thermodynamic limit, see Fig. 1 for a sketch.

> *Technical Note:* The analogy between partition functions and the Loschmidt amplitude can be made more concrete by interpreting
>
> $$\mathcal{G}(z) = \langle\psi_0|e^{-i\hat{H}z}|\psi_0\rangle\,, \qquad z \in \mathbb{C}, \tag{3.8}$$
>
> as a so-called 'boundary partition function' for a general complex parameter z. Non-analyticities of the associated rate function $g(z)$ are then associated to the zeros of $\mathcal{G}(z)$ in the complex plane (called Fisher zeros). A DQPT occurs whenever such Fisher zeros intersect the real axis, see Ref. [1] for more information.

As a final definition for this section, we note that one can generalize the Loschmidt echo $\mathcal{L}(t)$ for situations where the initial Hamiltonian $\hat{H}_0$ has more than one ground state, i.e. in the presence of spontaneous symmetry breaking: If $|\psi_i\rangle$ with $i = 0, \ldots, N_{gs} - 1$ labels all the N_{gs} states in the ground state manifold of $\hat{H}_0$, the

Loschmidt echo becomes

$$\mathcal{L}(t) = \sum_{i=0}^{N_{gs}-1} \left|\langle\psi_i|\psi_0(t)\rangle\right|^2, \tag{3.9}$$

and can be interpreted as the return probability to the ground state manifold of $\hat{H}_0$. In addition to the usual Loschmidt rate $\lambda(t) = -\lim_{N\to\infty} \frac{1}{N}\log\mathcal{L}(t)$, we can then also define the individual rates $\lambda_i(t)$ via

$$\left|\langle\psi_i|\psi_0(t)\rangle\right|^2 = e^{-N\lambda_i(t)}. \tag{3.10}$$

> ***Exercise 2:*** Based on the above definitions, show that in the thermodynamic limit $N \to \infty$, the Loschmidt rate $\lambda(t)$ reduces to the minimum function applied to the set of individual rates $\lambda_i(t)$, i.e. $\lambda(t) = \min_i[\lambda_i(t)]$.

4 The transverse field Ising model

We now consider a specific example to study DQPTs in the following , the paradigmatic transverse field Ising model (TFI). It is a ubiquitous model in the field of many-body physics and, due to its exact solvability, often used to benchmark numerical methods, including quantum computers.
In one dimension, the TFI consists of a linear chain of N qubits (or spin-1/2 sites) whose time evolution is generated by the following Hamiltonian,

$$\hat{H} = -\frac{1}{2}\sum_{\langle i,j\rangle} Z_i Z_j - \frac{g}{2}\sum_i X_i, \tag{4.1}$$

where $\langle i,j\rangle$ denotes nearest neighbor sites. We set the strength of the ZZ - Ising interaction to unity, but vary the strength g of the transverse field. The computational basis can be chosen as the z-basis in the following.
Before delving into the implementation as a quantum circuit, let us first perform a proper analysis of the model's symmetries and zero temperature (i.e. ground state) phase diagram.

> ***Exercise 3:*** Show that the parity operator
>
> $$\hat{P} = \prod_i X_i \tag{4.2}$$
>
> commutes with the Hamiltonian, i.e. $[\hat{H}, \hat{P}] = 0$. What is the spectrum of $\hat{P}$, i.e. its possible eigenvalues? Determine the exact ground states of $\hat{H}$ in the two limiting cases $g = 0$ and $g \to \infty$. Notice that for

> $g = 0$, the symmetry $\hat{P}$ is spontaneously broken, i.e. an infinitesimally small longitudinal field $\pm\epsilon \sum_i Z_i$ induces a finite expectation value of the ground state magnetization $m_z = \frac{1}{N}\sum_i Z_i$. Compute the expectation value of m_z in the respective symmetry broken ground states at $g = 0$, as well as for the ground state at $g \to \infty$.

From the solution to Exercise 3 we see that the TFI features a symmetry-broken ferromagnetic phase, i.e. $m_z(g) \neq 0$, for $g < g_c$ smaller than some so far unknown critical field strength g_c. As mentioned above, the TFI can be solved exactly and $m_z(g)$ can be computed analytically [2]. However, for our purposes, there is a more direct way to infer the critical field g_c:

> ***Exercise 4:*** Let us introduce the new Ising variables $\tilde{Z}_n = \prod_{i\leq n} X_i$ and $\tilde{X}_n = Z_n Z_{n+1}$, that can be thought of as living on the bonds of the lattice. Show that indeed $\{\tilde{Z}_n, \tilde{X}_n\} = 0$ as required, and express the Hamiltonian in terms of these new variables. Compare the result with the form of the Hamiltonian in terms of the original variables X_i, Z_i. How can you conclude from the resulting self-dual structure that the critical field assumes the value $g_c = 1$? The above transformation is known as the Kramers-Wannier duality transformation.

The previous analysis sets the stage for the investigation of DQPTs in the TFI model. In particular, we now know that for $g < 1$, the ground state manifold contains two degenerate ferromagnetic states $|\psi_0\rangle$ and $|\psi_1\rangle$ which are related by flipping all spins, $|\psi_1\rangle = \hat{P}\,|\psi_0\rangle$. For $g > 1$ the unique ground state is a paramagnet. Before shifting to numerics, let us answer the following final question:

> ***Exercise 5:*** We have shown the presence of ferromagnetic order in the ground state for $g < 1$. Does this order survive at finite temperatures? *Hint:* How much energy does a domain wall between two ferromagnetic patches cost? Accordingly, how many domain walls as a function of system size do you expect at finite temperatures? Use your result to argue that there can be no long-range ferromagnetic order at finite temperatures.

5 DQPTs in the TFI using Cirq

We now turn to the simulation of the time evolution generated by $\hat{H}$ using the methods introduced on Day 1 of the practical. In addition to this manual, your instructors will provide you with a python notebook containing several pre-defined functions that you may use to perform the following tasks.

5.1 Implementing the time evolution

Given $\hat{H}$ from Eq. (4.1), we want to perform a Trotter decomposition of the unitary operator $U(t) = e^{-i\hat{H}t}$, based on the elementary exponentials $e^{-iZ_iZ_j\,dt}$ and $e^{-igX_i\,dt}$ that can directly be implemented in Cirq. To this end, we can rewrite

$$U(t) = e^{-i\hat{H}t} = \left(e^{-i\hat{H}\,dt}\right)^{t/dt} = \left(e^{\hat{A}+\hat{B}}\right)^{t/dt}, \tag{5.1}$$

with $\hat{A} = -\frac{g}{2}dt\sum_i X_i$ and $\hat{B} = -\frac{1}{2}dt\sum_{\langle i,j\rangle} Z_iZ_j$.

> ***Exercise 6:*** Based on the functions expZZ(t) and expX(t) provided in the python notebook, define
> **a)** A function evolve_basic(circ,qubits,g,dt) that takes the input circuit circ and appends a set of operations that evolve the system of qubits by a small time step dt based on the first order Trotterization $e^{\hat{A}+\hat{B}} \approx e^Ae^B$.
> **b)** A function evolve_symmetric(circ,qubits,g,dt) that takes the circuit circ and appends a set of operations that evolve the system by dt based on the second order Trotterization $e^{\hat{A}+\hat{B}} \approx e^{A/2}e^Be^{A/2}$.
>
> In both cases, for the nearest neighbor Ising interaction you can approximate
>
> $$e^{-i\delta t\sum_{\langle i,j\rangle} Z_iZ_j} \approx \prod_{i=0}^{N/2-1} e^{-i\delta t\, Z_{2i}Z_{2i+1}} \prod_{i=0}^{N/2-2} e^{-i\delta t\, Z_{2i+1}Z_{2i+2}}. \tag{5.2}$$

> ***Exercise 7:*** To benchmark the results from Exercise 6, simulate the time evolution and extract the magnetization $m_z(t)$ as a function of time for a system of $L = 10$ qubits at $g = 2$, starting from an initial state $|\psi_0\rangle = |0...0\rangle$ of all spins down. Compare your results to the magnetization obtained through exact diagonalization of the Hamiltonian, which is provided by the pre-defined function magn_exact_diagonalization(L,g,t,dt) in the python notebook. How small do you have to choose the time step dt in the circuit to obtain a good approximation to the exact results for a final time of $t = 5$?

5.2 Extracting DQPTs

We now want to evaluate the Loschmidt rate as described at the beginning of this manual. The initial Hamiltonian is chosen as $\hat{H}|_{g=0}$, where the two degenerate ferromagnetic ground states are $|\psi_0\rangle = |0...0\rangle$ and $|\psi_1\rangle = |1...1\rangle$.

Exercise 8: Using the same time evolution as above, extract the simulated Loschmidt rate by projecting back onto the ground state manifold, see Eq. 3.9. Plot $\lambda(t)$, as well as the individual rates $\lambda_{1/2}(t)$, for
a) Different system sizes $L = 6, 8, 10, 12$ and $g = 2.0$. Can you see DQPTs emerging?
b) Different values of g between $0.5 \leq g \leq 1.5$. For which of these values are there DQPTs and for which not?

In the actual quantum device, we do not have access to the simulated wavefunction directly, and have to determine observables by repeated measurements.

Exercise 9: Determine both the magnetization and Loschmidt echo again, this time via repeated measurements. Plot your results for $L = 10$ and $g = 2.0$. How much sampling (as a function of system size) is in principle necessary to obtain accurate results?

Exercise 10: Compare the time evolution of the magnetization $m_z(t)$ with the Loschmidt rate $\lambda(t)$ (you may put both quantities into the same plot). Can you see any correspondence between $m_z(t)$ and $\lambda(t)$?

5.3 Tracking the entanglement production I: State Tomography

During the time evolution following the quantum quench, the system builds up entanglement. The increase in entanglement is responsible for the limitations on using classically efficient methods (e.g. matrix product states) to simulate the dynamics up to arbitrary times. To quantify the amount of entanglement in the system we consider the half-chain entanglement von Neumann entropy as well as the second Rényi entropy. For this purpose, we first define the reduced density matrix ρ_ℓ of the left half of the system,

$$\rho_\ell = \mathrm{Tr}_r\,[\rho], \tag{5.3}$$

with ρ the density matrix of the full system. The von Neumann entanglement entropy S is then defined as

$$S = -\mathrm{Tr}[\rho_\ell \log(\rho_\ell)]. \tag{5.4}$$

The second Rényi entropy on the other hand is obtained from ρ_ℓ via

$$S^{(2)} = -\log(\mathrm{Tr}[\rho_\ell^2]). \tag{5.5}$$

Exercise 11: We first extract the entanglement from the simulated wave function of the quantum circuit. Do so in the following way:
a) Define two functions vN_entropy(rho) and renyi2_entropy(rho) that return S and $S^{(2)}$ from Eqs. (5.4) and (5.5) for a given density matrix ρ, respectively. *Hint:* In order to compute S, express the trace in Eq. (5.4) in the eigenbasis of ρ.
b) Use the functions from part a) to compute and plot both $S(t)$ and $S^{(2)}(t)$ as a function of time for $g = 2$ (again starting from the ferromagnetic initial state $|\psi_0\rangle = |0...0\rangle$). Verify that both $S(t)$ and $S^{(2)}(t)$ grow *linearly* in time.

In order to obtain the reduced density matrix from the measured data, one needs to use state tomography. The principle of state tomography is readily explained (following Ref. [3]): Consider a single qubit (compare also with Day 1 of the practical), whose state is described by a point on (or within for mixed states) the Bloch sphere, i.e. three independent coordinates. Measuring the state of the qubit in the computational z–basis yields the z–component $\langle z\rangle = p_o - p_1$ on the Bloch sphere. In order to determine the y–component, we simply rotate around the x–axis by $\pi/2$ before measuring, in order to bring the y–axis upright. Analogously, rotation around the y–axis by $-\pi/2$ brings the x–axis upright and allows to determine the x–component.
For a system of N qubits with density matrix ρ, the above Ansatz is straightforwardly generalized: We apply all 3^N possible combinations of rotations of the individual qubits before measuring, denoted by U_j with $j \in \{1, ..., 3^N\}$. For a given rotation U_j, we then measure the probabilities p_{jk} of all basis states $|k\rangle$, with $k = 1, ..., 2^N$. This can be expressed as

$$p_{jk} = \langle k|\, U_j^\dagger\, \rho\, U_j\, |k\rangle\,. \tag{5.6}$$

Using the measured probabilities p_{jk} as inputs, Eq. (5.6) can straightforwardly be solved for the density matrix ρ by linear inversion. Notice however, that due to the limited amount of measurements, the resulting ρ is not guaranteed to be physical. Therefore, the ρ extracted from the measurements has to be be post-processed in order to find the physical density matrix that provides the best fit to the measured ρ. In the python notebook, we provide the function MLE_minimal(rho) that performs this task.

Exercise 12: For $L = 10$ and $g = 2$, perform a state tomography experiment with tomo_rep = 2000 repeated measurements (you can try out smaller numbers first) for each basis rotation. From the resulting probabilities p_{jk}, extract the associated density matrix and feed it into the provided function MLE_minimal(rho) in order to obtain the physical density matrix that best describes your measurement outcomes. Extract the corresponding von Neumann and second Rényi entropies as a function of time and compare them to the above exact results.

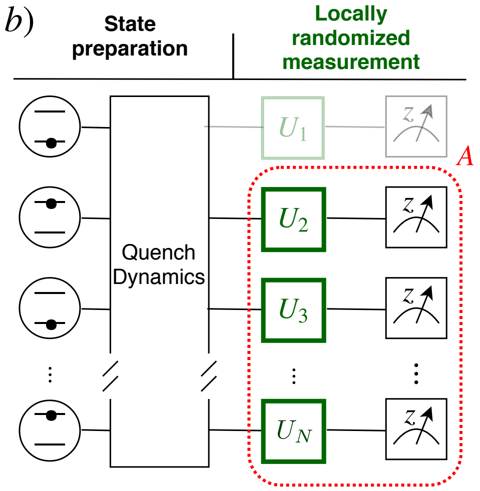

Figure 2: ***Randomized Measurements*** *(Figure 1b from Ref. [5]) Before measuring the qubits of subsystem A, random unitaries U_i are applied to the individual qubits. Correlations between these randomized measurement outcomes yields the second Rényi entropy according to Eq. (5.8).*

> ***Exercise 13:*** *(optional)* If you are interested, it is instructive to read through Ref. [4] (accessible at https://arxiv.org/abs/1106.5458) to see how the algorithm used in the function MLE_minimal(rho) works.

5.4 Tracking the entanglement production II: Randomized Measurements

Above, we extracted the von Neumann and second Rényi entanglement entropies through state tomography. However, for the second Rényi entropy (as well as for even higher Rényi entropies), there is a less costly way to extract $S^{(2)}(t)$ via *randomized measurements*. In this procedure, following Ref. [5], before the qubits are measured, we apply a random unitary operation $U = \prod_i U_i$ (with $i = 1, ..., L$) to each individual qubit of the chain, see Fig. 2, and use the statistical correlations between such randomized measurements in order to determine $S^{(2)}(t)$.

Let us go through the individual steps of this scheme more specifically:
1) Each U_i is chosen randomly from the so-called *circular unitary ensemble* (CUE). This can be achieved by the following parametrization,

$$U_i(a, b, c) = \begin{pmatrix} e^{ia}\cos(b) & e^{ic}\sin(b) \\ e^{-ic}\sin(b) & e^{-ia}\cos(b) \end{pmatrix}, \tag{5.7}$$

where a and c are uniformly sampled from $[0, 2\pi]$, and $b = \arcsin(k)$ with k uniformly in $[0, 1]$. In Cirq, we can realize Eq. (5.7) via the decomposition $U_i(a, b, c) = e^{i(a+c)Z/2}e^{ibY}e^{i(a-c)Z/2}$.

2) Once a set of U_i has been chosen, we perform repeated measurements with this given instance of random unitaries to extract the probabilities $P_U(s_A)$ of measuring a bitstring s_A of the subsystem A.
3) Finally, we repeat the above procedure many times and compute the probabilities $P_U(s_A)$ for different random operations $U = \prod_i U_i$. The second Rényi entropy of the subsystem A is then obtained from [5]

$$S^{(2)} = -\log \bar{X}, \qquad \text{with } X = 2^{L_A} \sum_{s_A, s'_A} (-2)^{-D(s_A, s'_A)} P_U(s_A) P_U(s'_A), \tag{5.8}$$

where the overbar $\bar{X}$ denotes the average over different random unitaries $U = \prod_i U_i$ with the U_i chosen independently from the CUE. $D(s_A, s'_A)$ denotes the *hamming distance* between the two bitstrings s_A and s'_A, which is defined as the number of bits where the two bitstring differ, i.e. $D(s_A, s'_A) = \#\{i : s_A(i) \neq s'_A(i)\}$.

> ***Exercise 17:*** To implement the above algorithm, go through the following steps.
> **a)** Write a function U2_CUE(qubit,symbs) that returns a parametrized single-qubit operation from Eq. (5.7), applied to the qubit qubit. The symbs[0], symbs[1], symbs[2] are the parameter symbols for the values a, b, c from Eq. (5.7).
> **b)** Write a function hamming(s1,s2) that implements the above hamming distance.
> **c)** Choose around 2000 measurement repetitions for each instance of the CUE and around 200 different instances of the CUE to compute $S^{(2)}(t)$ for selected time steps between $t = 0$ and $t = 5$, again for $g = 2$ (running the algorithm with these parameters might take a while, so you can experiment with smaller values first). Use the parametrized circuits introduced in the first day of the practical. Plot and compare your result to the exact value of $S^{(2)}(t)$ from the simulated wave function.

5.5 The effects of noise

Dearling with noise in the quantum machine is in general a very tricky task and requires lots of knowledge about the device. Here, we illustrate some basic ideas of quantum error mitigation using the simplest digital error model assuming independent spin flips on each qubit, which indeed seems too simple in many real devices. But in some cases, it turns out such a simple model is found to reasonably account for the effective noise from the device, such as the device used for recent quantum supremacy paper by Google [6]. A general form of single qubit noise is realized by a quantum channel, which is a result of the interaction with the environment (from Stinespring dilation theorem). Let us consider the following form of a single-qubit depolarizing channel,

$$\rho \mapsto (1 - p_x - p_y - p_z)\rho + p_x X\rho X + p_y Y\rho Y + p_z Z\rho Z. \tag{5.9}$$

Such a channel can be simulated by sampling a wavefunction in Cirq using the same simulator as before, where Cirq will apply the Pauli errors with the specified probabilities. Note that for other, non-unitary channels we would need to use the (more costly) density matrix simulator instead.
Here, we will try to mitigate the above type of readout error by unfolding the readout matrix obtained in a calibration process before the experiment, as explained in the instructions for day 1. Let us go through the implementation in the following:

Exercise 14: First, we create a circuit that includes the noise from Eq. (5.9) in our state tomography.
a) Take a small system of 5 qubits and create a circuit 'noise' that implements the above depolarization noise using the gates

cirq.asymmetric_depolarize(px,py,pz)

which is applied to all qubits in the chain. Choose $p_x = 0.01$ and $p_y = p_z = 0.005$ for concreteness for the following tasks.
b) Simulate the time evolution up to $t = 1$ at $g = 2$ and determine the reduced density matrix of the *three rightmost qubits* from the probabilites p_{jk} that you can obtain via state tomography. However, in contrast to before, insert the circuit noise directly before performing the measurements. You can use 1000 repetitions for the tomography.
c) From the resulting reduced density matrix (use MLE_minimal(rho)!), compute the von Neumann and second Rényi entanglement entropy and compare with the result that you obtain without noise.

Exercise 15: Now, we will try to mitigate the impact of noise via the scheme introduced on the first day of the practical.
a) To perform the calibration, create a circuit that first prepares a given computational basis state $|k\rangle$ of the three qubits in the subsystem ($k = 1, ..., 2^3$), and add the depolarization noise right before measuring the three qubits. Perform repeated measurements to determine the probabilities

$$P_{lk} = \Pr\left(\text{Measured bitstring } l \mid \text{Prepared bitstring } k\right) \tag{5.10}$$

for all k and l.
b) To perform the unfolding, use P_{lk} (recall that $l, k = 1, ..., 2^3$) to convert the measured, noisy probabilites p_{jk} (recall that $j = 1, ..., 3^3$; $k = 1, ..., 2^3$) that you obtained from the tomography in Exercise 14.b) into corrected probabilities $\tilde{p}_{jk}$ by inverting the relation $p_{jk} = \sum_l P_{kl}\, \tilde{p}_{jl}$ (using numpy).
c) Extract a new, error-mitigated density matrix from the probabilities $\tilde{p}_{jk}$ and compute $S(t)$ and $S^{(2)}(t)$. Compare your result with the unmitigated as well as the noise-free entanglement entropies to verify that your error-mitigation was effective.

Exercise 16: We can also investigate the impact of the above depolarization noise on the observables crucial for detecting DQPTs. Choose parameters that you deem sensible and include the noise in order to compute the Loschmidt rate for a quench across the phase transition. Compare mitigated, un-mitigated, and noise-free Loschmidt rates within a plot.

References

[1] M. Heyl, "Dynamical quantum phase transitions: a review," *Reports on Progress in Physics*, vol. 81, p. 054001, apr 2018.

[2] P. Pfeuty, "The one-dimensional Ising model with a transverse field," *Annals of Physics*, vol. 57, no. 1, pp. 79 – 90, 1970.

[3] M. G. Neeley, *Generation of Three-Qubit Entanglement Using Josephson Phase Qubits*. PhD thesis, 2010.

[4] J. A. Smolin, J. M. Gambetta, and G. Smith, "Efficient Method for Computing the Maximum-Likelihood Quantum State from Measurements with Additive Gaussian Noise," *Phys. Rev. Lett.*, vol. 108, p. 070502, Feb 2012.

[5] A. Elben, B. Vermersch, C. F. Roos, and P. Zoller, "Statistical correlations between locally randomized measurements: A toolbox for probing entanglement in many-body quantum states," *Phys. Rev. A*, vol. 99, p. 052323, May 2019.

[6] F. Arute, K. Arya, R. Babbush, *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.