Homework 12: mini IBM Q project
Traitement Quantique de l'Information

In this project, we are going to study the problem of sampling a classical probability distribution thanks to a NISQ device. At the end of this handout we provide some motivations for looking at such problems and a pointer to recent literature. The project is entirely self-contained and it is not necessary to read any literature beyond what was taught in class.

Login to https://quantum-computing.ibm.com/ and create an account with your epfl email. Accept the End User Agreement and at some point ask for a "token" that you can save (you can use the same one afterwards). Start getting familiar with this website after going through the material reviewed in class.

Useful (but not mandatory to go through) material is also https://youtu.be/a1NZC5rqQD8 (a series of short videos to start working with qiskit) and https://qiskit.org/textbook/ch-algorithms/teleportation.html (this link shows how to implement quantum teleportation, and it is good for learning how to build circuits and run experiments).

**Deadline Friday 18 december at 23h59:** *You will follow the guided questions in section* **??** *below, and hand in a small report. You are allowed to collaborate but your report and results of experiments must be individual. If you collaborated with somebody indicate it on your report. Detail the theoretical calculations and give the various plots asked for. Document what you do as well as the plots (machine used, number of runs etc) and write down short comments on your observations. Hand in also your Qiskit code. Upload your material in a single pdf file on moodle.*

**Important: Since you will have to run jobs on the IBM Q real devices you should start well in advance in order not to miss the deadline.** *This means that you should count one week to understand the theory and write a first version of the code. Then count one week to run the jobs and perform experiments. Typically your job might be in a queue for a few minutes or a few hours. Also depending when the machines are periodically calibrated it might be necessary to run your jobs a few times to get nice results. Note also that the practical part can be done without having succeeded at all the theoretical calculations.*

# 1 The general principle

Consider a probability distribution $p(c_1, c_2, \dots, c_n)$ over $n$ binary variables $c_i \in \{0, +1\}$. In other words there are $2^n$ entries and for each entry $0 \leq p(c_1, c_2, \dots, c_n) \leq 1$ and $\sum_{c_1,\dots,c_n \in \{0,+1\}^n} p(c_1, c_2, \dots, c_n) = 1$. Suppose we have (as a ressource) $n$ quantum bits. They live in the Hilbert space $\mathbb{C}^{\otimes n}$. We denote by $|c_1 c_2 \dots c_n\rangle$ the states of the computational basis where $c_i \in \{0, 1\}$. Given the probability distribution $p$, suppose we can find a unitary matrix $U : \mathbb{C}^{\otimes n} \to \mathbb{C}^{\otimes n}$ such that

$$|\langle c_1 c_2 \dots c_n | U | 0, 0, \dots 0\rangle|^2 = p(c_1, c_2, \dots, c_n). \tag{1}$$

If we can implement $U$ by a quantum circuit with reasonable (say polynomial in $n$) number of elementary logical gates then we can produce samples of the probability distribution $p$ in polynomial time (for each sample).

In other words we seek a quantum circuit such that: 1) the input is $|0\rangle^{\otimes n}$; 2) the output is $U|0\rangle^{\otimes n}$; 3) the measurement of the output in the computational basis yields $|c_1 c_2 \ldots c_n\rangle$ with probability given by equation (**??**).

The quantum circuit implements a sampler. For $n$ less than approximately 20 we will be able to implement it physically on NISQ devices.

# 2 A probability distribution

We consider the following probability distribution for $n \geq 3$:

$$p(c_1, \ldots, c_n) = \frac{1}{2^{2n}} \left| \sum_{b_1, \ldots, b_n \in \{0,1\}^n} (-1)^{\sum_{i=1}^n c_i b_i} \exp\left\{ i\theta \Big( \sum_{i=1}^{n-1} (-1)^{b_i} (-1)^{b_{i+1}} + (-1)^{b_n} (-1)^{b_1} \Big) \right\} \right|^2 \tag{2}$$

where $\theta \in [0, \pi]$. Of particular interest is

$$p(0, \ldots, 0) = \frac{1}{2^{2n}} \left| \sum_{b_1, \ldots, b_n \in \{0,1\}^n} \exp\left\{ i\theta \Big( \sum_{i=1}^{n-1} (-1)^{b_i} (-1)^{b_{i+1}} + (-1)^{b_n} (-1)^{b_1} \Big) \right\} \right|^2 \equiv \frac{|Z(i\theta)|^2}{2^{2n}} \tag{3}$$

The quantity $Z(i\theta)$ is the normalisation factor (or "partition function") of the so-called Ising model probability distribution where $s_i \in \{-1, +1\}$

$$\pi(s_1, s_2, \ldots, s_n) = \frac{1}{Z(J)} \exp\left\{ J \Big( \sum_{i=1}^{n-1} s_i s_{i+1} + s_n s_1 \Big) \right\} \tag{4}$$

computed for $J = i\theta$. Here the partition function can be computed exactly and we will admit the following result (see appendix for the details)

$$Z(J) = 2^n \Big( (\cosh J)^n + (\sinh J)^n \Big) \quad \text{and} \quad Z(i\theta) = 2^n \Big( (\cos\theta)^n + i^n (\sin\theta)^n \Big) \tag{5}$$

This gives us the explicit formula

$$p(0, \ldots, 0) = |(\cos\theta)^n + i^n (\sin\theta)^n|^2. \tag{6}$$

# 3 The quantum circuit for sampling $p(c_1, \ldots, c_n)$

Consider the circuit in figure **??**. The input state is $|0\rangle^{\otimes n}$, then we apply an Hadamard gate on each qubit, then we apply a unitary matrix $D$, and then a second series of Hadamard gates. Finally we perform measurements in the computational basis, which output a random string of classical bits $c_1, \ldots, c_n$.

In section **??** we guide you through a series of questions whose main goals are:

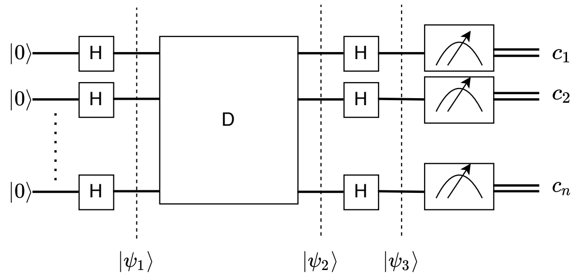


Figure 1: Circuit for sampling: time evolves from left to right

- To discover which unitary matrix $D$ is such that the measurement outputs are samples of the distribution (**??**). In other words for which $D$ do we have:

$$|\langle c_1, \dots, c_n| H^{\otimes n} D H^{\otimes n} |0, \dots, 0\rangle|^2 = p(c_1, \dots, c_n) \tag{7}$$

- To find a simple circuit for $D$ ?
- To implement the circuit on the publicly available NISQ machines of IBM Q experience and to produce histograms for small values of $n$ and a few values of $\theta$.
- To compare the theoretical curve for $p(0, \dots, 0)$ as a function of $\theta$ with the experimental curve (points).

# 4 Guided questions

## 4.1 Circuit ouput for diagonal $D$

Consider the class of diagonal matrices $D$. In other words on computational basis states we have $D|b_1, \dots, b_n\rangle = e^{i\varphi(b_1, \cdots, b_n)}|b_1, \dots, b_n\rangle$ for some real-valued function $\varphi : \{0,1\}^n \to \mathbb{R}$.

**a)** Compute $|\psi_1\rangle$ and then $|\psi_2\rangle$. *Hint*: start with $n = 1$, $n = 2$, and then generalize to any $n$.

**b)** Prove that the output of the circuit just before the measurement is

$$|\psi_3\rangle = \frac{1}{2^n} \sum_{c_1,\dots,c_n \in \{0,1\}} \left\{ \sum_{b_1 \dots b_n \in \{0,1\}^n} (-1)^{\sum_{i=1}^n b_i c_i} e^{i\varphi(b_1 \dots b_n)} \right\} |c_1 \dots c_n\rangle \tag{8}$$

*Hint*: check that $H|b_i\rangle = \frac{1}{\sqrt{2}} \sum_{c_i=0,1} (-1)^{b_i c_i} |c_i\rangle$ and use this formula.

**c)** What is the probability distribution produced by the measurements in the computational basis ?

## 4.2 Finding the matrix $D$

Comparing with equation (**??**) we must find a $2^n \times 2^n$ matrix $D$ such that $\varphi(b_1, \dots, b_n) = \theta\Big(\sum_{i=1}^{n-1} (-1)^{b_i} (-1)^{b_{i+1}} + (-1)^{b_n} (-1)^{b_1}\Big)$.

**d)** Check that for two qubits $e^{i\theta Z_1 \otimes Z_2}|b_1 b_2\rangle = e^{i\theta(-1)^{b_1}(-1)^{b_2}}|b_1 b_2\rangle$ where $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. Deduce $D$ in terms of Pauli matrices $Z_i$ acting on the qubits $i = 1, \dots, n$.

**e)** The circuit for $D$ is represented on figure **??** where denotes the $e^{i\theta Z\otimes Z}$ gate. Show that this gate is equivalent (up to a global unimportant phase) to CNOT$(I \otimes R(\theta))$CNOT where $R(\theta)$ is the "phase gate" $\begin{pmatrix} 1 & 0 \\ 0 & e^{-2i\theta} \end{pmatrix}$.

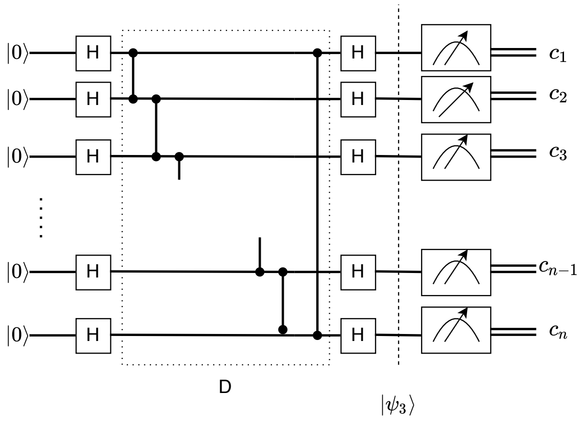


Figure 2: circuit for sampling with the implementation of $D$

### 4.3 Implementation on the IBM Q experience

First you familiarize yourself with the web site of the IBM Q experience https://quantum-computing.ibm.com/ . Play with the *composer* to get familiar with circuits and use the simulator as well as *real NISQ machines*. Get familiar also with the basics of the Qiskit language thanks to examples given in class and through the numerous tutorials.

**f)** First use the *composer* (graphical interface) to implement the circuit for a small number of qubits and use the *simulator* to produce the theoretical histograms $p(c_1, \ldots, c_n)$. Then run the *experiment* on a real machine and compare the histograms. Do this for $n = 3, \theta = \pi/5, \pi/3$ and $n = 4, \theta = \pi/5, \pi/3$ (the machine can have more than 3 or 4 qubits; you just ignore the extra bits and do not operate on them; however you may still do measurement operations on these qubits).

*Remark*: the $e^{i\theta Z\otimes Z}$ gate is already implemented in Qiskit. However we prefer that you use the implementation of question **e)** above instead.

**g)** Write a Qiskit code implementing the circuit. Check that

$$p(0, \cdots, 0) = |(\cos\theta)^n + i^n(\sin\theta)^n|^2$$

by running the *simulator* and the *experiment* on real NISQ machines for $\theta \in [0, \pi]$ for $n = 4, 5, 6, 7, 8, 9, 10$. Take a step size of $\delta\theta = \pi/32$ for the experimental points (i.e. for each $n$ there are 32 experimental points). We leave open to you to use various machines and compare the levels of noise (for up to 5 qubits there are lots of machines available, so choose two or three, but for $n \geq 6$ currently there is only one).

# 5 Motivations and further information

Recently Google announced that it demonstrated a *quantum advantage* (they called it *quantum supremacy*) in the problem of sampling a distribution made out of the *permanent* of a special class of matrices. They used a NISQ device with about 50 qubits and therefore sampled from a distribution with $2^{50}$ (classical) states. Since it is unknown how to compute permanents efficiently with classical methods and since the state space is huge it is a non-trivial problem to produce certificates that certify that the NISQ device truly outputs correct samples.

The problem studied in this mini-project is too simple to address questions related to a quantum advantage. Nevertheless it is a non-trivial illustration how a quantum device can in principle be used as a sampler.

*For further information see the review article: "Quantum sampling problems, Boson-Sampling and quantum supremacy" by A. P. Lund, M. J. Bremner and T. C. Ralph; npj Quantum Information (2017)3:15; www.nature.com/npjqi*

# 6 Appendix: partition function of the one-dimensional Ising model

Reading this section is not needed for the project and is here for completeness.

The Ising model is a mathematical model in statistical physics for the phenomenon of magnetism. We consider that a magnet is constituted by the collection of a macroscopic number of magnetic degrees of freedom of atoms. These magnetic degrees of freedom are grossly modelled by $n$ variables $s_i$, $1 \leq i \leq n$, taking values in $\{\pm 1\}$. Here we consider just a simple version of the model where the atoms are arranged in a "one-dimensional" cyclic cristaline array (see figure **??**) and each atom interacts with its two neighbours (this model was introduced by Lenz and studied by Ising in the 1920's).

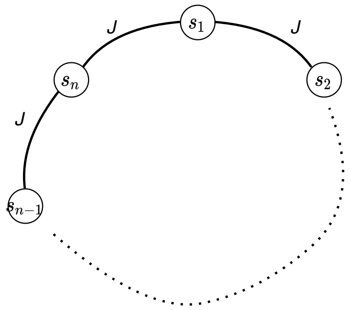


Figure 3: Ising model with periodic boundary

Here we show how to compute the normalisation factor of the distribution (**??**). This normalization factor is called the "partition function" and plays a central role in the theory. It equals:

$$Z(J) = \sum_{s_1,\dots,s_n \in \{-1,+1\}^n} \exp\Big\{ J\Big( \sum_{i=1}^{n-1} s_i s_{i+1} + s_n s_1 \Big) \Big\}.$$

We use a method called the *transfer matrix* method. Consider the matrix $T$,

$$T = \begin{pmatrix} e^J & e^{-J} \\ e^{-J} & e^J \end{pmatrix} \equiv \begin{pmatrix} T_{++} & T_{+-} \\ T_{+-} & T_{--} \end{pmatrix}$$

With a slight change in notation that $s_i = +, -$, and $s_{n+1} = s_1$ the partition function can be written as

$$Z(J) = \sum_{\underline{s} \in \{\pm\}^n} \prod_{i=1}^{n} T_{s_i, s_{i+1}}$$

One notices that this is equivalent to

$$Z = tr[T^n]$$

To compute the trace of $T^n$ one computes first the eigenvalues of $T$, and call them $\lambda_1$ and $\lambda_2$. Note that

$$T = e^J I + e^{-J} X$$

where $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ is the identity matrix and $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ one of the Pauli matrices. The eigenvalues are therefore $\lambda_1 = e^J + e^{-J} = 2\cosh J$ and $\lambda_2 = e^J - e^{-J} = 2\sinh J$. Thus

$$Z(J) = trT^n = \lambda_1^n + \lambda_2^n = 2^n\big((\cosh J)^n + (\sinh J)^n\big)$$