Educational Innovation Project UCM-UPM
Quantum Computing, Exercise 3

Ángel Rivas Vargas, Ignacio González, Felipe J. Llanes-Estrada, Juan J. Gálvez-Viruet, Rafael L. Delgado, Rafael Hernández Heredero, etc.

April 2024

1 Introduction

In this session you will solve three exercises.

- First you can run on a simulator, as in session 2, just to make sure that your code is working, at least two of the programs (you should already be familiar with the random numbers so we'll skip that one).
- Second, you will submit each job to one of IBM's actual quantum computers. Note that we are charged 1.6 $ per second of computation time on those. Normally each of these exercises should run in two seconds, but sometimes they can exceed ten seconds due to malfunction. We can not allow excess computations due to the lab's limited budget, so you will be requested to keep a strict limit of $t < 9$ seconds in your submissions.
- Also, you can also send your jobs to an actual quantum computer on the free allowance of the account which you created on session 2. These free jobs wait in a queue with low priority, so you may not have the results on time for the lab session, but you can check the outcome from home.

Please submit each exercise to the pay queue only once, when your lab instructor authorises you to do so, and making sure that you have not modified the statement limiting the running time. If you fail to comply, you will not pass the laboratory course.

2 Working environment

The easiest for this 2024 session is that you please reopen your IBM Quantum Lab from the last session with the same password.

You may also use your local `jupyter-notebook` if you installed Qiskit there. The laboratory computers at UCM also have a local installation ready for your use.

3 Notebooks

We provide you three python files at the online campus. Each of them is a file which you can run at either the lab or at home.

Please download the three files and upload them onto IBM quantum lab or whatever environment you are using. Follow the instructions in each python notebook.

Your lab instructor will provide you with an additional three files which are basically the same exercises, but with the data needed to run on the pay machines. This will be possible during the laboratory session only, your permission will afterwards be revoked (but you will retain the take-home versions of the exercises).

The three exercises are as follows.

1. **Random number generator**. In this first attempt you will obtain random numbers from a true quantum computer, as you learnt in the second lesson.

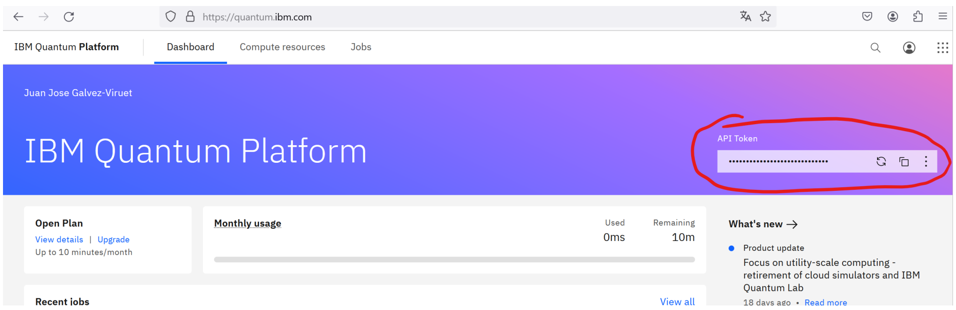
2. **Deutsch algorithm**. Here you will learn the most basic algorithm that presents a "quantum advantage" in which a quantum computer outperforms a classical computer.

3. **Bell inequalities**. In this exercise you will see the experimental realization on a quantum computer that the logic of quantum physics is different from classical logic, due to the fact that two quantum objects at different points can be entangled, and that a quantum object can be in a superposition of two states at the same time. A way to quantify this difference in logic is to write inequalities that are exactly satisfied in classical reasoning, but not in quantum mechanics, and try an experiment to see whether they are violated. This important test rules out "hidden variable" theories proposed by defenders of philosophical local realism.

4 Some vocabulary

- *Backend*: this is the quantum hardware or a simulator thereof that executes the quantum circuits.
- *Estimator*: this is a class of objects which can be "run" (so they are subprograms, see also *Sampler* below) that execute a quantum circuit together with some quantum operators and return the probability distribution of measuring those observables in the final-state qubits of the circuit.
- *Metadata*: various data about the computing job sent to a queue that are stored, beyond ("meta") its output results.
- *Sampler*: this is a class of objects which can be "run" (so they are subprograms, see also *Estimator* above) that execute a quantum circuit and give you a sample of the probability distribution of the final-state qubits.
- Token: in common language, it is a sort of coin (to put on a slot) used as credit to run electrical machines such as a car washing machine or the box cars at a fair ground. Here it is a character string (numbers and letters) which you will feed the IBM machines to run your programs on them.

  You will need your personal token which identifies you. Please find it on the upper right corner of your quantum lab homepage. You can copy it and paste it into a plain text file if you want to see it. Each account has its own identification "token".



5 Some generalities

You can check the **status of the jobs** that you have submitted to the classical simulators as well as to the quantum computers of IBM through the webpage `https://quantum.ibm.com/jobs`. To do this you need to note the "job number" of each submission.

You should download all your files from IBM's **Quantum Lab** to your personal computer, as the company will cancel this service on May 15th and all accesses after that date will have to be from your own installation of Jupyter-notebook+Qiskit.