QUANTUM ENGINEERING
Qiskit Simulations
2022-2023

Qiskit is an open-source SDK for simulating Quantum Circuits and programing quantum computers developed by IBM. You will find all the information you need to install Qiskit and get started at https://qiskit.org/

Once you have installed Qiskit start by following the introductory tutorial at https://qiskit.org/documentation/intro_tutorial1.html

Work to do :
The work consists of 9 exercises to solve using Qiskit from a very basic level to more advanced quantum protocols. For each exercise, please answer the question, provide the circuit, the results and the code. All the students must succeed !
If your need help, there are many tutorials available on the web, including on youtube.
Your reports need to be deposited on the **Moodle before the 21/05.** The grade will count for 50% of the final grade (50% for the exam).

Important remark on qubit ordering: Note that in Qiskit the last qubit is the leftmost qubit and the first one is the rightmost qubit which is the reversed order than that used in the class (and in most of the textbook)

EXERCISES

Exercise 1: Program a 1-qubit quantum circuit outputting a superposition state $1/\sqrt{2}\,(|0\rangle + |1\rangle)$ when the initial state is $|0\rangle$. Visualize the circuit, the initial quantum state and the final quantum state in the Bloch sphere.

Exercise 2: Program a 1-qubit quantum circuit outputting a superposition state $1/\sqrt{2}\,(|0\rangle - |1\rangle)$ when the initial state is $|1\rangle$. Visualize the circuit, the initial quantum state and the final quantum state in the Bloch sphere.

Exercise 3: Program a 1-qubit quantum circuit inducing a pi rotation on the state $|0\rangle$ i.e. outputting $|1\rangle$. Visualize the circuit, the initial quantum state and the final quantum state in the Bloch sphere.

Exercise 4: Program a 1-qubit quantum circuit outputting a superposition state $1/\sqrt{2}\,(|0\rangle - |1\rangle)$ stating from state $|0\rangle$. Visualize the circuit, the initial quantum state and the final quantum state in the Bloch sphere. Is this result equal to that of a previous exercise?

Exercise 5: Program a 2-qubit quantum circuit with a Bell state β_{00} . Visualize the circuit and use statename.draw('latex') to visualize the final state. Is it possible to visualize the initial state for each qubit in the Bloch sphere? And the final state in the Bloch sphere? If not, use the 'qsphere' representation.

Exercise 6: Program a 2-qubit quantum circuit with a Bell state β_{01} . Visualize the circuit, use statename.draw('latex') to visualize the final state and the 'qsphere' representation.

Exercise 7: Program a quantum circuit including a 3-qubit quantum register and a 3-bit classical register so that the initial state is $|010\rangle$ and the final state is $|001\rangle$. (Note that the qubit order in Qikist's kets is $|q_3\, q_2\, q_1\rangle$). Include a measurement for qubits 2 and 3 and send the result through classical registers 2 and 3 respectively. Visualize the final state both in the Bloch sphere and the Qsphere. Is it a pure state? Then execute the following code lines to simulate the circuit and obtain the result of the measurements:

```
sim = Aer.get_backend('aer_simulator')
qc=save_statevector()
 out_vector = sim.run(qc).result().get_statevector()
 plot_bloch_multivector(out_vector)
 qobj = assemble(qc)
result = sim.run(qobj).result()
counts = result.get_counts() plot_histogram(counts)
```

**Note: variable names in italics can be changed (and must sometimes to remain consistent with the first part of the exercise.

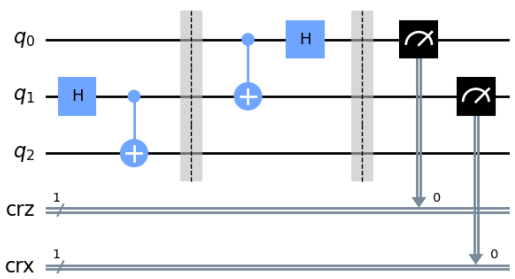
Are the results of the out_vector and counts' histogram the expected ones from the first part of the exercise i.e. consistent with the result of state_evolve(qc))?.

Exercise 8 : Program a quantum circuit with a 1-qubit quantum register and also one classical register. Generate a random superposition state $|\psi\rangle$ using 'random_statevector()' and visualize it in the Bloch sphere. Initialize your quantum circuit with the radom state as input for the quantum register using the following code lines:

```
init_gate = Initialize(psi) qc.append(init_gate, [0])
```

Add a barrier and a measurement after the circuit. Run the simulation and visualize the 'out_vector' in the Bloch sphere and the count's histogram. Are the input vector, outing vector and histogram consistent with each other? What happens if you repeat the simulation?

Exercise 9: Program the following quantum teleportation circuit:



Note that for more complex circuits it can be convenient to define the quantum registers and classical registers separately, merging them together afterwards as:

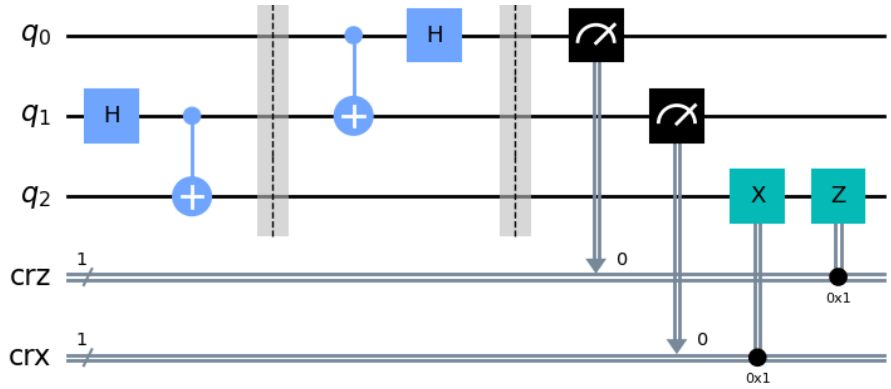
qr = QuantumRegister(3, name="q")

crz, crx = ClassicalRegister(1, name="crz"), ClassicalRegister(1,

name="crx")

teleportation_circuit = QuantumCircuit(qr, crz, crx)

Add conditional gates for Bob's qubit (q2) where the conditions are if crz=1 apply Zgate to q 2 and/or if crx=1 apply X-gate to q2 . The resulting circuit is:



Then initialize q 0 with a random superposition state and run the simulation. Is teleportation achieved? All commands needed to complete this exercise have been used in previous exercises with the exception of the conditional gates X and Z. See the beginning of this manual for the syntax enabling adding conditions to the gates.