

CSE622/622A IQC: Homework 1

Instructor: Debajyoti Bera (IIIT-Delhi)

Due: 20 January 2026

Use bra-ket notation to the extent possible. You may need to do linear-algebraic calculations; either do them by hand (submit calculations with the homework) or use numpy (submit python or ipynb/Jupyter notebook file as attachment on Classroom).

Q1 [5 points] Consider the quantum money protocol that we learnt and answer the below questions from the point of view of a malicious person who is traditionally named Eve. Eve receives a note from a legitimate channel, say with denomination D , serial number S , and qubit in state $|\psi\rangle$; she prepares a forged note in the following manner.

She randomly decides between two measurement X or Z and measures $|\psi\rangle$ using that measurement. Suppose the outcome is $|\psi'\rangle$. She creates a qubit in the state $|\psi'\rangle$ for her duplicate note. She uses creates a fake note using $|\psi'\rangle$ with denomination D and serial number S .

Then, to legitimise her note, she presents the note to the bank for verification. This exercise will help you compute Eve probability of success.

1. Suppose $|\psi\rangle = |0\rangle$. What is the probability that $|\psi'\rangle = |\psi\rangle$?
2. Prove that the bank always successfully verifies the note if $|\psi'\rangle = |\psi\rangle$.
3. What is the probability that the bank successfully verifies the note if $|\psi'\rangle \neq |\psi\rangle$ when $|\psi\rangle = |0\rangle$?
4. Prove that the bank successfully verifies the note with only $\frac{1}{2}$ probability if $|\psi'\rangle \neq |\psi\rangle$ for all possible $|\psi\rangle$.
5. Now combine both the cases of $|\psi'\rangle = |\psi\rangle$ and $|\psi'\rangle \neq |\psi\rangle$ to compute the probability of the bank successfully verifying the note.

Q2 [3 points] Write down the action of \sqrt{X} , \sqrt{Y} , \sqrt{Z} gates in any one of these basis: standard basis, Hadamard basis, $\{|+i\rangle, |-i\rangle\}$ (we will refer to this as the CP-basis – CP stands for circular polarization).

Q3 [3 points] Recall that X acts like a NOT-operator in the standard basis. Show that Z , Y and H too act like a NOT-operator in some basis. Write down the states of all the three bases in either the standard basis, the Hadamard basis, or the i basis.

Q4(a) [0 points, do not submit] Read about 3D rotations using Euler angles.

Q4(b) [0 points, do not submit] Understand how $R_Z(\theta)$ and $R_Y(\theta)$ modifies the above basis states for different values of θ .

Q5 [1+3+4+3+1+2=14 points] In this question you will design and implement the following operator:

$$U = |+\rangle\langle 0| + |-i\rangle\langle 1|$$

1. Write down the matrix form of U .
2. Apply U on the states in the standard basis, states in the Hadamard basis, and states in the CP-basis. You should write the output states in the standard basis.
3. Draw a Bloch sphere and show these states on the sphere: $U|0\rangle$, $U|1\rangle$, $U|+\rangle$, $U|-\rangle$.
4. Any 3D rotation can be decomposed as three rotations: about Z by γ , about Y by δ , about Z by β . With the help of trial-and-error, identify β, δ, γ such that the sequence performs the same mapping as U on $|0\rangle$, $|1\rangle$, $|+\rangle$, $|-\rangle$. The angles should also satisfy $U = e^{i\alpha} R_Z(\beta) R_Y(\delta) R_Z(\gamma)$ for some α . *Hint: Use 3D rotation ideas. The angles are either $0, \pi/2$ or π , so there are only a few combinations.*
5. Use Qiskit composer to create a 1-qubit circuit that first applies $R_Z(\gamma)$, then applies $R_Y(\delta)$, and finally $R_Z(\beta)$. Add a measurement to a classical register.
6. Copy the python code that created the circuit. Then paste it inside this boiler-plate code. You can either install qiskit on your computer or use Google colab for the same.

```
from qiskit import QuantumCircuit, transpile, QuantumRegister, ClassicalRegister
from qiskit_aer import AerSimulator
from qiskit_aer.noise import NoiseModel, pauli_error, depolarizing_error
from qiskit.providers.basic_provider import BasicSimulator
import numpy as np
import math

# add your circuit here
# make sure the object for QuantumCircuit is named 'circuit'

backend = BasicSimulator()

# Run the circuit on |0> and get the results
job = backend.run(circuit, shots=1000, initial_statevector = np.array([1, 0]))
result = job.result()
counts = result.get_counts()

print(counts)
```

Print the counts that the code outputs for different `initial_statevector` corresponding to $|0\rangle$, $|1\rangle$, $|+\rangle$, and $|-\rangle$.