Report on Quantum Information and Quantum Machine Learning
Project 1

Dmytro Romaniv 151958

October 25, 2025

Chapter 1

Task 1: Z-Type Projection Measurement

1.1 Minimal code snippet

```
from qiskit import QuantumCircuit
from qiskit_aer import Aer

SHOTS = 2048
backend = Aer.get_backend("qasm_simulator")

qc = QuantumCircuit(4, 4, name="Task1_Z")
# start in |0000>, measure only q0 -> c0
qc.measure(0, 0)

result = backend.run(qc, shots=SHOTS).result()
counts = result.get_counts(qc)          # {'0000': 2048}
print("Z-projection counts:", counts)
```

1.2 Counts (Histogram)

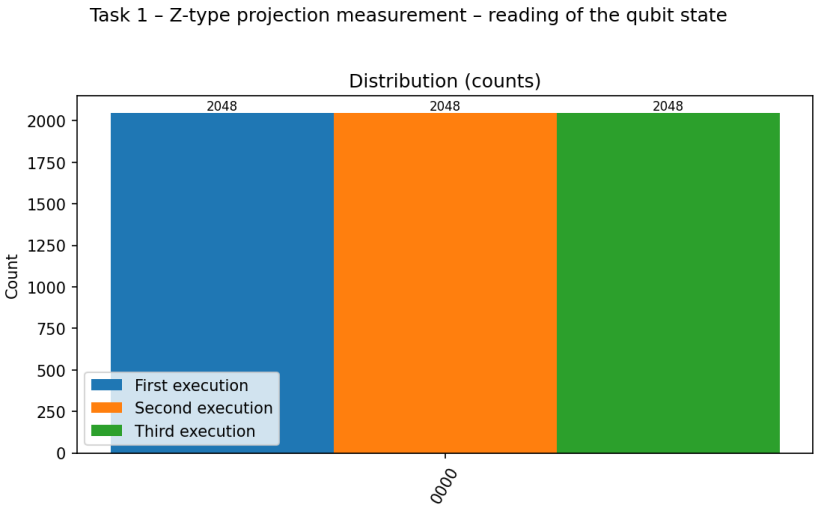

Figure 1.1: Distribution of outcomes for Z-type projection.

1.3 Probabilities

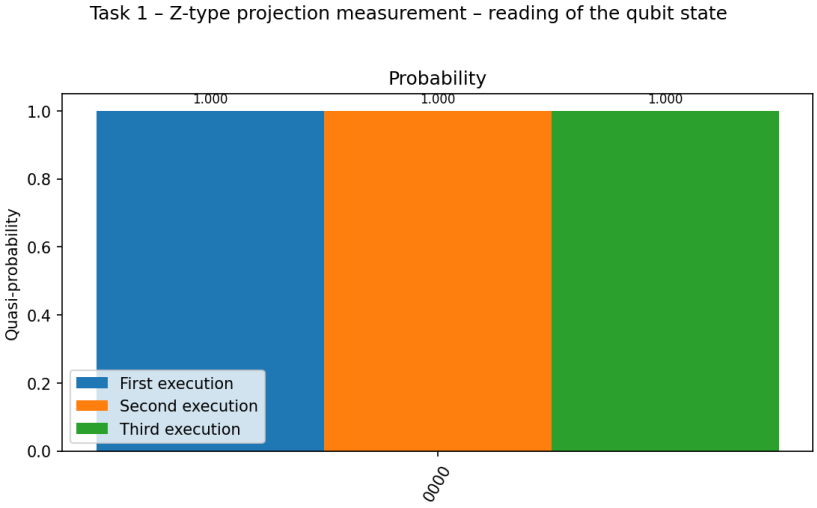

Figure 1.2: Measured probabilities for Z-type projection.

Chapter 2

Task 2: Operation and Reading of the Result of Quantum Gate X

2.1 Minimal code snippet

```
from qiskit import QuantumCircuit
from qiskit_aer import Aer

SHOTS = 2048
backend = Aer.get_backend("qasm_simulator")

qc = QuantumCircuit(4, 4, name="Task2_X")
qc.x(0)                  # flip |0> -> |1>
qc.measure(0, 0)

res = backend.run(qc, shots=SHOTS).result()
counts = res.get_counts(qc)              # e.g. {'0001': 2048}
p1 = counts.get("0001", 0) / SHOTS
print(f"X gate: counts={counts}, P(0001)={p1:.3f}")
```

2.2 Counts (Histogram)

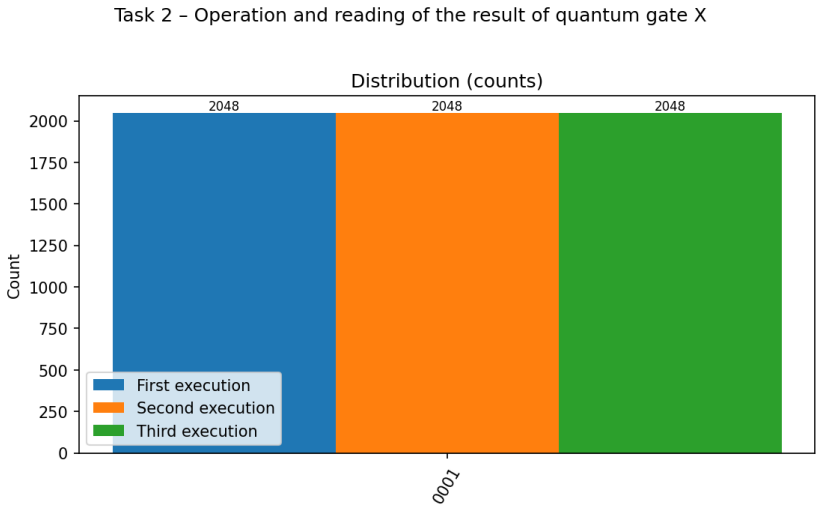

Figure 2.1: Counts after applying the X gate.

2.3 Probabilities

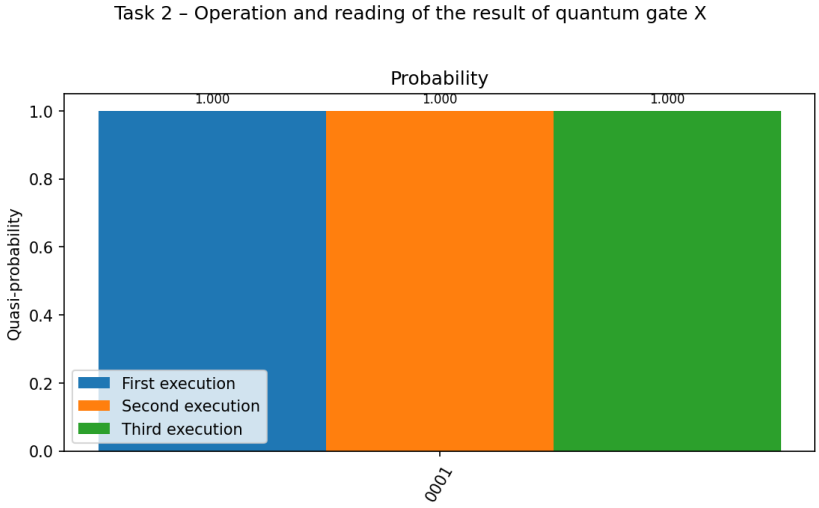

Figure 2.2: Probabilities after applying the X gate.

Chapter 3

Task 3: Operation and Reading of the Result of the Hadamard Gate (H)

3.1 Minimal code snippet

```
from qiskit import QuantumCircuit
from qiskit_aer import Aer

SHOTS = 2048
backend = Aer.get_backend("qasm_simulator")

qc = QuantumCircuit(4, 4, name="Task3_H")
qc.h(0)                  # (|0> + |1>)/sqrt(2)
qc.measure(0, 0)

res = backend.run(qc, shots=SHOTS).result()
counts = res.get_counts(qc)              # ~{'0000': ~1024, '0001':
    ~1024}
p0 = counts.get("0000", 0) / SHOTS
p1 = counts.get("0001", 0) / SHOTS
print(f"H gate: P(0000)={p0:.3f}, P(0001)={p1:.3f}")
```

3.2 Counts (Histogram)

Task 3 – SUPERPOSITION OF STATES – Operation and reading of the result of Hadamard quantum gate (H gate)

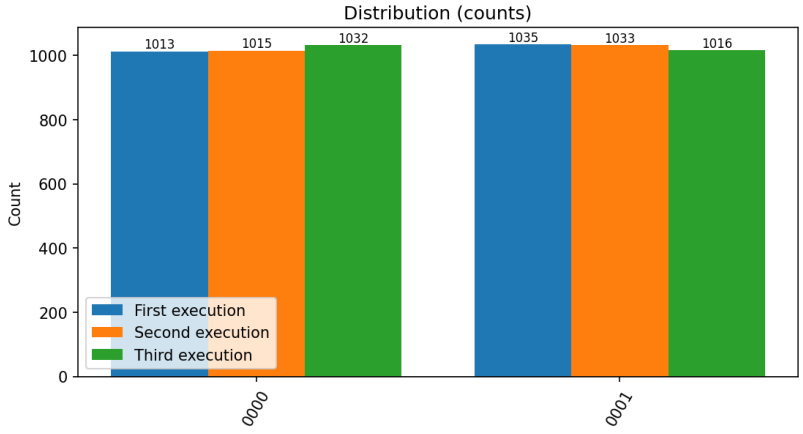

Figure 3.1: Counts after applying the Hadamard gate.

3.3 Probabilities

Task 3 – SUPERPOSITION OF STATES – Operation and reading of the result of Hadamard quantum gate (H gate)

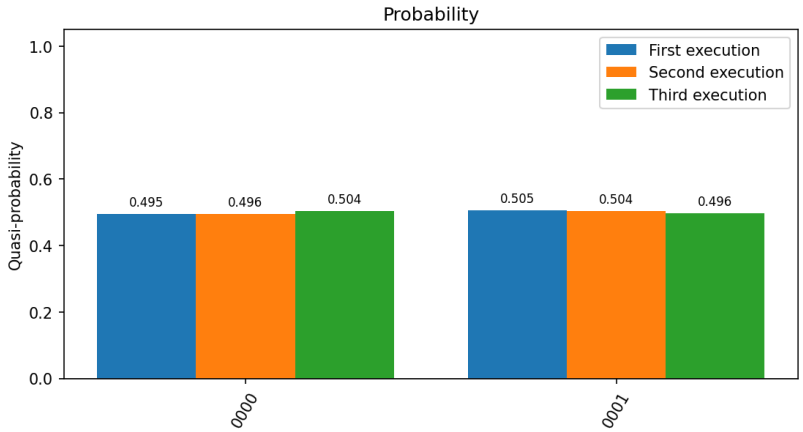

Figure 3.2: Probabilities after applying the Hadamard gate.

Chapter 4

Task 4: State Tomography of One Qubit

4.1 Preparation and basis change (snippets)

```python
from qiskit import QuantumCircuit
import numpy as np

qc = QuantumCircuit(1, 1, name="Task4_Basis")

# prepare same state for all bases
qc.ry(np.pi/2, 0)
qc.p(np.pi/2, 0)          # RY(pi/2) + P(pi/2)

# --- X base ---
qc.h(0)
qc.measure(0, 0)

# --- Y base ---
qc.sdg(0)
qc.h(0)
qc.measure(0, 0)

# --- Z base ---
# no extra gate, direct measurement
qc.measure(0, 0)
```

4.2 Measurement in the X Base

4.2.1 Counts

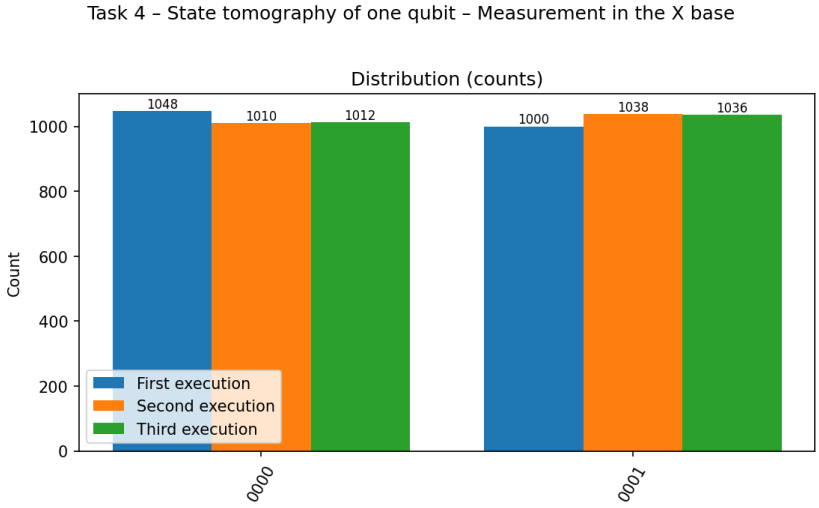

Figure 4.1: Counts for measurement in the X base.

4.2.2 Probabilities

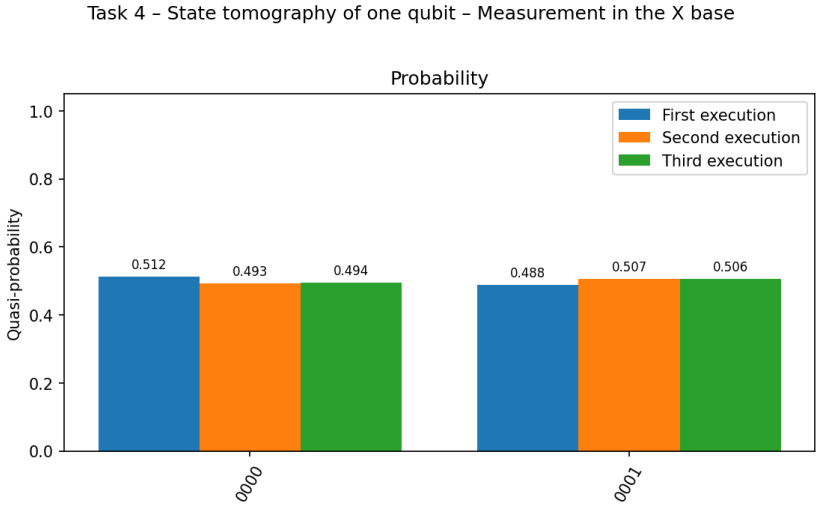

Figure 4.2: Probabilities for measurement in the X base.

4.3 Measurement in the Y Base

4.3.1 Counts

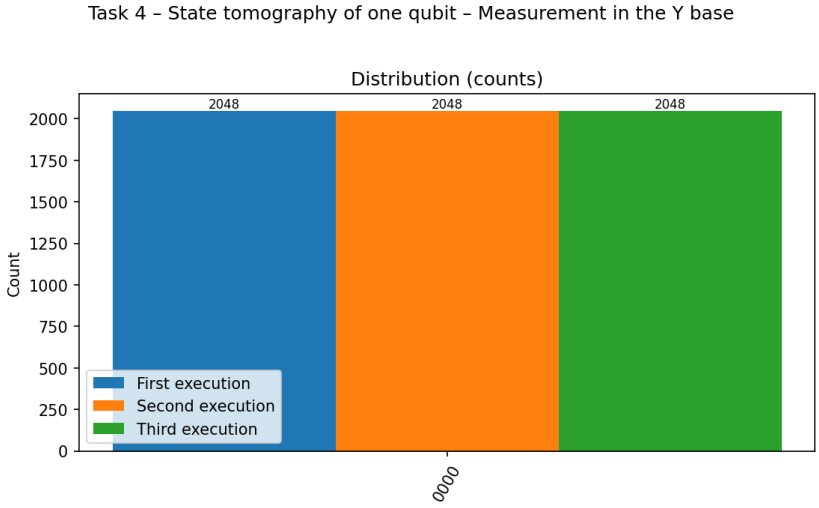

Figure 4.3: Counts for measurement in the Y base.

4.3.2 Probabilities

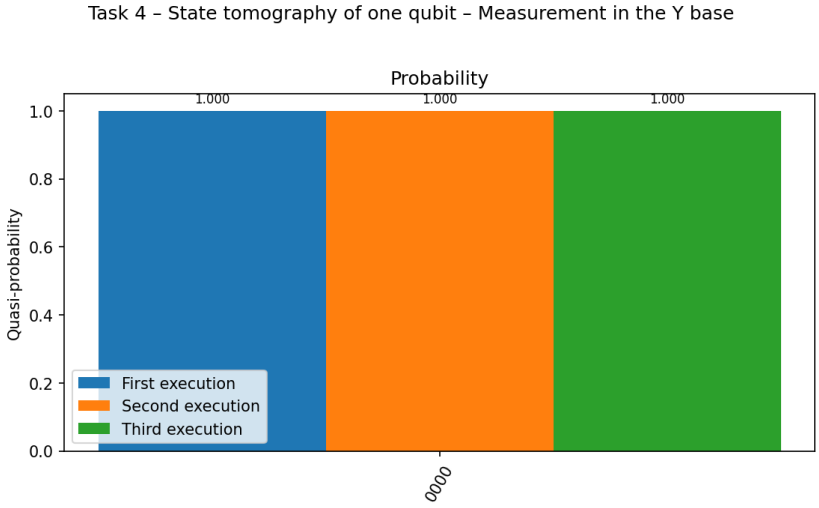

Figure 4.4: Probabilities for measurement in the Y base.

4.4 Measurement in the Z Base

4.4.1 Counts

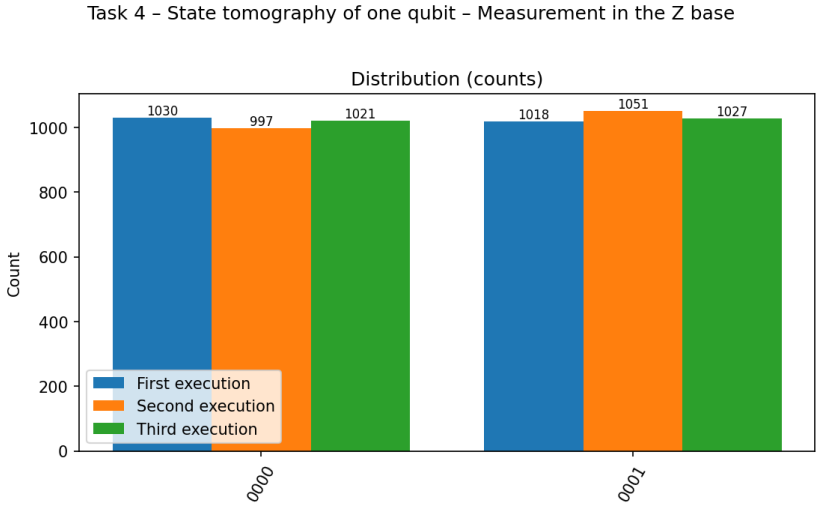

Figure 4.5: Counts for measurement in the Z base.

4.4.2 Probabilities

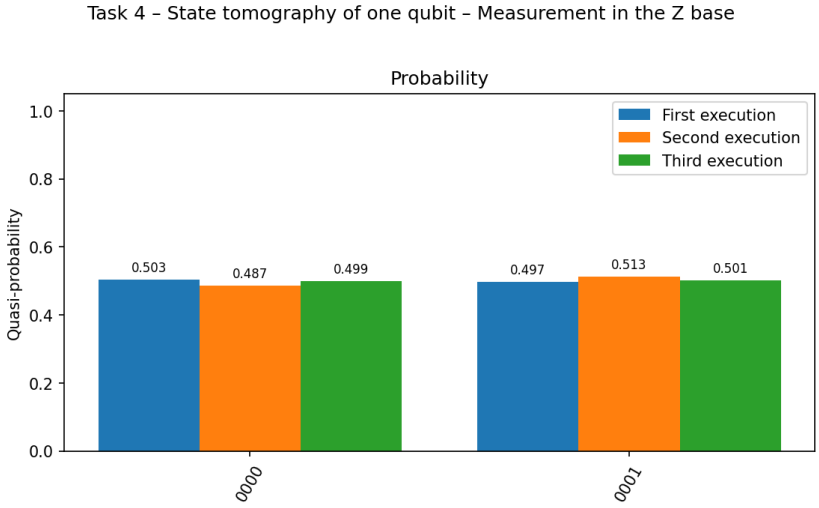

Figure 4.6: Probabilities for measurement in the Z base.

Chapter 5

Task 5: Extra Visualisations for Three Quantum Circuits

5.1 Minimal code snippet

```
from qiskit import QuantumCircuit
from qiskit_aer import Aer
from qiskit.visualization import (
    plot_state_city, plot_state_hinton,
    plot_state_qsphere, plot_bloch_multivector
)

sv_backend = Aer.get_backend("statevector_simulator")

def psi_of(label="|0>"):
    qc = QuantumCircuit(1, name=label)
    if label == "|1>": qc.x(0)
    elif label == "|+>": qc.h(0)
    return sv_backend.run(qc).result().get_statevector(qc)

psi = psi_of("|0>")
plot_state_city(psi);        plot_state_hinton(psi)
plot_state_qsphere(psi);     plot_bloch_multivector(psi)
```

5.2 $|0\rangle$

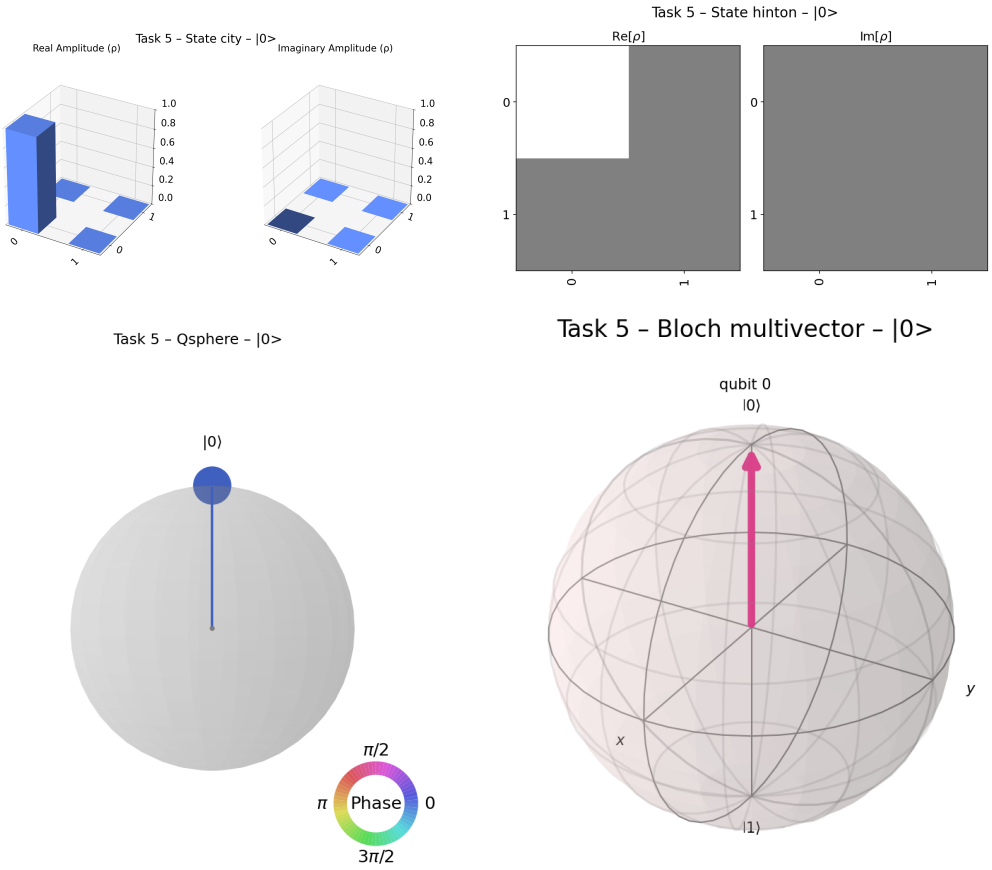

Figure 5.1: Visualisations for $|0\rangle$: state city, Hinton, qsphere, Bloch multivector.

5.3 $|1\rangle$

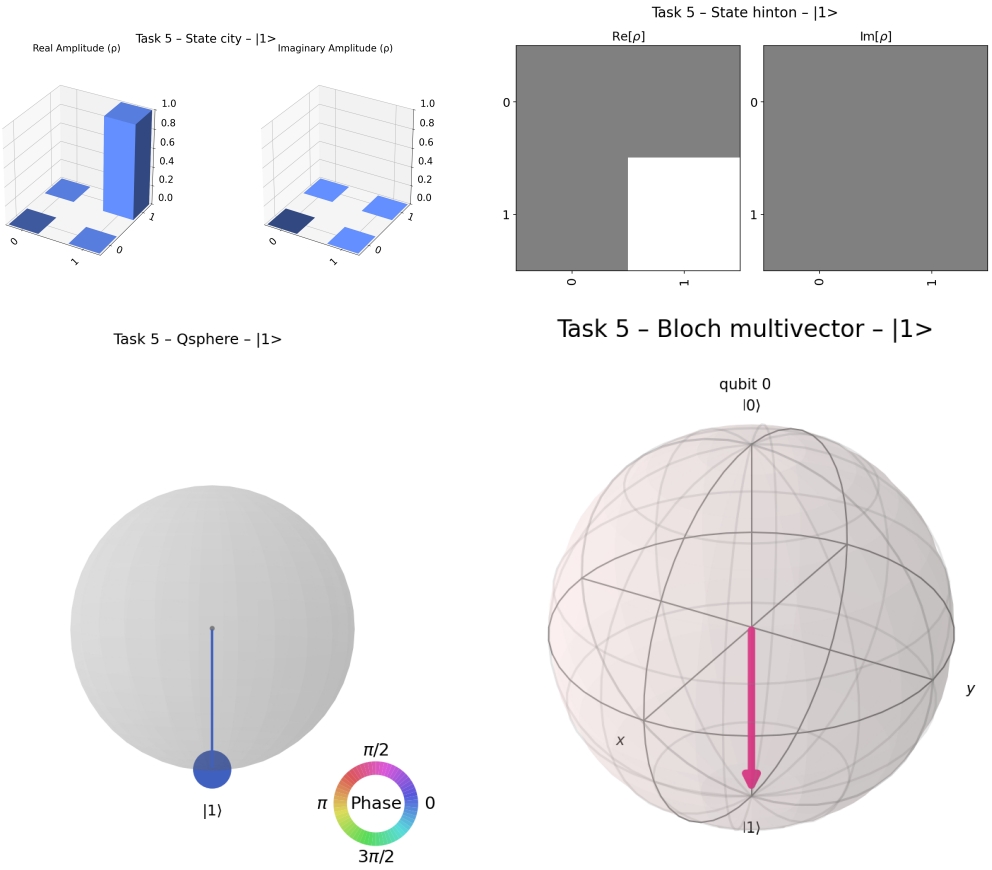

Figure 5.2: Visualisations for $|1\rangle$.

5.4 $|+\rangle$

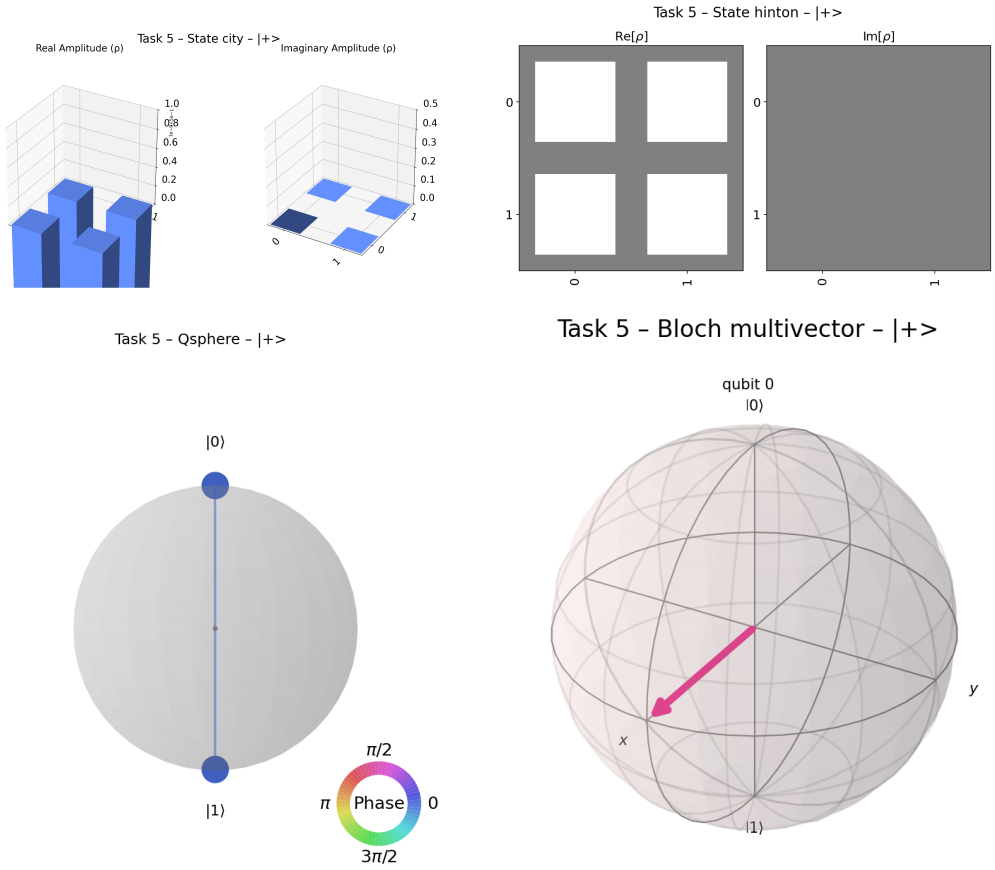

Figure 5.3: Visualisations for $|+\rangle$.