LS: Quantum Computing: Week-0 Assignment Submission

Sravan K Suresh

July, 2023

1 Solution to Question 1

KEY IDEAS:

- Each of the three 4×4 matrices can be factored as the *tensor product* of two 2×2 matrices!

- **Theorem:** *Let A be an $m \times m$ matrix with eigenvalue λ and the corresponding eigenvector $\mathbf{u}$. Let B be an $n \times n$ matrix with eigenvalue μ and the corresponding eigenvector $\mathbf{v}$. Then the matrix $A \otimes B$ has the eigenvalue $\lambda\mu$ with the corresponding eigenvector $u \otimes v$.*

  Proof: From the eigenvalue equations

  $$A\mathbf{u} = \lambda\mathbf{u} \tag{1}$$

  $$B\mathbf{v} = \mu\mathbf{v} \tag{2}$$

  we obtain

  $$A\mathbf{u} \otimes B\mathbf{v} = \lambda\mu\,(\mathbf{u} \otimes \mathbf{v})\} \tag{3}$$

  Since

  $$A\mathbf{u} \otimes B\mathbf{v} \equiv (A \otimes B)(\mathbf{u} \otimes \mathbf{v}) \tag{4}$$

  (from Reference-1)

  we get

  $$(A \otimes B)(\mathbf{u} \otimes \mathbf{v}) = \lambda\mu\,(\mathbf{u} \otimes \mathbf{v}) \tag{5}$$

  This equation is an eigenvalue equation. Consequently, $(\mathbf{u} \otimes \mathbf{v})$ is an *eigenvector* of $(A \otimes B)$ with eigenvalue $\lambda\mu$.

a)

$$\begin{bmatrix} 0 & 5 & 0 & 4 \\ 5 & 0 & 4 & 0 \\ 0 & 3 & 0 & 2 \\ 3 & 0 & 2 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 3 & 2 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Since the Eigenvalues of

$$\begin{bmatrix} 5 & 4 \\ 3 & 2 \end{bmatrix}$$

are $\boldsymbol{\lambda_1} = \frac{1}{2}(7 + \sqrt{57})$ and $\boldsymbol{\lambda_2} = \frac{1}{2}(7 - \sqrt{57})$ and the Eigenvalues of

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

are $\boldsymbol{\mu_1} = -1$ and $\boldsymbol{\mu_2} = 1$

Therefore the *Eigenvalues* of matrix given in a) are:

$\lambda_1\mu_1 = -\frac{1}{2}(7 + \sqrt{57})$

$\lambda_1\mu_2 = \frac{1}{2}(7 + \sqrt{57})$

$\lambda_2\mu_1 = -\frac{1}{2}(7 - \sqrt{57})$

$\lambda_2\mu_2 = \frac{1}{2}(7 - \sqrt{57})$

b) Upon shuffling columns 2 and 3 followed by shuffling rows 2 and 3 of the matrix b), we get back the matrix given in part a). Since Eigenvalues remain unaffected by these operations, all of the Eigenvalues for part b are the same as that of part a).

c)

$$\begin{bmatrix} 25 & 20 & 20 & 16 \\ 15 & 10 & 12 & 8 \\ 15 & 12 & 10 & 8 \\ 9 & 6 & 6 & 4 \end{bmatrix} = \begin{bmatrix} 5 & 4 \\ 3 & 2 \end{bmatrix} \otimes \begin{bmatrix} 5 & 4 \\ 3 & 2 \end{bmatrix}$$

Since the Eigenvalues of

$$\begin{bmatrix} 5 & 4 \\ 3 & 2 \end{bmatrix}$$

are $\boldsymbol{\lambda_1} = \frac{1}{2}(7 + \sqrt{57})$ and $\boldsymbol{\lambda_2} = \frac{1}{2}(7 - \sqrt{57})$

Therefore the *Eigenvalues* of matrix given in a) are:

$\lambda_1^2 = \frac{1}{2}(53 + 7\sqrt{57})$

$\lambda_1\lambda_2 = -2$

$\lambda_2\lambda_1 = -2$

$\lambda_2^2 = \frac{1}{2}(53 - 7\sqrt{57})$

2 Solution to Question 2

Since

$$\boldsymbol{O} = \sum_{\lambda} \lambda P_{\lambda} \tag{6}$$

and

$$\boldsymbol{I} = \sum_{\lambda} P_{\lambda} \tag{7}$$

(from Reference-2)

we get

$$\boldsymbol{O} = P_1 - P_{\text{-}1} \tag{8}$$

and

$$\boldsymbol{I} = P_1 + P_{\text{-}1} \tag{9}$$

Thus, from equations (8) and (9), we obtain $P_{\pm 1} = \frac{\boldsymbol{O} \pm \boldsymbol{I}}{2}$

3 Solution to Question 3

To prove the equivalence between a norm-preserving operator and a unitary operator, we will show that the three conditions stated in the hint are indeed equivalent. Let's denote the norm-preserving operator as $\mathbf{A}$.

Norm Preservation ⇒ Inner Product Preservation:
Assume $\mathbf{A}$ is norm preserving, i.e., $|\mathbf{A}x| = |x| \forall x \in \mathbf{V}$, to prove that $\langle \mathbf{A}x, \mathbf{A}y \rangle = \langle x, y \rangle \ \forall\, x, y \in \mathbf{V}$, let's consider the inner product $\langle \mathbf{A}x, \mathbf{A}y \rangle$.
Using the properties of inner products, we can rewrite this as follows:
$\langle \mathbf{A}x, \mathbf{A}y \rangle = \langle \mathbf{A}^{\dagger}(\mathbf{A}x), y \rangle$ (Taking the adjoint of the first argument).
Now, since $\mathbf{A}$ is norm-preserving, $|\mathbf{A}x| = |x| \ \forall\, x \in \mathbf{V}$. This implies that $(\mathbf{A}^{\dagger}\mathbf{A}x) = x \ \forall\, x \in \mathbf{V}$. Substituting this in the previous expression:

$$\langle \mathbf{A}x, \mathbf{A}y \rangle = \langle x, y \rangle \ \forall\, x, y \in \mathbf{V}$$

Hence, the norm-preserving property implies inner product preservation.
Inner Product Preservation ⇒ Norm Preservation:
Assume $\mathbf{A}$ is norm preserving, i.e., $|\mathbf{A}x| = |x| \ \forall\, x \in \mathbf{V}$, to prove that $\langle \mathbf{A}x \rangle = |x| \ \forall\, x \in \mathbf{V}$.
Taking x = $\mathbf{A}$x in the inner product preservation condition, we have:

$$\langle \mathbf{A}x, \mathbf{A}x \rangle = \langle x, \mathbf{A}x \rangle \ \forall\, x \in \mathbf{V}$$

Now, since $\langle \mathbf{A}x, \mathbf{A}x \rangle = ||\mathbf{A}x||^2$ and $\langle x, \mathbf{A}x \rangle = \langle \mathbf{A}^{\dagger}x, x \rangle$ (adjoint property), we can rewrite the equation as:

$$||\mathbf{A}x||^2 = \langle \mathbf{A}^{\dagger}x, x \rangle$$

Since the inner product is always non-negative, we conclude that $||\mathbf{A}x||^2 \geq 0$.
Additionally, if we choose x = $\mathbf{A}$x, then the equation becomes:

$$||\mathbf{A}x||^2 = \langle \mathbf{A}^\dagger \mathbf{A}x, \mathbf{A}x \rangle$$

Since $\mathbf{A}^\dagger \mathbf{A}x = x$, we have:

$$||\mathbf{A}x||^2 = \langle x, \mathbf{A}x \rangle \; \forall \, x \in \mathbf{V}$$

Taking the square root of both sides, we have:

$$||\mathbf{A}x|| = |x|$$

Hence, the inner product preservation property implies norm preservation.

Inner Product Preservation $\Rightarrow$ Unitary Operator:
Assume $\langle \mathbf{A}x, \mathbf{A}y \rangle = \langle x, y \rangle \; \forall \, x \in \mathbf{V}$, to prove that $\mathbf{A}$ is a unitary operator, i.e.: $\mathbf{A}^\dagger \mathbf{A} = \mathbf{A}\mathbf{A}^\dagger = I$, where I is an identity operator, consider the inner product of $\mathbf{A}^\dagger \mathbf{A}x$ and y for any vectors $x, y \in \mathbf{V}$:

$$\langle \mathbf{A}^\dagger \mathbf{A}x, y \rangle = \langle \mathbf{A}x, \mathbf{A}y \rangle$$

Since we know that $\langle \mathbf{A}x, \mathbf{A}y \rangle = \langle x, y \rangle$, we have:

$$\langle \mathbf{A}^\dagger \mathbf{A}x, y \rangle = \langle \mathbf{A}x, \mathbf{A}y \rangle$$

This implies that $\mathbf{A}^\dagger \mathbf{A}x = x \, \forall \, x \in \mathbf{V}$, using the properties of inner products. Similarly, by considering the inner product of x and $\mathbf{A}^\dagger \mathbf{A}y$, we can show that $\mathbf{A}^\dagger \mathbf{A}y = y \, \forall \, x \in \mathbf{V}$. Hence, $\mathbf{A}^\dagger \mathbf{A} = \mathbf{A}\mathbf{A}^\dagger = I$, where I is an identity operator. Thus, we have shown that norm preservation, inner product preservation, and being a unitary operator are all equivalent conditions.

4 Solution to Question 4

Analysis of the directions of the given quantum states:

- $|0\rangle$: The state $|0\rangle$ represents the Qubit in the state where it is certain to be in the $|0\rangle$ state. On the Bloch sphere, this state corresponds to the **North pole**, pointing in the **positive z-direction**.
- $|1\rangle$: The state $|1\rangle$ represents the Qubit in the state where it is certain to be in the $|1\rangle$ state. On the Bloch sphere, this state corresponds to the **South pole**, pointing in the **negative z-direction**.
- $\frac{(|0\rangle+|1\rangle)}{\sqrt{2}}$: This state is the superposition of $|0\rangle$ and $|1\rangle$, with equal amplitudes. Geometrically, this corresponds to a state located **on the equator** of the Bloch sphere. It is equally distant from the north and south poles, and the direction is **orthogonal to the z-axis**.
- $\frac{(|0\rangle\text{-}|1\rangle)}{\sqrt{2}}$: This state is also a superposition of $|0\rangle$ and $|1\rangle$, but with different relative signs. This state is also located **on the equator** of the Bloch sphere, but in the **opposite direction compared to the previous state**. It is also equally distant from the north and south poles, but the direction is **orthogonal to the z-axis in the opposite direction**.

Similarities among the first two and the last two states:

- The first two states, $|0\rangle$ and $|1\rangle$, are the **basis states of the computational basis**. They are **orthogonal** to each other and located at the **opposite ends of the z-axis**.
- The last two states, $\frac{(|0\rangle+|1\rangle)}{\sqrt{2}}$ and $\frac{(|0\rangle\text{-}|1\rangle)}{\sqrt{2}}$, are superpositions of $|0\rangle$ and $|1\rangle$ with equal amplitudes but different relative signs. Geometrically, they are located on the equator of the Bloch sphere, equidistant from the north and south poles. The only difference between them is the direction of their equatorial position, which is opposite to each other.

Therefore, the main similarity among the first two and the last two states is that they represent orthogonal states on the Bloch sphere, with the last two states being equidistant from the north and south poles but in opposite directions.

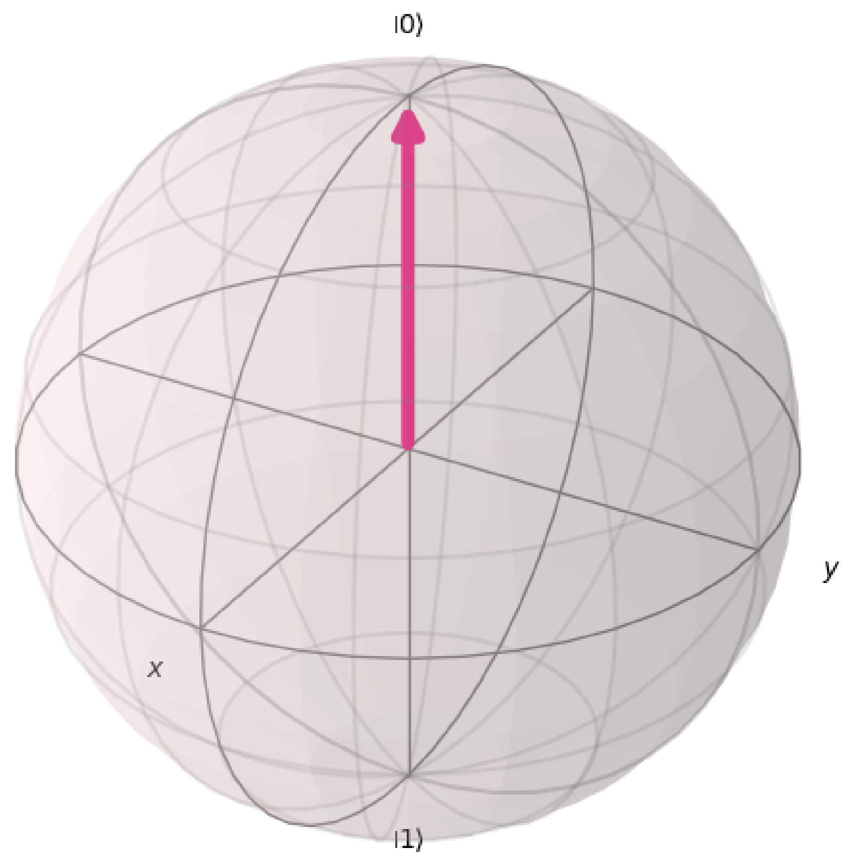
Week_0_Assignment_Q4

July 9, 2023

[6]:
```
from math import pi
from qiskit_textbook.widgets import plot_bloch_vector_spherical
```

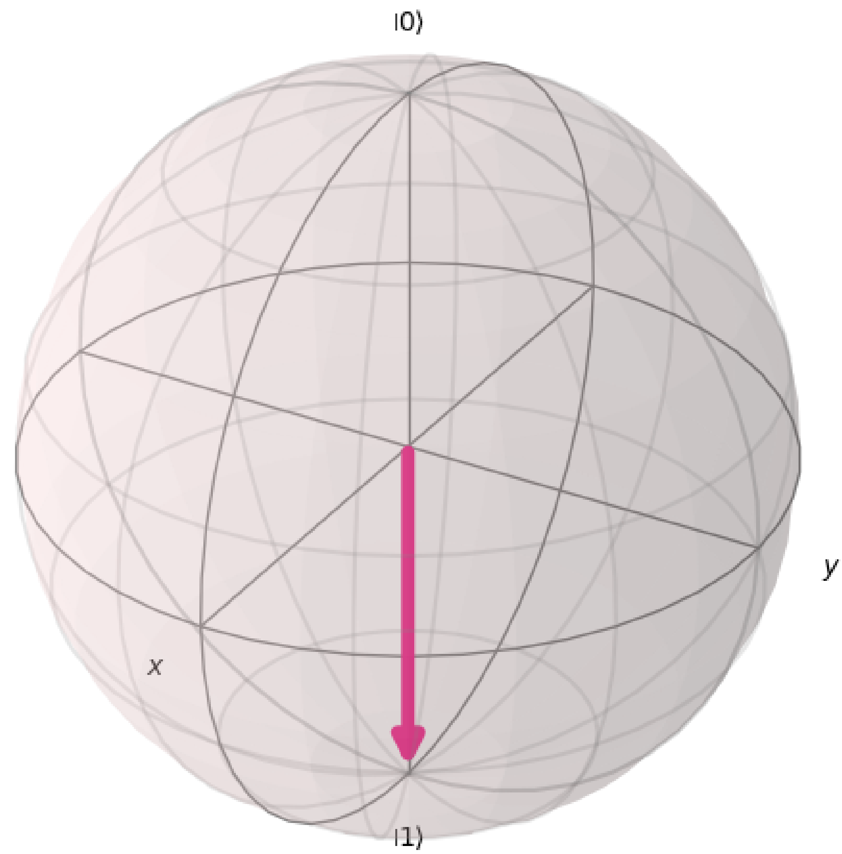
[7]:
```
coords = [0,0,1] # [Theta, Phi, Radius]
plot_bloch_vector_spherical(coords)
```

[7]:



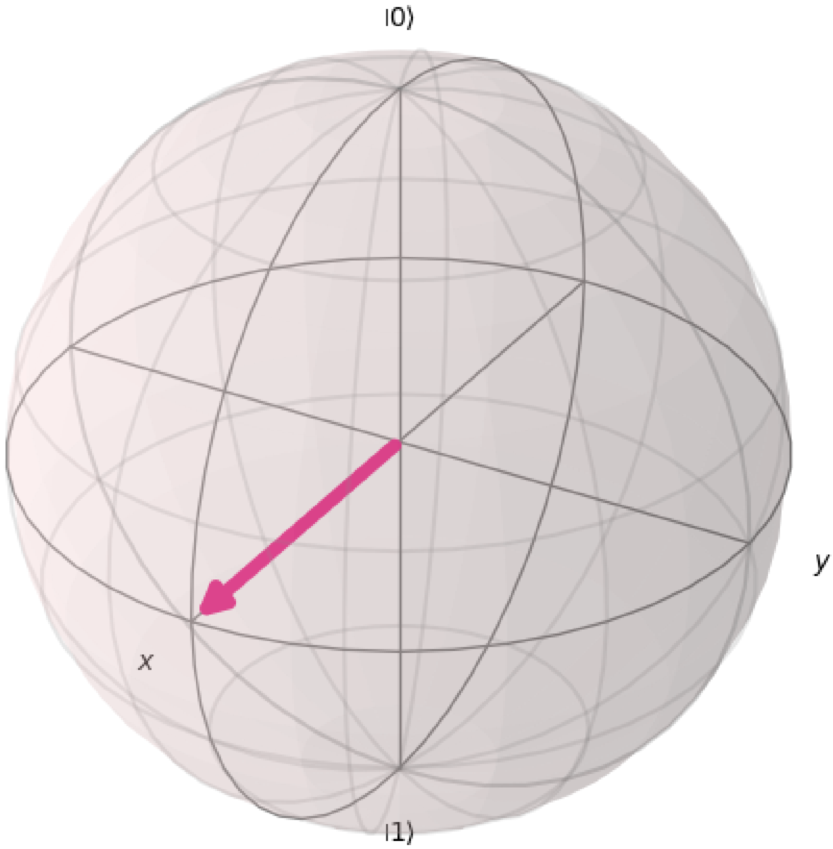
[8]:
```
coords = [pi,0,1] # [Theta, Phi, Radius]
plot_bloch_vector_spherical(coords)
```

[8]:



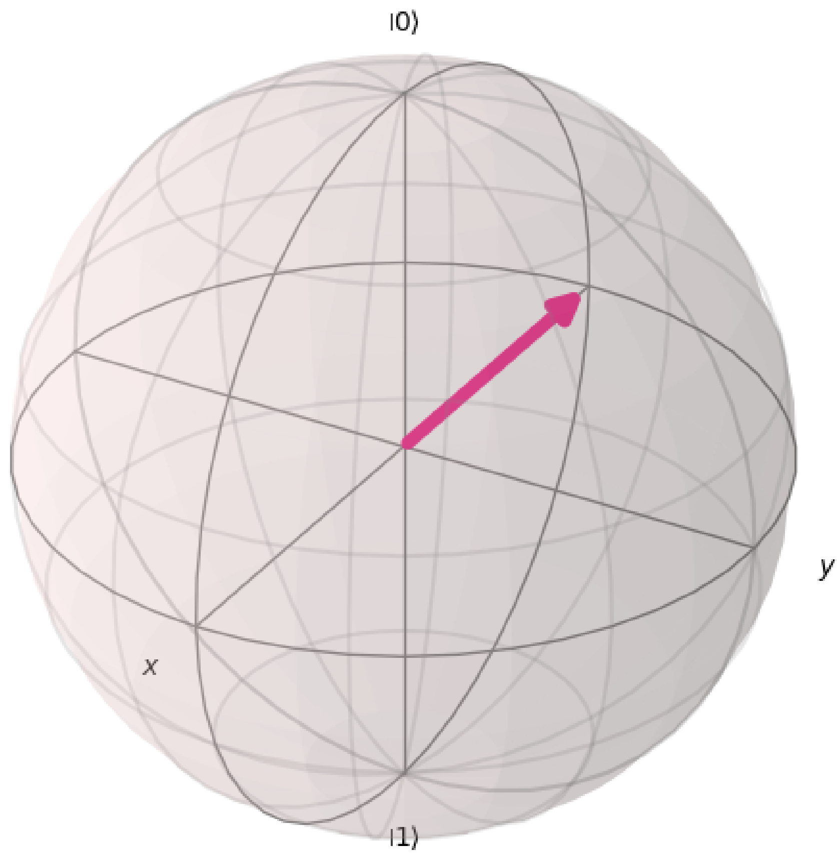
[9]:
```
coords = [pi/2,0,1] # [Theta, Phi, Radius]
plot_bloch_vector_spherical(coords)
```

[9]:



[10]:
```
coords = [pi/2,pi,1] # [Theta, Phi, Radius]
plot_bloch_vector_spherical(coords)
```

[10]:



|0⟩
y
x
|1⟩

5 Solution to Question 5

The circuit starts with the three qubits in the $|0\rangle$ state. The **Hadamard gate** (**H**) is applied to the first qubit, which creates the superposition state $\frac{(|0\rangle+|1\rangle)}{\sqrt{2}}$. Then the two **CNOT gates** (**CX**) are used to entangle the first qubit with the second and third qubits, resulting in the desired state $\frac{(|000\rangle+|111\rangle)}{\sqrt{2}}$.

Generalising this pattern for the case of **'n' Qubits**, the circuit, as before would begin with the n qubits in the $|0\rangle$ state. Just as for the previous cases, the **Hadamard gate** (**H**) would be applied to the first qubit, which creates the superposition state $\frac{(|0\rangle+|1\rangle)}{\sqrt{2}}$. Then the $(n-1)$ **CNOT gates** (**CX**) are used to entangle the first qubit with each of the other qubits, resulting in the desired state $\frac{(|0\cdots0\rangle+|1\cdots1\rangle)}{\sqrt{2}}$.

Week_0_Assignment_Q5

July 9, 2023

[1]:
```python
import qiskit
```

[12]:
```python
# For preparing Bell state \beta_{00} using |00⟩
from qiskit import QuantumCircuit, transpile, assemble, Aer, execute
from math import sqrt

# Create a quantum circuit with 2 qubits
circuit = QuantumCircuit(2)

# Apply Hadamard gate (H) to the first qubit
circuit.h(0)

# Apply CNOT gate (CX) between the first qubit and the second qubit
circuit.cx(0, 1)

# Visualize the circuit
print(circuit)

# Simulate the circuit
simulator = Aer.get_backend('statevector_simulator')
job = execute(circuit, simulator)
result = job.result()
statevector = result.get_statevector()

# Print the resulting statevector
print("Resulting Statevector:")
print(statevector)
```

```
q_0:  H

q_1:  X

Resulting Statevector:
Statevector([0.70710678+0.j, 0.        +0.j, 0.        +0.j,
             0.70710678+0.j],
            dims=(2, 2))
```

[13]:

```python
# Create a quantum circuit with 3 qubits
circuit = QuantumCircuit(3)

# Apply Hadamard gate (H) to the first qubit
circuit.h(0)

# Apply CNOT gate (CX) between the first qubit and the second qubit
circuit.cx(0, 1)

# Apply CNOT gate (CX) between the first qubit and the third qubit
circuit.cx(0, 2)

# Visualize the circuit
print(circuit)

# Simulate the circuit
simulator = Aer.get_backend('statevector_simulator')
job = execute(circuit, simulator)
result = job.result()
statevector = result.get_statevector()

# Print the resulting statevector
print("Resulting Statevector:")
print(statevector)
```

```
     ┌───┐     
q_0: ┤ H ├──■──
     └───┘┌─┴─┐
q_1: ─────┤ X ├
          └───┘

Resulting Statevector:
Statevector([0.70710678+0.j, 0.        +0.j, 0.        +0.j,
             0.        +0.j, 0.        +0.j, 0.        +0.j,
             0.        +0.j, 0.70710678+0.j],
            dims=(2, 2, 2))
```