**Tier 3 Week 2: Computational Formalism & Phase Mastery**

Audience: Undergraduate / Advanced

*Objective: Transition from a geometric understanding of the qubit (the Bloch Sphere) to the underlying computational engine: Linear Algebra.*

Recall that the qubit state can be represent in Bloch sphere, but in this module, we will learn how quantum gates are not just 'rotations' but are precise matrices that operate on state vectors mathematical language: Linear Algebra. Our focus shifts from visual intuition to computational calculation. We will introduce the most important but subtle concept in quantum computing: Phase.

**Recap: The Qubit as a State Vector**

**In Week 1**, we learned that any qubit state can be described by its position on the Bloch Sphere. Mathematically, this position is defined by a state vector, which we write in Dirac Notation as

This vector is a linear combination (a superposition) of the two basis states,

Where and are complex numbers called probability amplitudes. In linear algebra, we represent this abstract vector as a 2x1 column vector:

The Born Rule (covered in Week 1) states that the probability of measuring or is the magnitude squared of its amplitude:

For a valid state, the probabilities must sum to 1: (Normalized).

**Quantum Gates as Matrices**

We can apply gates like (Hadamard) and (Tilt) to visually rotate the qubit on the Bloch Sphere.

Now, we are going to treat these gates as what they truly are: matrices operator. Applying a gate to a qubit is simply performing matrix multiplication.

If a qubit is in an initial state , and we apply a gate the final state is:

All quantum gates must be unitary matrices, which means they preserve the vector's length (normalization): they rotate the vector without shrinking or stretching it, ensuring probabilities still sum to .

**Recap: The Core Gate Matrices**

Here are the matrix forms of the gates that are commonly used, then we are going to introduce the  gate.

*Pauli-X Gate (NOT Gate)*

This is equivalent to an rotation(rotate by  around x axis). It flips to and vice-versa.

*Example:*

*Hadamard Gate (Superposition Gate)*

This is the fundamental gate for creating superposition. It rotates from the Z-basis to the X-basis .

*Example:*

This is one of the rotation gate to tilt the qubit state represent in bloch sphere where it rotates around x-axis of bloch sphere.

*Example:*

The above example, we ignore the  coefficient, which in quantum computing language known as **global phase**, we will discuss it shortly later oh what is global phase.

This is another rotation gate similar function with  Gate by tilting the qubit state in bloch sphere, where the rotation is around y-axis of bloch sphere.

**Checkpoint:**

How does  gate act on qubit when the rotation angle is

**The Gate and Phase**

We now introduce the gate. They are different than  and gates in which they visually change the 'tilt' of the qubit, which directly changes the measurement probabilities (the 'percentage chance' of getting or ).

The gate is different. It represents a rotation around the Z-axis of the Bloch Sphere.

The general matrix is defined as:

gate is another general form of other specific gate. Let's look at two important examples:

**The Gate:**

When setting , we get and

*Recap: Note that* Euler's identity:

So,

This is the  gate, but with a global phase of If we factor out we get. Since global phase is unobservable, we define the standard gate as:

The gate flips the sign (the phase) of the component.

**Checkpoint:**

Try  and  and see what is the difference in both result. Use matrices calculation to help you.

The Gate:

When setting we get:

 For simplicity, the standard gate is defined as:

The gate applies a phase of to the component. This is our key tool for studying phase.

**Phase vs. Probability: The Effect**

What does an gate actually do? Why considering phase?

Let's start with a qubit in the state ( superposition):

Now, let's apply an to this state:

**Did the Measurement Probability Change?**

Let's use the Born Rule on our new state,

The measurement probabilities are exactly the same as the state we started with.

From this example, we can see that  rotation is "hidden" or "subtle." It does not change the Z-basis measurement probabilities (the 'tilt' up or down). Instead, it changes the relative phase(different than the global phase) between the and coefficients.

This phase is invisible if you only measure this one qubit, but it is CRITICAL when this qubit interacts with other qubits (e.g., in a CNOT gate), as it determines whether the states interfere constructively or destructively.

**Global phase and relative phase**

Phase is a fundamental property of complex numbers, and it's what separates quantum computation from classical probability. We must carefully distinguish between these two types:

**Global Phase**

A Global Phase is a phase factor, that multiplies the *entire* state vector.

You already saw this in the example, where the state is considered equivalent to the state What this means is that when we measure the state, we cannot know the difference between  and

This is why we ignore the global phase, because it is unobservable when we apply Born Rule which only cares about the *magnitude squared* of the amplitudes.

For example:

Let's check the probability of measuring any state $|k\rangle$:

Since is just a complex number, we can separate the magnitudes:

Since all measurement probabilities are identical, the global phase has no physical consequence.

**Relative Phase**

A Relative Phase is a phase difference *between* the complex amplitudes of the basis states.

For example, the states and  are different states which have different relative phases.

Why is it critical? While it doesn't change the Z-basis measurement probabilities, it changes the measurement probabilities in a different basis (like the X-basis

**Note:**

This is the mathematical root of **interference**.

Let's prove this by measuring both states in the -basis (i.e., finding the probability of measuring ):

Case 1:

(The state *is* so the probability of measuring it is .)

Case 2:

Now, let's find the probability:

Because the relative phase changed, the probability of measuring changed from to This proves relative phase is physically real, measurable, and the key to quantum algorithms.

**Checkpoint**

Note: To find the result of a sequence of gates, we apply the matrices one by one, starting from the right.

If we apply then the operation is:

Example:

A qubit starts in the state. We apply a Hadamard gate, followed by a gate. What is the final state vector?

*1. Initial State:*

*2. Apply First Gate (Hadamard):*

*3. Apply Second Gate (Z):*