1. Vectors

What it is:

A vector is an ordered list of numbers that can represent a point or a direction in space.

Why it matters in quantum computing:

- **Quantum states** are represented as **vectors** in a *complex vector space* (Hilbert space).
  Example: A qubit's state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{bmatrix}\alpha \\ \beta\end{bmatrix}$$

where $\alpha$and $\beta$are complex numbers.

- The **length (norm)** of this vector must be 1 — representing total probability = 1.
- Superposition and entanglement are expressed using vector addition and tensor products of vectors.

Real example:

If a qubit has a 70% chance of being $|0\rangle$and 30% chance of being $|1\rangle$,
then $|\psi\rangle = \sqrt{0.7}|0\rangle + \sqrt{0.3}|1\rangle$.

---

1. What is a Vector?

A **vector** is simply an **ordered list of numbers** that can represent:

- A **point** in space,
- A **direction**, or
- A **state** of a system.

In **classical computing**, a state is *binary* — for example, a bit is either **0** or **1**.
In **quantum computing**, a state is **continuous and complex**, so we use a **vector** to represent it.

---

2. Example: Classical Bit vs Quantum Bit (Qubit)

Type	**Possible States**	**Representation**
Classical Bit	0 or 1	A single number (0 or 1)
Quantum Bit (Qubit)	Superposition of 0 and 1	A **vector** with two components

A **qubit** is represented mathematically as:

$$|\psi\rangle = \begin{bmatrix}\alpha \\ \beta\end{bmatrix}$$

where

- α= amplitude for state **|0⟩**,
- β= amplitude for state **|1⟩**,
- both $\alpha$and βcan be **complex numbers**, and
- $|\alpha|^2 + |\beta|^2 = 1$(total probability must be 1).

---

3. Why Vectors?

Because in quantum mechanics:

- Every **quantum state** is a **vector** in a mathematical space called **Hilbert Space**.
- Operations (quantum gates) are **matrix transformations** acting on these vectors.

The vector gives us both:

- The **amplitude** (magnitude = probability),
- The **phase** (complex angle = interference effect).

---

4. Simple Example 1: The Basic States

There are two basic vectors that form the foundation of all qubit states:

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \text{and } |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$$

Think of these as the **X and Y axes** — all other qubit states are **combinations** (superpositions) of these two.

---

5. Example 2: Superposition State

Let's take a qubit that is equally likely to be 0 or 1.
That's called a **superposition state**:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix}$$

This means:

- Probability of measuring **0** = $|\frac{1}{\sqrt{2}}|^2 = 0.5$
- Probability of measuring **1** = $|\frac{1}{\sqrt{2}}|^2 = 0.5$

So there's a **50–50 chance** when you measure it.
This is created using the **Hadamard gate**, which acts as a "superposition generator."

---

6. Example 3: Including Complex Numbers

Now consider a qubit:

$$|\psi\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ i\end{bmatrix}$$

Here $i = \sqrt{-1}$(the imaginary unit).

Although the magnitudes are equal (so probabilities are still 50–50),
the **phase difference** (due to i) affects how this qubit **interferes** with others in a quantum circuit.

That's why **complex vectors** are essential — they encode both magnitude and phase information.

---

7. Visualizing a Qubit Vector (Bloch Sphere)

Even though vectors are mathematical, we can **visualize** a single qubit on a sphere:

$$|\psi\rangle = \cos(\frac{\theta}{2})\ |0\rangle + e^{i\phi}\sin(\frac{\theta}{2})\ |1\rangle$$

- θ= angle from the north pole (controls amplitude)
- ϕ= angle around the sphere (controls phase)

Every quantum state corresponds to a **point on the Bloch Sphere**,
and rotations on this sphere represent **quantum gates**.

2. Matrices

What it is:

A **matrix** is a rectangular array of numbers that can represent a **linear transformation** — that is, something that **changes or rotates** a vector.

$$A = \begin{bmatrix}a & b\\ c & d\end{bmatrix}$$

When we multiply a matrix with a vector, we get a **new vector** — the **transformed version** of the original.

$$\text{New Vector} = A \times \text{Old Vector}$$

2. Why Matrices in Quantum Computing?

Because in quantum mechanics:

- A **quantum gate** is a **linear transformation** acting on a **quantum state vector**.
- This transformation must preserve total probability (the length or norm of the vector).

- Hence, every valid quantum gate is represented by a **unitary matrix**, satisfying:

$$U^{\dagger}U = I$$

where:

- $U^{\dagger}$= conjugate transpose of U
- I= identity matrix

This condition ensures **no information is lost** — a critical rule of quantum mechanics.

---

3. Matrix-Vector Multiplication = Applying a Quantum Gate

Suppose you have a qubit in state:

$$| \psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

and you apply a quantum gate Urepresented by a **matrix**:

$$| \psi'\rangle = U | \psi\rangle$$

Then $| \psi'\rangle$is the **new quantum state**.

---

4. Basic Quantum Gates (Matrix Examples)

Let's look at the most important **2×2 matrices** that represent **single-qubit gates**.

◈ **a. Identity Gate (I)**

Does nothing — it keeps the state unchanged.

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

$I | 0\rangle =| 0\rangle$,
$I | 1\rangle =| 1\rangle$.

◈ **b. Pauli-X Gate (Quantum NOT)**

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

This gate **flips** $|0\rangle \leftrightarrow |1\rangle$, like a NOT gate in classical logic.

Example:

$$X \mid 0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \mid 1\rangle$$

The bit is flipped!

---

c. Pauli-Y Gate

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

This gate flips the qubit **and adds a phase rotation**.

It’s a combination of the X gate and a 90° rotation around the Y-axis of the **Bloch sphere**.

---

d. Pauli-Z Gate

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

The Z gate **changes the phase** of the $|1\rangle$ state by 180° (adds a negative sign).

Example:

$$Z \mid 1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} = -\mid 1\rangle$$

Same probability, but the phase (sign) changes — which affects interference.

◆ **e. Hadamard Gate (H)**

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Creates **superposition**.

Example:

$$H \mid 0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{\mid 0\rangle + \mid 1\rangle}{\sqrt{2}}$$

Equal chance of being 0 or 1 — the famous “quantum superposition” state!

---

5. Why Must Quantum Gates Be Matrices (Not Functions)?

In classical computing:

- A logic gate (like AND, OR, NOT) acts on **definite bits**.
- Each input has **one definite output**.

In quantum computing:

- A gate must handle **superpositions**.
- That means it must **linearly transform** vectors — because the **superposition principle** says:

$$U(\alpha \mid 0\rangle + \beta \mid 1\rangle) = \alpha U \mid 0\rangle + \beta U \mid 1\rangle$$

Only **matrix operations** preserve this rule.

---

6. Multi-Qubit Systems (Larger Matrices)

If we have 2 qubits, the combined state is a **4-dimensional vector** (2^2 = 4 elements).

For example:

$$\mid 00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \mid 11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

A two-qubit gate like **CNOT** (controlled NOT) is represented by a **4×4 matrix**:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

It flips the **second qubit** (target) only when the **first qubit** (control) is 1.

---

7. Unitary and Hermitian Matrices

Type	**Property**	**Meaning in Quantum**
Unitary	$U^\dagger U = I$	Preserves probability and reversibility
Hermitian	$H^\dagger = H$	Represents observable/measurable quantities

All quantum gates are **unitary**; all measurements are represented by **Hermitian** matrices.

---

8. Step-by-Step Simple Example

Let’s apply a **Z gate** to a superposition state.

$$| \psi\rangle = \frac{1}{\sqrt{2}}[\begin{smallmatrix}1\\1\end{smallmatrix}]$$
$$Z = [\begin{smallmatrix}1 & 0\\0 & -1\end{smallmatrix}]$$
$$Z | \psi\rangle = \frac{1}{\sqrt{2}}[\begin{smallmatrix}1 & 0\\0 & -1\end{smallmatrix}][\begin{smallmatrix}1\\1\end{smallmatrix}] = \frac{1}{\sqrt{2}}[\begin{smallmatrix}1\\-1\end{smallmatrix}]$$

This new state still has equal probabilities for 0 and 1,
but now has a **phase difference** — important for interference in quantum algorithms.

Why You Must Learn Matrices

Because matrices are:

- The **language** of quantum operations.
- The only way to **mathematically describe** how a qubit changes.
- The bridge between **abstract physics** and **computable quantum circuits**.

Without matrices, you can't:

- Build quantum gates,
- Simulate qubit evolution,
- Understand interference or entanglement.

---

3. Complex Numbers

What it is:

A complex number has a **real** part and an **imaginary** part:

$$z = a + bi$$

Why it matters in quantum computing:

- Quantum probability amplitudes (like α, β) are **complex**, not just real.
- The imaginary part allows for **interference**, a uniquely quantum phenomenon.
- Measuring a quantum system involves the **magnitude squared** of complex numbers:

$$| \alpha |^2 + | \beta |^2 = 1$$

- Phases (arguments of complex numbers) influence how quantum states interfere — critical for algorithms like **Grover's** and **Shor's**.

Real example:

A quantum gate might rotate a qubit's phase by multiplying by $e^{i\theta}$.
This complex rotation changes **interference patterns** between qubits.

---

4. Conjugate Transpose (Hermitian Adjoint)

What it is:

The **conjugate transpose** of a matrix is found by:

1. Transposing it (flipping rows ↔ columns),
2. Taking the complex conjugate of each entry.

$$A^{\dagger} = (\bar{A})^{T}$$

Why it matters in quantum computing:

- **Unitary operators** (quantum gates) satisfy $U^{\dagger}U = I$, ensuring no loss of information.
- **Hermitian operators** (like observables for measurement) have real eigenvalues, corresponding to measurable quantities (like spin, energy, etc.).
- The **inner product** between quantum states uses the conjugate transpose to calculate probability amplitudes.

Example:

If $\mid \psi \rangle = [\begin{matrix} \alpha \\ \beta \end{matrix}]$,
then $\langle \psi \mid = [\alpha^{*}, \beta^{*}]$.
The inner product $\langle \psi \mid \psi \rangle = \mid \alpha \mid^{2} + \mid \beta \mid^{2} = 1$.

---

5. Inner and Outer Products

Inner Product:

$$\langle \phi \mid \psi \rangle$$

Measures **similarity** or **overlap** between two quantum states.

- If $\langle \phi \mid \psi \rangle = 0$, states are **orthogonal** (completely distinguishable).
- Used to compute **probabilities** and **measurement outcomes**.
- Core to quantum state normalization and interference.

Outer Product:

$$\mid \psi \rangle \langle \phi \mid$$

Produces an **operator** (matrix) from two state vectors.

- Used to construct **projectors**, **density matrices**, and **quantum measurements**.
- Essential for representing mixed states and quantum decoherence in quantum information theory.

Example:

Projector onto state $|\psi\rangle$:

$$P = |\psi\rangle\langle\psi|$$

helps calculate measurement probabilities.

Summary Table

Topic	Quantum Role	Example
Vectors	Represent quantum states (qubits, superpositions)	
Matrices	Represent quantum gates (unitary transformations)	$H, X, Y, Z, CNOT$
Complex Numbers	Encode phase & probability amplitude	$e^{i\theta}$,
Conjugate Transpose	Ensures unitarity, used in Hermitian operators	$U^\dagger U = I$
Inner Product	Measures overlap (probability amplitude)	
Outer Product	Forms operators and projectors	

In simple terms

Quantum computing **uses math as a language** — and these concepts are the **alphabet**.
Without understanding vectors, matrices, and complex numbers, it's impossible to describe:

- What a qubit *is*,
- How it *evolves* under a quantum gate,
- How we *measure* and extract information.

Why You Must Learn Matrices

Because matrices are:

- The **language** of quantum operations.
- The only way to **mathematically describe** how a qubit changes.
- The bridge between **abstract physics** and **computable quantum circuits**.

Without matrices, you can't:

- Build quantum gates,
- Simulate qubit evolution,
- Understand interference or entanglement.

In summary:

Vectors represent quantum states.
Matrices represent quantum gates.
Matrix × Vector = New Quantum State.

Let's now explore **Complex Numbers in Quantum Computing** in a clear, practical, and example-rich way — connecting the math directly to **quantum states**, **phases**, and **interference**.

Complex Numbers in Quantum Computing

---

1. What is a Complex Number?

A **complex number** has two parts:

$$z = a + bi$$

where

- a= **real part**
- b= **imaginary part**
- $i = \sqrt{-1}$

You can think of a complex number as a **point on a 2D plane**, with:

- The **x-axis** = real part
- The **y-axis** = imaginary part

Example:

$$z = 3 + 4i$$

This can be represented as the point (3, 4) in the complex plane.

Its **magnitude (length)** is:

$$| z |= \sqrt{3^2 + 4^2} = 5$$

Its **angle (phase)** is:

$$\theta = \tan^{-1}(\frac{4}{3})$$

2. Complex Numbers as Rotations

Complex numbers can be written in **polar form**:

$$z = re^{i\theta} = r(\cos\theta + i\sin\theta)$$

where:

- r= magnitude (length)
- θ= phase angle

Multiplying by $e^{i\theta}$**rotates** a complex number on the plane —
this idea is directly used to rotate **quantum states** in phase space.

---

3. Why Complex Numbers in Quantum Computing?

Quantum mechanics is inherently **wave-like**.
Every quantum state behaves like a **wave** that can **interfere** — and waves naturally use **complex numbers** to represent:

- **Amplitude (height)** — determines probability
- **Phase (angle)** — determines interference pattern

So, **complex numbers** are used to express both the **magnitude** and **phase** of a quantum state.

---

4. Qubit State with Complex Numbers

A single qubit can be written as:

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

where both $\alpha$and $\beta$are **complex** numbers.

Normalization condition:

$$|\alpha|^2 + |\beta|^2 = 1$$

Here:

$$|\alpha|^2 = \alpha^*\alpha, |\beta|^2 = \beta^*\beta$$

which means we use **complex conjugates** to compute real probabilities.

Example:

$$| \psi \rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ i \end{bmatrix}$$

$$| \alpha |^2 = | \frac{1}{\sqrt{2}} |^2 = 0.5$$

$$| \beta |^2 = | \frac{i}{\sqrt{2}} |^2 = 0.5$$

Total probability = 1.
So, the qubit has equal chances of being 0 or 1,
but the **imaginary phase (i)** changes how it interferes with other qubits.

---

5. Phase and Interference — Why It Matters

Quantum computing power comes from **interference** — how probability amplitudes **add or cancel**.

Consider two quantum paths:

- $+\frac{1}{\sqrt{2}}$
- $-\frac{1}{\sqrt{2}}$

If you add them:

$$\frac{1}{\sqrt{2}} + (-\frac{1}{\sqrt{2}}) = 0$$

They cancel out — **destructive interference**.

Now add:

$$\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} = \sqrt{2}$$

They reinforce — **constructive interference**.

Complex phases control this interference pattern, allowing quantum algorithms (like Grover's) to amplify the correct answer and suppress the wrong ones.

---

6. Quantum Phase Representation

You can think of each qubit amplitude as a **rotating vector** (phasor) on a circle.

- 1→ points along +x axis
- i→ points along +y axis
- -1→ points along -x axis
- $-i$→ points along -y axis

When quantum gates act, they **rotate** these vectors around the circle —
changing the **phase**, which later determines interference during measurement.

Example: Phase Gate (S Gate)

$$S = \begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}$$

This gate multiplies the $|1\rangle$ amplitude by **i**, rotating it by 90° on the complex plane.

Before:

$$| \psi\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix}$$

After S gate:

$$S | \psi\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ i\end{bmatrix}$$

Same probabilities, but different phase → changes future interference effects.

---

7. Global vs Relative Phase

Type	Effect	Example	Result
Global Phase	Multiplies whole state by same complex number	(e^{i\theta}	\psi\rangle)
Relative Phase	Changes phase of one component	$[\alpha, e^{i\phi}\beta]$	Changes **interference**, affects computation

Only **relative phase** affects results — it's how algorithms like **QFT** (Quantum Fourier Transform) work.

---

8. Step-by-Step Example

Let's see how complex numbers modify interference.

Start:

$$| \psi\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix}$$

Apply a **Z gate**:

$$Z = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$$

$$Z \mid \psi\rangle = \frac{1}{\sqrt{2}}[\begin{smallmatrix}1\\-1\end{smallmatrix}]$$

Now, apply **Hadamard**:

$$H(Z \mid \psi\rangle) = H(\frac{1}{\sqrt{2}}[\begin{smallmatrix}1\\-1\end{smallmatrix}]) = \frac{1}{2}[\begin{smallmatrix}1-1\\1+1\end{smallmatrix}] = [\begin{smallmatrix}0\\1\end{smallmatrix}] = \mid 1\rangle$$

By introducing a **phase difference**, we made the probability of $|1\rangle$ = 100%.
That's the power of **complex phases** in quantum interference!

---

9. Visualization Summary

Representation	Meaning
Real part	Probability amplitude's real component
Imaginary part	Phase rotation component
Magnitude	Measurement probability
Angle (phase)	Determines interference behavior

---

10. Intuitive Analogy

Think of each quantum amplitude like a **wave** in the ocean:

- **Height (magnitude)** = how strong it is
- **Direction (phase)** = which way it's moving
- When waves meet, they **add up** or **cancel out**
- Complex numbers are the **language of these waves**

That's exactly how qubits behave — **they're waves of probability**, and complex numbers encode their **shape and direction**.

In summary:

- Quantum states are complex vectors.
- Complex numbers allow us to represent **phase** and **interference**.
- Without them, quantum effects like **superposition**, **entanglement**, and **interference** could not exist.

Would you like me to continue next with
Conjugate Transpose (Hermitian Adjoint) — explaining what it is, why it's needed in quantum math, and how it ensures probability preservation (with simple examples)?

Understanding Global vs Relative Phase in Quantum Computing

In quantum computing, every **quantum state** can be represented as a **vector** in complex space.
For a single qubit, the general state is:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where
α, β are **complex numbers**, and $|\alpha|^2 + |\beta|^2 = 1$.

---

1. What is a *Phase*?

A **phase** is the *angle* part of a complex number — the rotation in the complex plane.

For a complex number:

$$z = re^{i\theta}$$

- r → magnitude (length)
- $e^{i\theta}$ → phase rotation
  (think of it as turning a vector in the complex plane by angle θ)

---

2. Global Phase

A **global phase** is when you multiply the **entire quantum state** by the same complex number $e^{i\theta}$:

$$|\psi'\rangle = e^{i\theta}|\psi\rangle$$

This means:

$$|\psi'\rangle = e^{i\theta}(\alpha|0\rangle + \beta|1\rangle)$$

Key Idea:
Global phase does **not** change measurement probabilities because when you measure, you compute $|\alpha|^2$ and $|\beta|^2$, and the global factor $e^{i\theta}$ cancels out.

Think of it like turning the entire coordinate system — it looks different, but physics stays the same.

Example:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Multiply by $e^{i\pi/2}$:

$$|\psi'\rangle = e^{i\pi/2}\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Both states represent the **same physical situation** → same measurement results.

---

3. Relative Phase

A **relative phase** changes the phase of only **one part** of the superposition — this one *does* affect outcomes!

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Both have equal magnitudes, but the **sign (phase difference)** between $|0\rangle$and $|1\rangle$is different.

When you measure in the standard computational basis, both look 50–50.
But — in **interference experiments** (like applying a Hadamard gate again), they behave differently!

---

4. Why It Matters in Quantum Computing

Type	Effect on Computation	Measurement Impact	Example Algorithm
Global Phase	No effect	None	Ignored in all computations
Relative Phase	Changes interference patterns	Affects output	QFT, Grover's, Shor's, Phase Estimation

---

5. Real-world Analogy

Imagine two sound waves
if both waves shift together (global phase), your ear hears **no difference**.
But if one wave shifts relative to the other (relative phase), **interference** changes — you may hear silence or louder sound!

Same in quantum: interference patterns depend on *relative*, not *global*, phase.