QuantumKata course

Hugo Thomas - hugothomas@tuta.io

2020/2021

Contents

Complex Arithmetic

Basic operations

Powers of i

When raising i to an integer power, the result will vary according to a certain pattern as follows: $i^n = i^{n \bmod 4}$, where `mod` is the rest of the euclidian division of n by 4.

Here is the pattern:

Power of i	i^0	i^1	i^2	i^3	i^4	i^5	i^6	i^7	i^8	...
Result	1	i	-1	$-i$	1	i	-1	$-i$	1	...

Complex addition

Let $x = a + ib$ and $y = c + id \in \mathbb{C}$:

$$z = x + y = (a + ib) + (c + id) = \underbrace{(a + c)}_{\text{real}} + \underbrace{(b + d)}_{\text{imaginary}} i$$

Complex multiplication

Let $x = a + ib$ and $y = c + id \in \mathbb{C}$:

$$z = x \cdot y = (a + bi)(c + di) = a \cdot c + a \cdot di + c \cdot bi + bi \cdot di = \underbrace{a \cdot c - b \cdot d}_{\text{real}} + \underbrace{(a \cdot d + c \cdot b)i}_{\text{imaginary}}$$

Complex conjugate

Let $z = a + ib \in \mathbb{C}$, the conjugate of z, denoted $\overline{z}$, is:

$$\overline{z} = a - ib$$

To get the conjugate of a complex number, you change the sign of the imaginary part.

Complex division

Let $x = a + ib$ and $y = c + id \in \mathbb{C}$:

$$z = \frac{x}{y} = \frac{x \cdot \overline{y}}{y \cdot \overline{y}} = \frac{(a + bi)(c - di)}{(c + di)(c - di)} = \frac{a \cdot c + bi \cdot c - a \cdot di - bi \cdot di}{c \cdot c + di \cdot c - c \cdot di - di \cdot di} = \frac{a \cdot c + b \cdot d + (a \cdot (-d) + c \cdot b)i}{c^2 + d^2}$$

We finally can re-wrote our division to have a complex multiplication expression in the numerator and a real number in the denominator:

$$\frac{a + bi}{r} = \frac{a}{r} + \frac{b}{r}i$$

Modulus

The modulus of a complex number can be seen as the distance from the origin to the z point.

Let $z = a + ib \in \mathbb{C}$,

$$|z| = \sqrt{a^2 + b^2} = \sqrt{z \cdot \overline{z}}$$

Like the conjugate, the modulus distributes over multiplication.

$$|x \cdot y| = |x| \cdot |y|$$

Unlike the conjugate, however, the modulus doesn't distribute over addition. Instead, the interaction of the two comes from the triangle inequality:

$$|x + y| \leq |x| + |y|$$

Exponents

Imaginary exponents

To raise a real number to imaginary powers, we need to use the Euler's constant e, defined as follows:

$$e^{i\theta} = \cos\theta + i\sin\theta$$

Complex exponents

Let $z = a + ib \in \mathbb{C}$.

$$e^z = e^{a+ib} = e^a \cdot e^{ib} = e^a(\cos b + i\sin b) = \underbrace{e^a \cos b}_{\text{real}} + \underbrace{e^a \sin b}_{\text{imaginary}}\, i$$

Complex power of real numbers

Let $r \in \mathbb{R}$ and $z = a + ib \in \mathbb{Z}$.

First, we rewrite r^z into a product of two powers:

$$r^{a+bi} = r^a \cdot r^{bi}$$

Given that $r = e^{\ln r}$ (ln is the natural logarithm), we can rewrite the second part of the product as follows:

$$r^{bi} = e^{bi\ln r}$$

Now, given $e^{i\theta} = \cos\theta + i\sin\theta$, we can rewrite it further as follows:

$$e^{bi\ln r} = \cos(b \cdot \ln r) + i\sin(b \cdot \ln r)$$

When substituting this into our original expression, we get:

$$\underbrace{r^a \cos(b \cdot \ln r)}_{real} + \underbrace{r^a \sin(b \cdot \ln r) i}_{imaginary}$$

Cartesian and polar forms

Cartesian to polar conversion

Let $z = a + ib \in \mathbb{C}$, the polar representation of z is $z = re^{i\theta}$, i.e., the distance from origin r and phase θ.

- r should be non-negative: $r \geq 0$
- θ should be between $-\pi$ and π: $-\pi < \theta \leq \pi$

We need to calculate the r and θ values as seen in the complex plane. r should be familiar to you already, since it is the modulus of a number (exercise 6):

$$r = \sqrt{a^2 + b^2}$$

θ can be calculated using trigonometry: since we know that the polar and the Cartesian forms of the number represent the same value, we can write

$$re^{i\theta} = a + bi$$

Euler's formula allows us to express the left part of the equation as

$$re^{i\theta} = r\cos\theta + ir\sin\theta$$

So we get

$$a + bi = r\cos\theta + ir\sin\theta$$

For two complex numbers to be equal, their real and imaginary parts have to be equal. This gives us the following system of equations:

$$\begin{cases} a = r\cos\theta \\ b = r\sin\theta \end{cases}$$

To calculate θ, we can divide the second equation by the first one to get

$$\tan\theta = \frac{b}{a}$$

$$\theta = \arctan\left(\frac{b}{a}\right)$$

Polar multiplication

Let $x = r_1 e^{i\theta_1}$ and $y = r_2 e^{i\theta_2}$.

Multiplying two complex numbers in polar form can be done efficiently in the following way:

$$z = x \cdot y = r_1 e^{\theta_1 i} \cdot r_2 e^{\theta_2 i} = r_1 r_2 \cdot e^{(\theta_1 + \theta_2)i}$$

Arbitrary complex exponent

Let $x = a + ib$ and $y = c + id \in \mathbb{C}$:

Let's convert the number x to polar form $x = re^{i\theta}$ and rewrite the complex exponent as follows:

$$x^y = \left(re^{i\theta}\right)^{c+di} = e^{(\ln(r)+i\theta)(c+di)} = e^{\ln(r)\cdot c \,+\, \ln(r)\cdot di \,+\, i\theta\cdot c \,+\, d\theta i^2} = e^{(\ln(r)\cdot c \,-\, d\theta) \,+\, (\ln(r)\cdot d \,+\, \theta c)i}$$

Finally, this needs to be converted back to Cartesian form using Euler's formula:

$$e^{\ln(r)\cdot c - d\theta} \cdot (\cos(\ln(r) \cdot d + \theta c) + i\sin(\ln(r) \cdot d + \theta c))$$

Qubit

Matrix representation

The state of a qubit is represented by a complex vector of size 2:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

This vector is normalized: $|\alpha|^2 + |\beta|^2 = 1$.

Basis state

A qubit in state 0 would be represented by the following vector:

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

Likewise, a qubit in state 1 would be represented by this vector:

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

Note that you can use scalar multiplication and vector addition to express any qubit state as a sum of these two vectors with certain weights (known as **linear combination**):

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \beta \end{bmatrix} = \alpha \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Because of this, these two states are known as **basis states**.

These two vectors have two additional properties. First, as mentioned before, both are **normalized**:

$$\langle \begin{bmatrix} 1 \\ 0 \end{bmatrix} , \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rangle = \langle \begin{bmatrix} 0 \\ 1 \end{bmatrix} , \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rangle = 1$$

Second, they are **orthogonal** to each other:

$$\langle \begin{bmatrix} 1 \\ 0 \end{bmatrix} , \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rangle = \langle \begin{bmatrix} 0 \\ 1 \end{bmatrix} , \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rangle = 0$$

> As a reminder, $\langle V, W \rangle$ is the inner product of V and W.

This means that these vectors form an **orthonormal basis**. The basis of $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ is called the **computational basis**, also known as the **canonical basis**.

> There exist other orthonormal bases, for example, the **Hadamard basis**, formed by the vectors
>
> $$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \text{ and } \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$
>
> You can check that these vectors are normalized, and orthogonal to each other. Any qubit state can be expressed as a linear combination of these vectors:

$$\begin{bmatrix}\alpha \\ \beta\end{bmatrix} = \frac{\alpha + \beta}{\sqrt{2}} \begin{bmatrix}\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}}\end{bmatrix} + \frac{\alpha - \beta}{\sqrt{2}} \begin{bmatrix}\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}}\end{bmatrix}$$

The Hadamard basis is widely used in quantum computing, for example, in the BB84 quantum key distribution protocol.

Dirac Notation

Writing out each vector when doing quantum calculations takes up a lot of space, and this will get even worse once we introduce quantum gates and multi-qubit systems. **Dirac notation** is a shorthand notation that helps solve this issue. In Dirac notation, a vector is denoted by a symbol called a **ket**. For example, a qubit in state 0 is represented by the ket $|0\rangle$, and a qubit in state 1 is represented by the ket $|1\rangle$:

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

These two kets represent basis states, so they can be used to represent any other state:

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

Any symbol other than 0 or 1 within the ket can be used to represent arbitrary vectors, similar to how variables are used in algebra:

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

Several ket symbols have a generally accepted use, such as:

$$|+\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big) \quad |-\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - |1\rangle\big) \quad |i\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + i|1\rangle\big) \quad |-i\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle - i|1\rangle\big)$$

Q#: Qubit data type

```
// This statement allocates a qubit, and binds it to the variable q
use q = Qubit();
// You can work with the qubit here
// The qubit is deallocated once it's not used any longer
```

Freshly allocated qubits start out in state $|0\rangle$, and have to be returned to that state by the time they are released. If you attempt to release a qubit in any state other than $|0\rangle$, your program will throw a `ReleasedQubitsAreNotInZeroStateException`.

Bell states

```
operation BellState (qs : Qubit[]) : Unit is Adj {
    H(qs[0]);
    CNOT(qs[0], qs[1]);
}
```