An introduction lecture to quantum computing and time evolution

Meijian Li*

Instituto Galego de Fisica de Altas Enerxias (IGFAE),

Universidade de Santiago de Compostela, E-15782 Galicia, Spain

This lecture note is written for "Quantpostela: A Festa da Computación Cuántica no IGFAE", Oct 18-20, IGFAE, the IGFAE Quantum Computing Workshop 2023. It is intended to provide basic knowledge and selective perspectives on quantum computing in two 1-hour lectures.

CONTENTS

* meijian.li@usc.es

I. QUANTUM COMPUTING BASICS

[Comment] The material of this section is mainly based on the book of quantum computing by Nielsen and Chuang [1], Chapters 1-4. A good beginner reading can be found at Ref. [2].

A. An ultra-brief history

The development of quantum computing involves the contribution from many fields, including quantum mechanics, computer science, information theory, and cryptography. In the following, we will take an ultra-brief historical tour by visiting several milestones.

- Discovery of "Quantum"

    1. In 1905, Albert Einstein explains the photoelectric effect–shining light on certain materials can function to release electrons from the material, and suggests that light itself consists of individual quantum particles or photons.
    2. In 1924, the term "quantum mechanics" is first used in a paper by Max Born.

    The quantum nature of the physical world is unveiled, and thus proceeds the substantial progression of quantum mechanics, and later the quantum field theory.

- Development of "Computing"

    3. David Hilbert's 1928 problem: "what can humans know about mathematics, in principle, and what (if any) parts of mathematics are forever unknowable by humans?"
    4. To Tackle this problem, in 1936, Alan Turing described what we now call a Turing machine: a single, universal programmable computing device that Turing argued could perform any algorithm whatsoever.

    What follows is the continuous and extensive development of computers and computing techniques. The narrative also leads to another fascinating story of the incompleteness theorems, but we will not delve into that direction here.

- The concept of "Quantum Computing"

5. In 1982, Richard Feynman suggested that building computers based on the principles of quantum mechanics would allow us to avoid the essential difficulties in simulating quantum mechanical systems on classical computers.

6. In 1985, David Deutsch invented a new type of computing system, a quantum computer, with stating " 'quantum parallelism', a method by which certain probabilistic tasks can be performed faster by a universal quantum computer than by any classical restriction of it."

A significant motivation for developing quantum computers is to overcome the limitations of classical computers.

- Quantum advantage (over classical)

7. In 1994, Peter Shor demonstrated that the problem of finding the prime factors of an integer, and the 'discrete logarithm' problem could be solved efficiently on a quantum computer.

8. In 1995, Lov Grover showed that the problem of conducting a search through some unstructured search space could also be sped up on a quantum computer.

More and more studies demonstrated the advantages of quantum computing over classical computing with theoretical algorithms.

- NISQ era

The current state of quantum computing is referred to as the noisy intermediate-scale quantum (NISQ) era, characterized by quantum processors containing up to 1000 qubits which are not advanced enough yet for fault-tolerance or large enough to achieve quantum supremacy. The race for quantum supremacy continues, involving a wide range of companies, universities, and research institues. Noticeably, during the same time as the workshop, a mere 500 meters from the venue, the Galician Supercomputing Center (CESGA), installed the most powerful quantum computer in Spain and one of the first in Europe, "Qmio", a 32-qubit computer based on superconducting technology.

B. Basics of quantum mechanics

Quantum mechanics is a mathematical framework for the development of physical theories. As a brief overview, let us write out its postulates as listed in Ref. [2] the language of state vectors:

- Postulate 1: Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the state space of the system. The system is completely described by its state vector, which is a unit vector in the system's state space.

- Postulate 2: The evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2,

$$|\psi'\rangle = U\,|\psi\rangle \ . \tag{1}$$

- Postulate 3: Quantum measurements are described by a collection $\{M_m\}$ of measurement operators. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that result m occurs is given by

$$p(m) = \langle\psi|\, M_m^\dagger M_m\,|\psi\rangle \ , \tag{2}$$

and the state of the system after the measurement is

$$\frac{M_m\,|\psi\rangle}{\sqrt{\langle\psi|\, M_m^\dagger M_m\,|\psi\rangle}} \ . \tag{3}$$

The measurement operators satisfy the completeness equation,

$$\sum_m M_m^\dagger M_m = 1 \ . \tag{4}$$

- Postulate 4: The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through n, and system number i is prepared in the state $|\psi_i\rangle$, then the joint state of the total system is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$.

C. Quantum bits

1. *Single qubit*

A bit is a state of 0 or 1, a mathematical concept in classical computing. Bits are stored as tiny electric charges on nanometer-scale capacitors.

In analogy, a quantum bit, i.e., qubit [1], is a mathematical concept in quantum computing. It is a state of 2-dimensional vector,

$$|\psi\rangle = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} , \tag{5}$$

in which c_1 and c_2 are two complex numbers, and the constraint is $|c_1|^2 + |c_2|^2 = 1$, i.e., the normalization constraint. The state of the qubit can be stored on an electron, photon, or an atom. Two special states, the computational basis states, correspond to the two states of the classical bit 0 and 1,

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} , \qquad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix} . \tag{6}$$

A general qubit state is a *superposition*, or *linear combination* of them, $|\psi\rangle = c_1 |0\rangle + c_2 |1\rangle$ as in Eq. (5). The coefficients c_1 and c_2 are known as the *amplitudes*. If we measure a qubit of the state $|\psi\rangle$, then the outcome is a classical bit of 0 with probability $|c_1|^2$, or 1 with probability $|c_2|^2$. The qubit after the measurement is $|0\rangle$ or $|1\rangle$ respectively.

Because of the normalization constraint, we can also write Eq. (5) as

$$|\psi\rangle = e^{i\gamma} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \right) , \tag{7}$$

where γ, θ, and ϕ are real numbers. The factor $e^{i\gamma}$ is an overall phase and does not contribute to observable effects, so it is often ignored. The number θ and ϕ define a point on the unit three-dimensional sphere, namely the Bloch sphere,[2] as illustrated in Fig. 1.

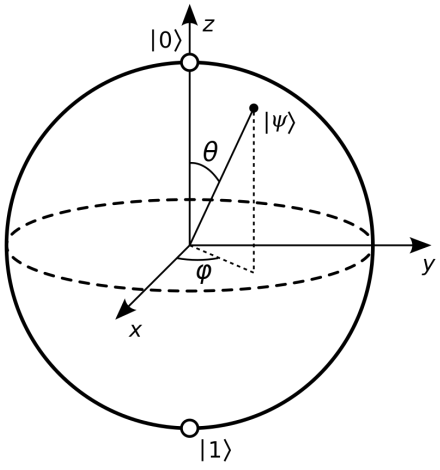


FIG. 1. Bloch sphere representation of a qubit.

[1] The coining of the term qubit is attributed to Benjamin Schumacher. In 1995, he defined the 'quantum bit' or 'qubit' as a tangible physical resource in the process of providing an analogue to Shannon's noiseless coding theorem.

[2] Note that in this representation, multiple qubits are not just multiple Bloch spheres, but high-dimensional hyperspheres, which are not easy to visualize.

Let us count the degrees of freedom in one qubit. The two complex coefficients, as in Eq. (5), give 4 real variables, which subtracted by the normalization condition, making it in total 3 independent real variables. Alternatively, we count the number of real variables in Eq. (7), which is also 3.

An example for a qubit state is,

$$\frac{1+i}{2}|0\rangle + \frac{i}{\sqrt{2}}|1\rangle = \begin{pmatrix} (1+i)/2 \\ i/\sqrt{2} \end{pmatrix} . \tag{8}$$

Two often used states, which we will also see later, are

$$|+\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) , \qquad |-\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) . \tag{9}$$

In summary, the quantum state of a qubit is a vector of unit length in a two-dimensional complex vector space known as state space.

2. *Multiple qubits*

Let us start by asking a naive question. If a single qubit state is represented by a Bloch sphere, is a 2-qubit(or more generally n-qubit) state represented by two(n) Bloch spheres?

At $n = 2$, in correspondence to the four classical two-bits state, 00, 01, 10, and 11, there are four computational basis states, $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. A general two-qubit state is a four-dimensional vector

$$|\psi\rangle = c_{00}|00\rangle + c_{01}|01\rangle\, c_{10}|10\rangle + c_{11}|11\rangle \ , \tag{10}$$

in which cs are complex variables, satisfying the normalization constraint $|c_{00}|^2 + |c_{01}|^2 + |c_{10}|^2 + |c_{11}|^2 = 1$. Similar to the single qubit case, upon measurement, the outcome is a classical bit of 00 with probability $|c_{00}|^2$, or 01 with probability $|c_{01}|^2$, and likewise for the other two. The qubit after the measurement is in the corresponding computational basis state. Interestingly, if we only measure the first qubit, we get 0 with probability $|c_{00}|^2 + |c_{01}|^2$ or 1 with probability $|c_{10}|^2 + |c_{11}|^2$, and the post-measurement state is

$$\frac{c_{i0}|i0\rangle + c_{i1}|i1\rangle}{\sqrt{|c_{i0}|^2 + |c_{i1}|^2}} \tag{11}$$

with the measurement outcome $i = 0, 1$ correspondingly.

An important two-qubit state is the Bell state or EPR pair

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} . \tag{12}$$

It has a special feature that measurement of the second qubit always gives the same result as the measurement of the first qubit.

Let us count the degrees of freedom in the two-qubits system. The four complex coefficients give 8 real variables, which subtracted by the normalization condition, making it in total 7 independent real variables. Apparently, this is larger than twice the number of a single qubit, or of two Bloch spheres, 6. Instead, the two-qubit state is represented by a hypersphere in the 7 dimensional space, which is not as intuitive as the Bloch sphere for visualization, so it is not often used for introduction purpose. The main point here is: where does the extra degree of freedom come from? When we count 6, we did not consider the correlation between the two qubits.

Generalizing to a system of n qubits. The computational basis states are of the form $|q_1 q_2 \dots q_n\rangle$ with $q_i = 0, 1$, so a quantum state of such a system is specified by 2^n amplitudes. The degrees of freedom is therefore $2^n - 1$, in contrast to the that of the classical bits $2n$. In this sense, the amount of information stored in the qubits as compared to the same amount of classical bits is exponential versus linear.

D. Quantum circuits

A classical computer is built from an electrical circuit containing wires and logic gates. In analogy, a quantum computer is built from a quantum circuit containing wires and elementary quantum gates to carry around and manipulate quantum information. The quantum information is the quantum state of a qubit or a collection of qubits.

1. *Quantum wire*

The simplest quantum circuit is just the quantum wire, denoted as

(13)

which does nothing at all. However, it is also the hardest to implement in practice. The reason is that quantum states are often incredibly fragile, as stored in a single photon or a single atom. In designing a quantum computer, there is a tension between maintaining and manipulating the qubits. The system should interact weakly most of the time as quantum wires, but can be caused to interact strongly some of the time as quantum gates.

2. *Single qubit operations*

We have seen that a single qubit is a vector $|\psi\rangle = c_1 |0\rangle + c_2 |1\rangle$ parameterized by two complex numbers satisfying $|c_1|^2 + |c_2|^2 = 1$. Operations on a qubit must preserve this norm, and thus are described by 2×2

unitary matrices. A matrix U being unitary means that $UU^\dagger = I$, with $U^\dagger$ the transpose and complex conjugate of U. An intuitive way to understand unitary matrices is that they preserve the length of their inputs. This *unitarity* constraint is the only constraint on quantum gates. Let us start looking at important single qubit gates.

1. The Pauli matrices

   The quantum NOT gate is a generalization of the classical NOT gate, on the computational basis states, it acts as

   $$\begin{aligned} NOT\,|0\rangle &= |1\rangle\,, \\ NOT\,|1\rangle &= |0\rangle\ . \end{aligned} \tag{14}$$

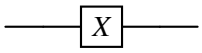
   The matrix form of the NOT gate is the Pauli matrix σ_X, and the NOT gate is also known and denoted as the X gate,

   $$X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{15}$$

   We can see that

   $$X(c_1\,|0\rangle + c_2\,|1\rangle) = c_1\,|1\rangle + |0\rangle\ . \tag{16}$$

   In the quantum circuit representation, the X gate reads

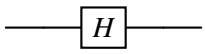
   (17)

   The other two Pauli matrices are

   $$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \qquad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{18}$$

2. The Hadamard gate

   The circuit representation of the Hadamard gate is

   (19)

   and its matrix representation is

   $$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{20}$$

Its action on the computational basis states are

$$\begin{aligned} H|0\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \\ H|1\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}} \ . \end{aligned} \tag{21}$$

3. The phase gate S

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} . \tag{22}$$

4. The T gate (also known as the $\pi/8$ gate)

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/8} \end{pmatrix} . \tag{23}$$

An arbitrary single qubit quantum gate can be decomposed as a product of rotations

$$U = e^{i\alpha} \begin{pmatrix} e^{-i\beta/2} & 0 \\ 0 & e^{i\beta/2} \end{pmatrix} \begin{pmatrix} \cos(\gamma/2) & -\sin(\gamma/2) \\ \sin(\gamma/2) & \cos(\gamma/2) \end{pmatrix} \begin{pmatrix} e^{-i\delta/2} & 0 \\ 0 & e^{i\delta/2} \end{pmatrix} , \tag{24}$$

where α, β, γ, and δ are real variables.

3. Multiple qubit gates

We know that in classical computing, there are important gates such as, AND, OR, XOR(exclusive-OR), NAND and NOR gates. What are their corresponding quantum gates, or are there? Think about unitarity.

1. The controlled-NOT gate

   The prototypical controlled operation is the controlled-NOT (CNOT). The CNOT gate has two input qubits, known as the *control* qubit and the *target* qubit. In the computational basis $|c, t\rangle$ ('c' for control and 't' for target), the action of CNOT is given by

$$|c, t\rangle \rightarrow |c, t \oplus c\rangle \ , \tag{25}$$

   in which $\oplus$ is addition modulo 2. The matrix representation of CNOT is

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \tag{26}$$

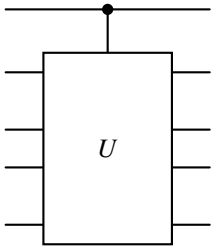
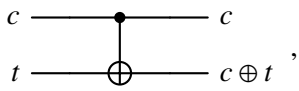


FIG. 2. Controlled-U gate.

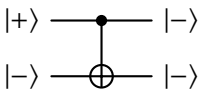
The circuit representation of CNOT is



, (27)

with the top line the *control* qubit, and the bottom the *target* qubit.

Note that the statement that the CNOT leaves the control qubit alone, and modifies the target qubit is true in the computational basis. Can you find single-qubit states $|a\rangle$ and $|b\rangle$ so that applying a CNOT to the combined state $|ab\rangle$ changes the first qubit, i.e., the control qubit? An example is the $|+,-\rangle$ state. Check that



(28)

This circuit is also an illustration of the phase kickback method.

Notably, any multiple qubit logic gate may be composed from CNOT and single qubit gates.

2. The controlled-U gate

We can define a controlled-U gate as a generalization of the CNOT gate, for any quantum gate U acting on n qubits. As illustrated in Fig. 2, such a gate has a single control qubit, indicated by the line with the black dot, and n target qubits, indicated by the boxed U. If the control qubit is set to 0 then nothing happens to the target qubits. If the control qubit is set to 1 then the gate U is applied to the target qubits.

3. Toffoli gate

Any classical circuit can be replaced by an equivalent circuit containing only reversible elements, by making use of a reversible gate known as the *Toffoli* gate, as shown in Fig. 3. Two of the bits are control bits that are unaffected by the action of the Toffoli gate. The third bit is a target bit that is flipped if both control bits are set to 1, and otherwise is left alone.

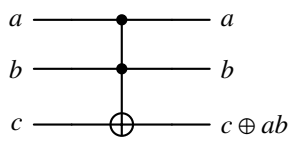


FIG. 3. The circuit representation of Toffoli gate. Here, ab means the product of a and b.

4. *Measurement*

Given a quantum state $|\psi\rangle = c_1|0\rangle + c_2|1\rangle$, is there a way to figure out the value of c_1 and c_2 through experiment? The answer is no. Recall the measurement outcome and the post-measurement states in the previous discussions.

The quantum circuit symbol for measurement is shown in Fig. 4. The double line coming out of the measurement carry classical bit.

FIG. 4. Quantum circuit symbol for measurement.

5. *Illustrative example I: The Bell states*

The Bell states or the EPR states are four states,

$$\begin{aligned} |\beta_{00}\rangle &= \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \\ |\beta_{01}\rangle &= \frac{|01\rangle + |10\rangle}{\sqrt{2}}, \\ |\beta_{10}\rangle &= \frac{|00\rangle - |11\rangle}{\sqrt{2}}, \\ |\beta_{11}\rangle &= \frac{|01\rangle - |10\rangle}{\sqrt{2}}, \end{aligned} \tag{29}$$

and we have already seen the first one in Eq. (12). They can be created by the quantum circuit in Fig. 5 with input states the computational basis states.

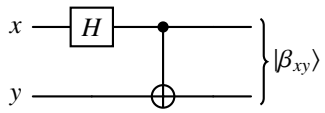


FIG. 5. Quantum circuit to create Bell states.

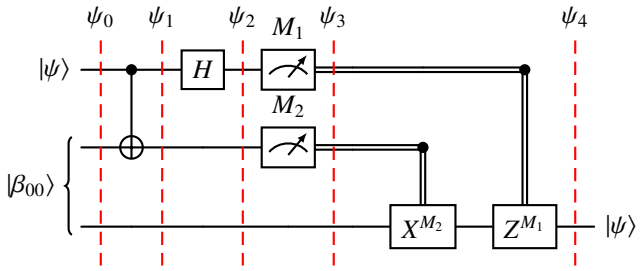


FIG. 6. Quantum circuit for teleporting a qubit.

6. *Illustrative example II: quantum teleportation*

Alice and Bob met long ago but now live far apart. While together they generated an EPR pair, each taking one qubit of the EPR pair when they separated. Many years later, Alice tries to deliver a qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ to Bob. She does not know the state of the qubit, and moreover can only send classical information to Bob. The circuit to carry out this mission is illustrated in Fig. 6. The two top lines represent Alice's system, while the bottom line is Bob's system. The meters represent measurement with outcomes M_1 and M_2, and the double lines coming out of them carry classical bits. Let us write down the states in the circuit step by step. Initially,

$$|\psi_0\rangle = |\psi\rangle|\beta_{00}\rangle = \frac{1}{\sqrt{2}}\left[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|00\rangle + |11\rangle)\right] . \tag{30}$$

Alice applies a CNOT gate to her qubit,

$$|\psi_1\rangle = \frac{1}{\sqrt{2}}\left[\alpha|0\rangle(|00\rangle + |11\rangle) + \beta|1\rangle(|10\rangle + |01\rangle)\right] . \tag{31}$$

Then a Hadamard gate,

$$\begin{aligned} |\psi_2\rangle =& \frac{1}{2}\left[|00\rangle(\alpha|0\rangle + \beta|1\rangle) + |01\rangle(\alpha|1\rangle + \beta|0\rangle) + |10\rangle(\alpha|0\rangle - \beta|1\rangle) + |11\rangle(\alpha|1\rangle - \beta|0\rangle)\right] \\ =& \frac{1}{2}\left[|00\rangle|\psi\rangle + |01\rangle X|\psi\rangle + |10\rangle Z|\psi\rangle + |11\rangle XZ|\psi\rangle\right] . \end{aligned} \tag{32}$$

Now, by measuring the first two qubits, the state of the third qubit is determined. Alice sends the measurement outcome M_1M_2 to Bob, and Bob applies $Z^{M_1}X^{M_2}$ to his qubit accordingly, recovering $|\psi\rangle$. Because the process requires Alice sending classical messages to Bob, we see that quantum teleportation does not enable faster than light communication.

E. The density operator

Suppose a quantum system is in one of a number of states $|\psi_i\rangle$ with respective probabilities p_i, where i is an index. We call $\{p_i, |\psi_i\rangle\}$ an ensemble of *pure* states. The *density operator*, also known as the *density*

matrix, for the system is defined as

$$\rho \equiv \sum_i p_i \left|\psi_i\right\rangle \left\langle\psi_i\right| \ . \tag{33}$$

If the state of the system is known exactly, i.e., an ensemble of $\{1, |\psi\rangle\}$, we say the system is in a *pure* state, and $\rho = |\psi\rangle\langle\psi|$. Otherwise, the system is a mixture of different pure states $|\psi_i\rangle$, and we say it is in a *mixed* state. **[Exercise]** Show that $\mathrm{Tr}(\rho^2) \leq 1$, with equality if and only if ρ is a pure state.

1. *Operations with the density operator*

If the evolution of the system is given by the unitary operator U,

$$\left|\psi_i\right\rangle \xrightarrow{U} U\left|\psi_i\right\rangle \ , \tag{34}$$

that of the density operator follows as

$$\rho = \sum_i p_i \left|\psi_i\right\rangle \left\langle\psi_i\right| \xrightarrow{U} \sum_i p_i U\left|\psi_i\right\rangle \left\langle\psi_i\right| U^\dagger = U\rho U^\dagger \ . \tag{35}$$

For a measurement with operators M_m, the probability of getting result m given the initial state $|\psi_i\rangle$ is

$$p(m|i) = \left\langle\psi_i\right| M_m^\dagger M_m \left|\psi_i\right\rangle = \mathrm{Tr}(M_m^\dagger M_m \left|\psi_i\right\rangle \left\langle\psi_i\right|) \ . \tag{36}$$

The total probability of getting m by measuring the system is then

$$p(m) = \sum_i p_i p(m|i) = \sum_i p_i \,\mathrm{Tr}(M_m^\dagger M_m \left|\psi_i\right\rangle \left\langle\psi_i\right|) = \mathrm{Tr}(M_m^\dagger M_m \rho) \ . \tag{37}$$

After the measurement, the state that is initially $|\psi_i\rangle$ with measurement result m becomes

$$\left|\psi_i^m\right\rangle = \frac{M_m\left|\psi_i\right\rangle}{\sqrt{\left\langle\psi_i\right| M_m^\dagger M_m \left|\psi_i\right\rangle}} \ , \tag{38}$$

thus the subsystem becomes an ensemble of $\{p(i|m), |\psi_i^m\rangle\}$. Here, $p(i|m) = p(i,m)/p(m) = p(m|i)p_i/p(m)$. The density operator is therefore

$$\begin{aligned} \rho_m &= \sum_i p(i|m) \left|\psi_i^m\right\rangle \left\langle\psi_i^m\right| \\ &= \sum_i \frac{p(m|i)p_i}{p(m)} \frac{M_m\left|\psi_i\right\rangle \left\langle\psi_i\right| M_m^\dagger}{\left\langle\psi_i\right| M_m^\dagger M_m \left|\psi_i\right\rangle} \\ &= \sum_i p_i \frac{M_m\left|\psi_i\right\rangle \left\langle\psi_i\right| M_m^\dagger}{\mathrm{Tr}(M_m^\dagger M_m \rho)} \\ &= \frac{M_m \rho M_m^\dagger}{\mathrm{Tr}(M_m^\dagger M_m \rho)} \ . \end{aligned} \tag{39}$$

Consider a quantum system that is prepared in the state ρ_i with probability p_i. We can write ρ_i as the density matrix of some ensemble $\{p_j^{(i)}, |\psi_j^{(i)}\rangle\}$ for a fixed i,

$$\rho_i = \sum_j p_j^{(i)} |\psi_j^{(i)}\rangle \langle\psi_j^{(i)}| \ . \tag{40}$$

Then the full ensemble is $\{p_i p_j^{(i)}, |\psi_j^{(i)}\rangle\}$, so the density matrix is

$$\rho = \sum_{i,j} p_i p_j^{(i)} |\psi_j^{(i)}\rangle \langle\psi_j^{(i)}| = \sum_i p_i \rho_i \ . \tag{41}$$

With this relation, we can write the density matrix of the measured system, using Eqs. (37) and (39),

$$\rho = \sum_m p(m)\rho_m = \sum_m M_m \rho M_m \ . \tag{42}$$

2. *Properties of the density operator*

The class of operators that are density operators are characterized by the following theorem of *characterization of density operators*: An operator ρ is the density operator associated to some ensemble $\{p_i, |\psi_i\rangle\}$ if and only if it satisfies the conditions:

(1) Trace condition: ρ has trace equal to one.

(2) Positivity condition: ρ is a positive operator.

We have seen that in Sec. I B, quantum mechanics as a mathematical framework in the language of state vectors. It is effectively equivalent to use the language of density operators.

- Postulate 1: Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the state space of the system. The system is completely described by its density operator, which is a positive operator ρ with trace one, acting on the state space of the system. If a quantum system is in the state ρ_i with probability p_i, then the density operator for the system is $\sum_i p_i \rho_i$.

- Postulate 2: The evolution of a closed quantum system is described by a unitary transformation. That is, the state ρ of the system at time t_1 is related to the state ρ' of the system at time t_1 by a unitary operator U which depends only on the times t_1 and t_2,

$$\rho' = U\rho U^\dagger \ . \tag{43}$$

- Postulate 3: Quantum measurements are described by a collection $\{M_m\}$ of measurement operators. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is ρ immediately before the measurement then the probability that result m occurs is given by

$$p(m) = \mathrm{Tr}(M_m^\dagger M_m \rho) \ , \tag{44}$$

and the state of the system after the measurement is

$$\frac{M_m \rho M_m^\dagger}{\sqrt{M_m^\dagger M_m \rho}} \ . \tag{45}$$

The measurement operators satisfy the completeness equation,

$$\sum_m M_m^\dagger M_m = 1 \ . \tag{46}$$

- Postulate 4: The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through n, and system number i is prepared in the state ρ_i, then the joint state of the total system is $\rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_n$.

Both the language of the state vector and that of the density operator can describe the framework of quantum mechanics. An advantage of the latter is that it can deal with both the pure and the mixed states.

II. TIME EVOLUTION

[Comment] The material of this section is mainly based on the book of quantum computing by Nielsen and Chuang [1], Chapter 4, and the book of quantum mechanics by J. J. Sakurai [3], Chapter 5.

One of the most important practical applications of computation is the simulation of physical systems. It allows access to the real-time dynamics of a target quantum system by simulating them in another controllable quantum system.

A. Time evolution in quantum theory

We have seen that in the framework of quantum mechanics, the evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi(t_0)\rangle$ of the system at time t_0 is related to the state $|\psi(t)\rangle$ of the system at a later time t by a unitary operator U which depends only on the times t_0 and t,

$$|\psi(t)\rangle = U(t; t_0)\,|\psi(t_0)\rangle \ . \tag{47}$$

This statement is actually another way of saying that the evolution of a quantum state obeys the Shrödinger equation,

$$H\,|\psi(t)\rangle = i\hbar\frac{\partial}{\partial t}\,|\psi(t)\rangle\ , \tag{48}$$

with H the Hamiltonian. The solution is given in Eq. (47) providing

$$U(t;t_0) = e^{-iH(t-t_0)/\hbar}\ . \tag{49}$$

We can prove it by expanding the exponential as

$$\begin{aligned}\exp\left[\frac{-iH(t-t_0)}{\hbar}\right] &= \sum_{n=0}^{\infty}\frac{1}{n!}\left[\frac{-iH(t-t_0)}{\hbar}\right]^n \\ &= 1 + \frac{-iH(t-t_0)}{\hbar} + \frac{1}{2}\left[\frac{-iH(t-t_0)}{\hbar}\right]^2 + \cdots\ .\end{aligned} \tag{50}$$

Then we can apply the time derivative to each term,

$$\begin{aligned}\frac{\partial}{\partial t}\exp\left[\frac{-iH(t-t_0)}{\hbar}\right] &= \sum_{n=0}^{\infty}\frac{1}{n!}\frac{\partial}{\partial t}\left[\frac{-iH(t-t_0)}{\hbar}\right]^n = \frac{-iH}{\hbar} + \left[\frac{-iH}{\hbar}\right]^2(t-t_0) + \cdots \\ &= \sum_{n=1}^{\infty}\frac{1}{n!}\left[\frac{-iH}{\hbar}\right]^n n(t-t_0)^{n-1} \\ &= \frac{-iH}{\hbar}\sum_{n=1}^{\infty}\frac{1}{(n-1)!}\left[\frac{-iH}{\hbar}\right]^{n-1}(t-t_0)^{n-1} \\ &= \frac{-iH}{\hbar}\exp\left[\frac{-iH(t-t_0)}{\hbar}\right]\ ,\end{aligned} \tag{51}$$

which is

$$H\,U(t;t_0) = i\hbar\frac{\partial}{\partial t}U(t;t_0)\ . \tag{52}$$

Applying both side to the initial state $|\psi(t_0)\rangle$, we recover Eq. (48). An alternative way to obtain Eq. (49) is to compound successively infinitesimal time-evolution operators:

$$\exp\left[\frac{-iH(t-t_0)}{\hbar}\right] = \lim_{N\to\infty}\left[1 - \frac{-iH(t-t_0)}{\hbar N}\right]^N\ . \tag{53}$$

Then we can apply the Leibniz product rule to calculate the time derivative,

$$\begin{aligned}\frac{\partial}{\partial t}\exp\left[\frac{-iH(t-t_0)}{\hbar}\right] &= \lim_{N\to\infty}\sum_{k=1}^{N}\frac{-iH}{\hbar N}\left[1 - \frac{-iH(t-t_0)}{\hbar N}\right]^{N-1} \\ &= \lim_{N\to\infty}\frac{-iH}{\hbar}\left[1 - \frac{-iH(t-t_0)}{\hbar N}\right]^{N-1} \\ &= \frac{-iH}{\hbar}\exp\left[\frac{-iH(t-t_0)}{\hbar}\right]\ .\end{aligned} \tag{54}$$

Note that to get to the last step, we use the limit $N \to \infty$ such that $N - 1$ is effectively N.

What if the Hamiltonian is time dependent, say $H(t)$? The time-dependent Schrödinger equation reads

$$H(t)\,|\psi(t)\rangle = i\hbar \frac{\partial}{\partial t}\,|\psi(t)\rangle \ . \tag{55}$$

To address the problem, let us ask a further question, do the Hamiltonian's at different times commute?

Suppose $H(t)$'s commute at different times, i.e., $[H(t_1), H(t_2)] = 0$ for $t_1 \neq t_2$. The solution to Eq. (55) is

$$U(t; t_0) = \exp\left[-\frac{i}{\hbar}\int_{t_0}^{t} \mathrm{d}t' H(t')\right] . \tag{56}$$

We can make the proof by replacing $H(t - t_0)$ by $\int_{t_0}^{t} \mathrm{d}t' H(t')$ in Eq. (50) and onward, with using

$$\frac{\partial}{\partial t}\int_{t_0}^{t} \mathrm{d}t' H(t') = H(t) \ . \tag{57}$$

Suppose $H(t)$'s do not commute at different times, i.e., $[H(t_1), H(t_2)] \neq 0$ for $t_1 \neq t_2$. The solution to Eq. (55) is given by the Dyson series,

$$\begin{aligned} U(t; t_0) =& I + \sum_{n=1}^{\infty}\left(\frac{-i}{\hbar}\right)^n \int_{t_0}^{t} \mathrm{d}t_1 \int_{t_0}^{t_1} \mathrm{d}t_2 \cdots \int_{t_0}^{t_{n-1}} \mathrm{d}t_n H(t_1)H(t_2)\cdots H(t_n) \\ \equiv& \mathcal{T} e^{-i\int_{t_0}^{t} \mathrm{d}\tau H(\tau)/\hbar} \ . \end{aligned} \tag{58}$$

Here we have $t_1 > t_2 > \cdots > t_n$, so we can say that the H's are time-ordered, and we use $\mathcal{T}$, the time-ordering operator, in writing the expression. We can prove that it is the solution of Eq. (55) by checking order by order on both sides of the equation. The way to obtain the solution is through an iterative procedure. To start with, we integrate over both sides of the evolution equation,

$$\begin{aligned} & H(t)U(t; t_0) = i\hbar \frac{\partial}{\partial t} U(t; t_0) \ , \\ \to\ & \mathrm{d}t H(t)U(t; t_0) = i\hbar\, \mathrm{d}U(t; t_0) \ , \\ \to\ & \int_{t_0}^{t} \mathrm{d}t' H(t')U(t'; t_0) = i\hbar \int_{t_0}^{t} \mathrm{d}U(t'; t_0) \ , \\ \to\ & \frac{-i}{\hbar}\int_{t_0}^{t} \mathrm{d}t' H(t')U(t'; t_0) = U(t; t_0) - U(t_0; t_0) \ . \end{aligned} \tag{59}$$

With the initial condition $U(t_0; t_0) = I$, we have

$$U(t; t_0) = 1 - \frac{i}{\hbar}\int_{t_0}^{t} \mathrm{d}t' H(t')U(t'; t_0) \ . \tag{60}$$

We can then plug Eq. (60) into the $U(t'; t_0)$ part of itself, and iteratively for the integrand U's, ultimately arriving at Eq. (58).

The evolution operator has the following properties:

- Unitarity: $U(t;t_0)^\dagger U(t;t_0) = I$
- Identity and normalization: $U(t;t) = I$
- Composition: $U(t;t_0) = U(t;t_1)U(t_1;t_0)$
- Time Reversal: $U(t;t_0)^{-1} = U(t_0;t)$

B. Simulation algorithm

Solving the time evolution problem is thus solving an ordinary differential equation, or equivalently carrying out matrix exponentiation. To reduce the complexity of simulation, one usually breaks down the evolution operator into pieces that are easier and more efficient to perform numerically. The reduction is done in two levels as the following.

1. Decomposing the time duration

   Using the composition property, we can write the evolution operator as a sequential product of many small timesteps,

   $$U(t;t_0) = U(t_n;t_{n-1})\dots U(t_2;t_1)U(t_1;t_0)\ , \tag{61}$$

   in which $t_k = k\delta t$ ($k = 0, 1, \dots, n$), $t_n = t$, and $\delta t = (t-t_0)/n$ is the duration of each timestep. Note that Eq. (61) is exact, including in the cases with time-dependent Hamiltonian. With this decomposition, we can perform the calculation in a step-by-step fashion. Then within each timestep, we can make proper approximations on $U(t_k;t_{k-1})$. The Hamiltonian within each timestep can be treated as time-constant,

   $$U(t_k;t_{k-1}) = \mathcal{T}\, e^{-i\int_{t_{k-1}}^{t_k} \mathrm{d}\tau H(\tau)/\hbar} \approx e^{-iH(t_k)\delta t/\hbar}\ . \tag{62}$$

   The approximation can be considered reasonable when δt is sufficiently small, or more strictly speaking, when $H\delta t/\hbar$ is small. The calculation can be done on a classical computer, and one primary group of numerical methods is the finite difference method (FDM). FDM approximates the derivatives with finite differences in each small time step. Note that though we are writing the evolution operator as an exponential of the Hamiltonian matrix, it is equivalent to solving the ordinary differential equation. Typical methods of the group include the Euler method, and Runge-Kutta (RK) methods. For example, with the most straightforward method, the forward Euler, one would treat the evolution as

   $$U(t_k;t_{k-1}) \approx 1 - iH(t_k)\delta t/\hbar\ . \tag{63}$$

This method is also known for its in-stability. The operation is no longer unitary, and the norm of the state changes over time. Higher-order and more balanced FDMs, such as the 4th order RK, can provide relatively stable region for calculation on classical computers. However, the current quantum computing implementation requires strict unitarity, and FDMs do not apply directly.

Alternatively to FDMs, one can treat the unitary operator U by exponentiating the Hamiltonian matrix, though the computation is expensive in general especially for a large basis space.

2. Decomposing the Hamiltonian

One way to reduce the computational complexity of matrix exponentiation is to split the matrix into pieces, and each individual piece can be exponentiated in a more feasible way. This approach is based on the *Trotter formula*. Suppose we write the Hamiltonian as $H = A + B$, in which A and B are also Hermitian operators. Then for a any real t (you should take it as δt in the context of our discussion here), we have

$$e^{i(A+B)t} = \lim_{n\to\infty} \left(e^{iAt/n} e^{iBt/n}\right)^n \ . \tag{64}$$

I am dropping the constant $\hbar$ for simplicity, which can be absorbed into the Hamiltonian. Let us prove the formula. We can write down the product unit on the left-hand side in the series form, as what we have done in Eq. (50), thus getting

$$\begin{aligned} e^{iAt/n} e^{iBt/n} =& \left[I + \frac{iAt}{n} + O\left(\frac{t^2}{n^2}\right)\right]\left[I + \frac{iBt}{n} + O\left(\frac{t^2}{n^2}\right)\right] \\ =& I + \frac{i}{n}(A+B)t + O\left(\frac{t^2}{n^2}\right) \ . \end{aligned} \tag{65}$$

For the purpose of proving the Trotter formula, one cares about the scaling in the power n, not the constant t, so the t^2 in red can simply be taken as 1 in the above expression. For reasons we will see later, I am also keeping track of the scaling on t throughout the derivation. Taking products of these gives

$$\begin{aligned} \left(e^{iAt/n} e^{iBt/n}\right)^n =& \left[I + \frac{i}{n}(A+B)t + O\left(\frac{t^2}{n^2}\right)\right]^n \\ =& \left[I + \frac{i}{n}(A+B)t\right]^n + O\left(\frac{t^2}{n}\right) \\ =& \sum_{k=0}^{n} \binom{n}{k} I^{n-k} \left[\frac{i}{n}(A+B)t\right]^k + O\left(\frac{t^2}{n}\right) \\ =& I + \sum_{k=1}^{n} \binom{n}{k} \frac{1}{n^k} \left[i(A+B)t\right]^k + O\left(\frac{t^2}{n}\right) \ . \end{aligned} \tag{66}$$

In the second line, the asymptotic behavior on n changes since we are dropping out terms including

$$\binom{n}{1} I^{n-1} O\left(\frac{t^2}{n^2}\right) = O\left(\frac{t^2}{n}\right) . \tag{67}$$

For the n-dependent part,

$$\begin{aligned}
\lim_{n\to\infty} \binom{n}{k} \frac{1}{n^k} &= \lim_{n\to\infty} \frac{n!}{k!(n-k)!} \frac{1}{n^k} \\
&= \lim_{n\to\infty} \frac{n(n-1)(n-2)\dots(n-k+1)}{k! n^k} \\
&= \lim_{n\to\infty} \frac{1}{k!} \left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\dots\left(1-\frac{k-1}{n}\right) \\
&= \lim_{n\to\infty} \frac{1}{k!} \left[1 + O\left(\frac{1}{n}\right)\right] .
\end{aligned} \tag{68}$$

Back to Eq. (66), we have

$$\begin{aligned}
\lim_{n\to\infty} \left(e^{iAt/n} e^{iBt/n}\right)^n =& I + \lim_{n\to\infty} \sum_{k=1}^{n} [i(A+B)t]^k \frac{1}{k!} \left[1 + O\left(\frac{1}{n}\right)\right] + O\left(\frac{t^2}{n}\right) \\
=& \lim_{n\to\infty} \sum_{k=0}^{n} [i(A+B)t]^k \frac{1}{k!} \left[1 + O\left(\frac{1}{n}\right)\right] + O\left(\frac{t^2}{n}\right) \\
=& \lim_{n\to\infty} e^{i(A+B)t} \left[1 + O\left(\frac{1}{n}\right)\right] + O\left(\frac{t^2}{n}\right) \\
=& e^{i(A+B)t} .
\end{aligned} \tag{69}$$

In getting the last line, we apply the $n \to \infty$ limit, thus proving the Trotter formula.

As aforementioned, in the context of our evolution problem, the t for us is actually the small timestep duration δt. In order to quantity the accuracy of the approximation, we would like to know the asymptotic in δt at a certain but not necessarily large n. Let us get back to the last step of Eq. (69), taking $n = 1$ instead of $n \to \infty$, we get

$$e^{i(A+B)\delta t} = e^{iA\delta t} e^{iB\delta t} + O\left(\delta t^2\right) . \tag{70}$$

In terms of our problem, this means that when exponentiation of the full Hamiltonian matrix (e.g., $A + B$) is difficult but that of the individual pieces (e.g., A and B) are easier, we can carry out the latter with an accuracy of $O\left(\delta t^2\right)$. Moreover, one could use higher-order Trotter formula to get higher accuracy, e.g.,

$$e^{i(A+B)\delta t} = e^{iA\delta t/2} e^{iB\delta t} e^{iA\delta t/2} + O\left(\delta t^3\right) . \tag{71}$$

[Exercise] Prove the above equation. One could consider using a similar reasoning of the proof for the Trotter formula above by expanding each exponential into a series of δt.

To summarize, the time evolution simulation algorithm first decompose the whole evolution operator into a sequential product of small timesteps, then trotterizes each timestep into sequential operations. The two steps share the same spirit of decomposing, so both are referred to as *trotterization* in various places. But one should not confuse the two.

The discussed procedure also applies to quantum field theory in the Hamiltonian formalism, which inherited the time-dependent Shrödinger equation. For a concrete example, an interested reader is referred to the simulation of a quark state traversing a time-dependent background field [4–6]. Readers who are interested in further studying different algorithms for solving differential equations on a classical computer are referred to the book Numerical Recipes [7], Chapter 16.

C. The quantum simulation algorithm

Having understand the fundamental theory of the time evolution in quantum theory, let us review the corresponding digital quantum simulation algorithm in its five generic steps with a simple example:

1. **Input**: description of the target quantum system in terms of the system Hamiltonian and underlying Hilbert space. If the original system lives in a infinitely large Hilbert space, an adequate discretized version must be provided. To put it simple, this means determining the Hamiltonian and choose the basis space. For example, consider a single particle living on a line, in a one dimensional potential $V(x)$,

$$\hat{H} = \frac{\hat{p}^2}{2m} + \hat{V}(x) \ . \tag{72}$$

   The "$\hat{O}$" notation means that O is an operator. The system state $|\psi\rangle$ resides in the infinite dimensional Hilbert space. To proceed, we only keep a finite region of interest and discretize it. In this way, the continuous position basis space $\{|x\rangle\}$ is approximated by $\{|k\Delta x\rangle\}$, with range $-d \le x \le d$ and interval $\Delta x = d/N$ (N is a positive integer). The system state is approximated as

$$|\psi\rangle = \int_{-\infty}^{\infty} |x\rangle \langle x|\psi\rangle \, \mathrm{d}x \to |\tilde{\psi}\rangle = \sum_{k=-N}^{N} a_k \, |k\Delta x\rangle \ . \tag{73}$$

   The wavefunction $\langle x|\psi\rangle$ is now described by the state vector a_k.

2. **Encoding**: mapping the degrees of freedom of the problem to qubits on the quantum computer. In the example, we are making a one-to-one correspondence between the basis states $|k\Delta x\rangle$ and the qubit states $|q_0 q_1 \cdots q_{m-1}\rangle$, in which $q_i = 0, 1$ ($i = 0, 1, \ldots, m-1$). The number of qubits m is on the order of $\log_2 N$ with a compact encoding.

3. **Initial state preparation**: preparing the computational initial state $|\psi(t_i)\rangle$ as qubit states on the quantum computer. In the example, we assign the initial state by providing the input state vector $a_k(t_i)$,

$$|\psi(t_i)\rangle = \sum_{k=-N}^{N} a_k(t_i)\,|k\Delta x\rangle \ . \tag{74}$$

4. **Time evolution**: building the series of quantum gates representing the evolution operator $U(t_f; t_i)$. This is the core part of the simulation. Applying the procedure as formulated in the previous section, we can carry out U in a step-by-step fashion, then in each time step, we apply a sequential operations

$$e^{-i\frac{\hat{p}^2}{2m}\delta t} e^{-i\hat{V}(x)\delta t} \ . \tag{75}$$

   The left term with the kinetic energy is diagonal in the momentum space, and is therefore efficient to calculate by exponentiating each diagonal element. Analogously, the right term can be done conveniently in the x space. In order to take advantages of both features, we can perform the calculation in each operation's preferred basis, and use quantum Fourier transform to connect the two.

5. **Measurement**: extracting information from the final quantum state $|\psi(t_f)\rangle$ by performing measurements. With the final state as a qubits state living on the circuit, we can perform measurements multiple times (which means multiple simulations/shots) to get a probability distribution of the state in the basis space, e.g., $f(k\Delta x) = |a_k(t_f)|^2$.

For a more complicated and involving example, an interested reader is referred to the quantum simulation of a quark state traversing a time-dependent background field [8, 9]. While the theoretical framework might seem complex to those not well-versed in quantum chromodynamics or jet physics, the overall structure of the Hamiltonian and the simulation framework share a similar essence with the example provided above.

---

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).

[2] A. Matuschak and M. Nielsen, "Quantum computing for the very curious," https://quantum.country/qcvc.

[3] J. J. Sakurai and J. Napolitano, *Modern Quantum Mechanics*, Quantum physics, quantum information and quantum computation (Cambridge University Press, 2020).

[4] M. Li, X. Zhao, P. Maris, G. Chen, Y. Li, K. Tuchin, and J. P. Vary, Phys. Rev. D **101**, 076016 (2020), arXiv:2002.09757 [nucl-th].

[5] M. Li, T. Lappi, and X. Zhao, Phys. Rev. D **104**, 056014 (2021), arXiv:2107.02225 [hep-ph].

[6] M. Li, T. Lappi, X. Zhao, and C. A. Salgado, Phys. Rev. D **108**, 036016 (2023), arXiv:2305.12490 [hep-ph].

[7] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, USA, 2007).

[8] J. a. Barata, X. Du, M. Li, W. Qian, and C. A. Salgado, Phys. Rev. D **106**, 074013 (2022), arXiv:2208.06750 [hep-ph].

[9] J. a. Barata, X. Du, M. Li, W. Qian, and C. A. Salgado, Phys. Rev. D **108**, 056023 (2023), arXiv:2307.01792 [hep-ph].