

Quantum Information (ELEC-C9440) Lecture 1-2

Matti Raasakka

Aalto University

Spring 2023

Quantum information

Quantum information and computation is the study of information processing tasks that can be accomplished by using quantum systems. Important questions:

- ▶ How much classical information can be transmitted over a quantum channel?
- ▶ How about quantum information instead?
- ▶ Or how might noise affect the channel's capacity?

Why we need physics for studying information? Because information is physical:

- ▶ Information is encoded in physical systems
- ▶ It is processed by physical devices acting on those systems
- ▶ Measurement in general disturbs the system and affects subsequent measurements (uncertainty principle)
- ▶ Erasure of information requires energy (Landauer's principle)
- ▶ Perfect copying of information is forbidden by quantum mechanics (no-cloning theorem)
- ▶ Etc.

Quantum computation

Quantum computers are machines which execute quantum circuits (made up of wires and quantum logic gates) to process information.

Why bother with quantum computation when we already have classical computers?

- ▶ There are problems that can be solved efficiently on a quantum computer but not on a classical computer (Shor's algorithm for integer factorization)
- ▶ If you want to simulate quantum mechanics efficiently, your computer needs to be quantum as well! Important applications in chemistry
- ▶ Some day energy efficiency of quantum computers might be better than traditional computers?

Technology for building quantum computers still has a long way to go but the rate of progress is fast.

In the meantime we can

- ▶ use the small quantum computers that are available
- ▶ use simulators on classical computers to study quantum algorithms

Linear algebra

Lightning fast recap of necessary algebra

Notation

Notation	Description
z^*	Complex conjugate of the complex number z . $(1 + i)^* = 1 - i$
$ \psi\rangle$	Vector. Also known as a <i>ket</i> .
$\langle\psi $	Vector dual to $ \psi\rangle$. Also known as a <i>bra</i> .
$\langle\varphi \psi\rangle$	Inner product between the vectors $ \varphi\rangle$ and $ \psi\rangle$.
$ \varphi\rangle \otimes \psi\rangle$	Tensor product of $ \varphi\rangle$ and $ \psi\rangle$.
$ \varphi\rangle \psi\rangle$	Abbreviated notation for tensor product of $ \varphi\rangle$ and $ \psi\rangle$.
A^*	Complex conjugate of the A matrix.
A^T	Transpose of the A matrix.
A^\dagger	Hermitian conjugate or adjoint of the A matrix, $A^\dagger = (A^T)^*$. $\begin{bmatrix} a & b \\ c & d \end{bmatrix}^\dagger = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}.$
$\langle\varphi A \psi\rangle$	Inner product between $ \varphi\rangle$ and $A \psi\rangle$. Equivalently, inner product between $A^\dagger \varphi\rangle$ and $ \psi\rangle$.

Vectors and operators

Typically we will use the Dirac notation in which vectors are denoted by $|v\rangle$, but sometimes it is useful to write out the components explicitly w.r.t. some basis. For example

$$|v\rangle = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = c_1 |v_1\rangle + c_2 |v_2\rangle , \quad (1)$$

where c_1 and c_2 are the components of $|v\rangle$ in the basis $|v_1\rangle = (1, 0)$, $|v_2\rangle = (0, 1)$. Linear operators act on these vectors as

$$A|v\rangle = A\left(\sum_i c_i |v_i\rangle\right) = \sum_i c_i A|v_i\rangle . \quad (2)$$

A linear operator A can be written as a matrix in some basis on V . Then the action of A on vectors is given by the usual matrix-vector multiplication

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \rightarrow A|v\rangle = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} a_{1,1}c_1 + a_{1,2}c_2 \\ a_{2,1}c_1 + a_{2,2}c_2 \end{pmatrix} . \quad (3)$$

Tensor products

We will use the tensor product to combine smaller vector spaces into larger ones. Suppose we have two vector spaces V and W with orthonormal bases $\{|i\rangle\}$ and $\{|j\rangle\}$, respectively. If the dimension of V is n and the dimension of W is m , then $V \otimes W$ is a vector space of dimension nm . The vectors $|i\rangle \otimes |j\rangle$ is an orthonormal basis in $V \otimes W$. For example, take $n = m = 2$, then the vector

$$|z\rangle = |0\rangle \otimes |0\rangle - |1\rangle \otimes |0\rangle + 2|1\rangle \otimes |1\rangle \quad (4)$$

would be a vector in $V \otimes W$. In the component notation, the tensor product of vectors is (2 dimensional case)

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} a \begin{pmatrix} c \\ d \end{pmatrix} \\ b \begin{pmatrix} c \\ d \end{pmatrix} \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}. \quad (5)$$

The basis vectors would be

$$|0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \quad (6)$$

Tensor products

Tensor products of operators work similarly. Consider the operators $A : V \mapsto V$ and $B : W \mapsto W$. Their tensor product operates

$$(A \otimes B)(|v\rangle \otimes |w\rangle) = (A|v\rangle) \otimes (B|w\rangle) . \quad (7)$$

If A and B have the matrix representations

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} , \quad B = \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} , \quad (8)$$

then the operator $A \otimes B$ could be written

$$A \otimes B = \begin{pmatrix} a_{1,1} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} & a_{1,2} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} \\ a_{2,1} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} & a_{2,2} \begin{pmatrix} b_{1,1} & b_{1,2} \\ b_{2,1} & b_{2,2} \end{pmatrix} \end{pmatrix} \quad (9)$$

$$= \begin{pmatrix} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & a_{1,2}b_{1,1} & a_{1,2}b_{1,2} \\ a_{1,1}b_{2,1} & a_{1,1}b_{2,2} & a_{1,2}b_{2,1} & a_{1,2}b_{2,2} \\ a_{2,1}b_{1,1} & a_{2,1}b_{1,2} & a_{2,2}b_{1,1} & a_{2,2}b_{1,2} \\ a_{2,1}b_{2,1} & a_{2,1}b_{2,2} & a_{2,2}b_{2,1} & a_{2,2}b_{2,2} \end{pmatrix} . \quad (10)$$

Eigenvalues and vectors

A non-zero vector $|\nu\rangle$ is an eigenvector of A if

$$A|\nu\rangle = \lambda|\nu\rangle . \quad (11)$$

$\lambda \in \mathbb{C}$ is the eigenvalue of $|\nu\rangle$. One can solve for the eigenvalues by

$$A|\nu\rangle = \lambda|\nu\rangle \rightarrow (A - \lambda\mathbb{I})|\nu\rangle = 0 \rightarrow \det(A - \lambda\mathbb{I}) = 0 . \quad (12)$$

The rightmost equation is known as the characteristic equation and it can be solved for the eigenvalues λ . Then the corresponding eigenvectors can be found for each eigenvalue λ_i by solving the linear system of equations $A|\nu_i\rangle = \lambda_i|\nu_i\rangle$ for the components of $|\nu_i\rangle$.

Spectral decomposition: Any normal operator ($A^\dagger A = AA^\dagger$) is diagonal w.r.t. some orthogonal basis. Any such operator A can be written

$$A = \sum_i \lambda_i |\nu_i\rangle\langle\nu_i| , \quad (13)$$

where $|\nu_i\rangle$ is the eigenvector of A with eigenvalue λ_i .

Operator functions

Later we will encounter expressions such as $\log A$, where A is a linear operator, so we will need a way to define what it means to apply a function to an operator. In general, for any function on complex numbers

$$f : \mathbb{C} \mapsto \mathbb{C} \quad (14)$$

we can define a corresponding function on normal matrices. The definition uses the spectral decomposition of A :

$$f(A) \equiv \sum_i f(\lambda_i) |v_i\rangle\langle v_i| . \quad (15)$$

So in practice one will first compute the eigenvalues λ_i and eigenvectors $|v_i\rangle$ of a matrix and then apply any complex function on it using the above definition.

Quantum mechanics

Postulates, quantum bits, superdense coding

Postulate 1: States

Postulate (States)

Associated to any isolated physical system is a complex vector space (a Hilbert space \mathcal{H}) known as the state space. The system is completely described by a unit vector $|\psi\rangle \in \mathcal{H}$

The states are normalized to unit vectors: $\langle\psi|\psi\rangle = 1$

We often expand states in some orthogonal basis $\{|i\rangle\}$ s.t. $\langle i|j\rangle = \delta_{ij}$

$$|\psi\rangle = \sum_i a_i |i\rangle \quad (16)$$

Overall phase has no physical significance: $|\psi\rangle$ and $e^{i\varphi} |\psi\rangle$ are the same physical state for every $\varphi \in \mathbb{R}$. Relative phases are however physically significant: the states $a|\psi\rangle + b|\phi\rangle$ and $a|\psi\rangle + e^{i\varphi} b|\phi\rangle$ are physically different.

Postulate 2: Evolution of states

Postulate (Evolution of states)

The evolution of a closed quantum system is described by an unitary transformation. That is, if $|\psi_t\rangle$ and $|\psi_{t'}\rangle$ are the states of a system at two different times, they are related by

$$|\psi_{t'}\rangle = U |\psi_t\rangle , \quad (17)$$

where U is an unitary operator.

In our use cases we can always represent U as a matrix. Remember that a matrix is unitary if $U^\dagger U = \mathbb{I}$, or in other words $U^{-1} = U^\dagger$. Another common way to state this postulate is to give the Schrödinger equation

$$i\hbar \frac{d|\psi\rangle}{dt} = H |\psi\rangle , \quad (18)$$

which gives the evolution of our quantum state in terms of its Hamiltonian H .

Postulate 3: Quantum measurement

Postulate (Quantum measurement)

Any measurement is described by a collection of measurement operators $\{M_m\}$ acting on the state space, one operator for each possible measurement outcome m . The probability of outcome m is

$$p(m) = \langle \psi | M_m^\dagger M_m | \psi \rangle \quad (19)$$

and the state collapses to

$$|\psi\rangle \mapsto \frac{M_m |\psi\rangle}{\sqrt{p(m)}} . \quad (20)$$

These measurement operators satisfy the completeness relation $\sum_m M_m^\dagger M_m = \mathbb{I}$ which guarantees that probabilities sum to one

$$\sum_m p(m) = \sum_m \langle \psi | M_m^\dagger M_m | \psi \rangle = \langle \psi | \left(\sum_m M_m^\dagger M_m \right) | \psi \rangle = 1 . \quad (21)$$

Postulate 3: Quantum measurement

We often talk about measurements in terms of *observables*. Observables represent properties of physical systems which can be measured (e.g. position, momentum, charge, etc). Mathematically, observables are Hermitian/self-adjoint operators ($A^\dagger = A$). By the spectral decomposition of A ,

$$A = \sum_m m |m\rangle\langle m| , \quad (22)$$

where $|m\rangle$ is the eigenstate of A with eigenvalue m . This observable corresponds to the measurement operators $M_m = |m\rangle\langle m|$. Observables make it easy to compute expected values of its corresponding measurement:

$$\mathbb{E}(A) = \sum_m m p(m) = \sum_m m \langle \psi | m \rangle \langle m | \psi \rangle \quad (23)$$

$$= \langle \psi | \left(\sum_m m |m\rangle\langle m| \right) | \psi \rangle = \langle \psi | A | \psi \rangle . \quad (24)$$

Postulate 4: Composite systems

Postulate (Composite systems)

The state space of a composite system is the tensor product of the state spaces of its components.

So, if we have n systems, with the i th system prepared in the state $|\psi_i\rangle$, then the joint system is in the product state $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_n\rangle$.

Accordingly, if we have operators U_i each acting on the i th subsystem, their joint action is described by the matrix

$$U = U_1 \otimes U_2 \otimes \dots \otimes U_n . \quad (25)$$

Note that in the case of states, often one omits the \otimes -sign in order to save ink

$$|\psi_1\rangle \otimes |\psi_2\rangle \otimes \dots \otimes |\psi_n\rangle \equiv |\psi_1\rangle |\psi_2\rangle \dots |\psi_n\rangle . \quad (26)$$

Then, for example,

$$(U_1 \otimes U_2)(|\psi_1\rangle \otimes |\psi_2\rangle) = (U_1 |\psi_1\rangle) \otimes (U_2 |\psi_2\rangle) . \quad (27)$$

Quantum bits

Just as bits are the fundamental building blocks of classical computation and information, quantum bits (*qubits*) are the analogous concept in quantum computation and information. A qubit is a vector in the two-dimensional Hilbert space \mathbb{C}^2 . In terms of the *computational basis states* the state of a general qubit is

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad |\alpha|^2 + |\beta|^2 = 1 \quad (28)$$

$$\langle 0|0\rangle = \langle 1|1\rangle = 1, \quad \langle 0|1\rangle = 0. \quad (29)$$

Whereas a bit has two states (0 and 1), the qubit has infinitely many states since any linear combination of $|0\rangle$ and $|1\rangle$ yields a valid quantum state. One often measures qubits in the computational basis, that is, uses the measurement operators

$$M_0 = |0\rangle\langle 0| \quad M_1 = |1\rangle\langle 1|. \quad (30)$$

According to Postulate 3, measurement would give 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$.

Quantum bits - Bloch sphere

A general qubit state has two complex parameters or four real parameters. Because of the normalization condition $|\alpha|^2 + |\beta|^2 = 1$, only 3 of those parameters are free. Let's then parametrize our state with

$$\alpha = e^{i\gamma} \cos \frac{\theta}{2} \quad \beta = e^{i(\gamma+\varphi)} \sin \frac{\theta}{2} . \quad (31)$$

This parametrization automatically satisfies the normalization condition. Then our qubit is

$$|\psi\rangle = e^{i\gamma} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle \right) . \quad (32)$$

Notice that $e^{i\gamma}$ is a global phase and therefore isn't physically significant. Therefore the physically distinct qubit states can be written in terms of two real parameters, θ and φ :

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle . \quad (33)$$

Quantum bits - Bloch sphere

The *Bloch sphere* is useful for visualizing the state of a qubit. We obtain the Bloch sphere by interpreting the parameters θ and φ as coordinates on a sphere.

$$\theta = 0 \mapsto |0\rangle \quad (34)$$

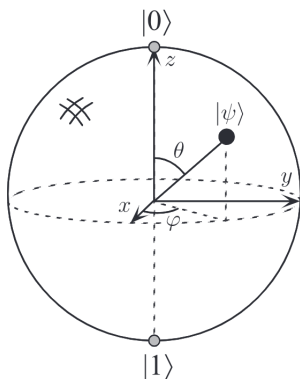
$$\theta = \pi \mapsto |1\rangle \quad (35)$$

$$\theta = \frac{\pi}{2}, \phi = 0 \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad (36)$$

$$\theta = \frac{\pi}{2}, \phi = \pi \mapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (37)$$

$$\theta = \frac{\pi}{2}, \phi = \frac{\pi}{2} \mapsto \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \quad (38)$$

$$\theta = \frac{\pi}{2}, \phi = \frac{3\pi}{2} \mapsto \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \quad (39)$$



We will see that the states along the x , y , and z -axes corresponds to eigenstates of the Pauli X , Y , and Z -matrices.

Multiple qubits

Now consider a system of 2 qubits. By Postulate 4, the state space is now 4 dimensional and is spanned by the computational basis states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. Now a general 2 qubit state can be written

$$|\psi\rangle = \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle , \quad (40)$$

where $|01\rangle = |0\rangle \otimes |1\rangle$ etc. This can easily be generalized for n qubits. The state space would be \mathbb{C}^{2^n} with states

$$|\psi\rangle = \sum_{x=0}^{2^n-1} \alpha_x |x\rangle , \quad (41)$$

where we have used the following useful shorthand for computational basis states: consider a basis state $|x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle$, where each $x_i \in \{0, 1\}$. This is the binary representation of the integer

$$x = x_1 2^{n-1} + x_2 2^{n-2} + \dots + x_n 2^0 . \quad (42)$$

Now if we define

$$|x\rangle \equiv |x_1\rangle \otimes |x_2\rangle \otimes \dots \otimes |x_n\rangle , \quad (43)$$

we can write down multi-qubit basis states very easily. For example, $|6251\rangle$ instead of $|1100001101011\rangle$.

Distinguishability of quantum states

A consequence of Postulate 3 is that quantum states can't necessarily be distinguished from each other with perfect accuracy. This is an important difference between classical and quantum information.

Example: You are given a quantum state $|\psi\rangle$ with the promise that it is either $|\psi_0\rangle = |0\rangle$ or $|\psi_1\rangle = |1\rangle$. Can you determine which it is? Yes you can, because the two possibilities are orthogonal $\langle\psi_0|\psi_1\rangle = 0$. You would make the measurement

$$M_0 = |0\rangle\langle 0| \quad M_1 = |1\rangle\langle 1| \quad (44)$$

and you would get the outcome 0 if and only if the state was $|\psi\rangle = |\psi_0\rangle$.

Example: Consider the same problem with $|\psi_0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $|\psi_1\rangle = |1\rangle$. Now if you make the same measurement you would get

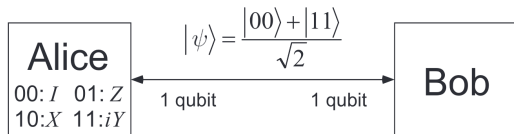
$$\text{If } |\psi\rangle = |\psi_0\rangle : \quad p(0) = \langle\psi_0|0\rangle \langle 0|\psi_0\rangle = \frac{1}{2} \quad p(1) = \langle\psi_0|1\rangle \langle 1|\psi_0\rangle = \frac{1}{2} \quad (45)$$

$$\text{If } |\psi\rangle = |\psi_1\rangle : \quad p(0) = \langle\psi_1|0\rangle \langle 0|\psi_1\rangle = 0 \quad p(1) = \langle\psi_1|1\rangle \langle 1|\psi_1\rangle = 1, \quad (46)$$

therefore if your measurement gave the result 0, then you know you were given ψ_0 . However if the result is 1, then you can't know which state you had.

Example: superdense coding

One bit can be encoded in a qubit, one can always encode $x_1 x_2 \dots x_n \mapsto |x_1 x_2 \dots x_n\rangle$ for n bits encoded in n qubits. Qubits can however store even more information with *superdense coding*.



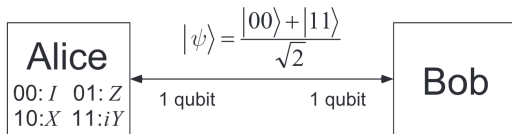
Alice and Bob share a Bell state $|\Phi^+\rangle$. Bell states are

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \quad |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \quad (47)$$

$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \quad |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) . \quad (48)$$

These are all orthogonal and such that Alice can transform the initial state to each one at will.

Example: superdense coding



Suppose Alice wants to send two bits of information x_1x_2 . In superdense coding she would do the following:

1. If $x_1 = 1$, then bit flip $|0\rangle \leftrightarrow |1\rangle$
2. If $x_2 = 1$, then phase flip $|0\rangle \mapsto |0\rangle$ and $|1\rangle \mapsto -|1\rangle$.
3. If $x_1 = x_2 = 1$, then do both: first bit flip and then phase flip
4. Otherwise do nothing

In other words the two-qubit state after Alice's manipulation is

$$x_1x_2 = 00 : |\Phi^+\rangle \quad x_1x_2 = 10 : |\Psi^+\rangle \quad (49)$$

$$x_1x_2 = 01 : |\Phi^-\rangle \quad x_1x_2 = 11 : |\Psi^-\rangle . \quad (50)$$

Then Alice mails her qubit to Bob. Finally Bob can measure both qubits and determine which Bell state he has and therefore which two-bit string Alice wanted to send. Note that distinguishing between the possible states is always possible with perfect fidelity since they are orthogonal,

Quantum computation

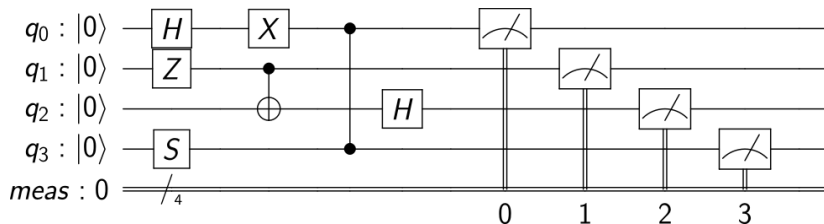
Quantum circuits, quantum gates

Quantum circuits

As stated by Postulate 2, quantum states evolve unitarily. Since we are interested in manipulating qubits, we need a useful way to describe unitary transformations acting on qubits. *Quantum circuits* give a pictorial representation of unitary operators which consist of:

1. Wires representing individual qubits
2. Boxes representing simple unitary operators
3. Symbols representing measurement

For example,



Reversibility, Hilbert space size

Important difference to classical computation is that quantum circuits are *reversible* (if no measurements are made). It means that no information is lost during computation. For example, consider the classical AND-operation which maps two bits to one bit







a	b	a AND b
0	0	0
0	1	0
1	0	0
1	1	1

Only looking at the output bit, one can't deduce what was the input: irreversible and information is lost. A quantum circuit can always be reversed by reading it backwards and replacing gates with their conjugates:

- Read the circuit right-to-left
- Replace each gate $U \mapsto U^\dagger$

This produces a circuit which perfectly undoes the effect of the original circuit. We can simulate only small quantum circuits on classical computers. Why? Because even to keep track of the state of n qubits you need to store the 2^n amplitudes of the computational basis states. Even for $n = 500$ the number of amplitudes is more than the number of atoms in the universe!

Common one-qubit gates

Hadamard		$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Pauli-X		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Phase		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$

All gates are in the computational Z-basis.

Common two-qubit gates

controlled-NOT



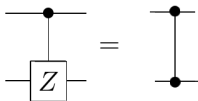
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

swap



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

controlled-Z



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

controlled-phase

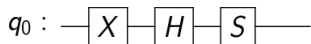


$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}$$

All gates are in the computational Z-basis.

Matrix representation of quantum gates

Quantum circuits are read from left to right. Note that operator expressions work differently: rightmost operator acts first! For example, the circuit

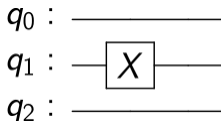


would correspond to the operator

$$U = SHX \quad (51)$$

$$U|q_0\rangle = SHX|q_0\rangle. \quad (52)$$

When we for example “apply X on the 2nd qubit”



we mean that we act on our qubits with:

$$U = \mathbb{I} \otimes X \otimes \mathbb{I} \quad (53)$$

Examples of gate actions

Let's see a few examples of how different gates act on qubits.

$$\alpha |0\rangle + \beta |1\rangle \longrightarrow \boxed{X} \longrightarrow \beta |0\rangle + \alpha |1\rangle$$

$$\alpha |0\rangle + \beta |1\rangle \longrightarrow \boxed{Z} \longrightarrow \alpha |0\rangle - \beta |1\rangle$$

$$\alpha |0\rangle + \beta |1\rangle \longrightarrow \boxed{H} \longrightarrow \alpha \frac{|0\rangle + |1\rangle}{\sqrt{2}} + \beta \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Another useful example is the action of $H^{\otimes n}$ on $|0\rangle^{\otimes n}$:

$$H^{\otimes 3} |0\rangle^{\otimes 3} = \left(\frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \right)^{\otimes 3} \quad (54)$$

$$= \frac{1}{2^{3/2}} (|000\rangle + |001\rangle + |010\rangle + |011\rangle \quad (55)$$

$$+ |100\rangle + |101\rangle + |110\rangle + |111\rangle) . \quad (56)$$

For n qubits, one can use Hadamards to easily create an equal superposition of all computational basis states.

Pauli matrices

The Pauli matrices occur often in quantum algorithms. They are

$$\sigma_0 \equiv \mathbb{I} \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_1 \equiv \sigma_x \equiv X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (57)$$

$$\sigma_2 \equiv \sigma_y \equiv Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 \equiv \sigma_z \equiv Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (58)$$

They are both Hermitian ($U^\dagger = U$), unitary ($U^\dagger U = \mathbb{I}$). All of them (except the identity) are traceless ($\text{tr } \sigma_a = 0$) and have eigenvalues ± 1 . Their eigenvectors are

$$X : |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \quad (59)$$

$$Y : |i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle), \quad |-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle) \quad (60)$$

$$Z : |0\rangle, \quad |1\rangle. \quad (61)$$

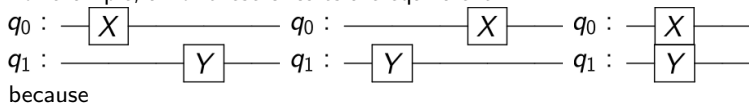
They form a basis in the space of 2×2 Hermitian matrices, so any 2×2 Hermitian matrix can be written

$$H = a_0 \sigma_0 + a_1 \sigma_1 + a_2 \sigma_2 + a_3 \sigma_3, \quad (62)$$

for some coefficients a_i .

Reordering gates, linearity

There is a certain amount of freedom in reordering gates in quantum circuits. For example, all of these circuits are equivalent



$$(X \otimes \mathbb{I})(\mathbb{I} \otimes Y) = X \otimes Y = (\mathbb{I} \otimes Y)(X \otimes \mathbb{I}) . \quad (63)$$

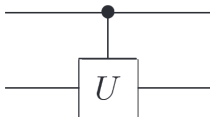
In other words, the X and Y gates commute because they act on different wires/qubits.

Notice that in general we need to only specify the action of a gate on the computational basis states. If this is known, then we automatically know how the gate acts on arbitrary superpositions

$$U \sum_{x=0}^{2^n-1} \alpha_x |x\rangle = \sum_{x=0}^{2^n-1} \alpha_x U |x\rangle . \quad (64)$$

Controlled gates

In traditional computation, flow control structures are vitally important: for example, the IF-statement. The same is true in quantum computation. Controlled operations is how this is accomplished in quantum computing. A general controlled unitary is drawn



The first qubit is the *control qubit* and the second is the *target qubit*. The action of the above circuit is

$$|c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes U^c |t\rangle , \quad (65)$$

that is, the one-qubit unitary U is applied on the target qubit $|t\rangle$ if and only if the control qubit is set $|c\rangle = |1\rangle$.

Controlled gates

One of the most common controlled gates is the controlled-X/controlled-NOT gate.



The traditional notation \oplus refers to the fact that if the control qubit is set, then the second qubit is combined with the first by the XOR-operation

$$|c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes |c \oplus t\rangle . \quad (66)$$

In other words, the target is flipped if the control is set. Another important controlled gate is the controlled-Z:

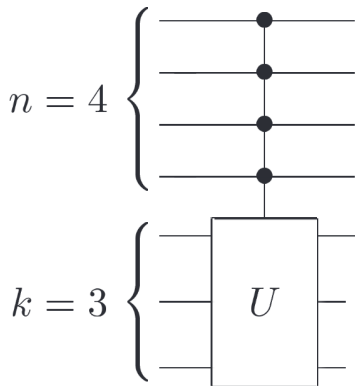


The notation is symmetric because in this case it doesn't matter which wire is the control and which is the target because a minus-sign is produced only when both qubits are set

$$|c\rangle \otimes |t\rangle \mapsto |c\rangle \otimes Z^c |t\rangle = |c\rangle \otimes ((-1)^t)^c |t\rangle = (-1)^{ct} |c\rangle \otimes |t\rangle . \quad (67)$$

Controlled gates

In general, we can have unitaries controlled on any number of qubits:



Now the unitary U is applied on the last 3 qubits only if all of the first 4 qubits are set

$$|x_1 x_2 \dots x_n\rangle |\psi\rangle \mapsto |x_1 x_2 \dots x_n\rangle U^{x_1 x_2 \dots x_n} |\psi\rangle . \quad (68)$$

Universal quantum computation

Arbitrary classical programs can be built from just a few basic gates: e.g. AND, OR, and NOT -gates. One might wonder if quantum computation is similar in this regard: yes it is. Different quantum programs correspond to different unitary operators U , so the appropriate question is how many and what kind of gates we need to build up any unitary transformation on qubits. Like in the classical case, different choices exist but one is

Universal gates: CNOT, H, S, T (69)

Universality means that any unitary operator U can be approximated with arbitrary accuracy by using a finite sequence of universal gates. Now we'll state two important facts about universal circuits without proof:

- ▶ Solovay-Kitaev theorem: Any single-qubit unitary U can be accurately and efficiently approximated with universal gates (for accuracy ϵ , circuit length $\sim \text{poly}(\log(1/\epsilon))$)
- ▶ There exist unitary transformations U on n -qubits which cannot be efficiently approximated with universal gates (circuit length $\sim 4^n$)

Quantum circuit model of computation

Now we can summarize the core elements of quantum computation

1. **State space:** The quantum circuit operates on n qubits which form a 2^n -dimensional complex Hilbert space. The states $|x_1 x_2 \dots x_n\rangle$ where $x_i = \{0, 1\}$ are known as computational basis states.
2. **Ability to prepare computational basis states:** we may assume that any computational basis state can be prepared by the quantum computer efficiently.
3. **Ability to realize universal gates:** universal gates can be applied on any qubits. For example, two-qubit gates such as the CNOT can be applied to any two qubits.
4. **Ability to measure in computational basis:** useful information can be extracted from the quantum computer when it is measured. We may assume the ability of the computer to measure any qubit in the computational basis.

Using Qiskit

What is Qiskit, programming examples

What is Qiskit

Qiskit is an open source Python package for quantum programming. There are many tools for quantum computation but Qiskit is the most popular one.

Qiskit can

- ▶ build quantum circuits
- ▶ simulate quantum computers
- ▶ run programs on real hardware
- ▶ visualize circuits and experiment results
- ▶ model hardware noise
- ▶ etc

We will have some Qiskit-programming exercises on this course. You can use Qiskit in one of the following ways

- ▶ Aalto JupyterHub: <https://jupyter.cs.aalto.fi/> (login with Aalto account)
- ▶ IBM Quantum Lab: <https://quantum-computing.ibm.com/lab> (needs an IBM account)
- ▶ Running locally:
https://qiskit.org/documentation/getting_started.html

Programming examples

Examples in separate Python notebooks.