Introduction to quantum computing

Polytech ET5 IIM

Marc Baboulin
(marc.baboulin@inria.fr)

université
PARIS-SACLAY

Outline

Outline

Outline

Outline

Outline

Outline

Why a course on quantum computing

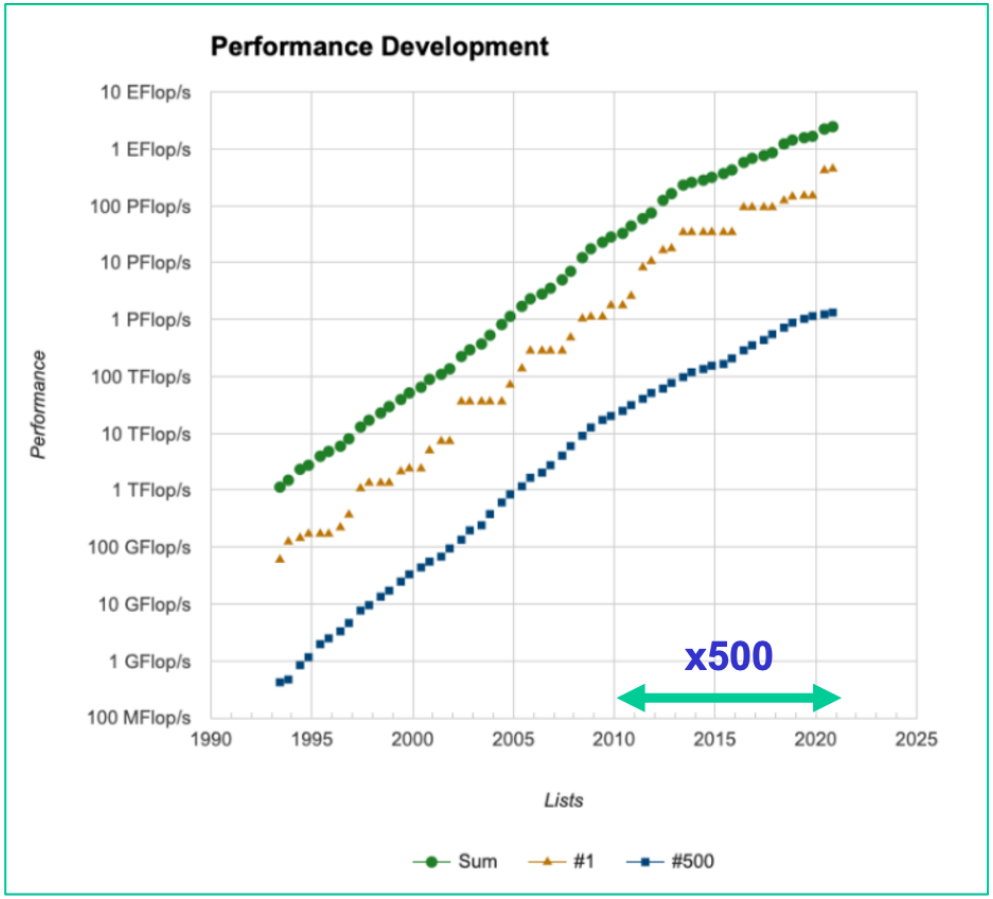
- Fast development of quantum computing since 1994 (Schor).
- Potential impact on critical fields (cryptography, artificial intelligence,...).
- Major players in the computer science industry (Google, IBM, Microsoft, Amazon, Eviden,...) and governments (national quantum initiative in France) invest in quantum computing.
- In France, many companies invest (and hire) in quantum technologies (Total, EDF, Thalès, Airbus, BNP, ...) and the startups in the field have a very strong growth (PasQal, Quandela, Alice & Bob, Welinq,...).

Course organization

- Lecturers: Marc Baboulin (UPSaclay), Océane Koska (Eviden).
- **Contents:**
  1. Presentation of the main concepts for quantum computing (qubits, circuits, algorithms).
  2. Practical work on machines using the simulation library Qiskit (IBM).
- Prerequisite: Python programming, basic notions of linear algebra.
- There is no requirement for knowing quantum physics.

Outline

Computing capabilities in 2024

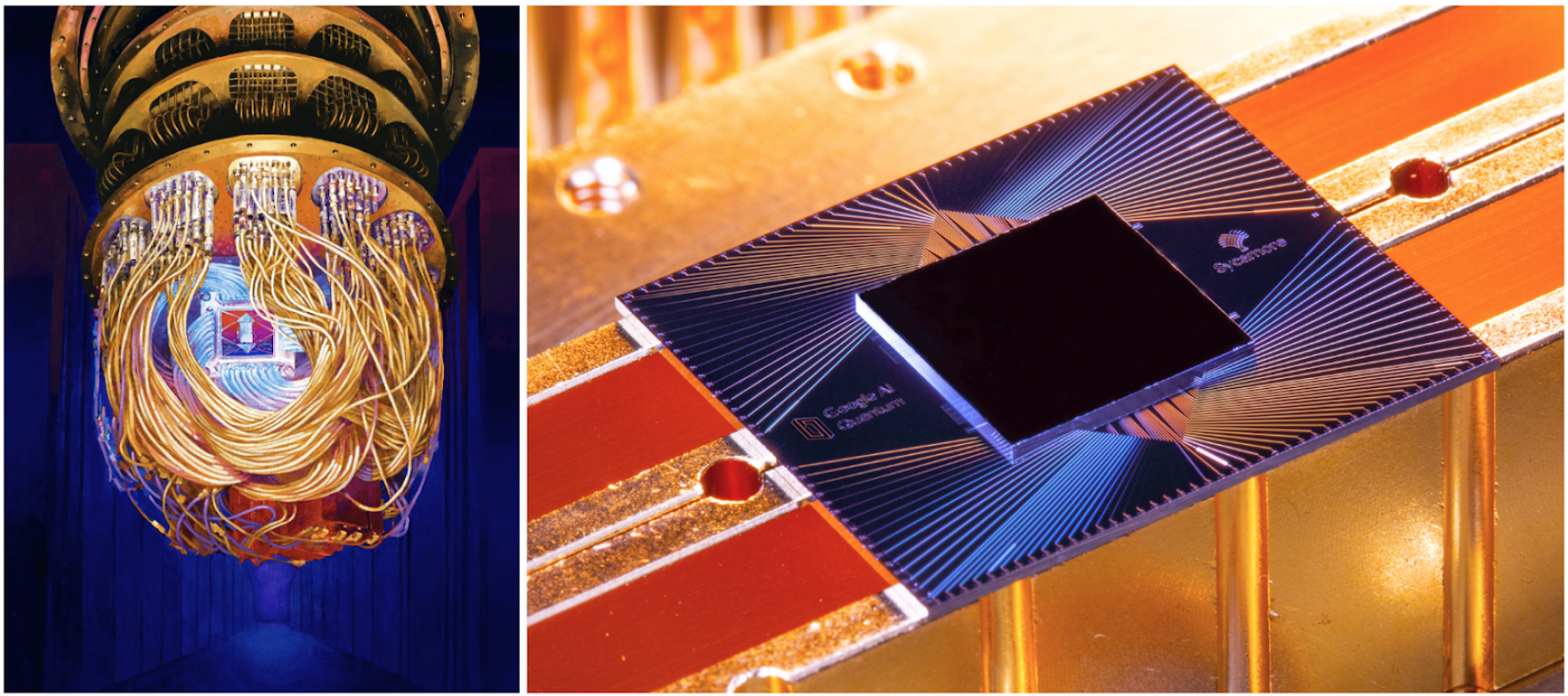


← 60 qubits

- HPC: we are at the time of Exascale (10^{18} op/sec).
- Exascale supercomputer probably beaten with only 60 (not noisy) qubits.

Computing capabilities in 2024

- 105 qubits claimed in Dec. 2024 by Google (superconducting).
- Quantum technologies are much more energy-efficient:
  21 MW (Frontier) vs 10^{-4} MW (Sycamore, 54 qubits).



Quantum processor Sycamore (Google).

Quantum computing

Quantum computing = exploiting specific properties of quantum mechanics (superposition, entanglement) to perform computations.

Some problems can be solved **faster** with a quantum computer (quantum superiority):

- find the prime factors of large integers, [Shor, 1994]
- predict the state of molecules, [Lloyd, 1996]
- search in a database, [Grover, 1996]
- and many others: `http://math.nist.gov/quantum/zoo/`.

Some are beyond the reach of the most powerful existing supercomputers (quantum supremacy).

Targeted fields: quantum chemistry, biology, artificial intelligence (machine learning), cryptography, optimization; scientific computing...

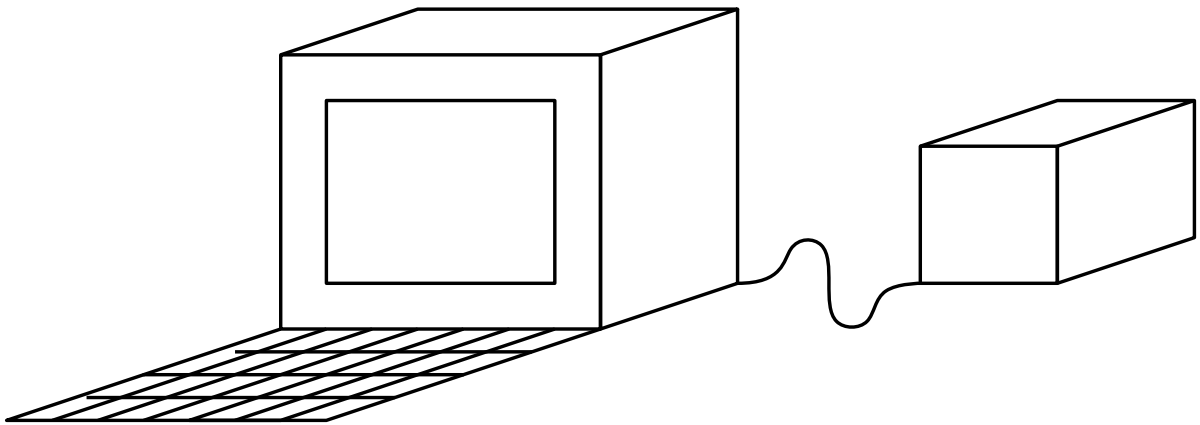
Quantum promises

- **Principle:** a quantum device has a state (made up of qubits instead of bits) which is initialized and then evolves by applying given operations (algorithm).
  At the end, informations about the state are measured.
- **Parallelism:** operations on n qubits are computed on the whole superposition at the same time (2^n coeff. for n qubits) $\rightarrow$ potentially exponential speedup (e.g., in combinatorial optimization, linear systems).
- **But:** return a single probabilistic result and the current quantum computers are still experimental and NISQ (noisy + Intermediate Scale).
- 2 computational models: gate/circuit model (in this course) and quantum annealing model.

Future computers will be hybrid

Classical CPU

Communicates with the quantum coprocessor



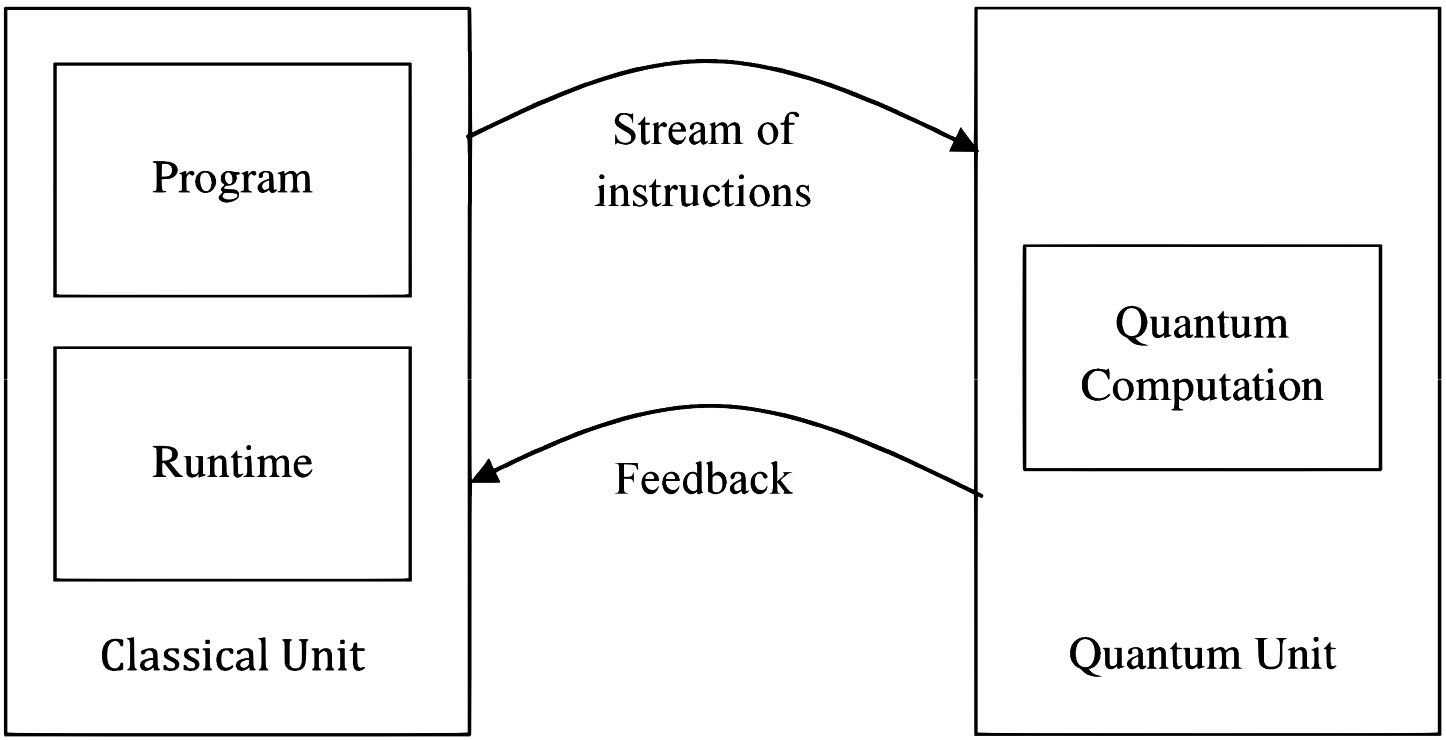
Quantum accelerator

"Specialized" device, affected by noise
(in state evolution or measurements)

Hybrid classical-quantum computer

Quantum / Classical interaction

Typical execution flow:



Outline

Notation and inner product

We work in the vector space $\mathbb{C}^n$. The main difference with classical linear algebra comes from the specific notation out of quantum mechanics. Let $v, w \in \mathbb{C}^n$,

- **Dirac (bra-ket) notation:**
  $|v\rangle$ is a column vector (ket),
  $\langle v|$ is a row vector (bra) with $\langle v| = |v\rangle^*$ (conjugate transpose).
- $\langle v|w\rangle$ denotes the inner product of $|v\rangle$ and $|w\rangle$.

$$\langle v|w\rangle = \begin{pmatrix} \bar{v_1}, \ldots, \bar{v_n} \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_n \end{pmatrix} = \sum_{i=1}^{n} \bar{v_i} w_i.$$

- $\||v\rangle\|_2 = \sqrt{\langle v|v\rangle}$ is the Euclidean norm of $|v\rangle$.

Outer product

Let $v, w \in \mathbb{C}^n$, the outer product of $|v\rangle$ and $|w\rangle$ is :

$$|v\rangle\langle w| = \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} \begin{pmatrix} \bar{w}_1, \ldots, \bar{w}_n \end{pmatrix} = \begin{pmatrix} v_1\bar{w}_1 & \ldots & v_1\bar{w}_n \\ \vdots & & \vdots \\ v_m\bar{w}_1 & \ldots & v_m\bar{w}_n \end{pmatrix}.$$

Properties:

- $(|u\rangle\langle v|)|w\rangle = |u\rangle\langle v|w\rangle = \langle v|w\rangle|u\rangle$
- $\forall |v\rangle,\ |v\rangle = \sum_i v_i |e_i\rangle = \sum_i |e_i\rangle\langle e_i|v\rangle = (\sum_i |e_i\rangle\langle e_i|)|v\rangle$
  $\Rightarrow \sum_i |e_i\rangle\langle e_i| = I$ (completeness relation).
- Outer product representation of a matrix A, with $(e_i)_i$ and $(f_j)_j$ orthonormal bases in input and output spaces, respectively:

$$A = \sum_{ij} \langle f_j| A |e_i\rangle |f_j\rangle\langle e_i|.$$

The $|f_j\rangle\langle e_i|$ matrices have only one nonzero coeff. (equal to 1) at position j, i. They form a basis of the space of matrices.
If same input and output spaces then $A = \sum_{ij} a_{ij} |e_i\rangle\langle e_j|$.

Expression of AB with Dirac notations

If we have the same basis $(e_i)_i$ in the input and output spaces of A and B then:

$$\begin{aligned} AB &= \left(\sum_{i,\ell} a_{i\ell} \left|e_i\right\rangle \left\langle e_\ell\right|\right) \left(\sum_{k,j} b_{kj} \left|e_k\right\rangle \left\langle e_j\right|\right) \\ &= \sum_{i,l,k,j} a_{i\ell} b_{kj} \left|e_i\right\rangle \left\langle e_\ell\right| \left|e_k\right\rangle \left\langle e_j\right| \\ &= \sum_{i,k,j} a_{ik} b_{kj} \left|e_i\right\rangle \left\langle e_j\right| \\ &= \sum_{i,j} \left(\sum_{k} a_{ik} b_{kj}\right) \left|e_i\right\rangle \left\langle e_j\right| . \end{aligned}$$

We recover the usual expression for the matrix-matrix multiply.

Definitions

- A^* denotes the conjugate transpose of A ($A^* = \bar{A}^T$).
  Other notations: $A^\dagger$ or A^H.
- The spectral norm (or 2-norm) of A is:

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \max_{\|x\|_2 = 1} \|Ax\|_2 .$$

- The trace of a square matrix A is defined by:

$$Tr(A) = \sum_{i=1}^{n} a_{ii}.$$

- Exponential of a square matrix:

$$e^A = \sum_{k=0}^{+\infty} \frac{A^k}{k!} = \lim_{k \to \infty} \left(I + \frac{A}{k} \right)^k$$

Matrices properties

$A \in \mathbb{C}^{n \times n}$,

- A is Hermitian (or self-adjoint) if $\forall u, v \in \mathbb{C}^n,\ \langle Au|v\rangle = \langle u|Av\rangle$ or equivalently if $A^* = A$.
  A is a positive operator if $\forall u \in \mathbb{C}^n,\ \langle Au|u\rangle \geq 0$ (definite positive is > 0). A positive operator is necessarily Hermitian.
- A is unitary if $A^*A = AA^* = I$ or equivalently if $A^* = A^{-1}$.

  Unitary matrices preserve the inner product $(\forall u, v \in \mathbb{C}^n,\ (Au)^*(Av) = u^*v)$ and the Euclidean norm $(\forall v \in \mathbb{C}^n,\ \|Au\|_2 = \|u\|_2)$.
- A is normal if $A^*A = AA^*$.

Eigenvalues/eigenvectors

- $\lambda \in \mathbb{C}$ is an eigenvalue of $A \in \mathbb{C}^{n \times n}$ if there exists a nonzero vector $|v\rangle \in \mathbb{C}^n$ such that $A|v\rangle = \lambda |v\rangle$.
  $|v\rangle$ is then called an eigenvector of A.
- The eigenvalues of A are the solution of the characteristic equation $det|A - \lambda I| = 0 \Rightarrow A$ has at least one eigenvalue.
- A is diagonalizable if there exists P invertible and D diagonal such that $A = PDP^{-1}$. The columns of P correspond to eigenvectors of A.
- Spectral decomposition:
  If A diagonalizable, $A = \sum_i \lambda_i |v_i\rangle \langle v_i|$, where the $|v_i\rangle$ form an orthonormal set of eigenvectors of A, with eigenvalues λ_i.
- If A is Hermitian, its eigenvalues are all real and A is diagonalizable.
- If A is unitary, all its eigenvalues have modulus one and thus can be written in the form $e^{i\theta}$, $\theta \in \mathbb{R}$.

Connection between Hermitian and unitary matrices

- If A is Hermitian then $e^{iA} = \sum_{k=0}^{+\infty} \frac{(iA)^k}{k!}$ is a unitary matrix. For instance $e^{-i\hat{H}t/\hbar}$ is unitary, where $\hat{H}$ is the Hamilton operator.
- Eigenvalues of e^{iA} are the exponential of the eigenvalues of A.

Some special matrices

- The 4 Pauli matrices:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

  The Pauli matrices are Hermitian and unitary (cf exercise).

- The Hadamard matrix: $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.

  Will be used to put a vector into the superposition of 2 other vectors (quantum states).

- The general phase shift matrix: $R_\theta = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$.

  Will be used to rotate the vector (quantum state) $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ by the angle (phase) θ.
  Rem: $R_\pi = Z$.

Singular value decomposition (SVD)

Let $A \in \mathbb{C}^{m \times n}$, there exists $U \in \mathbb{R}^{m \times m}$ et $V \in \mathbb{R}^{n \times n}$ unitary matrices such that :

$$A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^* = U_1 \, \Sigma \, V^*, \text{ if } m \geq n \text{ and } U = \begin{pmatrix} U_1 & U_2 \end{pmatrix}$$

$$A = U \begin{pmatrix} \Sigma & 0 \end{pmatrix} V^* = U \, \Sigma \, V_1^*, \text{ if } m \leq n \text{ and } V = \begin{pmatrix} V_1 & V_2 \end{pmatrix}$$

where $\Sigma = diag(\sigma_1, \ldots, \sigma_p)$, $\sigma_1 \geq \cdots \geq \sigma_p \geq 0$, $p = min(m, n)$.

- The σ_i are called the singular values of A .
- The columns of U and V are called the left and right singular vectors, respectively.
- The rank of A corresponds to the number of nonzero singular values.
- If $m = n$, we have $U = U_1$ and the SVD is $A = U \, \Sigma \, V^*$.

Properties of the SVD

- Developed expression:

$$\boxed{A = \sum_{i=1}^{rank(A)} \sigma_i \ |u_i\rangle \ \langle v_i|}$$

- A and A^* have the same singular values.
- If $A = U \ \Sigma \ V^*$ and A invertible then $A^{-1} = V \ \Sigma^{-1} \ U^*$.
- If A is Hermitian then we have $\forall i, \sigma_i = |\lambda_i|$.
- The σ_i^2 are the eigenvalues of A^*A (or AA^*).
- The $|v_i\rangle$ are the eigenvectors of A^*A.
- The $|u_i\rangle$ are the eigenvectors of AA^*.

Tensor product 1/3

The tensor product is at the core of most quantum operations.

- The tensor product enables us to construct larger vector spaces from existing vector spaces.
- Let V and W be two vector spaces over a field $\mathbb{F}$, with bases $e_1, \ldots, e_m$ and $f_1, \ldots, f_n$.
  The tensor product $V \otimes W$ is a vector space over $\mathbb{F}$ of dimension $m \times n$ with basis $(e_i \otimes f_j)_{i,j}$. It defines a bilinear operation.

Tensor product 2/3

- In QC, vector spaces are complex Euclidean spaces with canonical basis and then the tensor product coincides with the Kronecker product defined as follows:
  if $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{p\times q}$ then

$$A \otimes B = \begin{pmatrix} a_{11}B & \dots & a_{1n}B \\ a_{21}B & \dots & a_{2n}B \\ \vdots & & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{pmatrix} \in \mathbb{C}^{mp\times nq}.$$

- Example, if $u = \begin{pmatrix} a \\ b \end{pmatrix}$ and $v = \begin{pmatrix} c \\ d \end{pmatrix}$ then

$$u \otimes v = \begin{pmatrix} a\begin{pmatrix} c \\ d \end{pmatrix} \\ b\begin{pmatrix} c \\ d \end{pmatrix} \end{pmatrix} = \begin{pmatrix} ac \\ ad \\ bc \\ bd \end{pmatrix}.$$

Tensor product 3/3

Properties:

$A, B \in \mathbb{C}^{m\times m}$, $C, D \in \mathbb{C}^{n\times n}$, $u, v \in \mathbb{C}^{m}$, $w, x \in \mathbb{C}^{n}$, $a, b \in \mathbb{C}$, then

- $(A \otimes C)^* = A^* \otimes C^*$
- $(A \otimes C)(B \otimes D) = AB \otimes CD$
- $(A \otimes C)(u \otimes w) = Au \otimes Cw$
- $(av) \otimes (bw) = ab(v \otimes w)$
- $(u + v) \otimes w = u \otimes w + v \otimes w$
- $u \otimes (w + x) = u \otimes w + u \otimes x$
- If A and B are unitary then $A \otimes B$ is unitary.

Notations:

- For a matrix A: $A^{\otimes n} = \underbrace{A \otimes A \otimes \cdots \otimes A}_{n \text{ times}}$
- For a space vector $\mathbb{S}$: $\mathbb{S}^{\otimes n} = \underbrace{\mathbb{S} \otimes \mathbb{S} \otimes \cdots \otimes \mathbb{S}}_{n \text{ times}}$

Outline

Notations

Binary strings:

$s = s_1 \dots s_n \in \{0,1\}^n$ denotes a binary string on n digits.

- s corresponds to the decimal number $\sum_{k=1}^{n} s_k 2^{n-k}$.
  s will be used to index the elements of 2^n-dimensional vectors.
- bitwise XOR (modulo 2 addition): $s \oplus t = v$ with

$$v_i = \begin{cases} 0 \text{ if } s_i = t_i, \\ 1 \text{ otherwise.} \end{cases}$$

- bitwise dot product: $s \bullet t = \sum_{i=1}^{n} s_i t_i \ (\in \mathbb{R})$.

Qubits

A quantum computer has a quantum register that contains qubits.

The state of an n-qubit register is a unit vector of $(\mathbb{C}^2)^{\otimes n}$.

1-qubit state:

- bit $= \{0, 1\} \rightarrow$ qubit $\in \mathbb{C}^2$
- $|\psi\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \alpha |0\rangle + \beta |1\rangle$, with $|\alpha|^2 + |\beta|^2 = 1$.

  $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ = state of the qubit = superposition of two basis states.
- α, β are the amplitudes.

2-qubit state:

- $|\psi\rangle \in \mathbb{C}^4 = (\mathbb{C}^2)^{\otimes 2}$
- $|\psi\rangle = \alpha_{00} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \alpha_{01} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \alpha_{10} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} + \alpha_{11} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$
  $= \alpha_{00} |0\rangle \otimes |0\rangle + \alpha_{01} |0\rangle \otimes |1\rangle + \alpha_{10} |1\rangle \otimes |0\rangle + \alpha_{11} |1\rangle \otimes |1\rangle$
  $= \alpha_{00} |00\rangle + \alpha_{01} |01\rangle + \alpha_{10} |10\rangle + \alpha_{11} |11\rangle \in \mathbb{C}^{2^2}, \||\psi\rangle\| = 1.$
- For the binary string $s \in \{0,1\}^2$, $|s\rangle$ is the basis vector of $\mathbb{C}^4$ with a 1 in position $d+1$ where d is the decimal number corresponding to s.

n-qubit state:

$$|\psi\rangle = \alpha_{00\ldots0} \underbrace{|0\rangle \otimes |0\rangle \otimes \cdots \otimes |0\rangle}_{n \text{ times}} + \alpha_{00\ldots1} \underbrace{|0\rangle \otimes |0\rangle \otimes \cdots \otimes |1\rangle}_{n \text{ times}} + \cdots +$$

$$\alpha_{11\ldots1} \underbrace{|1\rangle \otimes |1\rangle \otimes \cdots \otimes |1\rangle}_{n \text{ times}}.$$

$$= \alpha_{00\ldots0} |\underbrace{00\ldots0}_{n \text{ digits}}\rangle + \alpha_{00\ldots1} |\underbrace{00\ldots1}_{n \text{ digits}}\rangle + \cdots + \alpha_{11\ldots1} |\underbrace{11\ldots1}_{n \text{ digits}}\rangle.$$

The state of n qubits can be expressed as

$$\boxed{|\psi\rangle = \sum_{s\in\{0,1\}^n} \alpha_s |s\rangle,}$$

with $\alpha_s \in \mathbb{C}, s \in \{0,1\}^n$ and $\sum_{s\in\{0,1\}^n} |\alpha_s|^2 = 1$.

$|\psi\rangle$ will be the input of the quantum algorithm.

Qubits

n-qubit state:

$|\psi\rangle \in \mathbb{C}^{2^n} = (\mathbb{C}^2)^{\otimes n} \Rightarrow$ dimension of state space grows exponentially with the number of qubits

In classical computing, state in $\{0,1\}^n$ (n-dimensionnal space).

$\Rightarrow$ much more information in quantum computing.

However we cannot extract more than n bits of information from a n-qubit register (see *measurement*).

Expression of tensor products with Dirac notations

- If $s = s_1 \dots s_n \in \{0,1\}^n$ is a binary string then $|s\rangle = \bigotimes_i |s_i\rangle$ (for instance $|010\rangle = |0\rangle \otimes |1\rangle \otimes |0\rangle$).
- More generally, if $s \in \{0,1\}^m$ and $s' \in \{0,1\}^n$ then $|ss'\rangle = |s\rangle \otimes |s'\rangle$, often denoted as $|s\rangle\,|s'\rangle$, where ss' is the concatenation of s and s'.
- Using the outer product representation of matrices we have

$$\begin{aligned} A \otimes B &= \left(\sum_{s,t} a_{st} |s\rangle \langle t| \right) \otimes \left(\sum_{s',t'} b_{s't'} |s'\rangle \langle t'| \right) \\ &= \sum_{s,t,s',t'} a_{st} b_{s't'} (|s\rangle \langle t|) \otimes (|s'\rangle \langle t'|) \\ &= \sum_{s,t,s',t'} a_{st} b_{s't'} |s\rangle |s'\rangle \langle t| \langle t'| \\ &= \sum_{s,t,s',t'} a_{st} b_{s't'} |ss'\rangle \langle tt'| . \end{aligned}$$

Superposition

Definition

n qubits corresponding to the state $|\psi\rangle = \sum_{s \in \{0,1\}^n} \alpha_s |s\rangle$ *are in a basis state if* $\exists t$ *such that* $|\alpha_t| = 1$ *and* $|\alpha_s| = 0, \ \forall s \neq t$.

Otherwise, these qubits are in superposition.

Proposition

An n-qubit state vector is in a basis state iff it can be expressed as the tensor product of n 1-qubit vectors in basis state.

Remark

Superposition does not exist in classical computing. An n-bit register is always in a basis state (i.e contains a binary string in $\{0,1\}^n$*).*

Entanglement

Definition

A quantum state $|\psi\rangle \in (\mathbb{C}^2)^{\otimes n}$ that cannot be expressed as a product state of 1-qubit states $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$ is said to be entangled.

Example

The 2-qubit state $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ (Bell state or EPR pair) is an entangled state. Measurement of first qubit gives 0 or 1 with probability 1/2 and measurement of second qubit gives the same result.

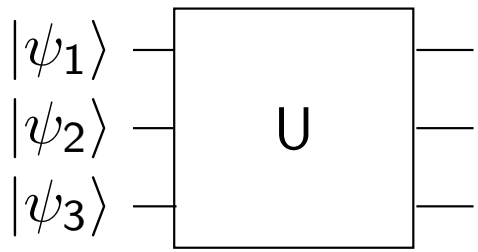
Operations on qubits

Quantum operations are linear, norm-preserving and reversible.

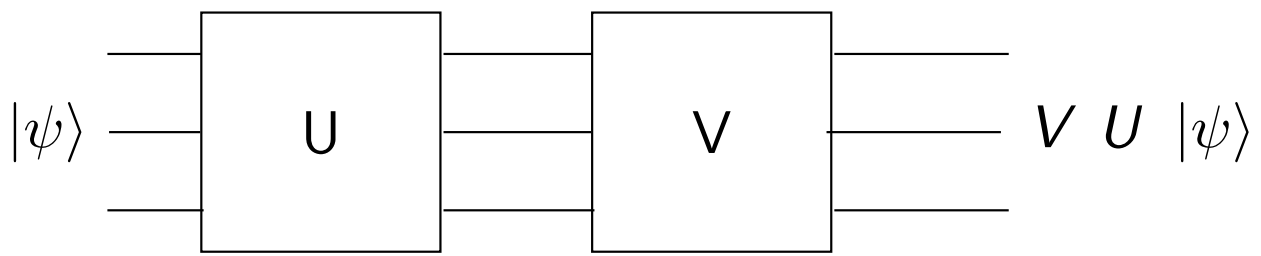
- qubits can be put side-by-side using tensor product:
$$\underbrace{|\psi_3\rangle}_{\mathbb{C}^{2^{n+m}}} = \underbrace{|\psi_1\rangle}_{\mathbb{C}^{2^n}} \otimes \underbrace{|\psi_2\rangle}_{\mathbb{C}^{2^m}}$$
- The evolution of a quantum state is governed by unitary transformations through matrix/vector products.
$$|\psi_{t_2}\rangle = U\,|\psi_{t_1}\rangle \iff |\psi_{t_1}\rangle = U^*\,|\psi_{t_2}\rangle$$

Operations on qubits

- Quantum operations are represented as gates:

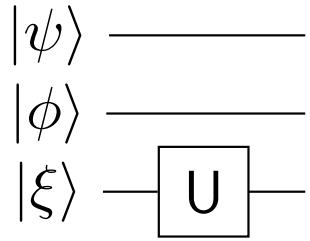


- Diagrams are read from left to right and the corresponding matrices are multiplied from right to left:



Operations on qubits

- Gates can be applied on single qubits ($U \in \mathbb{C}^{2\times 2}$ acts on $|\xi\rangle$):



  Empty lines correspond to identity gate I,
  $I \otimes I \otimes U$ is applied to the 3-qubit state.

- Effect on product states:
  $(I \otimes I \otimes U)(|\psi\rangle \otimes |\phi\rangle \otimes |\xi\rangle) = |\psi\rangle \otimes |\phi\rangle \otimes U|\xi\rangle$

- Effect on entangled state: the $2^n \times 2^n$ matrix $I \otimes I \otimes U$ is applied to the 3-qubit state $|\psi\rangle$, which may require exponential memory and time.

Outline

Gate-based quantum computer

- The unitary matrix corresponding to the circuit is specified as a combination of gates (instructions) out of a given set.
- These gates correspond to elementary unitary matrices that can be efficiently implementable by the hardware.
- The basic gates can operate on one, two, or three qubits...at the same time, resulting in so-called unary, binary, ternary... operators.

The Pauli gates (Pauli group)

- $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
- $X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
- $Y = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$
- $Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

- They form a basis for $\mathbb{C}^{2\times 2}$, they are unitary, Hermitian and we have $XYZ = iI$.
- X corresponds to the NOT gate (or bit flip),

$$X|0\rangle = |1\rangle \text{ and } X|1\rangle = |0\rangle.$$

- We also have

$$X = |0\rangle\langle 1| + |1\rangle\langle 0| \text{ and } X|j\rangle = |j \oplus 1\rangle, \text{ with } j \in \{0, 1\}.$$

- The Z gate is called the phase flip gate,

$$Z|0\rangle = |0\rangle \text{ and } Z|1\rangle = -|1\rangle.$$

The Hadamard gate

- $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.
- Useful to put a qubit into a superposition of 2 states:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \stackrel{\text{def}}{=} |+\rangle\,, \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \stackrel{\text{def}}{=} |-\rangle\,.$$

Proposition

If we have an n-qubit quantum computing device initially in the state $|0\cdots 0\rangle$ and apply the Hadamard gate to all qubits (or equivalently the matrix $H^{\otimes n}$), then we obtain the uniform superposition of basis states $\frac{1}{\sqrt{2^n}} \sum_{s\in\{0,1\}^n} |s\rangle$.

Remark

Many algorithms (see Grover) start by setting the state to a uniform superposition.

One-qubit gates: phase shift operators

- $R_\varphi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{pmatrix}$: leaves $|0\rangle$ unchanged and rotates $|1\rangle$ by the angle/phase φ.
- The Pauli gate Z corresponds to R_π.
- Special case $\varphi = \pi/2$: $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$.
- Special case $\varphi = \pi/4$: $T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$.
- We observe that $S = T^2$.

Rotation gates

- $R_x(\theta) = e^{-i\theta X/2} = \begin{pmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$,

  $R_y(\theta) = e^{-i\theta Y/2} = \begin{pmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$,

  $R_z(\theta) = e^{-i\theta Z/2} = \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}$.

- $U_1 = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}$,

  $U_2(\phi,\lambda) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\phi+\lambda)} \end{pmatrix}$,

  $U_3(\theta,\phi,\lambda) = \begin{pmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i(\phi+\lambda)}\cos\frac{\theta}{2} \end{pmatrix}$ (universal gate).

One-qubit gates: parameterized representation

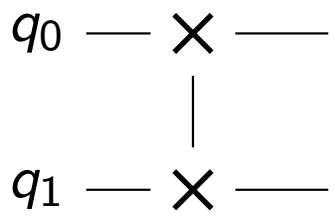
- All one-qubit gates (unitary 2-by-2 matrices) can be represented by the parameterized matrix

$$U(\theta,\phi,\lambda) = \begin{pmatrix} e^{-i(\phi+\lambda)/2}\cos(\theta/2) & -e^{-i(\phi-\lambda)/2}\sin(\theta/2) \\ e^{i(\phi-\lambda)/2}\sin(\theta/2) & e^{i(\phi+\lambda)/2}\cos(\theta/2) \end{pmatrix}.$$

- The representation is valid up to a global phase factor (i.e., to a factor of the form $e^{i\alpha}$ with $\alpha \in \mathbb{R}$).
- Examples: $X = e^{i\pi/2}U(\pi,\pi,0)$, $H = e^{i\pi/2}U(\frac{\pi}{2},0,\pi)$.
- In the simulation library Qiskit the gates can also be specified with θ,ϕ,λ.

Two-qubit gates: the SWAP gate

- The SWAP binary operator takes $|01\rangle$ to $|10\rangle$ and conversely (the states $|00\rangle$ and $|11\rangle$ remain unchanged).
- Matrix representation: $SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
- Circuit representation:



- If two qubits are in a product state $|\psi\rangle \otimes |\phi\rangle$, then

$$SWAP(|\psi\rangle \otimes |\phi\rangle) = |\phi\rangle \otimes |\psi\rangle \,.$$

- SWAP gates can be used to "move" qubits within a connectivity graph of the qubits, which can be useful when some qubits are not physically adjacent on the chip.

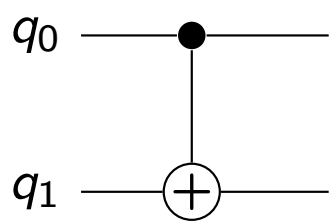
Two-qubit gates: the controlled-NOT (CNOT) gate

- The second qubit q_1 (target bit) is "controlled" by the first one q_0 (control bit): if $q_0 = |0\rangle$ then q_1 is unchanged, if $q_0 = |1\rangle$ then we apply a bit-flip on q_1 (X gate).
- Matrix representation: $CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

$$CNOT(|00\rangle) = |00\rangle \quad CNOT(|01\rangle) = |01\rangle$$

$$CNOT(|10\rangle) = |11\rangle \quad CNOT(|11\rangle) = |10\rangle$$

- Circuit representation ($CNOT_{01}$):



- Action of CNOT on basis states:
$|c\rangle\,|t\rangle \to |c\rangle\,|t \oplus c\rangle\,,\ (c, t \in \{0, 1\})$.

Properties of the controlled-NOT gate

- The CNOT is an entangling operator:

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \xrightarrow{CNOT} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq |\psi_1\rangle \otimes |\psi_2\rangle$$

- The CNOT can be used to swap two qubits since we have the identity:

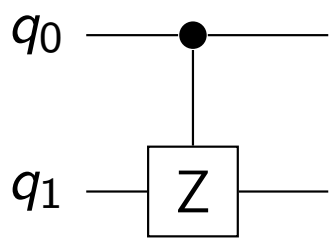
$$SWAP_{ij} = CNOT_{ij} CNOT_{ji} CNOT_{ij}.$$

  (cf exercise).

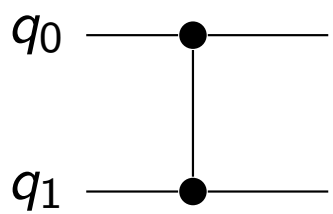
- It has been proved that the set of gates consisting of just H, T, and $CNOT$ is sufficient to construct any unitary matrix. This set of gates is said to be *universal* (see later).

Two-qubit gates: the CZ gate

- We have control and target qubits q_0 and q_1 like the CNOT but when $q_0 = |1\rangle$ then we apply the Z operator to q_1.
- Matrix representation: $CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$
- Circuit representation:



or



- CZ is symmetric: either qubit can be chosen as the control or the target, with the same result.

General *controlled* operation

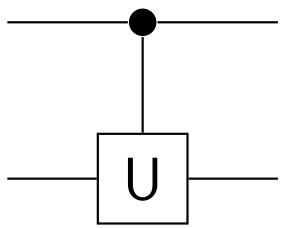
- For a single qubit operation U, the controlled-U operation is defined by:

$$|0\rangle \otimes |\psi\rangle \to |0\rangle \otimes |\psi\rangle \text{ and } |1\rangle \otimes |\psi\rangle \to |1\rangle \otimes (U|\psi\rangle)$$

or

$$|c\rangle |t\rangle \to |c\rangle U^c |t\rangle .$$

- Matrix representation: $\begin{pmatrix} I & 0 \\ 0 & U \end{pmatrix}$
- Circuit representation:



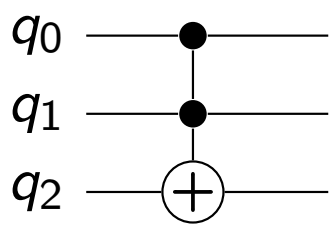
- This definition can be extended to $U \in \mathcal{U}(2^n)$ acting on n qubits.

Example of a 3-qubit gate: the CCNOT (Toffoli) gate

- The first two qubits are control and the third is the target, i.e q_0 and q_1 must be in state $|1\rangle$ to modify q_2: $(q_0, q_1, q_2) \Rightarrow (q_0, q_1, (q_2 \oplus q_0 q_1))$.
- If $q_2 = 0$, CCNOT computes the logical AND $q_0 \wedge q_1$.
- Matrix representation:

$$CCNOT = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

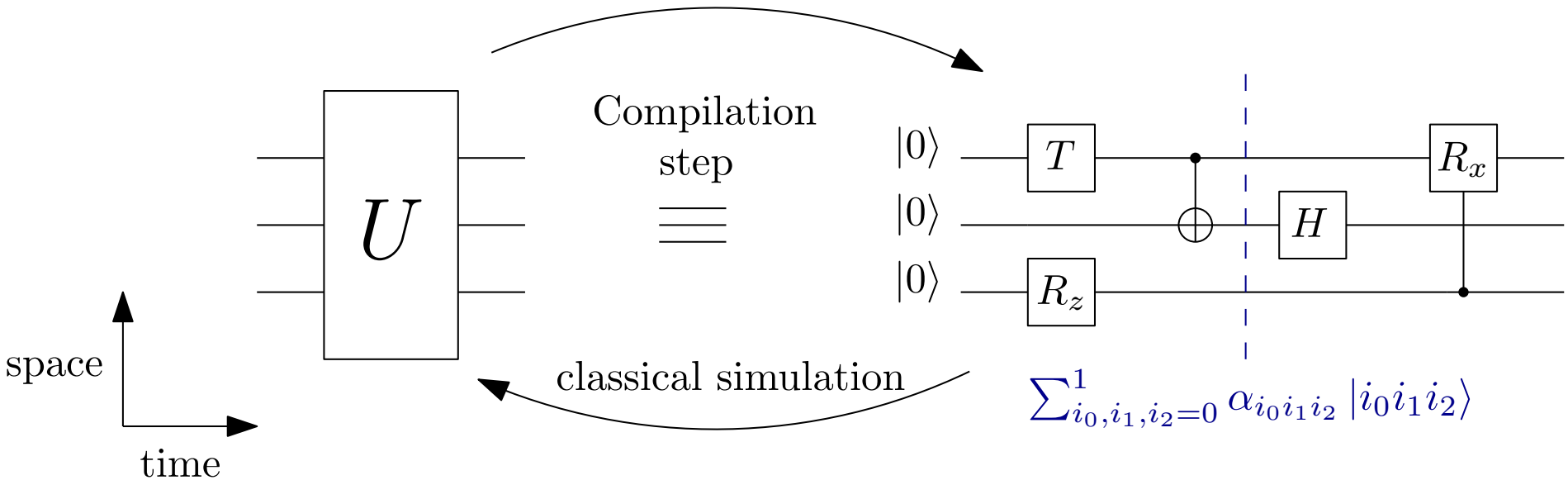
- Circuit representation:



Quantum circuit

Quantum algorithm $\equiv$ quantum circuit = series of quantum gates

Space composition $\rightarrow$ tensor product
Time composition $\rightarrow$ matrix multiplication



$$U = \Lambda_3(R_x) \times (I_2 \otimes H \otimes I_2) \times (CNOT \otimes I_2) \times (T \otimes I_2 \otimes R_z)$$

U is unknown to the hardware

What gates can we use ?

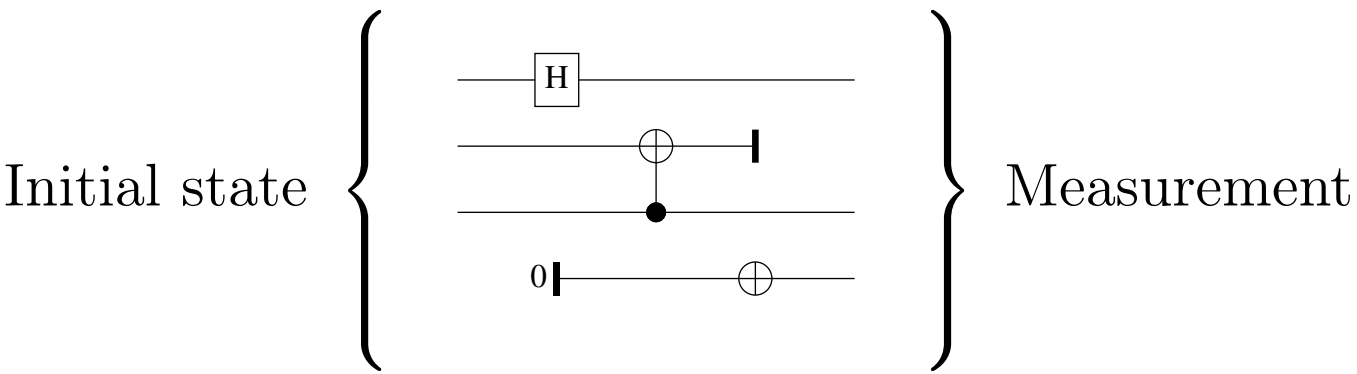
Universality: a set of gates is universal when this set is sufficient to construct any unitary matrix with arbitrary precision.

Hardware constraint: the gates available depend on the technology.

- Superconducting qubits (IBM, Google, Rigetti):
  {CNOT, H, T}
  Fault-tolerant computation: {Clifford, T}
- Trapped-ions circuits (IonQ, UIBK):
  $\{MS(\theta) = e^{-i\theta(\sum_{i=1}^{n} \sigma_x^i)^2/4}, R_z, R_x\}$

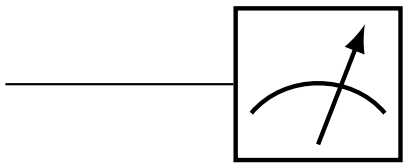
Model for quantum computations

- The input of a quantum circuit will be an initial quantum state (typically $|0\rangle$). This initialization is called "state preparation".
- The output is obtained via a measurement gate, we cannot "read" directly the state (contrary to classical computing).



Measurement

The measurement is performed via a special gate:



Given an n-qubit quantum state $|\psi\rangle = \sum_{s\in\{0,1\}^n} \alpha_s |s\rangle$:

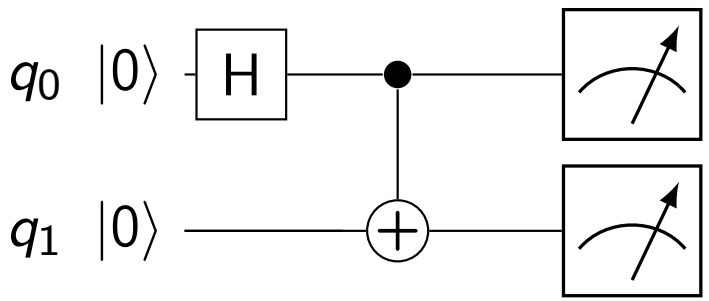
- Measurement gate on qubit $\#k$ outputs $x \in \{0,1\}$ such that:
  $x = 0$ with probability $\sum_{s\in\{0,1\}^n : s_k=0} |\alpha_s|^2$,
  $x = 1$ with probability $\sum_{s\in\{0,1\}^n : s_k=1} |\alpha_s|^2$.
- After measurement, if x is the measured value, the state becomes: $|\psi\rangle = \sum_{s\in\{0,1\}^n : s_k=x} \frac{\alpha_s}{\sqrt{\sum_{s:s_k=x} |\alpha_s|^2}} |s\rangle$.

  Example with 2 qubits: measuring the first qubit alone with result 0 gives a post-measurement state $|\psi\rangle = \frac{\alpha_{00}|00\rangle+\alpha_{01}|01\rangle}{\sqrt{|\alpha_{00}|^2+|\alpha_{01}|^2}}$.
- The original state cannot be recovered.
- The measurement of all the n qubits gives s with probability $|\alpha_s|^2$, for $s \in \{0,1\}^n$.

Measurement

- Contrary to classical physics, quantum measurement modifies deeply the observation.
- **Measurement postulate**: every measurable physical quantity is described by a Hermitian operator $\mathbb{H}$ acting on the state $|\psi\rangle$.
- Possible outputs of measurement are (real) eigenvalues of $\mathbb{H}$.
- State after measurement is represented by a unit eigenvector of $\mathbb{H}$.

Example: Bell circuit



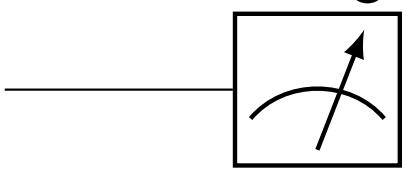
- q_0 and q_1 are prepared in state $|0\rangle$.
- Hadamard gate puts q_0 in state $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$.
- CNOT is applied to q_1 with q_0 as control qubit.
- Output state is $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ (entangled state).
- Measurement of q_0, q_1 has 50/50 chance of finding 00 and 11 (illustration with Qiskit).
- The measurement outputs for each qubit are correlated.

No-cloning theorem

- Hope: by making a copy of a state, we would keep track of the state before measurement (that destroys it). Unfortunately this is not possible, due to the properties of the quantum gates.
- Theorem: Let $|\psi\rangle$ be an arbitrary quantum state. There is no unitary matrix that maps $|\psi\rangle \otimes |0\rangle$ to $|\psi\rangle \otimes |\psi\rangle$.
- Remark: For a given state we could construct a specific unitary matrix to copy this state but there is no matrix that could do that for any state.
- As a consequence, we can reproduce the output state only by running again the circuit.

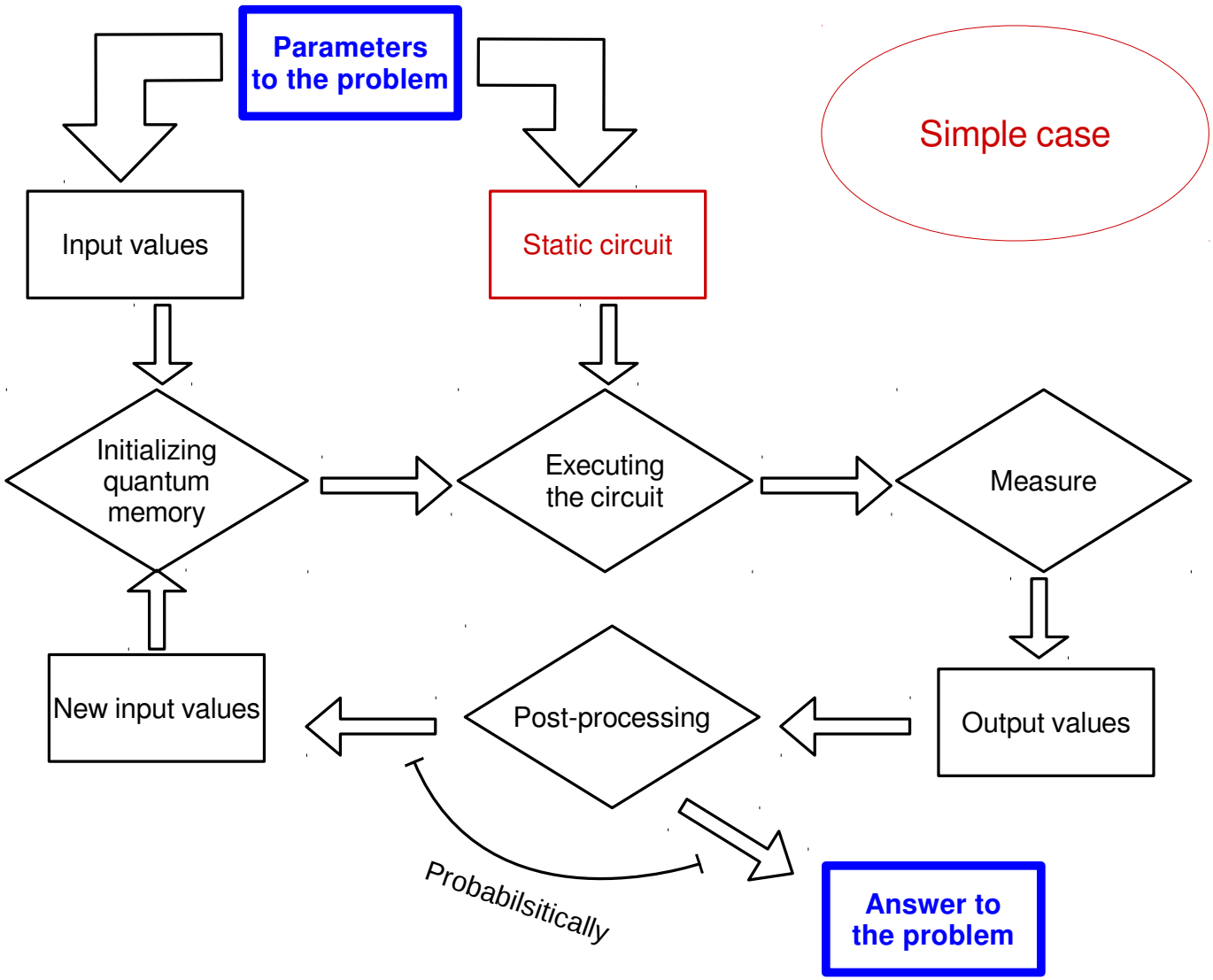
Summary: quantum algorithm

- Initialize quantum state (vector of dimension 2^n).
- Execute the circuit (matrix of size $2^n \times 2^n$) $\equiv$ series of elementary actions applied on the quantum memory.
- No "quantum loop" or "conditional escape".
- No copying (no-cloning theorem).
- No direct access to the quantum state.
- Output state is measured (and then modified!) via a measurement gate (probabilistic):



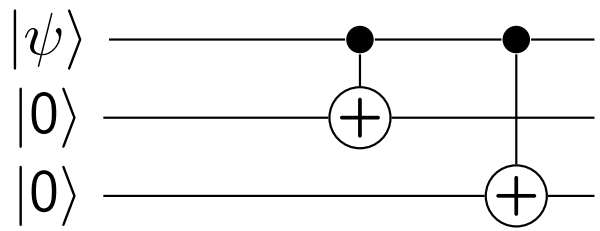
- Post-process and interact with classical computer.

Quantum / Classical interaction



Major research challenge: quantum error correction

- NISQ quantum computers: 50-100 noisy qubits
- Average gate fidelity: 0.1% to 1%
  (vs 1 failure/10^{17} operations on classical computers).
- Contrary to classical computing, no possibility for replicating states (no-cloning theorem).
- Types of error: bit flip ($|0\rangle \rightarrow |1\rangle$), phase flip ($|1\rangle \rightarrow e^{i\phi}|1\rangle$).
- Principle of Quantum Error Correcting Codes (QECC): encoding/decoding quantum states by using extra qubits.
  $\alpha|0\rangle + \beta|1\rangle \longrightarrow \alpha|000\rangle + \beta|111\rangle$



- Computationally very expensive, impact on performance, ignore errors from initialization and measurement.

Outline

The Grover's algorithm

- It belongs to the class of quantum search algorithms.
- **Pb to solve:** find a specific element in a set of $N = 2^n$ elements (the index can be stored in n bits).
- Can be used for a wide range of problems (e.g., search in an unstructured database).
- We suppose we have a test function $f : \{0,1\}^n \rightarrow \{0,1\}$: $f(x) = 1$ if x is the solution and 0 otherwise.
  We want to determine x.
- The complexity will be expressed in number of calls (queries) to the function f (*query complexity*).
- Grover showed that this problem can be solved in $\mathcal{O}(\sqrt{N})$ queries using a quantum algorithm (vs $\mathcal{O}(N)$ for a classical one) $\rightarrow$ quadratic speedup.

The oracle

- Oracle = black box (unitary operator $\equiv$ circuit) that will encode the test function f to recognize the solution.

$$O : \mathbb{C}^{2^n} \otimes \mathbb{C}^2 \to \mathbb{C}^{2^n} \otimes \mathbb{C}^2$$
$$|x\rangle |y\rangle \mapsto |x\rangle |f(x) \oplus y\rangle ,$$

  where $|x\rangle$ is the index register and $|y\rangle$ is the oracle qubit which is flipped if $f(x) = 1$ and remains unchanged otherwise.
  Exercise: verify that O is unitary.
- Let prepare $|x\rangle |0\rangle$. Then $|f(x) \oplus y\rangle = |f(x)\rangle$ and if y is flipped to 1, then x is the solution.
- We apply O to the state $|x\rangle (|0\rangle - |1\rangle)/\sqrt{2}$. If x is the solution, $|0\rangle$ and $|1\rangle$ are interchanged (unchanged otherwise).

$$O : |x\rangle (|0\rangle - |1\rangle)/\sqrt{2} \mapsto (-1)^{f(x)} |x\rangle (|0\rangle - |1\rangle)/\sqrt{2},$$

  that simplifies to

$$O : |x\rangle \mapsto (-1)^{f(x)} |x\rangle .$$

Search algorithm

Goal: find the solution using the smallest number of calls to the oracle. It requires $n+1$ qubits.

Idea: Increase iteratively the basis state coefficient that correspond to the binary string for which f gives 1.

Initialization: Prepare $|x\rangle$ in state $|0\rangle^{\otimes n}$ and apply Hadamard transform to obtain the state: $|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$.
The auxiliary qubit (oracle) is set to $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

Repeat:

1. Flip the sign of the vectors for which O gives 1.
2. Invert all coefficients of the quantum state around the average coefficients so that, after a certain number of iterations, the coefficient of the state $|s\rangle\,(|0\rangle - |1\rangle)/\sqrt{2}$ (s is the solution) has a coefficient much larger and thus will be measured with probability close to 1.

Quantum Fourier transform (QFT)

- Classical discrete Fourier transform:

$$x_0, \ldots, x_{N-1} \mapsto y_0, \ldots, y_{N-1} \text{ with } y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2i\pi jk/N}.$$

- Quantum Fourier transform: for the basis states of $\mathbb{C}^N$ (with $N = 2^n$ for an n qubits computer), we have

$$|j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2i\pi jk/N} |k\rangle$$

or equivalently, for an arbitrary state,

$$\sum_{j=0}^{N-1} x_j |j\rangle \mapsto \sum_{k=0}^{N-1} y_k |k\rangle ,$$

with the y_k's are the discrete Fourier transform of the x_j's. The QFT is a unitary transformation (exercise).

Quantum Fourier transform (QFT)

- Binary representation: $j_1 j_2 \dots j_n = j_1 2^{n-1} + j_2 2^{n-2} + \dots + j_n 2^0$
- Binary fraction: $0.j_l j_{l+1} \dots j_m = \frac{j_j}{2^1} + \frac{j_{l+1}}{2^2} \dots + \frac{j_m}{2^{m-l+1}}$
- The QFT can be expressed using the product representation

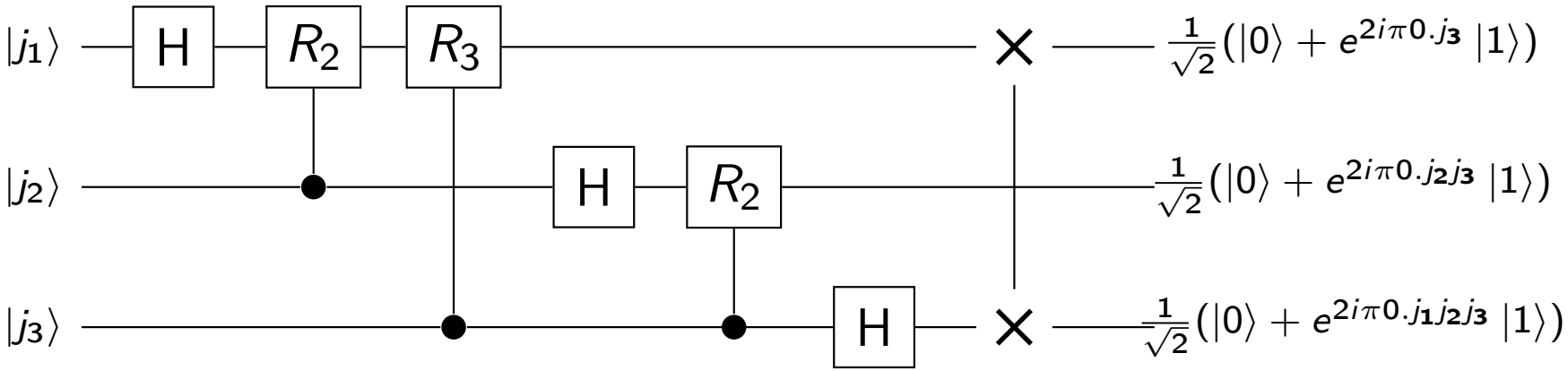
$$|j_1 \dots j_n\rangle \mapsto \frac{(|0\rangle + e^{2i\pi 0.j_n} |1\rangle)(|0\rangle + e^{2i\pi 0.j_{n-1} j_n} |1\rangle) \dots (|0\rangle + e^{2i\pi 0.j_1 j_2 \dots j_n} |1\rangle)}{\sqrt{2^n}}$$

- The QFT algorithm will compute successively the factors

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2i\pi 0.j_1 j_2 \dots j_n} |1\rangle), \frac{1}{\sqrt{2}}(|0\rangle + e^{2i\pi 0.j_2 \dots j_n} |1\rangle), \dots$$

using the gates H (to put initially each qubit in superposition) and controlled-R_k rotations where $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2i\pi/2^k} \end{pmatrix}$.
Swap gates at the end will reverse the order of the qubits. (see explanations).

Quantum Fourier transform (QFT)



Quantum circuit for the QFT (3 qubits).
(Recall that $R_2 = S$ and $R_3 = T$).

- The QFT requires $\mathcal{O}(n^2)$ gates, vs $\mathcal{O}(n2^n)$ for the best classical algorithm.
- It cannot be used for traditional applications like signal processing etc because the amplitudes cannot be accessed directly in the measurement phase **but** QFT is a key subroutine for some crucial quantum applications (phase estimation, Schor algorithm,...).

Quantum phase estimation (QPE)

- **Goal**: determine the eigenvalues of a unitary matrix U.
- Since they have modulus one, we want to estimate φ where $\lambda = e^{2i\pi\varphi}$, $0 \leq \varphi \leq 1$ ($\varphi = phase$).
- QPE is a useful routine in more general quantum algorithms (Shor, HHL,...).
- We want to estimate φ in its binary decimal representation $\varphi = 0.\varphi_1\varphi_2\cdots\varphi_n$ where $\varphi_k \in \{0,1\}$ and $\varphi = \sum_{k=1}^{n} \varphi_k 2^{-k}$.
- We use 2 registers, one for the n qubits prepared in state $|0\rangle$, one to store the eigenstate $|\psi\rangle$ of U.
- We apply a Hadamard transform to the first register and then successive controlled-U^{2^k}, $k = 0, \ldots, n-1$ operations to the second one, that remains unchanged during the computation.
- Finally we apply the inverse Fourier transform to the first register to obtain the state $|\varphi_1\rangle \otimes |\varphi_2\rangle \otimes \cdots \otimes |\varphi_n\rangle$, followed by measurement of the first register.

Outline

Summary

- Quantum computing gives promising results in some fields (artificial intelligence, cryptography, chemistry).
- Designing quantum algorithms requires a new way of thinking.
- HPC algorithms and platforms enable us to simulate and compile quantum circuits of intermediate size.
- Scalability remains limited so far.
- Quantum errors will make logical qubit hard to exploit despite progress in physical qubits.
- There is no agreement on the best qubit technology (superconducting, trapped ions, photons...).
- Special-purpose quantum processors are likely to arise commercially by the end of the 2020's.