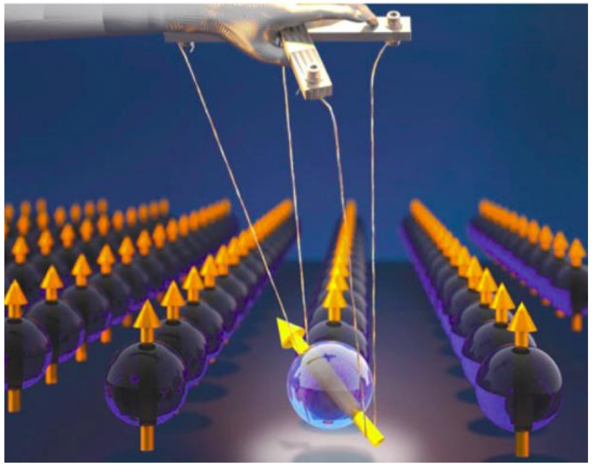
Introduction to Quantum Computing II

Advanced Quantum Algorithms



Diego Porras
Instituto de Física Fundamental
CSIC

CSIC
Spanish Council of Research

Outline

1.- Introduction

2.- Qubits and quantum gates

- (Reminder) Qubits, Bloch sphere and single qubit gates
- Multi-qubit gates
- Quantum measurements
- Many-qubit states and complexity

3.- Quantum algorithms

- Grover’s algorithm and quantum search
- Quantum Fourier Transform and other algorithms
- NISQ devices. Motivation for the quest for new QC paradigms

4.- Quantum annealing and quantum variational algorithms

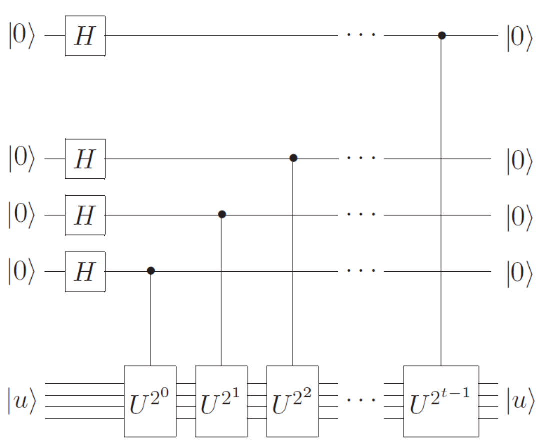
- QUBO problems, the Ising model and quantum annealing
- Quantum Approximate Optimization Algorithm (QAOA)
- Variational Eigensolver Algorithm (VQE)
- Quantum variational algorithms in this summer School

Two types of quantum computation

Universal Digital QC (Fault tolerant QC)

Relies on an ideal quantum computer with a large number of very good qubits.

Exploits the circuit model of QC and thus it **requires error correction**

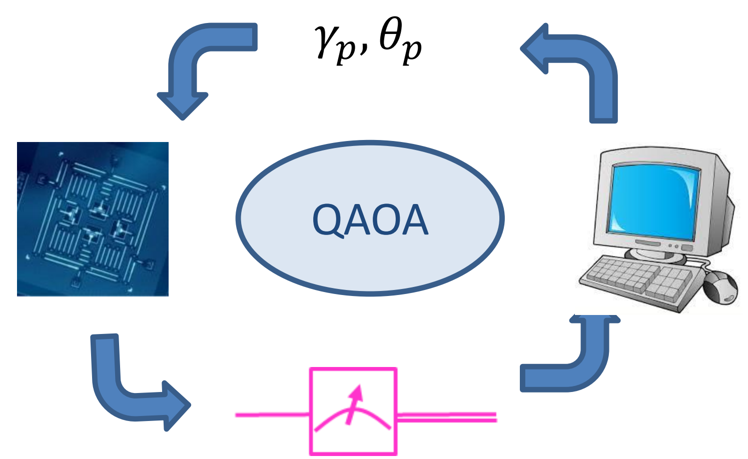


„Analogical“ QC (quantum annealers, hybrid methods)

Sacrifices universality for the sake of feasibility of implementation.

Tolerant to errors without the need of quantum error correction

Examples: Quantum annealers, hybrid quantum-classical methods, quantum simulators…



Qubits

- In classical computation, a *bit* is the basic unit of information

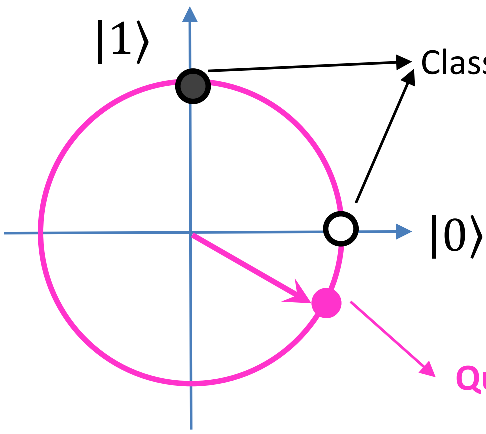
$$\text{BIT:} \quad 0{,}1$$

- In quantum physics, this is replaced by a mathematical object known as qubit

$$\text{QUBIT:}\, |\psi\rangle = \alpha\, |0\rangle + \beta\, |1\rangle$$

α, β are complex numbers and they have to fulfil that $|\alpha|^2 + |\beta|^2 = 1$

- A qubit's quantum state is analogous to a vector. Contrary to a classical (binary) state, a qubit can be in a continuum of quantum states:



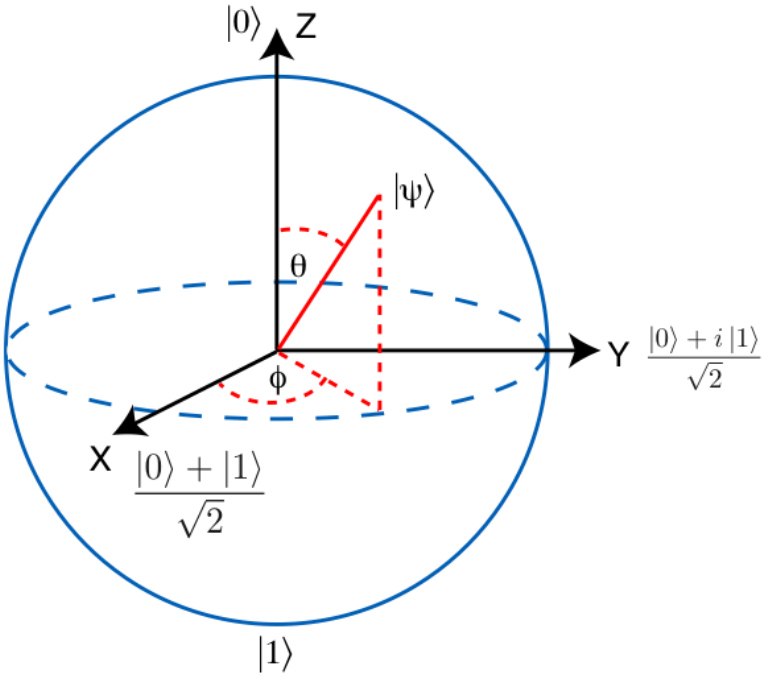
Quantum states (anywhere on the circumference)

$$|\psi\rangle = \binom{\alpha}{\beta} = \alpha|0\rangle + \beta\, |1\rangle$$

The Bloch Sphere

- A very convenient geometrical representation of a qubit's state is in terms of a vector in a sphere of unit radius (Bloch sphere):

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



- Bloch vector components: expectation values of each of the Pauli matrices (spin components)

$$\mathrm{X} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

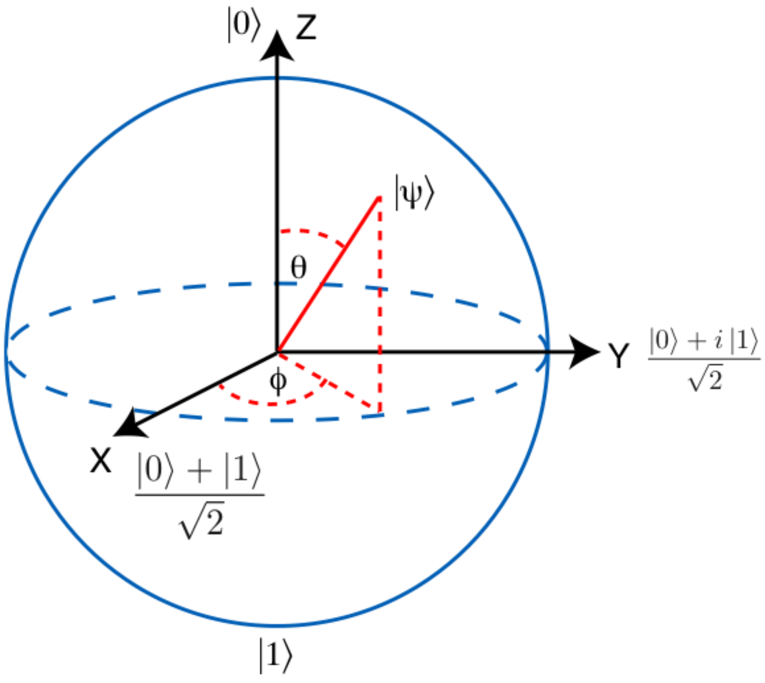
$$\mathrm{Y} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \qquad \vec{u} = \begin{pmatrix} \langle\psi|X|\psi\rangle \\ \langle\psi|Y|\psi\rangle \\ \langle\psi|X|\psi\rangle \end{pmatrix}$$

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

The Bloch Sphere

- A very convenient geometrical representation of a qubit's state is in terms of a vector in a sphere of unit radius (Bloch sphere):

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



- If you are not very familiar with Pauli matrices and expectation values, just think in terms of amplitude and phases of complex numbers!

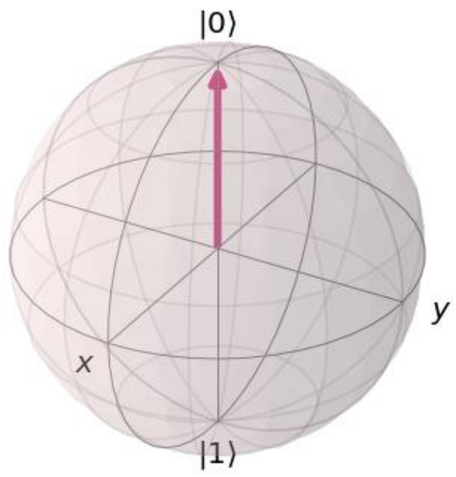
$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

$$\alpha = \cos\left(\frac{\theta}{2}\right)$$

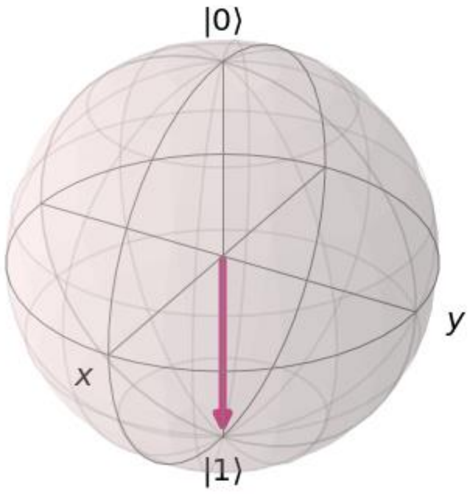
$$\beta = \mathrm{e}^{i\phi}\sin\left(\frac{\theta}{2}\right)$$

The Bloch Sphere

- “Classical” states



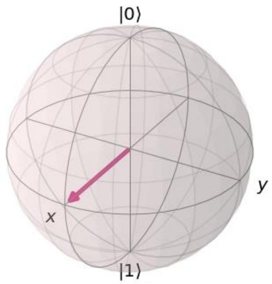
$$|\psi\rangle = |0\rangle$$



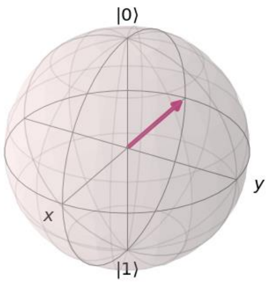
$$|\psi\rangle = |1\rangle$$

The Bloch Sphere

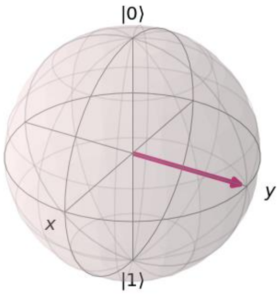
- Most “Quantum” states: linear superpositions that live in the equator of the Bloch sphere



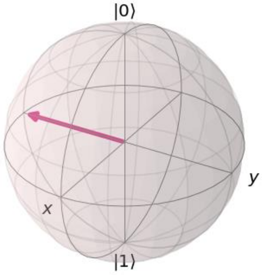
$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$



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



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



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

Quantum gates

- You can think of a quantum computation as a process in which the qubit state $|\psi\rangle$ undergoes some transformations (single qubit gates, if you only have one qubit)

- Those changes are unitary operations induced on the qubit (they must respect the normalization condition)

$$
\begin{tikzcd}
|0\rangle \arrow[r, "\text{create superposition}"] & \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \arrow[d, "\text{change sign of coefficient } \beta"] & \\
& \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \arrow[r, "\text{change phase}"] & \frac{1}{\sqrt{2}}\left(|0\rangle - e^{i\phi}|1\rangle\right)
\end{tikzcd}
$$

Quantum gates

- You can think of a quantum computation as a process in which the qubit state $|\psi\rangle$ undergoes some transformations (single qubit gates, if you only have one qubit)

- Those changes are unitary operations induced on the qubit (they must respect the normalization condition)

(multiply by)

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$$

$$\begin{pmatrix}1\\0\end{pmatrix} \longrightarrow \boxed{H} \longrightarrow \frac{1}{\sqrt{2}}\begin{pmatrix}1\\1\end{pmatrix}$$

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

(multiply by)

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$$

$$\boxed{Z} \longrightarrow \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix}$$

(multiply by)

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 0\\ 0 & e^{i\phi}\end{pmatrix}$$

$$\frac{1}{\sqrt{2}}\begin{pmatrix}1\\-1\end{pmatrix} \longrightarrow \boxed{R_\phi} \longrightarrow \frac{1}{\sqrt{2}}\begin{pmatrix}1\\-e^{i\phi}\end{pmatrix}$$

$$\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle - e^{i\phi}|1\rangle)$$

Summary of single-qubit quantum gates

- The Hadamard gate (creates superpositions)

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{cases} H|0\rangle = \dfrac{1}{\sqrt{2}}(\,|0\rangle + |1\rangle\,) = |+\rangle \\ H|1\rangle = \dfrac{1}{\sqrt{2}}(\,|0\rangle - |1\rangle\,) = |-\rangle \end{cases}$$

- The Pauli gates (define spatial coordinates, X,Y,Z)

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{cases} X|0\rangle = |1\rangle \\ X|1\rangle = |0\rangle \end{cases}$$

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

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{cases} Z|0\rangle = |0\rangle \\ Z|1\rangle = -|1\rangle \end{cases}$$

Summary of single-qubit quantum gates

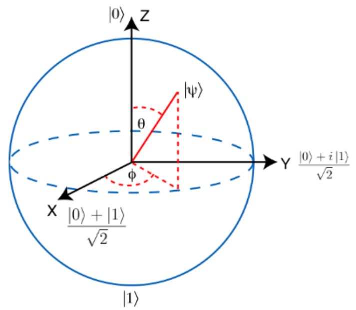
- Rotation in x-y plane (similar to the Z-Pauli gate, but complex phase)

$$R_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\,\phi} \end{pmatrix} \quad \begin{cases} R_\phi|0\rangle = |0\rangle \\ R_\phi|1\rangle = e^{i\phi}|1\rangle \end{cases}$$

- Also, rotations around x,y,z axis in the Bloch sphere:

$$R_z(\phi) = \begin{pmatrix} e^{-\frac{i\phi}{2}} & 0 \\ 0 & e^{\frac{i\phi}{2}} \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{pmatrix}$$

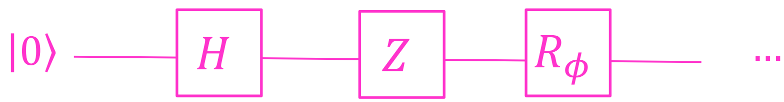


Very good intro at:

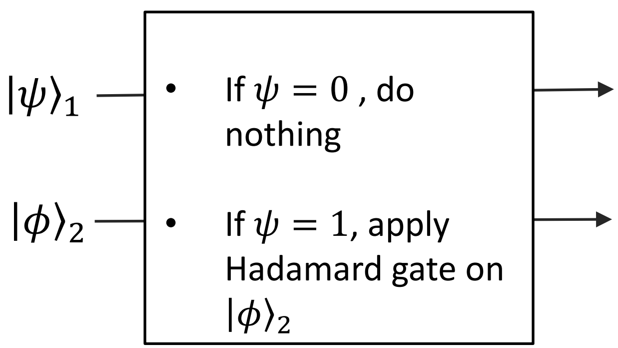
https://qiskit.org/textbook/ch-states/single-qubit-gates.html

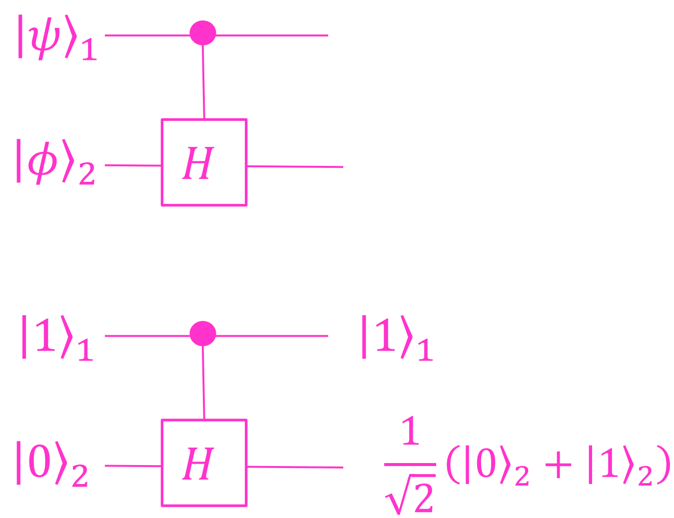
2-qubit quantum gates

- A quantum circuit looks like a set of boxes acting within the temporal line of a qubit evolution
- *Remember that this is just 2x2 matrices applied to a two-component vector (it just takes some time to get used the notation)*



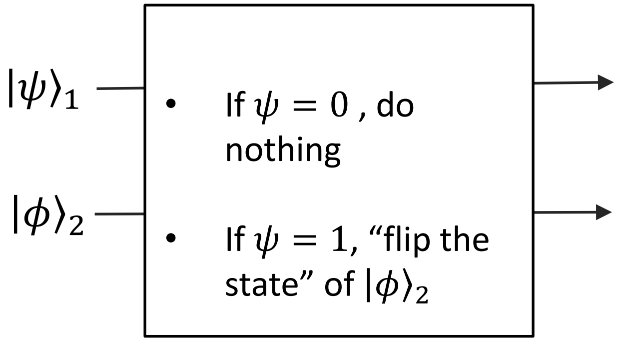
- We can also have 2-qubit operations acting simultaneously on two qubits. Example:

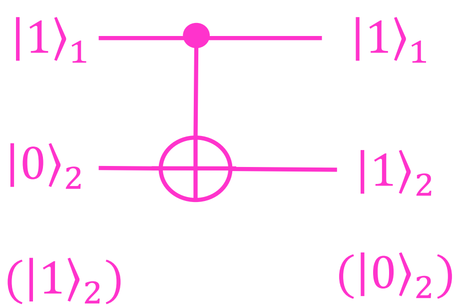




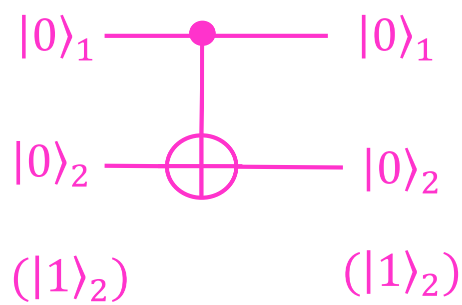
2-quibt quantum gates

- Another famous 2-quibt gate: the CNOT gate





"flip qubit 2"

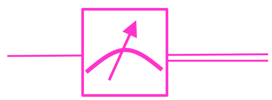


"do not flip qubit 2"

Quantum measurement

- How do we extract information from qubits after a quantum computation has been performed?
- One of the weirdest processes in quantum mechanics comes into play: the **quantum measurement**

$|\psi\rangle$



outcome

$|0\rangle$ with probability $p = |\alpha|^2$

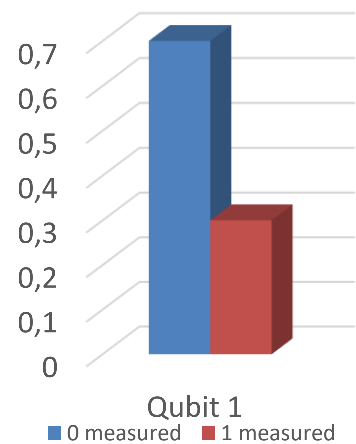
or

$|1\rangle$ with probability $p = |\beta|^2$

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

- Measurement apparatus

(e.g. photo detector measures energy emitted by the quibt only if it is in "excited" state 1)



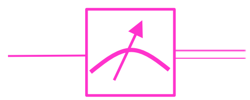
Repeat the experiment many-times.

The % of times that you measure 0 (or 1) tells you what is the value of $|\alpha|^2$ or $|\beta|^2$

Quantum measurement

- Just to see how this works in practice, consider the following two possible sequences of quantum operations:

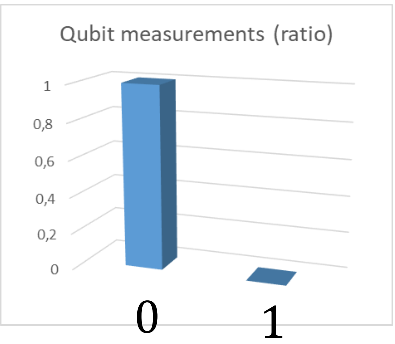
A – Qubit starts in $|0\rangle$, we just measure without any further operation

$|0\rangle$ 

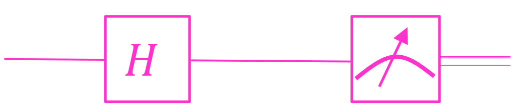
$|0\rangle$ with prob. $= |\alpha|^2 = 1$

or

$|1\rangle$ with prob. $= |\beta|^2 = 0$



B – Qubit starts in $|0\rangle$, then Hadamard gate (create superposition), then we measure:

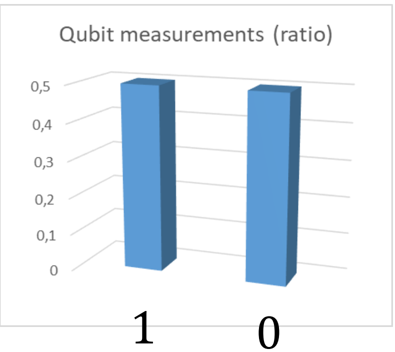
$|0\rangle$ 

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

$|0\rangle$ with prob. $= |\alpha|^2 = 1/2$

or

$|1\rangle$ with prob. $= |\beta|^2 = 1/2$



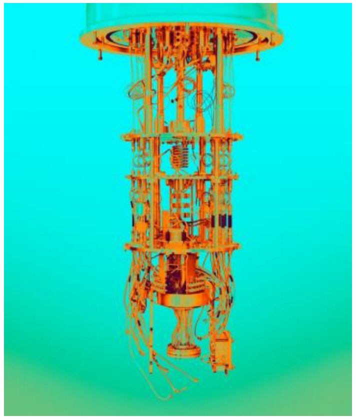
Quantum measurement

- Quantum *measurements are not a joke*. They are actually a very time-consuming way of extracting information from a quantum computer.

- This is actually **the bottleneck** of many current applications of quantum computers (e.g. in Data Science or Quantum Machine Learning, where quantum computers would need to process large amount of data)

$$|\Psi_{in}\rangle$$

Encode information
(2^N numbers)



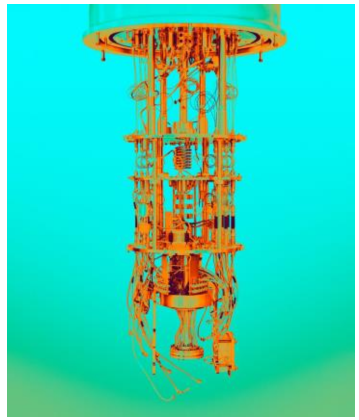
$$|\Psi_{out}\rangle$$

Read outcome
(2^N numbers)

Quantum measurement

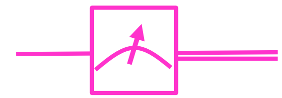
$|\Psi_{in}\rangle$

Encode information
(2^N numbers)



$U|\Psi_{in}\rangle$

$|\Psi_{out}\rangle$



Read outcome
(2^N numbers)

0111010010010 1011

1010010010101 0010

1110100100101 0100

0100111010010 01010

0001110100100101 0110

The “art of quantum software”
→ find smart ways to encode
information in these numbers

Many-qubit states

- If you have more than one qubit, the dimension of the vector (Hilbert) space increases,

$N = 2$

Qubit 1	Qubit 2
$\vert 0\rangle_1$	$\vert 0\rangle_2$
$\vert 1\rangle_1$	$\vert 1\rangle_2$

$\Rightarrow$

states
$\vert 0\rangle_1 \vert 0\rangle_2$
$\vert 1\rangle_1 \vert 0\rangle_2$
$\vert 0\rangle_1 \vert 1\rangle_2$
$\vert 1\rangle_1 \vert 1\rangle_2$

- Single qubit quantum gates act independently on each of the qubits

$$|0\rangle_1 \;-\!\boxed{H}\!-\; \frac{1}{\sqrt{2}}(|0\rangle_1 + |1\rangle_1)$$

$$|0\rangle_2 \;-\!\boxed{H}\!-\; \frac{1}{\sqrt{2}}(|0\rangle_2 + |1\rangle_2)$$

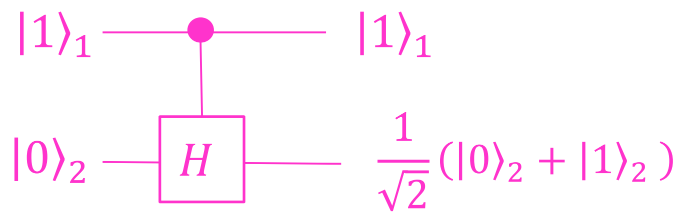
$$|0\rangle_1|0\rangle_2 \longrightarrow \frac{1}{2}(|0\rangle_1 + |1\rangle_1)(|0\rangle_2 + |1\rangle_2)$$

$$\frac{1}{2}(|0\rangle_1|0\rangle_2 + |0\rangle_1|1\rangle_2 + |1\rangle_1|0\rangle_2 + |1\rangle_1|1\rangle_2)$$

Entanglement = Complexity

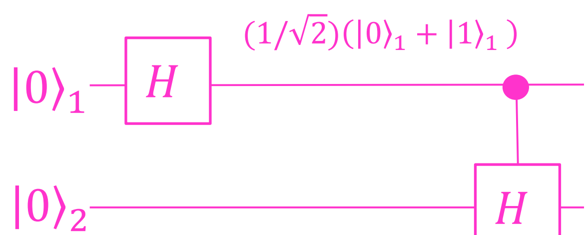
- If you think about the quantum state generated by a quantum circuit, in general, this is easily going to get complicated. For example:

$$|\Psi_{\mathrm{i}}\rangle = |1\rangle_1|0\rangle_2$$



- Consider now a sequence of gates:

$$|\Psi_{\mathrm{i}}\rangle = |0\rangle_1|0\rangle_2$$



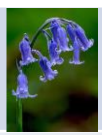
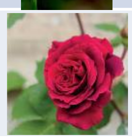
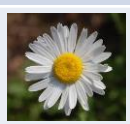
$$|\Psi_{\mathrm{f}}\rangle = \underbrace{\frac{1}{\sqrt{2}}|0\rangle_1|0\rangle_2}_{\text{do nothing}} + \underbrace{\frac{1}{2}|1\rangle_1(|0\rangle_2 + |1\rangle_2)}_{\text{Hadamard on qubit 2}}$$

$$= \frac{1}{\sqrt{2}}|0\rangle_1|0\rangle_2 + \frac{1}{2}|1\rangle_1|0\rangle_2 + \frac{1}{2}|1\rangle_1|1\rangle_2$$

Grover's algorithm and quantum search in unstructured data

- Search for an object in an unstructured set of N elements → requires $O(N)$ checks

- Example: You have a list of N items (flowers), each in a room number. The list has not been sorted and you have to look for one of the items (e.g. "Rose"), to find some information associated to it (e.g. "Room 4")

Name	Room
Bluebell 	1
Rose 	4
Poppy 	2
Daisy 	3

Classically you have to read the list an average of N times to find the searched item

Grover‘s algorithm and quantum search in unstructured data

- We will show that quantum mechanically we can create a superpositon of N elements and search efficiently with only N steps.

VOLUME 79, NUMBER 2 PHYSICAL REVIEW LETTERS 14 JULY 1997

Quantum Mechanics Helps in Searching for a Needle in a Haystack

Lov K. Grover*

3C-404A Bell Labs, 600 Mountain Avenue, Murray Hill, New Jersey 07974

(Received 4 December 1996)

Quantum mechanics can speed up a range of search applications over unsorted data. For example, imagine a phone directory containing N names arranged in completely random order. To find someone's phone number with a probability of 50%, any classical algorithm (whether deterministic or probabilistic) will need to access the database a minimum of $0.5N$ times. Quantum mechanical systems can be in a superposition of states and simultaneously examine multiple names. By properly adjusting the phases of various operations, successful computations reinforce each other while others interfere randomly. As a result, the desired phone number can be obtained in only $O(\sqrt{N})$ accesses to the database.

Grover's algorithm and quantum search in unstructured data

- Quantum search algorithms rely on Grover's algorithm. We need to create a quantum data base, which is a linear superposition of the form:

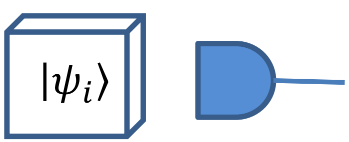
$$|\psi_i\rangle = \frac{1}{\sqrt{4}}\left(|\text{Bluebell}\rangle|1\rangle + |\text{Rose}\rangle|4\rangle + |\text{Poppy}\rangle|2\rangle + |\text{Daisy}\rangle|3\rangle\right)$$

name
(known)

Index
(what we are looking for)

- Brute force → perform quantum measurements of the name
  → repeat until you get the searched value ("Rose")
  → then, measure the index value ("room number")

On average $N = 4$ runs until you measure "Rose"



outcome	state after measurement, $|\psi_f\rangle$	prob.
Bluebell	$(|\text{Bluebell}\rangle|1\rangle$	¼
Rose	$|\text{Rose}\rangle|4\rangle$	¼
Poppy	$|\text{Poppy}\rangle|2\rangle$	¼
Daisy	$|\text{Daisy}\rangle|3\rangle$	¼

Grover‘s algorithm and quantum search in unstructured data

- Grover’s algorithm working principle is “amplitude amplification”. In the initial state all data-base entries have the same probability, but the algorithm enhances the amplitude that we detect the searched item.

- The algorithm works by applying Grover’s algorithm a number, n_{steps}, of times. We will show that only $n_{\text{steps}} \approx \sqrt{N}$ is needed, contrary to $n_{\text{steps}} \approx N$ in the brute force approach.

$$|\psi_i\rangle = \frac{1}{\sqrt{N}}\left(|\text{Bluebell}\rangle|1\rangle + {\color{red}|\text{Rose}\rangle|4\rangle} + |\text{Poppy}\rangle|2\rangle + \cdots \right)$$

$$|\psi_f\rangle = G^{n_{\text{steps}}}\,|\psi_i\rangle$$

$$|\psi_f\rangle \approx {\color{red}|\text{Rose}\rangle|4\rangle}$$

Grover‘s algorithm and quantum search in unstructured data

- Let’s see how it works!

- Imagine that we define our search over the $N = 2^n$ possible states of a system of n qubits.

$$|x = 0\rangle = |\, 0\; 0\; 0\; 0\; \ldots 0\; 0\, \rangle$$

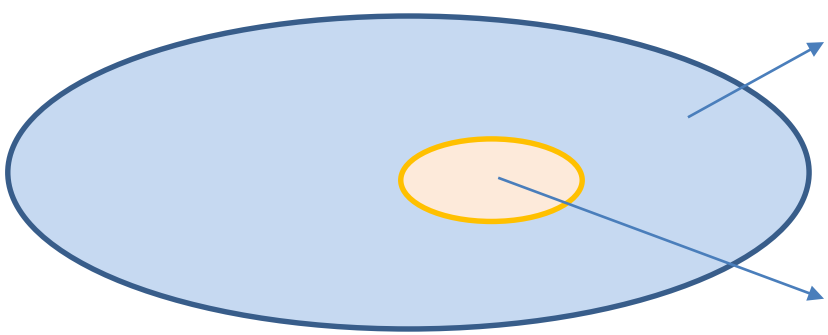
$$|x = 1\rangle = |\, 1\; 0\; 0\; 0\; \ldots 0\; 0\, \rangle$$

$$|x = 2\rangle = |\, 0\; 1\; 0\; 0\; \ldots 0\; 0\, \rangle$$

$$\vdots$$

$$|x = s_0\, 2^0 + s_1\, 2^1 + s_2\, 2^2 + \cdots\rangle = |\, s_0\; s_1\; s_2\; s_3\; \ldots s_{N-1}\, \rangle$$

- We are searching for a set of M values of x, amongst the total N values



$N - N$ values do not satisfy search (not OK)

M values satisfy search (OK)

Grover's algorithm and quantum search in unstructured data

- Imagine that we create quantum superposition of all posible states

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0,\dots,N-1} |x\rangle$$

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{x\,\in not\ OK} |x\rangle$$

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x\in OK} |x\rangle$$

$$|\psi_0\rangle = \underbrace{\sqrt{\frac{N-M}{N}}}_{a_0}|\alpha\rangle + \underbrace{\sqrt{\frac{M}{N}}}_{b_0}|\beta\rangle$$

Grover's algorithm and quantum search in unstructured data

- Imagine that we create quantum superposition of all posible states

$$|\psi_0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0,\ldots,N-1} |x\rangle$$

$$|\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{x\,\in not\,OK} |x\rangle$$

$$|\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x\in OK} |x\rangle$$

How can you create this state?

$$|\psi_0\rangle = H^{\otimes n}|0\rangle_1|0\rangle_2 \ldots |0\rangle_n$$

$$= \frac{1}{\sqrt{2}}(\,|0\rangle_1 + |1\rangle_1)\frac{1}{\sqrt{2}}(\,|0\rangle_2 + |1\rangle_2)\ldots\frac{1}{\sqrt{2}}(\,|0\rangle_n + |1\rangle_n)$$

$$= \frac{1}{\sqrt{2^n}}\underbrace{(|0\rangle_1|0\rangle_2 \ldots |0\rangle_n \;+\; |1\rangle_1|0\rangle_2 \ldots |0\rangle_n \;+\; |1\rangle_1|1\rangle_2 \ldots |0\rangle_n \;+ \cdots)}_{\text{All possible combinations}}$$

Grover‘s algorithm and quantum search in unstructured data

- Grover’s algorithm consists of a series of steps, such that the wavefunction of the quantum computer stays within the $\{\,|\alpha\rangle,\ |\beta\rangle\}$ subspace.

$$|\psi_m\rangle = G\,|\psi_{m-1}\rangle = G^2|\psi_{m-2}\rangle = \cdots$$

$$|\psi_m\rangle = a_m|\alpha\rangle + b_m|\beta\rangle$$

- We will see that by an appropriate choice of the operator G, the probability amplitude b_m can grow from $b_0 = \sqrt{M/N} \ll 1$ up to a value $b_m \approx 1$:

- How do we build the Grover operator? It is made out of two operators:

$$G = (2\,|\psi_0\rangle\langle\psi_0| - \mathbf{I})\,O$$

Reflection around the initial state $|\psi_0\rangle$

Oracle

Grover's algorithm and quantum search in unstructured data

$$G = (2\,|\psi_0\rangle\langle\psi_0| \; - \mathbf{I})\; O$$

- Let's start with the Oracle operator (cool name, isn't it?)

- The Oracle action is to add a minus sign only in front of those states which satisfy the search condition

$$O|x\rangle = \begin{cases} |x\rangle & \text{If } x \in \text{not OK} \\ -|x\rangle & \text{If } x \in \text{ OK} \end{cases}$$

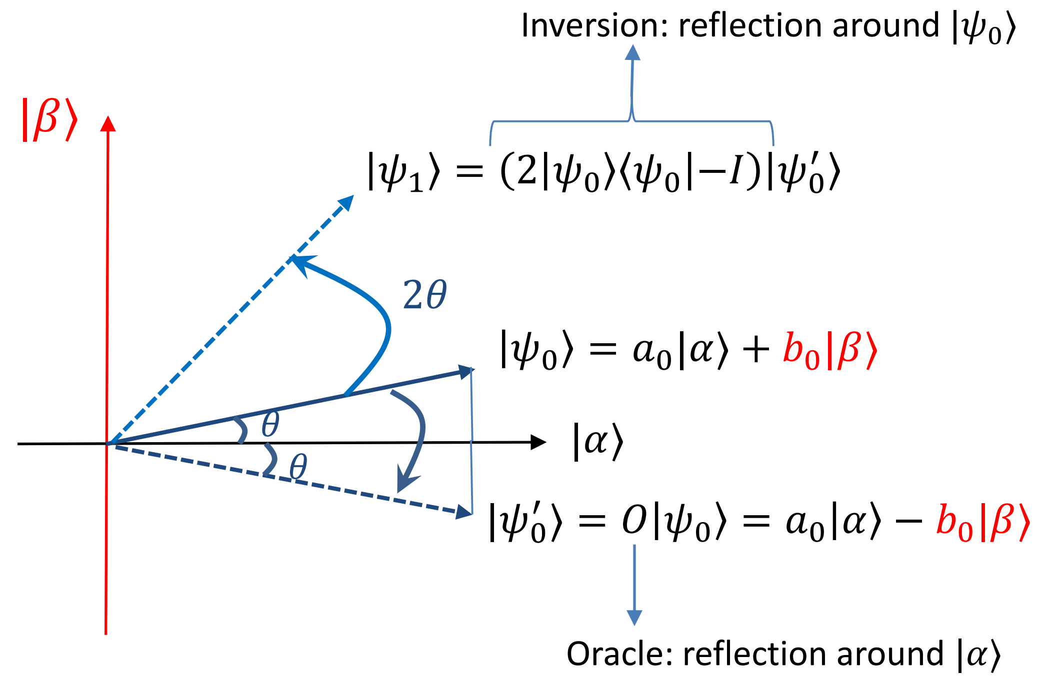
- The action of the Oracle on the initial wavefunction is

$$O\;|\psi_m\rangle = O(a_m|\alpha\rangle + b_m|\beta\rangle) = a_m|\alpha\rangle - b_m|\beta\rangle$$

Grover‘s algorithm and quantum search in unstructured data

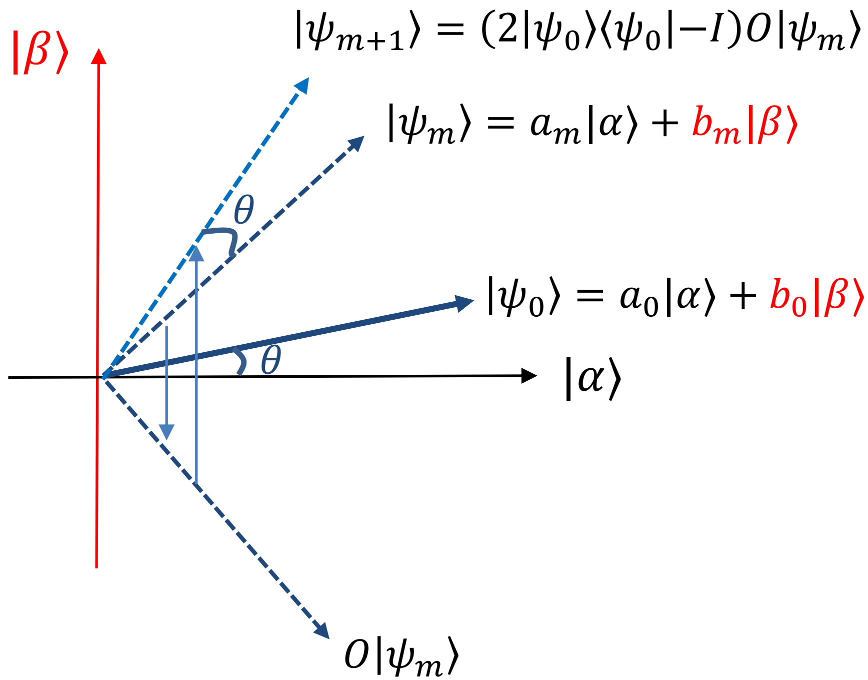
- To understand the action of the Grover operator, it is very useful to consider the following geometrical interpretation.

- Consider that the quantum computer's wavefunction is a vector in the two-dimensional subspace spanned by $\{\,|\alpha\rangle,\ |\beta\rangle\}$



Grover's algorithm and quantum search in unstructured data

- The joint action of the Oracle operator and the inversion operator implies that the wavevector is lifted up towards the searched state



Grover's algorithm and quantum search in unstructured data

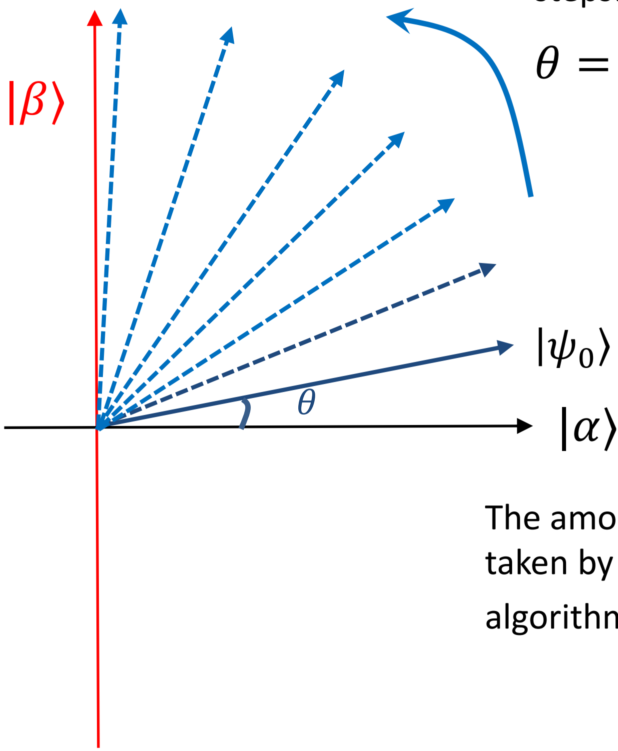
- The net effect is that the wavevector, approaches the searched state

$$|\psi_m\rangle = a_m|\alpha\rangle + b_m|\beta\rangle$$

$b_0 \approx 1$

State-vector evolves at steps:

$$\theta = \frac{b_0}{a_0} \approx \frac{\sqrt{M}}{\sqrt{N}}$$



$$|\psi_0\rangle = a_0|\alpha\rangle + b_0|\beta\rangle$$

$b_0 \ll a_0$

The amount of steps taken by the quantum algorithm is $\propto \frac{1}{\sqrt{N}}$

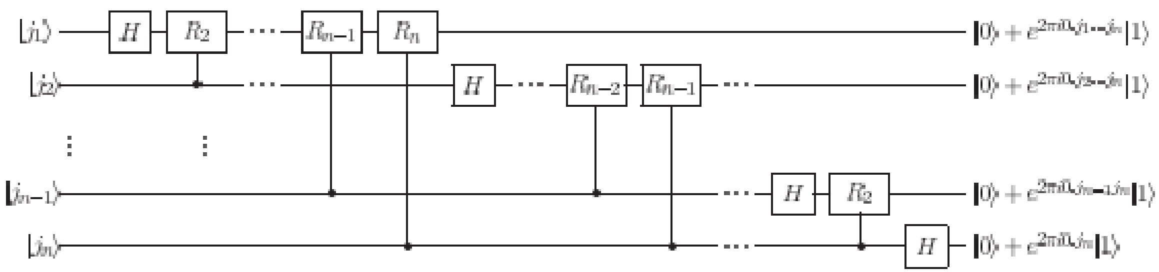
Question: What happens if the wavevector $|\psi_m\rangle$, is close but not exactly $|\beta\rangle$??

Can you say that the search was succesful?

The Quantum Fourier Transform

- Many other useful quantum algorithms are based on the Quantum Fourier Transform

$$\sum_{j=0}^{N-1} x_j \,|\, j \,\rangle \rightarrow \sum_{j=0}^{N-1} y_k \,|\, k \,\rangle = \frac{1}{\sqrt{N}} \sum_{j,k} e^{\,i\,\frac{2\,\pi\, j\cdot k}{N}} x_j |\, k\rangle$$



- (see Juan José García Ripoll's talks and the Quantum Fourier Transform jupyter notebook)

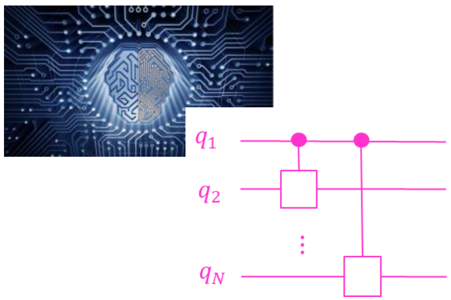
- Applications of the QFT: Quantum phase estimation and **Shor's factorization algorithm.**

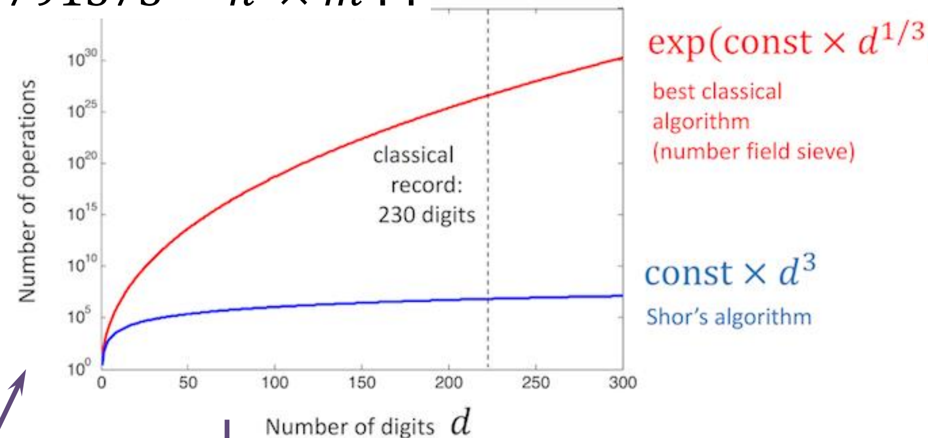
This is the most celebrated application of quantum computers and it would allow to break the RSA cryptosystems that are the basis for secure communication nowdays!

Quantum Computing: reality Vs. Expectations

Expectations

$$\overbrace{8135791573}^{d\text{ digits}} = n \times m\,??$$



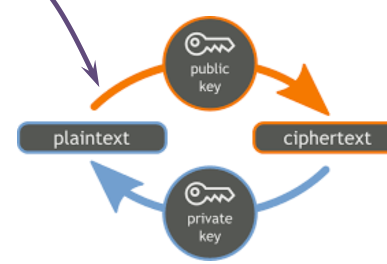


Ideal (Universal) quantum computer

- Highly precise gates (errors < $10^{-2}, 10^{-3}$)
- Quantum error correction (requires extra number of qubits)
- Any quantum algorithm, including the celebrated "quantum Shor's alrgorithm"

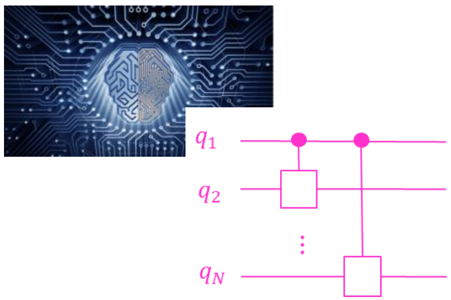
- Factorizing a number into primes is very hard computationally (grows exponentially with number of digits)
- But a quantum computer can do it much more easily!!
- In practice: more than 10^8 "good" qubits required to factorize a 2048-digit number.

Breaking RSA cryptosystems??



Quantum Computing: reality Vs. Expectations

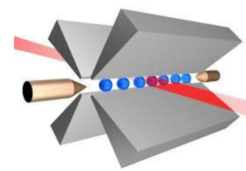
Expectations

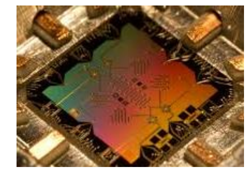


Ideal (Universal) quantum computer

- Highly precise gates (errors < $10^{-2}, 10^{-3}$)
- Quantum error correction (requires extra number of qubits)
- Any quantum algorithm, including the celebrated “quantum Shor’s alrgorithm”
- Also, it could simulate any quantum dynamics (and material)

Reality





Real quantum computer

- Quantum gates with large errors ($0.5 * 10^{-1}$ in superconducting qubits) or more accurate but difficult to scale up (ions)
- Very limited in number of qubits (about 20 qubits)
- Very unlikely to be able to run Shor’s algorithm (or similiar in complexity) in the near future

Quantum Computing: reality Vs. Expectations

Quantum Computing in the NISQ era and beyond

John Preskill

Institute for Quantum Information and Matter and Walter Burke Institute for Theoretical Physics,
California Institute of Technology, Pasadena CA 91125, USA

30 July 2018

Noisy Intermediate-Scale Quantum (NISQ) technology will be available in the near future. Quantum computers with 50-100 qubits may be able to perform tasks which surpass the capabilities of today's classical digital computers, but noise in quantum gates will limit the size of quantum circuits that can be executed reliably. NISQ devices will be useful tools for exploring many-body quantum physics, and may have other useful applications, but the 100-qubit quantum computer will not change the world right away — we should regard it as a significant step toward the more powerful quantum technologies of the future. Quantum technologists should continue to strive for more accurate quantum gates and, eventually, fully fault-tolerant quantum computing.

An alternative to circuit quantum computing: quantum annealing

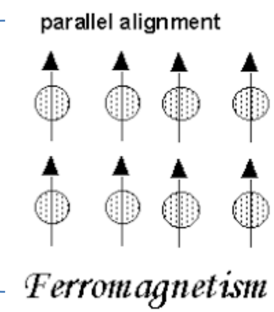
- Quantum annealing – an idea that can be used to solve optimization problems.

- Imagine that you have a set of binary variables, $s_1, s_2, \ldots, s_N = \pm 1$. You want to minimize a function of those variables (*objective function*),

$$f(s_1, s_2, \ldots, s_N)$$

- A simple example of this is the ferromagnetic Ising spin model (+/- is spin up/down)

$$f(s_1, s_2, \ldots, s_N) = -\sum_{j,k} s_j s_k \, ,$$

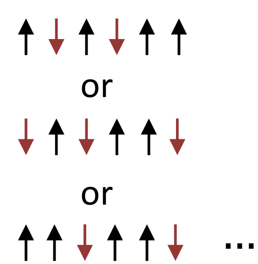


where the solution is simply given by all $s_j = +1$ or -1.

- Other problems may be more complicated, even if they have only pair-wise interactions (*QUBO*, Quadratic Unconstrained Binary Optimization, problems):

$$f(s_1, s_2, \ldots, s_N) = \sum_{j,k} J_{jk} s_j s_k$$

Coupling between sites *j* and *k*



Quantum annealing

- In general, the objective function can have a more complicated form with higher than quadratic term:

$$f(s_1, s_2, \dots, s_N) = \sum_j h_j s_j + \sum_{j,k} J_{jk}^{(2)} s_j s_k + \sum_{j,k,l} J_{jkl}^{(3)} s_j s_k s_l + \cdots$$

- A few examples of optimization problems that can be solved:
  - Portfolio optimization (financial sector)
  - Logistics and route planning
  - Website optimization

- Finding the minimum value of the objective function can be a very hard problem

- Classical Monte Carlo methods can be used here, but also computationally demanding

Quantum annealing

- How to transform any optimization problem to a binary optimization problem?
- Consider for example that your problema consists of minimizing a multi-variable function of an integer, x:

$$f(x) = a + b\ \ x + c\, x^2 + \cdots$$

- The variable x can be expressed as a binary number (we assume that the variable is bounded from above, such that only a finite number of powers of 2 are needed):

$$x = s_1 2^0 + s_2 2^1 + s_3\ 2^2 + \cdots + s_N 2^{N-1}$$

- Written in this way, the number x can be expressed in terms of a set of N binary variables, $\{s_0, s_1, \dots, s_{N-1}\}$, which can be either 0 or 1. This is of course assuming that $0 < x < 2^{N-1}$.
- After Taylor-expanding the function $f(x)$ in terms of those binary variables we get:

$$f(x) = f(s_1, s_2, \dots, s_N) = \sum_j h_j s_j + \sum_{j,k} J^{(2)}_{jk} s_j s_k + \sum_{j,k,l} J^{(3)}_{jkl} s_j s_k s_l + \cdots$$

Quantum annealing

- What if this was a quantum system?

- Imagine we consider that this is a Hamiltonian with additional terms (fields). Consider a quadratic binary problem for simplicity. We write it in terms of spin variables (+1,-1) instead of binary numbers (0, 1):

$$H = \underbrace{\sum_{j,k} J_{jk}^{(2)} \sigma_j^z \sigma_k^z}_{\text{Z-Pauli matrices (eigenvalues are +1 or -1)}} + h \sum_j \sigma_j^x \longrightarrow \text{X-Pauli matrices (induce quantum transitions between +1 and -1)}$$

- The idea is that we update the initial classical problem into a quantum problem. Before, we had classical bits, but now we have quantum bits, which can be in any linear superposition of +1 or –1.

- The minimum energy state of the Hamiltonian depends on the relative value of the parameters $J_{jk}^{(2)}$ and h:

$$H_z = \sum_{j,k} J_{jk}^{(2)} \sigma_j^z \sigma_k^z \longrightarrow \left|\Psi^{\mathrm{J}}_{\text{Ground State}}\right\rangle = \uparrow \downarrow \uparrow \downarrow \uparrow \uparrow \quad \text{unkown}$$

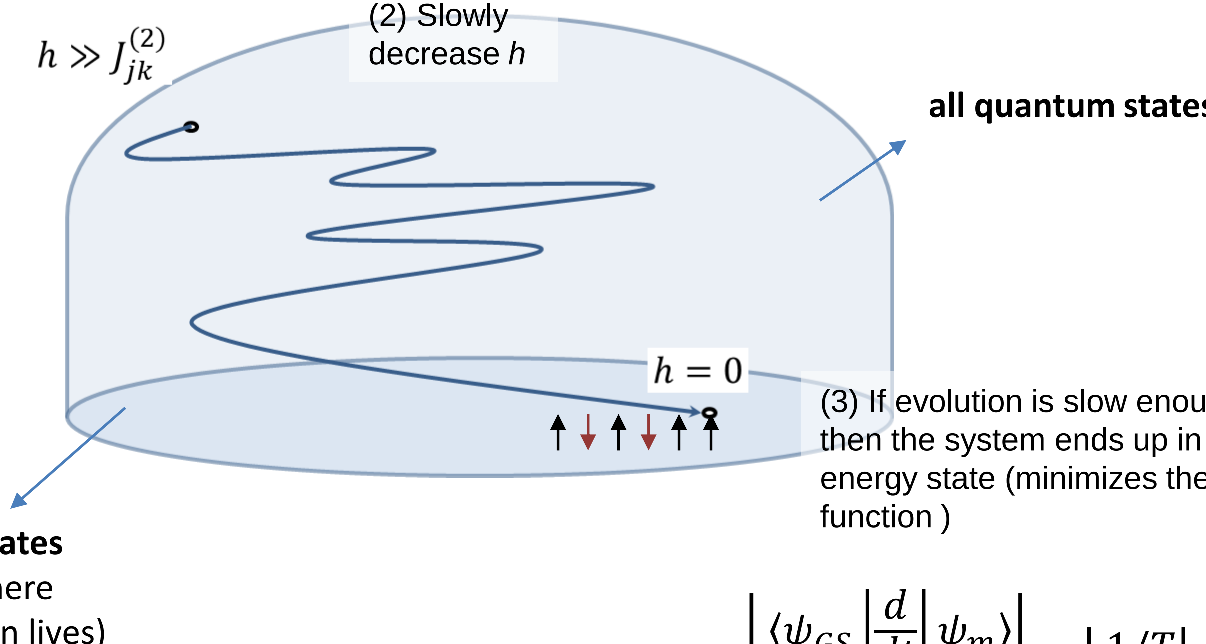
$$H_x = h \sum_j \sigma_j^x \longrightarrow \left|\Psi^{\mathrm{h}}_{\mathrm{GS}}\right\rangle = \frac{1}{\sqrt{2^N}} \underbrace{(|\uparrow\rangle_1 + |\downarrow\rangle_1)(|\uparrow\rangle_2 + |\downarrow\rangle_2) \ldots}_{\text{Ground state is a product of linear superpositons}}$$

Quantum annealing

- To carry out quantum annealing we assume that we can control in time at least the field $h(t)$,

$$H = \sum_{j,k} J_{jk}^{(2)} \sigma_j^z \sigma_k^z + h(t) \sum_j \sigma_j^x$$

(1) Start with high field (ground state is known)

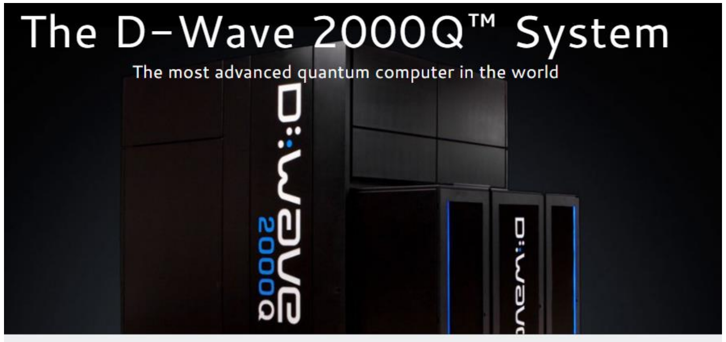


(3) If evolution is slow enough (adiabatic), then the system ends up in the mínimum energy state (minimizes the objective function)

$$\left| \frac{\langle \psi_{GS} \left| \frac{d}{dt} \right| \psi_m \rangle}{\Delta} \right| \approx \left| \frac{1/T}{\Delta} \right| \ll 1$$

Quantum annealing

- Early (theoretical) ideas by T. Kadowaki and H. Nishimori,Phys. Rev. E 58, 5355 (1998)
- The first commercial quantum devices (since 2008) by Canadian company D-Wave
- Today → Newest model D-Wave 2000Q has over 2000 qubits.



- The principles or even the quantum effects in the machine are controversial (high temperatures, noise..)
- Clients:
  - NASA
  - Google
  - Lockheed Martin
  - Los Alamos National Laboratory
  - Oak Ridge National Laboratory
  - Volkswagen

The D-Wave machine is not an universal quantum computer, but it actually can only solve binary optimization problems

Quantum annealing

- Qubit connectivity is very limited

- To minimize a general objective function requires some “tricks”

$$f(s_1, s_2, \dots, s_N) = \sum_j h_j s_j + \sum_{j,k} J^{(2)}_{jk} s_j s_k + \sum_{j,k,l} J^{(3)}_{jkl} s_j s_k s_l + \cdots$$

... which in the end require some qubits been used as “auxiliary” qubits to mediate interactions

- **How to translate a general optimization problem into D-Wave devices is still work in progress ...**

Chimera Graph

The Chimera architecture comprises sets of connected *unit cells*, each with four horizontal qubits connected to four vertical qubits via couplers. Unit cells are tiled vertically and horizontally with adjacent qubits connected, creating a lattice of sparsely connected qubits. See Figure 12.

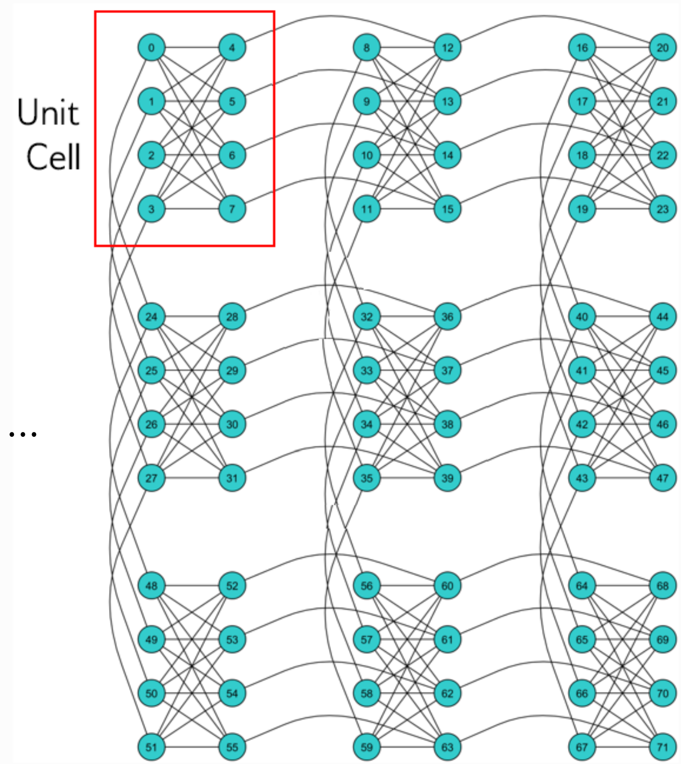


Fig. 12 A **3x3** *Chimera graph, denoted C3. Qubits are arranged in 9 unit cells.*

Documentation from https://www.dwavesys.com

Quantum Approximate Optimization Algorithm (QAOA)

- A very useful theoretical insight on the structure of the quantum annealing ansatz can be obtained by explicitly writing the form of the wave function:

$$|\Psi(\mathrm{t_f})\rangle = T e^{-i\int_0^{t_f} H(t')dt'} \prod_j |+\rangle_j$$

Evolution operator (dictates how the quantum state evolves)

Intial state (ground state of the Bx-field Hamiltonian)

$$H(t) = \sum_{j,k} J_{jk}^{(2)} \sigma_j^z \sigma_k^z + h(t) \sum_j \sigma_j^x \quad \begin{cases} h(0) \gg J_{jk}^{(2)} \\ h(t_f) = 0 \end{cases}$$

Quantum Approximate Optimization Algorithm (QAOA)

- If we Trotterize the time-evolution of the wave-function of a quantum annealing algorithm, we get the general form

$$|\Psi(\mathrm{t_f})\rangle = Te^{-i\int_0^{t_f} H(t')dt'} \prod_j |+\rangle_j$$

$$|\Psi\rangle = e^{-i\,\beta_n\,H_x}\, e^{-i\,\gamma_n\,H_{Ising}} \ldots e^{-i\,\beta_1\,H_x}\, e^{-i\,\gamma_1\,H_{Ising}} \prod_j |+\rangle_j$$

$$H_x = \sum_j \sigma_j^x$$

$$H_{Ising} = \sum_{jl} \tilde{J}_{jl}\sigma_j^z\sigma_l^z$$

$$\beta_k = h(t_k)\,\Delta t$$

$$\gamma_k = J(t_k)\,\Delta t$$

Quantum Approximate Optimization Algorithm (QAOA)

- However, one could just consider the parameters $\{\beta_k, \gamma_k\}$ as variational parameters

$$|\Psi_{\mathrm{QAOA}}(\vec{\beta},\vec{\gamma}\,)\rangle = e^{-i\,\beta_n\,H_x}\,e^{-i\,\gamma_n\,H_{Ising}} \ldots e^{-i\,\beta_1\,H_x}\,e^{-i\,\gamma_1\,H_{Ising}} \prod_j |+\rangle_j$$

$$\Downarrow$$

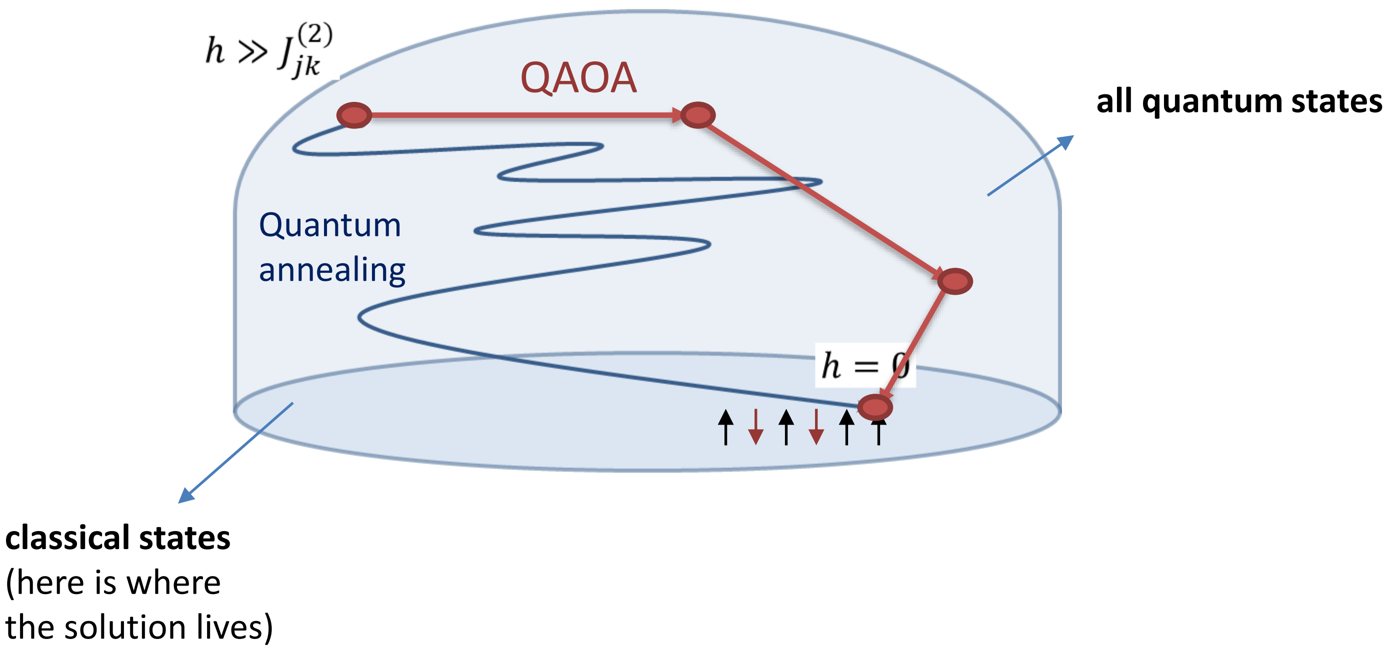
- QAOA:
    - Use quantum device to prepare the wavefunction $|\Psi(\beta,\gamma)\rangle$
    - Measure the expectation value $E_g(\vec{\beta},\vec{\gamma}) = \langle\Psi(\vec{\beta},\vec{\gamma})|H_{Ising}\,|\Psi(\vec{\beta},\vec{\gamma})\,\rangle$
    - Use a classical optimization algorithm to find the values $\vec{\beta}_{min}, \vec{\gamma}_{min}$ that minimize $E_g(\vec{\beta},\vec{\gamma})$
    - $|\Psi(\vec{\beta}_{min},\vec{\gamma}_{min})\rangle$ must be close to the optimal solution with large enough quantum circuit depth (actually it converges to the quantum annealing solution)

A Quantum Approximate Optimization Algorithm
Edward Farhi, Jeffrey Goldstone, Sam GutmannFahri, arXiv:1411.4028

Quantum Approximate Optimization Algorithm (QAOA)

- If we Trotterize the time-evolution of the wave-function of a quantum annealing algorithm, we get the general form

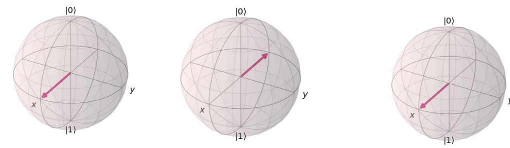
$$\left|\Psi_{\mathrm{QAOA}}(\vec{\beta},\vec{\gamma}\,)\right\rangle = e^{-i\,\beta_n\,H_x}\, e^{-i\,\gamma_n\,H_{Ising}} \ldots e^{-i\,\beta_1\,H_x}\, e^{-i\,\gamma_1\,H_{Ising}} \prod_j |+\rangle_j$$



Variational Quantum Eigensolver (VQE)

- Variational Quantum Eigensolver: this is another variational anstatz, however here the objective function to be minimized is not the same as the one creating the wavefunction.

$$\left|\Psi_{\mathrm{VQE}}(\theta_1, \dots \theta_n)\right\rangle = e^{-i\,\theta_1 \sigma_1^y} e^{-i\,\theta_2 \sigma_2^y} \dots e^{-i\,\theta_n \sigma_n^y} \prod_j |0\rangle_j$$



Variational parameters are now angles that determine a local rotation

$|\psi_1\rangle$ $|\psi_2\rangle$ ... $|\psi_n\rangle$

- VQE can be used to search for the minimum of a classical, QUBO problem: same idea as QAOA, only the ansatz changes

$$E_g(\vec{\theta}\,) = \langle \Psi_{VQE}(\vec{\theta}) | H_{Ising} \, | \Psi_{VQE}(\vec{\theta}) \rangle$$

The theory of variational hybrid quantum-classical algorithms
Jarrod R McClean, Jonathan Romero, Ryan Babbush and Alán Aspuru-Guzik
New Journal of Physics, Volume 18, 023023 (2016)

Variational Quantum Eigensolver (VQE)

- (entangled) Variational Quantum Eigensolver: the "search dynamics" change if we consider now entangling gates that coupled different qubits

$$\left|\Psi_{\mathrm{VQE}}(\vec{\theta}^{(1)}, \vec{\theta}^{(2)})\right\rangle$$
$$= \underbrace{\left(e^{-i\,\theta_1^{(2)}\sigma_1^y} e^{-i\,\theta_2^{(2)}\sigma_2^y} \dots e^{-i\,\theta_n^{(2)}\sigma_n^y} U_{ent}\right)}_{\text{second layer}} \underbrace{e^{-i\,\theta_1^{(1)}\sigma_1^y} e^{-i\,\theta_2^{(1)}\sigma_2^y} \dots e^{-i\,\theta_n^{(1)}\sigma_n^y}}_{\text{first layer}} \prod_j |0\rangle_j$$

Entangling gates: truly quantum search

$$U_{ent} = e^{-i\frac{\pi}{2}\sum_{j,l}\sigma_j^z\sigma_l^z}$$

(or other gates coupling all-to-all or nearest-neighbour qubits)

- This ansatz can be extended to p – layers (however, with increasing complexity)

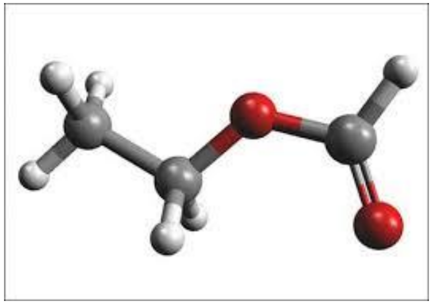
$$\left|\Psi_{\mathrm{VQE}}(\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \dots, \vec{\theta}^{(p)})\right\rangle$$

The theory of variational hybrid quantum-classical algorithms
Jarrod R McClean, Jonathan Romero, Ryan Babbush and Alán Aspuru-Guzik
New Journal of Physics, Volume 18, 023023 (2016)

Variational Quantum Eigensolver (VQE)

- The Variational Quantum Eigensolver can be actually be applied to more general problems than QUBO problems

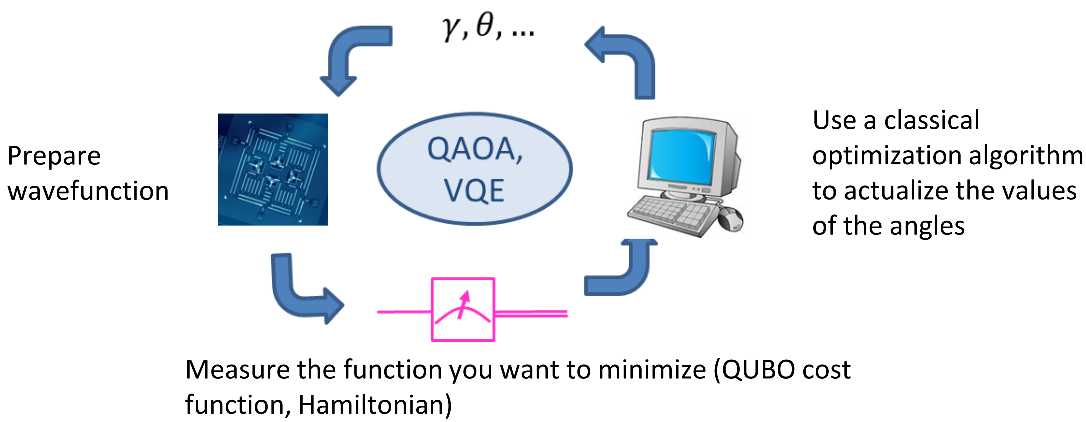
$$E_g\left(\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \ldots, \vec{\theta}^{(p)}\right) = \langle \Psi_{\mathrm{VQE}}(\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \ldots, \vec{\theta}^{(p)}) | H_{Molecule} | \Psi_{\mathrm{VQE}}(\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \ldots, \vec{\theta}^{(p)}) \rangle$$



- VQE:
  - Use quantum device to prepare the wavefunction $|\Psi_{\mathrm{VQE}}(\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \ldots, \vec{\theta}^{(p)})\rangle$
  - Measure the expectation value $E_g\left(\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \ldots, \vec{\theta}^{(p)}\right)$
  - Use a classical optimization algorithm to find the values $(\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \ldots, \vec{\theta}^{(p)})$that minimize $E_g\left(\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \ldots, \vec{\theta}^{(p)}\right)$
  - After convergence,$|\Psi_{\mathrm{VQE}}(\vec{\theta}^{(1)}, \vec{\theta}^{(2)}, \ldots, \vec{\theta}^{(p)})\rangle$, is an optimal description of the ground state

The theory of variational hybrid quantum-classical algorithms
Jarrod R McClean, Jonathan Romero, Ryan Babbush and Alán Aspuru-Guzik
New Journal of Physics, Volume 18, 023023 (2016)

Quantum Variational Algorithms



Summary:

- Variational quantum algorithms aim to solve optimization problems in the classical world (QAOA, VQE) or in the quantum world, like finding the grounds state of a molecule (VQE)

- These methods are under intense investigation: it is yet not fully clear under what conditions they can beat classical numerical methods

- Both QAOA and VQE rely on the intuition that a quantum search can more efficiently explore the space of possible solutions.

Quantum Variational Algorithms

Summer School on Quantum Computing: Software for Near Term Quantum Devices

	Mon 31 Aug	Tue 1 Sep	Wed 2 Sep	Thu 3 Sep	Fri 4 Sep
9:30-11:30	Germán Sierra Introduction to quantum computing I	Juan Sánchez Toural Tutorial QISKIT	Maria Schuld Quantum machine learning	Jens Eisert Quantum advantages and near-term quantum computing	Román Orús Applications of quantum computing in finance
12:00-14:00	Diego Porras Introduction to quantum computing II	Ivano Tavernelli Quantum algorithms for applications in quantum chemistry and physics	Juan José García Ripoll Applied mathematics with quantum computers + General questions	Pol Forn Introduction to Experimental Quantum Computation	Practical Session (D. Porras – Juan José García-Ripoll)
15:30-17:30	Ginés Carrascal de las Heras (IBM) Tutorial QISKIT	Stefan Woerner Variational quantum computing for classical optimization problems	Practical Session (D. Porras – J. J García-Ripoll)		

Applications of VQE/QAOA (with some review of the theory): implementations in Qiskit

Variational Quantum Computing in Finance/Quantum Chemistry

A few proposed projects on quantum annealing/QAOA/VQE with Qiskit:

- quantum_annealing_questions
- Portfolio opitmization
- Quantum chemistry tutorials