

Quantum computing and quantum machine learning: Introduction

PD Dr. habil. Jeanette Miriam Lorenz
Fraunhofer Institute for Cognitive Systems IKS
&
LMU Munich
09.08.2023



Fraunhofer Institute for Cognitive
Systems IKS



LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN



[<https://www.umb.ch/blog/news/detail/quantencomputing-fuer-den-praktischen-einsatz-rueckt-naeher>]

Airbus Quantum Computing Challenge

Bringing flight physics into the Quantum Era

ZEIT ONLINE

Suche

Politik Gesellschaft Wirtschaft Kultur • Wissen Gesundheit • **Digital** Campus • Arbeit Sport ZEITmagazin • mehr •

Technologie

Bund investiert zwei Milliarden Euro für Quantencomputer

Innerhalb von fünf Jahren soll in Deutschland ein Quantencomputer entstehen. Mithilfe von Qubits erzielt die Technologie weit höhere Leistungen als herkömmliche Rechner.

11. Mai 2021, 11:29 Uhr / Quelle: ZEIT ONLINE, dpa, [kzi](#) / [🔖](#)

QUANTUM
FLAGSHIP

Discover

The future is Quantum.

The Second Quantum Revolution is unfolding now, exploiting the enormous advancements in our ability to detect and manipulate single quantum objects. The Quantum Flagship is driving this revolution in Europe.

[LEARN MORE](#)



Computertechnologie

Ein Quantensprung für Deutschland?

Stand: 15.06.2021 18:25 Uhr

In Ehningen bei Stuttgart wurde Europas erster Quantencomputer eingeweiht. Der ultraschnelle Rechner der Firma IBM soll der Wirtschaft helfen, im Wettstreit mit China und den USA zu bestehen.

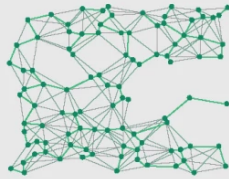
Von Michael Herr, SWR

Das "Wunderwerk der Technologie", wie Angela Merkel es nannte, ist wichtig: Immerhin eine Grundfläche von drei mal drei Metern bei einer Höhe von drei Metern soll der Quantencomputer haben, den IBM zusammen mit der Fraunhofer-Gesellschaft am Dienstag in der Deutschland-Zentrale des IT-Konzerns in Ehningen bei Stuttgart der Öffentlichkeit präsentiert hat. Der laut IBM "leistungsstärkste Quantencomputer im industriellen Umfeld".

Application fields

EXHIBIT 1 | Quantum-Advantaged Computational Problems

Type of problem



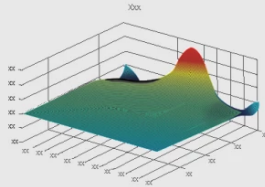
Combinatorial optimization

Useful for...

Minimizing or maximizing an objective function, such as finding the most efficient allocation of resources or the shortest total distance among a set of points (e.g., the traveling salesman problem)

Industry applications include...

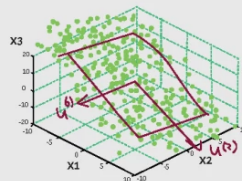
- Network optimization (e.g., for airlines, taxis)
- Supply chain and logistics optimization
- Portfolio optimization in financial services



Differential equations

Modeling the behavior of complex systems involving fundamental laws of physics (e.g., Navier Stokes for fluid dynamics and chemistry)

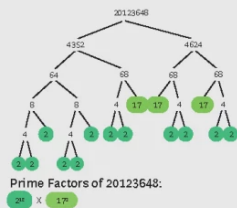
- Fluid dynamics simulations for automotive and aeronautical design and medical devices (e.g., blood flow analysis)
- Molecular simulation for specialty materials design and drug discovery



Linear algebra

Machine learning tasks involving matrix diagonalization, such as clustering, pattern matching, and principal components analysis, as well as support vector machines, which are ubiquitous in applications across industries

- Risk management in quantitative finance
- DNA sequence classification
- Marketing and customer segmentation



Factorization

Cryptography and computer security, where the most common protocols today (e.g., RSA) rely on the infeasibility (for classical computers) of factoring the product of two large prime numbers

- Decryption and code breaking (e.g., for governments)

Source: BCG analysis.

[<https://www.bcg.com/de-de/publications/2019/quantum-computers-create-value-when>]

Emerging technology:

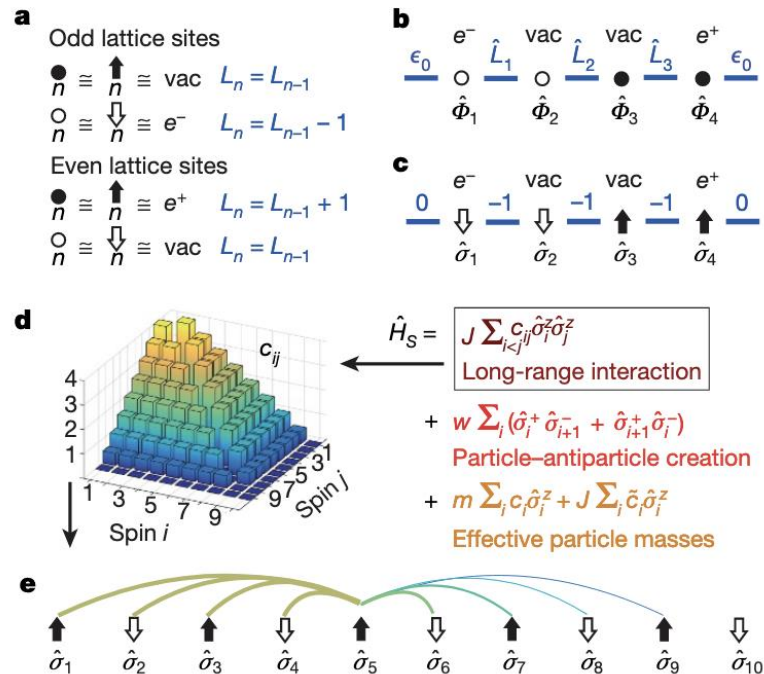
Quantum computing

Quantum computing (QC) has the potential to lead to disruptive changes in many industrial areas:

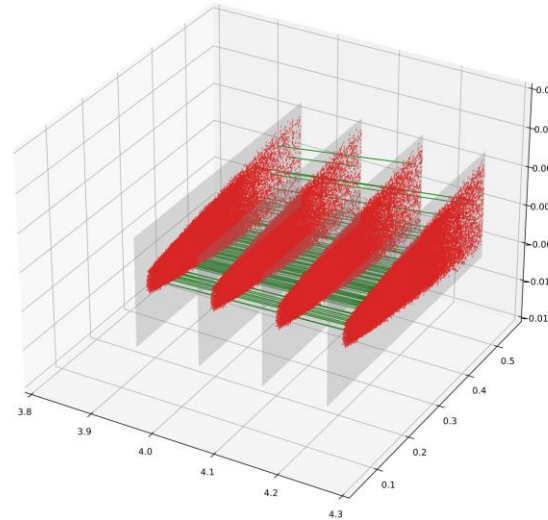
- **Simulation** of quantum mechanical systems
 - (Development of new drugs, chemical sector with battery development,...)
- **Optimization** problems (Logistics, production, pharma,...)
- **Quantum machine learning** (Computer vision, mobility,...)



Applications in particle physics

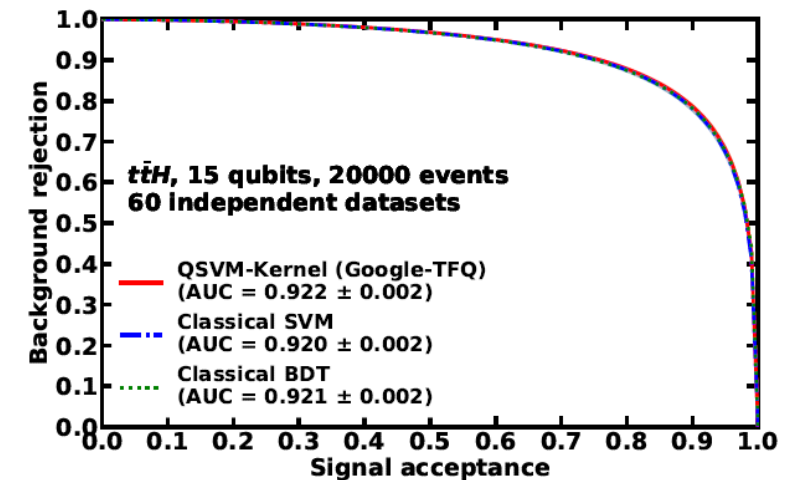


Lattice QCD
 – fundamental interactions in discretized space



Reconstruction of tracks
 - Optimization problem or
 quantum machine learning

Event selection
 - solved by quantum
 machine learning



Is it all just a hype?

„We help companies make better decisions in less time with quantum-hybrid computing“

„Quantum computing has the potential to drive the major breakthroughs needed to help solve the climate crisis.“

„Much like artificial intelligence in its early days, the reputation of quantum computing has been tarnished by grand promises and few concrete results. Talk of quantum computers is often closely flanked by promises of polynomial-time solutions to NP-Hard problems and other such implausible appeals to blind optimism..“

What quantum computing might be able to do and what not:

- **QC will not replace classical computers.**
- QC are expected to lead to exponential or polynomial speed-up for certain calculations (or more precisely subroutines).
- **Academic** quantum advantage has been claimed.
- **A practical quantum advantage has not been shown yet.**
- A claim on **quantum utility** has been made recently.

Plan for the lectures

1. Intro to quantum computing (Jeanette)
2. Fault-tolerant quantum computing (Federico)
3. Introduction to NISQ quantum computing (Jeanette)
4. Quantum computing for optimization problems (Federico)
5. Quantum machine learning (Jeanette)

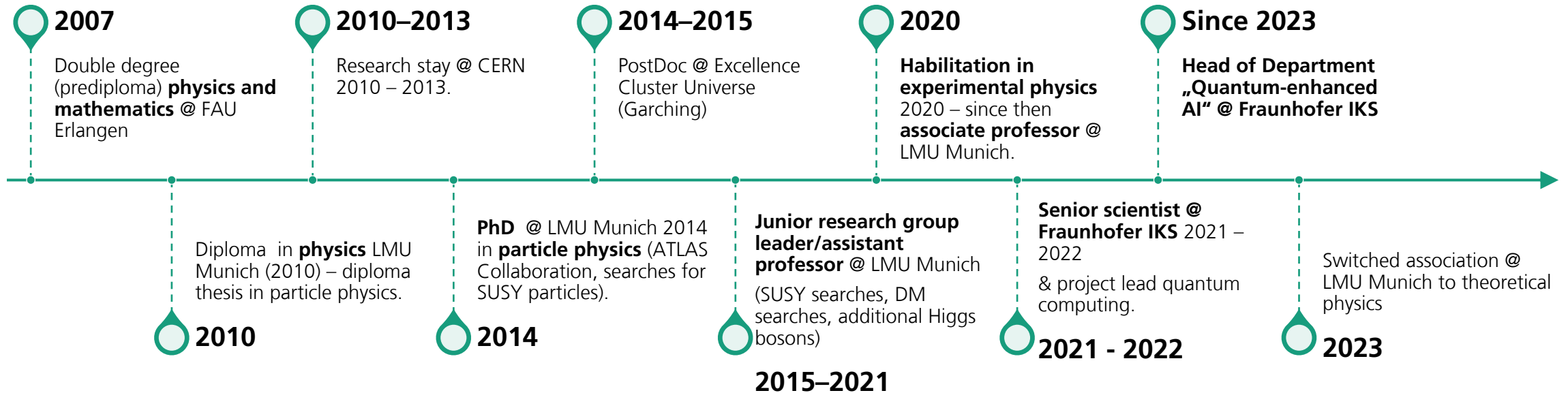
Lectures interleaved with tutorials
(typically first lecture, then tutorial)



[<https://www.umb.ch/blog/news/detail/quantencomputing-fuer>]

Jeanette

Searches for SUSY + DM, additional Higgs bosons @ ATLAS -> Quantum computing @ Fraunhofer/LMU



Introduction to quantum computing

1. What is quantum computing?
2. Basics:
 - a) History
 - b) Current research questions in quantum computing
 - c) Definitions
 - d) What is a qubit?
3. One and multi-qubit gates, simple algorithms
4. Computational complexity



[<https://www.umb.ch/blog/news/detail/quantencomputing-fuer>]



because nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical, and by golly it's a wonderful problem, because it doesn't look so easy.

Richard Feynman, Simulating Physics with Computers, International Journal of Theoretical Physics 21, 467 (1982)

History of QC

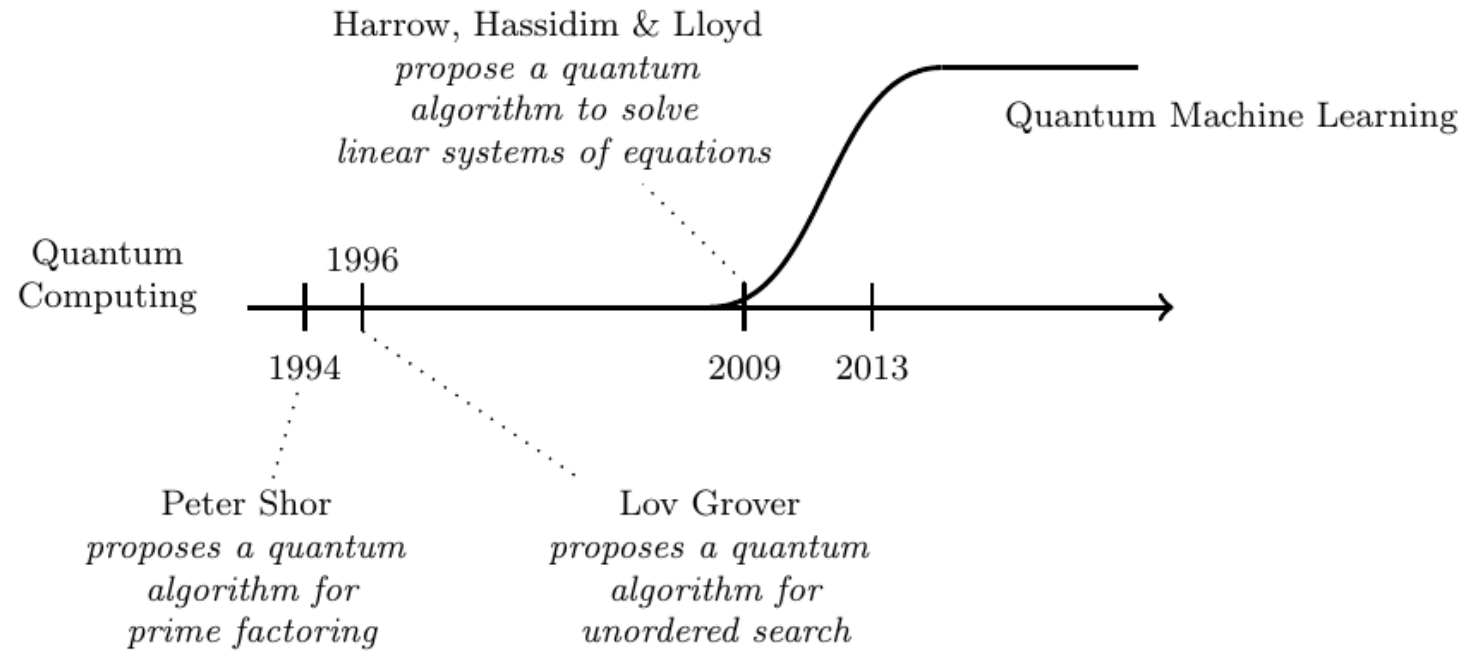


Fig. 3.3 Timeline of quantum computing and quantum machine learning

[M. Schuld & F. Petruccione, Machine Learning with Quantum Computers, Springer 2021]

Research questions

- What is quantum information?
 - Can we build a computer based on quantum systems?
 - How can we then formulate algorithms on such systems?
 - What does quantum theory mean for the limits of what is computable?
 - What distinguishes quantum computers from classical ones?
- With first small and noisy quantum computers starting to be available: for what can we use them? (Now and in perspective)
 - How can we control their imperfection?
 - How to organise the interplay between classical and quantum computers?

**Research topics of
Jeanette's department**

A few basic definitions

Qubit: A quantum system associated with two measurable states

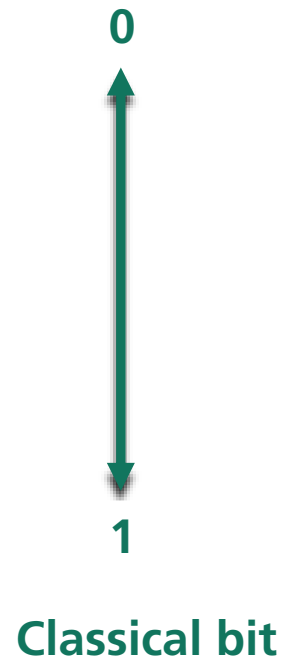
Quantum computer (QC): Physical implementation of n qubits with precise control on the evolution of the system

Quantum algorithm (QAlg): Controlled manipulations of a quantum system with subsequent measurement to retrieve information from the system

Quantum gates: Manipulation that act on one or two, ... qubits

A quantum algorithm can be formulated as **quantum circuits** of elementary gates

In a nutshell: What is quantum computing?



A **classical bit** can be either 0 or 1.

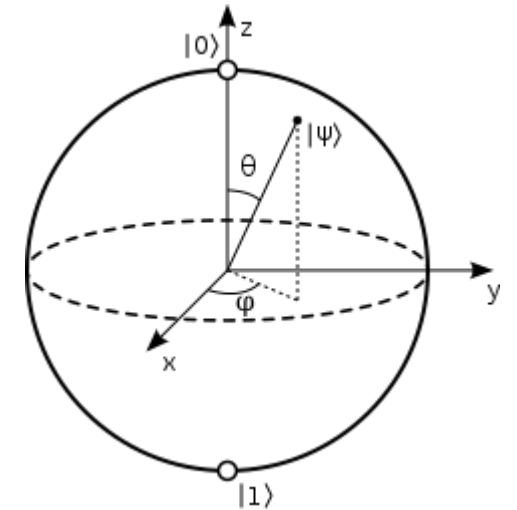
A **Quantum Bit** ($\hat{=}$ Qubit) is the superposition of two states $|0\rangle$ und $|1\rangle$ **at the same time**:

$$a |0\rangle + b |1\rangle$$

with $|a|^2 + |b|^2 = 1$

But: When **measuring** the qubit, only the classical states 0 and 1 can be measured, i.e. 0 is measured with a probability of $|a|^2$ and 1 with a probability of $|b|^2$.

A quantum computer returns probabilistic results.



Qubit: Bloch sphere

[Wikimedia](#)

The formal mathematical definition

A **qubit** is the fundamental unit of quantum information.

At any given time, it is in a **superposition state** represented by a linear combination of Dirac vectors $|0\rangle$ and $|1\rangle$ in \mathbb{C}^2 :

$$|\psi\rangle = a|0\rangle + b|1\rangle \text{ where } |a|^2 + |b|^2 = 1$$

And $a, b \in \mathbb{C}$

The vectors $|0\rangle$ and $|1\rangle$ form an orthonormal basis of a two-dimensional Hilbert space -> **Computational basis**.

a and b are called **probability amplitudes**.

$|0\rangle$ and $|1\rangle$ can be represented as the standard basis states of \mathbb{C}^2 :

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \in \mathbb{C}^2, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \in \mathbb{C}^2$$

A qubit in polar form

We can rewrite a qubit in polar form:

$$|\psi\rangle = a|0\rangle + b|1\rangle = r_1 e^{i\varphi_1} |0\rangle + r_2 e^{i\varphi_2} |1\rangle$$

Furthermore, we can identify two quantum states if they only differ by a multiple of a complex unit, i.e. by a factor $e^{i\varphi}$ for $0 \leq \varphi < 2\pi$.

That means the above qubit is effectively the same as:

$$|\psi\rangle = r_1 |0\rangle + r_2 e^{(\varphi_2 - \varphi_1)i} |1\rangle$$

r_1 and r_2 are in \mathbb{R} and $r_1^2 + r_2^2 = 1$.

Mapping to the Bloch sphere

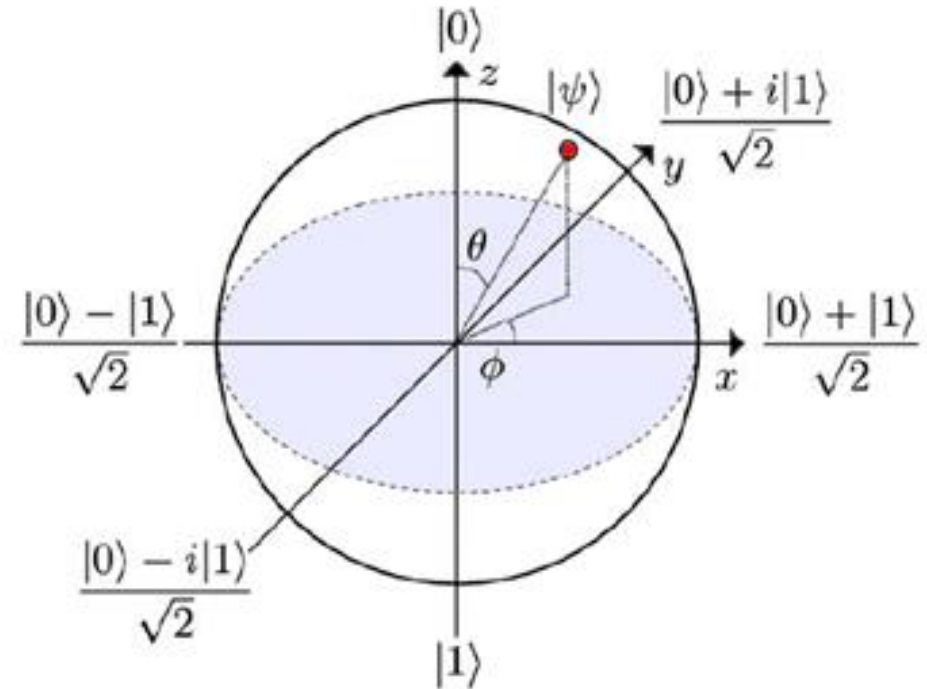
We can find $0 \leq \theta \leq \pi$ so that

$$r_1 = \cos \frac{\theta}{2} \text{ and } r_2 = \sin \frac{\theta}{2}$$

With this:

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

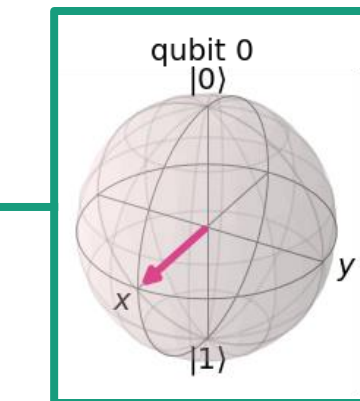
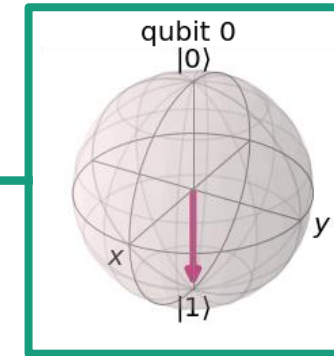
Using this, we can use a non-linear projection to get from the three-dimensional surface of the hypersphere in \mathbb{C}^2 (that one can think of as \mathbb{R}^4) to a two-dimensional surface of a Bloch sphere in \mathbb{R}^2 . The main point why this does work is because we can ignore global phases.



Basic operations acting on one qubit

Table 3.3 Some useful single-qubit logic gates and their representations

Gate	Circuit representation	Matrix representation	Dirac representation
X	$\text{---}[X]\text{---}$	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$ 1\rangle\langle 0 + 0\rangle\langle 1 $
Y	$\text{---}[Y]\text{---}$	$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$i 1\rangle\langle 0 - i 0\rangle\langle 1 $
Z	$\text{---}[Z]\text{---}$	$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$ 1\rangle\langle 0 - 0\rangle\langle 1 $
H	$\text{---}[H]\text{---}$	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$\frac{1}{\sqrt{2}}(0\rangle + 1\rangle)\langle 0 + \frac{1}{\sqrt{2}}(0\rangle - 1\rangle)\langle 1 $
S	$\text{---}[S]\text{---}$	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	$\frac{1}{\sqrt{2}} 0\rangle\langle 0 + \frac{1}{\sqrt{2}}i 1\rangle\langle 1 $
R	$\text{---}[R]\text{---}$	$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & e^{(-i\pi/4)} \end{pmatrix}$	$\frac{1}{\sqrt{2}} 0\rangle\langle 0 + \frac{1}{\sqrt{2}}e^{(-i\pi/4)} 1\rangle\langle 1 $



[M. Schuld et al, Machine Learning with Quantum Computers, Springer 2021]

Two qubits

Orthonormal basis formed by two qubits:

$$|\psi\rangle_1 = a_1 |0\rangle_1 + b_1 |1\rangle_1 \quad \text{where } |a_1|^2 + |b_1|^2 = 1$$

$$|\psi\rangle_2 = a_2 |0\rangle_2 + b_2 |1\rangle_2 \quad \text{where } |a_2|^2 + |b_2|^2 = 1$$

The orthonormal basis is then:

$$|0\rangle_1 \otimes |0\rangle_2, |0\rangle_1 \otimes |1\rangle_2, |1\rangle_1 \otimes |0\rangle_2, |1\rangle_1 \otimes |1\rangle_2$$

Or in short:

$$|00\rangle, |01\rangle, |10\rangle, |11\rangle$$

Example: Representation of numbers

To display numbers from 0 to 15 in bits, we need 4 bits:

0 0 0 0

0 0 0 1

0 0 1 0

0 0 1 1

0 1 0 0

...

1 1 1 1




Four qubits allow to represent all of these 16 states **at the same time**:

$|0000\rangle + |0001\rangle + |0010\rangle + |0011\rangle + |0100\rangle + |0101\rangle + |0110\rangle + |0111\rangle + |1000\rangle + |1001\rangle + |1010\rangle + |1011\rangle + |1100\rangle + |1101\rangle + |1110\rangle + |1111\rangle$

→ **Parallelisation of calculations possible.**

With n qubits 2^n states can be represented at the same time.

Basic operations acting on multiple qubits

Gate	Circuit representation	Matrix representation
CNOT		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
SWAP		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$
T		$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

[M. Schuld et al, Machine Learning with Quantum Computers, Springer 2021]

Multi-qubit gates are required to entangle qubits (and without entanglement some basic benefit of QC not available).

Definition:

A 2-qubit state in $\mathbb{C}^2 \otimes \mathbb{C}^2$ is called **entangled** if and only if it **cannot** be written as tensor product of two 1-qubit kets:

$$|\psi_1\rangle \otimes |\psi_2\rangle = (a_1|0\rangle_1 + b_1|1\rangle_1) \otimes (a_2|0\rangle_2 + b_2|1\rangle_2)$$

If a quantum state is not entangled, it is called **separable**.

Measurement of a two-qubit system

- Usually measurements in the computational basis
- Measurements are represented by projections onto the possible eigenspaces:
$$P_0 = |0\rangle\langle 0| \text{ or } P_1 = |1\rangle\langle 1|$$
- E.g., $p(0) = \text{tr}(P_0|\psi\rangle\langle\psi|) = \langle\psi|P_0|\psi\rangle = |a_1|^2$
- The full observable corresponding to the computational basis measurement is the Pauli-z observable:
$$\sigma_z = |0\rangle\langle 0| - |1\rangle\langle 1|$$
- With the eigenvalues:
$$\begin{aligned} &+1 \text{ for } |0\rangle \\ &-1 \text{ for } |1\rangle \end{aligned}$$

Measurement in practise

- How to obtain a expectation value? -> Sample. Run a QAlg s times. s is called the number of shots.
- How many shots are required to obtain an estimate $\langle \sigma_z \rangle$ with an error ε ?
-> Bernoulli experiment
- In case of large s and a probability of $p \approx 0.5$: Wald interval for $\langle \sigma_z \rangle = 0$:

$$\varepsilon = z \cdot \sqrt{\frac{\hat{p}(1 - \hat{p})}{s}}$$

With \hat{p} : estimator for the probability
 z : confidence level

- $O(\varepsilon^{-2})$ samples required for a given ε and z
- *Example: $\varepsilon = 0.1$ and $z = 0.99$: $s = 167$*

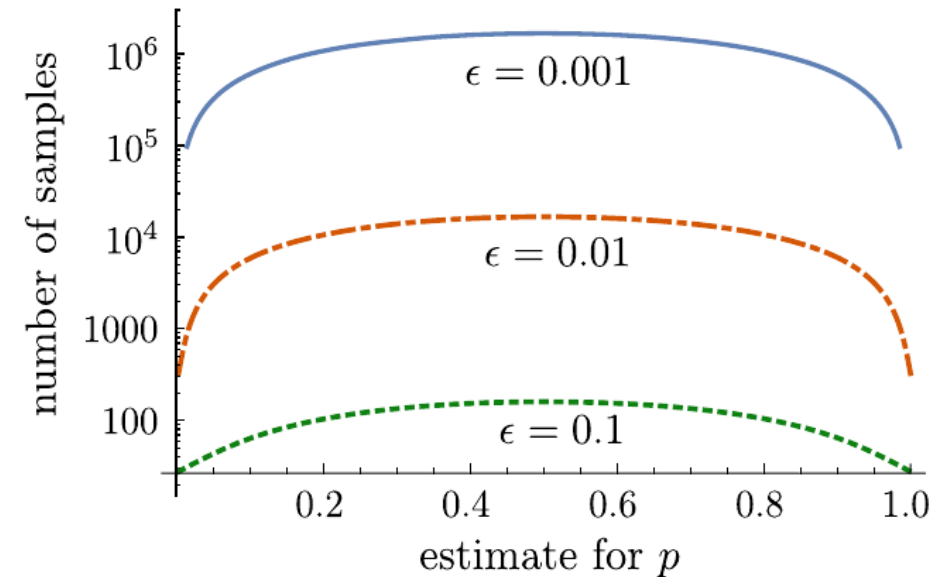
Measurement in practise

- If $\hat{p} \rightarrow 0$ or $\hat{p} \rightarrow 1$:

The Wald approximation is not valid, instead the Wilson score interval is required:

$$\varepsilon = \frac{z}{1 + \frac{z^2}{s}} \left(\frac{\hat{p}(1 - \hat{p})}{s} + \frac{z^2}{4s^2} \right)^{1/2}$$

=> For $\varepsilon = 0.1$ only 27 measurements are required for the same boundaries as before.



[M. Schuld et al, Machine Learning with Quantum Computers, Springer 2021]

Why is quantum computing promising?

Superposition of states

Entanglement of states – i.e. multiple qubits are connected/correlated.

Interference of qubits – i.e. states interfere → Enhancement or reduction of states (**see Grover!**)

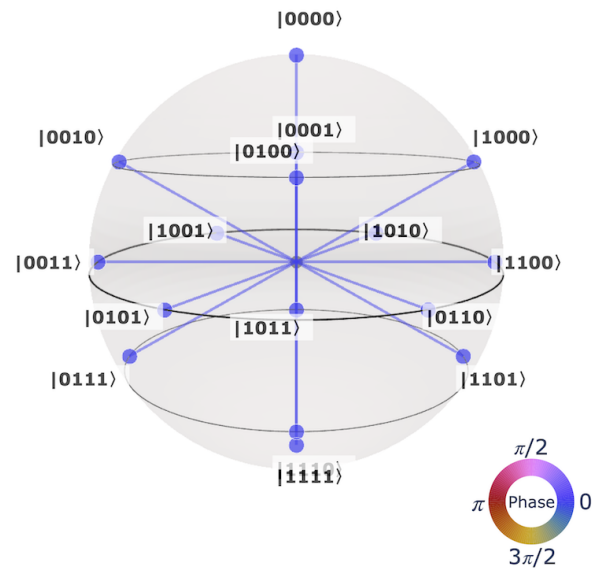
Quantum computing could potentially result in an **increased computing capacity** – thus leads to a more efficient solution of problems.

→ Simplified processing of complicated datasets, solution of **currently unsolvable** problems.

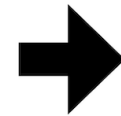
However: It is uncertain when the quality of quantum computers will be **sufficient** to fully profit from these advantages.

The flow of a quantum computation

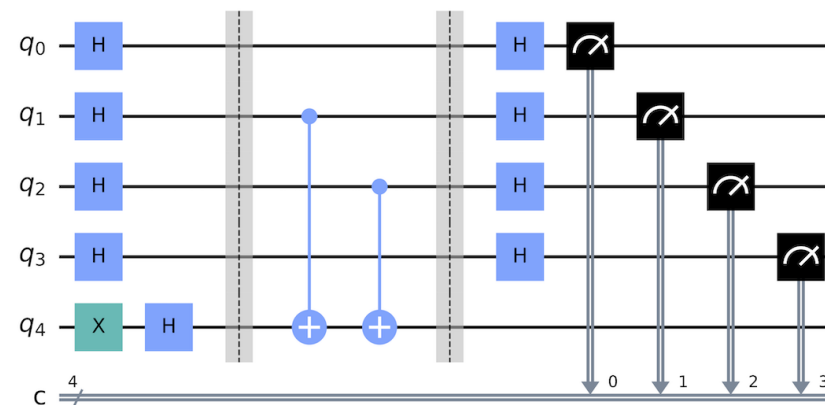
Simplified: Superposition/Entanglement + Interference -> Solution



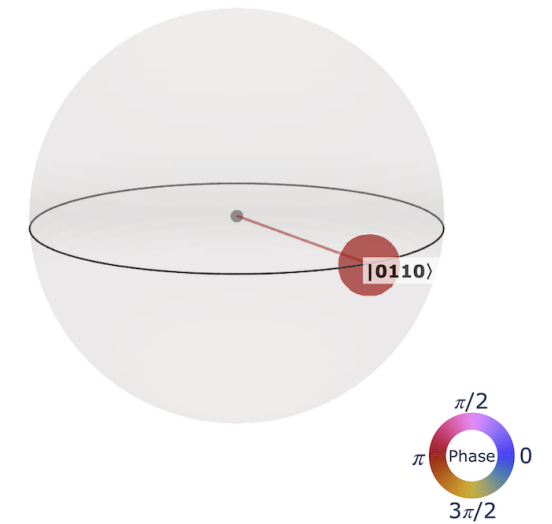
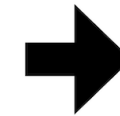
Superposition of
all possibilities



Quantum circuit



Computation driven interference



Solution

[https://qiskit.org/documentation/qc_intro.html]

What is a quantum computer?

Different criteria to assess quality of quantum computer + if it is a quantum computer:

- **Universality**
 - Universal quantum computer? -> **Di Vincenzo criteria**
- **Fidelity** (Quality of qubits)
- **Scalability** (Architecture scalable?)
- **Qubits**: number, architecture-specific limitations like nearest neighbor connections
- Logical **connectivity** (two-qubit gates possible for all qubits?)
- **Circuit depth** (How many operations possible?)
- **Cloud access** (and availability in general)

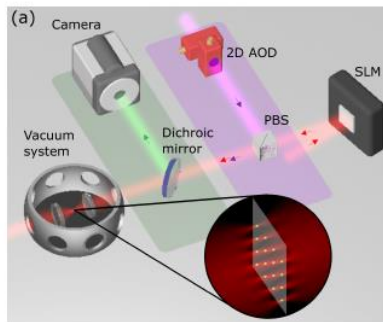
Di-Vincenzo criteria:

1. **Scalable** qubit system with distinct qubits?
2. Ability to **initialize** the state of any qubit to a definite state in the computational basis
3. The qubits must **hold** their states
4. Ability to **apply unitary operators to qubit states** and to two qubits at once
5. Ability for ‚strong‘ measurements, i.e. the **ability that the measurement measures the state of the qubit** for the property being measured

Different QC hardware concept (examples)

Neutral atoms

- Atom ensemble surrounded by laser system forming an magneto-optical trap, addressable arrays of atoms.
- Requires a specific way of programming (pulsar), good connectivity, more native to QUBO formulations?



L. Henriet et al, Quantum Computing with neutral atoms, arXiv:2006.12326v2

Superconducting qubits

- Superconducting Josephson junctions at cryogenic temperatures.
- Low connectivity, many SWAP operations needed for highly connected circuits, runtime environments in first attempts available

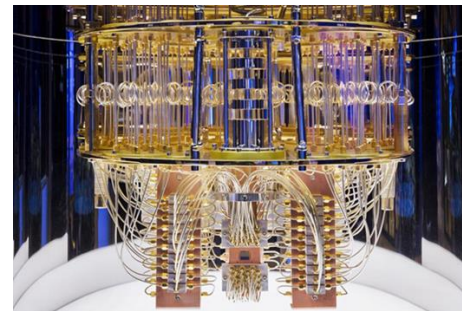


Image of the interior of IBM's Quantum Computer. Copyright IBM

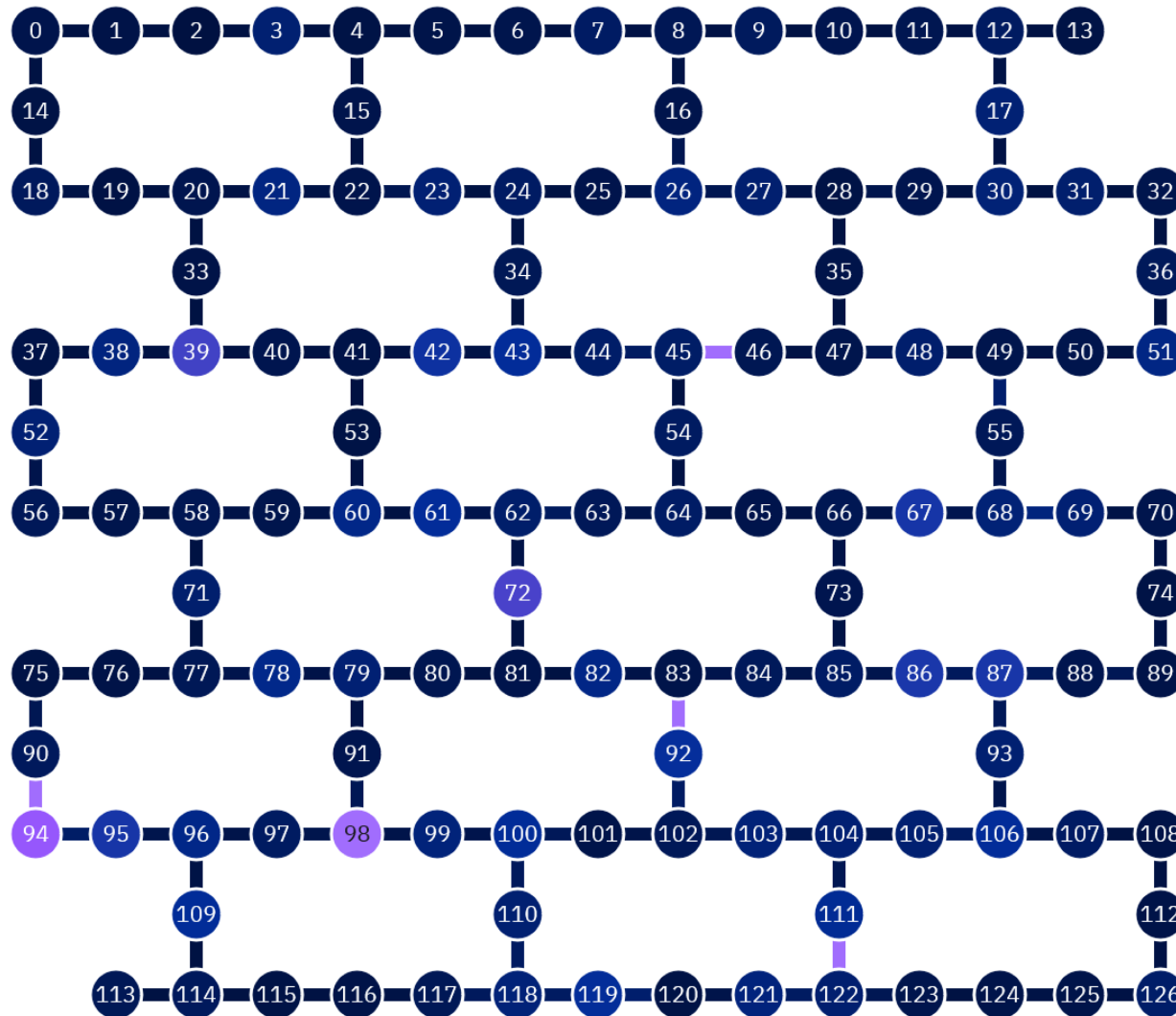
Trapped ions

- Ionized atoms trapped in electric potentials to form a line of qubits.
- Can be operated at room temperature, high connectivity, no runtime environment and running jobs relatively manual



Source: <https://www.aqt.eu/media-press/>, Dieter Kühl

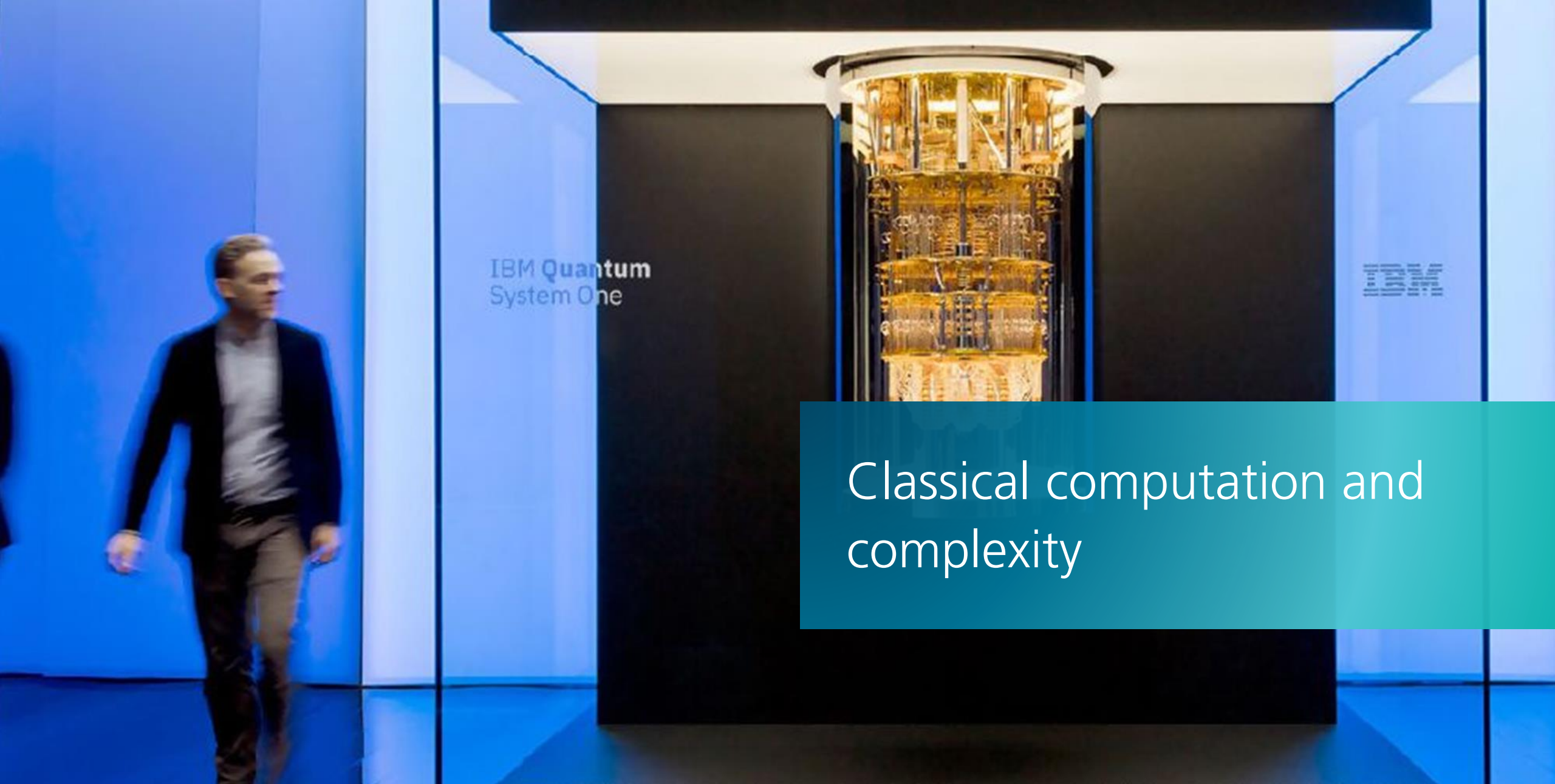
Example for a superconducting chip design



ibm_cusco

127 qubits

[Visit via the IBM Lab, e.g.,
<https://quantum-computing.ibm.com/services/resources>]



Classical computation and complexity

Classical computers

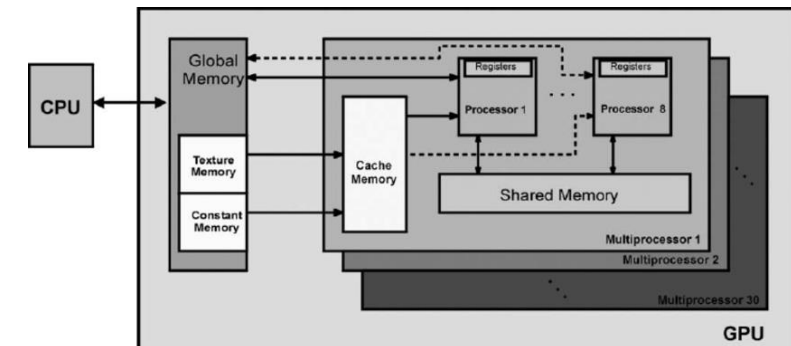
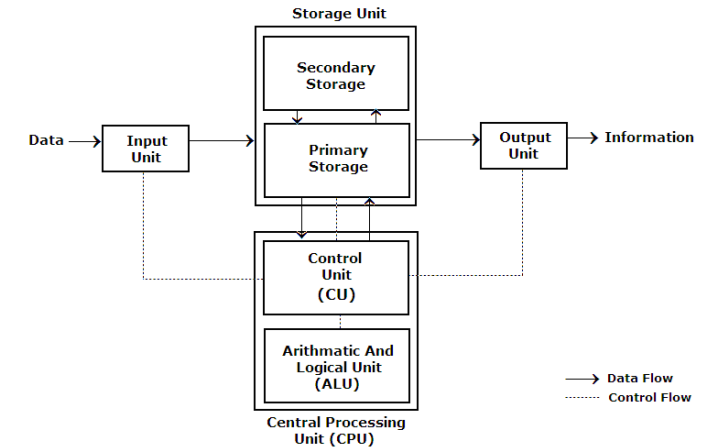
Components of a classical computer (simplified):

- **Processor** (Central Processing Unit – CPU)
- **Graphics Processing Unit** (GPU) for specialized calculations
- **Memory** for storing information during the calculation
- **Storage** for long-term preservation of data

Difference CPU and GPU:

- CPU good at performing different operations, but then maybe slow
- GPU good for a set of highly optimized operations – for these very fast (matrix multiplication)

Quantum computing to be included in these systems as
Quantum Processing Unit



[<https://kullabs.com/class-11/computer-science-1/computer-system/components-of-computer-system>, F. Massanes et al, Computer-unified device architecture implementation of a block-matching algorithm for multiple graphical processing unit cards]

Storage of information and operations

Classical storage of information in bits

- A bit is either 0 or 1
- At a low level a computer works with binary numbers
- Example: $1101 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$

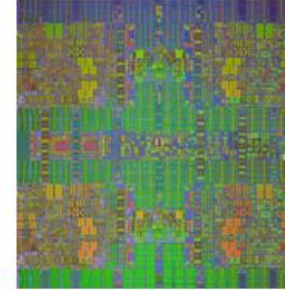
Capacity of storage:

- Measured in terabytes, gigabytes, ...
- 1 gigabyte = 1000 megabytes = ... = 10^9 bytes
- 1 byte $\hat{=}$ 8 bits

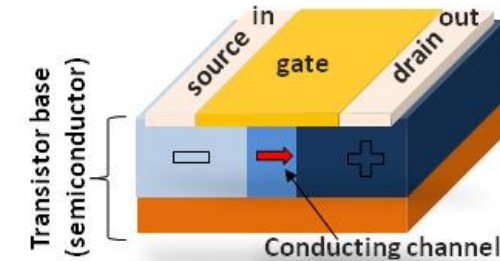
Operations by using transistors

A

Microprocessor Chip

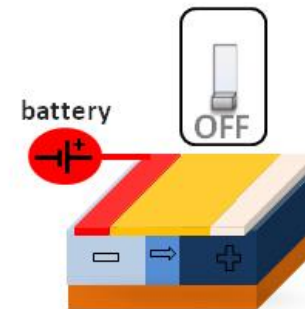


3D view of a transistor

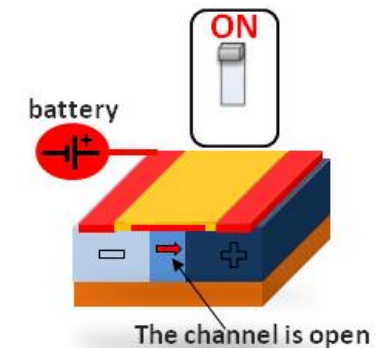


B

OFF state

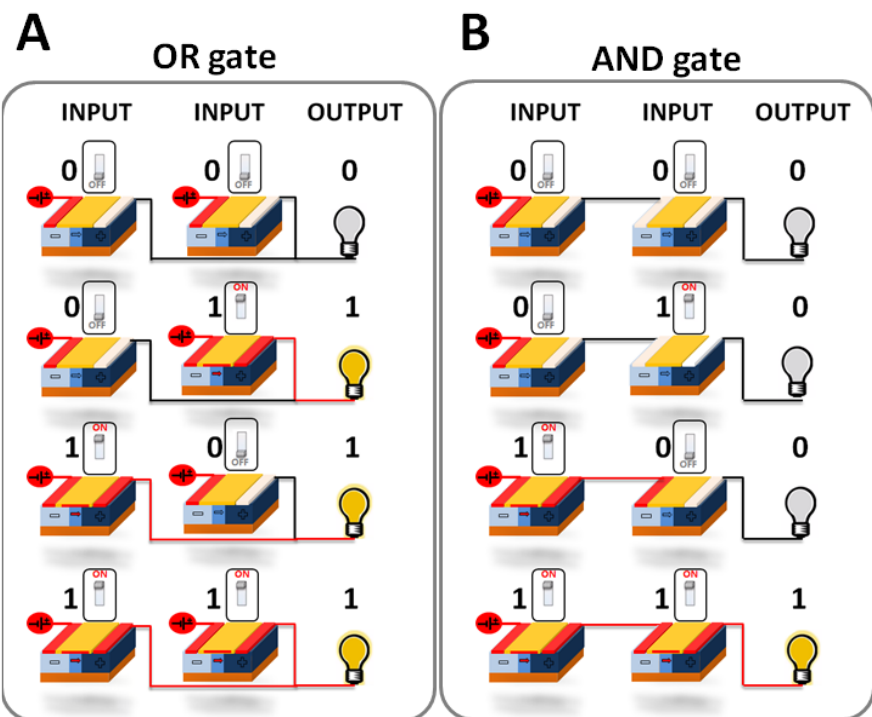


ON state



[Abels, Seth & Khisamutdinov, Emil. (2015). Nucleic Acid Computing and its Potential to Transform Silicon-Based Technology. DNA and RNA Nanotechnology. 2. 10.1515/rnan-2015-0003.]

Operations



YES

INPUT	OUTPUT
A	
0	0
1	1

NOT

INPUT	OUTPUT
A	
0	1
1	0

AND

INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

OR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

XOR

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

NAND

INPUT		OUTPUT
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

NOR

INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	0

XNOR

INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	1

[Abels, Seth & Khisamutdinov, Emil. (2015). Nucleic Acid Computing and its Potential to Transform Silicon-Based Technology. DNA and RNA Nanotechnology. 2. 10.1515/rnan-2015-0003.]

How to realize an addition

Task:

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 0 \text{ carry } 1$$

How to realize an addition

Task:

$$0 + 0 = 0$$


$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 0 \text{ carry } 1$$

Which logical gates are needed?

XOR



INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

Almost there!

How to realize an addition

Task:

$$0 + 0 = 0$$


$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 0 \text{ carry } 1$$

Which logical gates are needed?

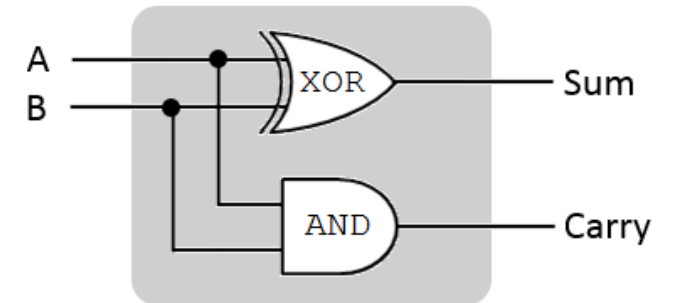
XOR



INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

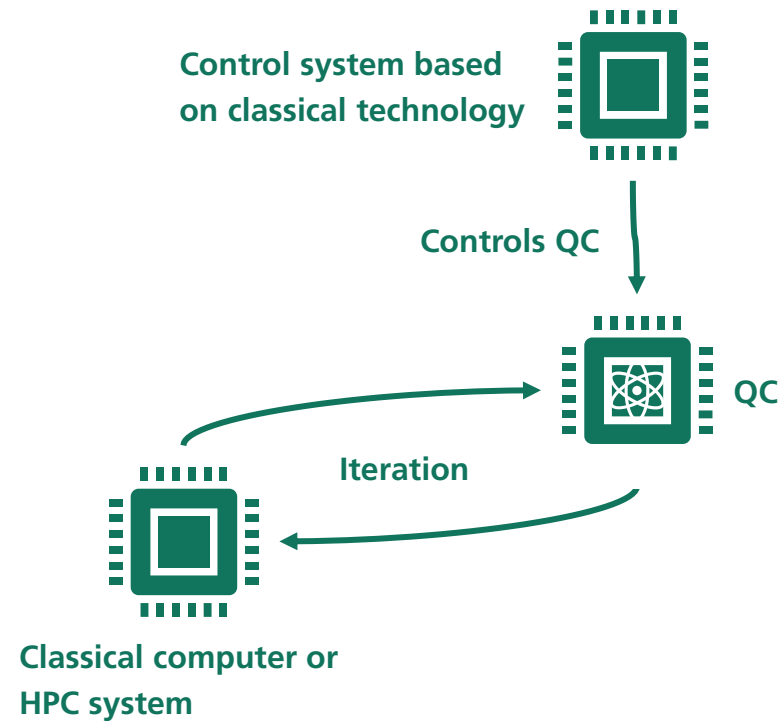
Almost there!

Half-Adder

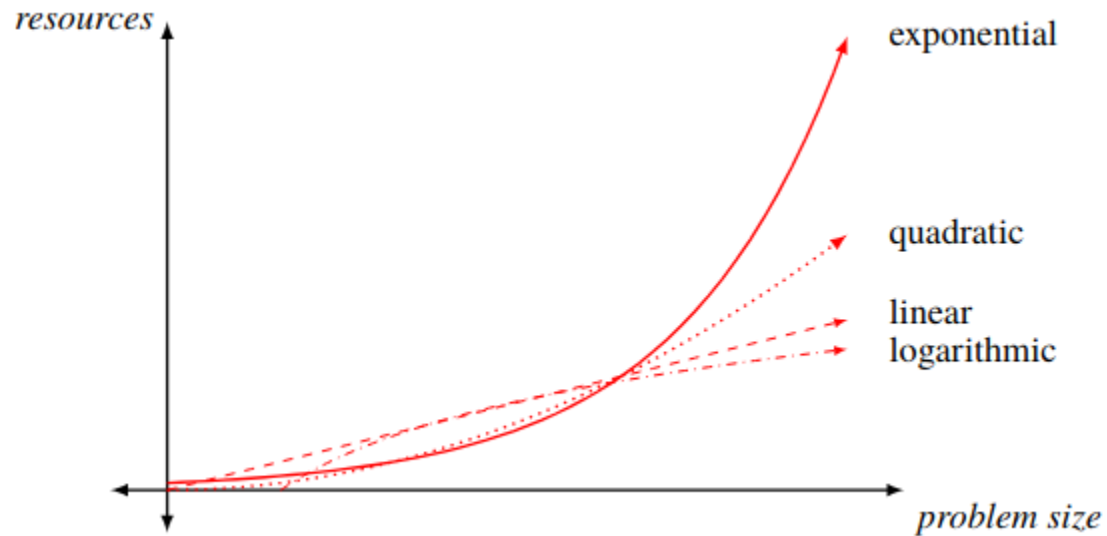


[Source: <https://www.101computing.net/binary-additions-using-logic-gates/>]

The interplay between classical and quantum computers



A word on complexity and what we need to achieve



[R. Sutor, Dancing with Qubits, Packt]

In QC we are interested in achieving at least a **polynomial speedup** with respect to classical algorithms.

How to compare algorithms? – The ‚Big – O‘ notation:

Let $f(n)$ and $g(n)$ be functions from positive integers to positive reals. We say $f = O(g)$ (which means that „f grows no faster than g“) if there is a constant $c > 0$ such that $f(n) \leq c \cdot g(n)$.

Example: Sorting

Sort [7, -2, 0, 3] – How do you do it?

In general number of swaps $\leq \frac{1}{2} n^2$

So $O(n^2)$



With the advent of near-term quantum devices, this monolithic reliance on complexity theory is slowly changing. Empirical studies and proof-of-principle experiments have to deal with the details of an implementation, and a constant factor in the runtime, for example, if we need $n = 20$ qubits or $cn = 1,000,000 * 20$ qubits (even if the constant c does not grow with the problem size), suddenly becomes crucial. This is an exciting development: classical computer science would be widely decimated (and machine learning hardly existent) if the only algorithms people are interested in were those for which we can prove efficient runtimes on paper.

Maria Schuld et al.

How to define quantum speedup?

Terminology developed by T.F. Rnnow et al. to benchmark QCs and quantum algorithms:

Provable quantum speedup:

- Proof required that there cannot be a classical algorithm that performs as well or better than the considered quantum algorithm
- This is the case for Grover's algorithm -> scales quadratically better than classical, given an oracle to mark the desired state

Strong quantum speedup:

- Comparison of the quantum algorithm with the best known classical algorithm
- For example Shor's algorithm

Common quantum speedup:

- Strong quantum speedup relaxed to comparing to best available classical algorithm

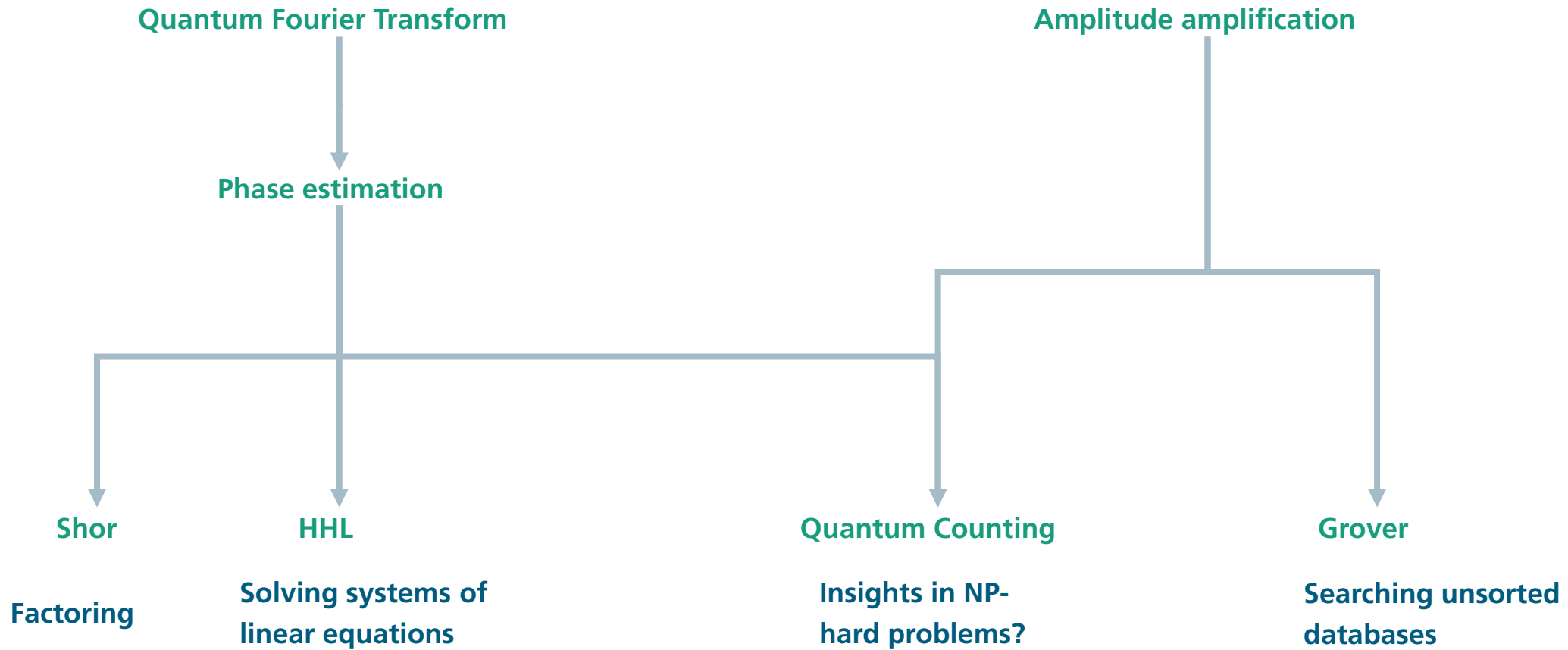
Potential quantum speedup:

- Only comparing two specific algorithms and just referring to this comparison.

Limited quantum speedup:

- Comparison of two conceptually equivalent algorithms.
- Example: quantum and classical annealing

Outlook: A map through fault-tolerant QC



Tutorial: the addition on a quantum computer

Tutorials attached to the agenda

Setup:

- Either linux installation, virtual box with linux installation, or google colab
- Install via pip:
 - pip install qiskit
 - pip install pennylane
 - (and probably a few more things will be requested during the installation process)
 - Installation commands in google colab have to be preceded by a !