Quantum Computing:
A New Approach to Algorithms

Yassine Hamoudi

Part 1

What is a
quantum computer?

Part 2

Some problems solved by
quantum algorithms

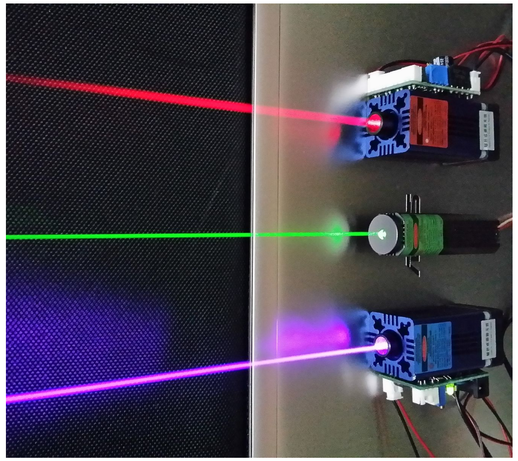
PART 1

What is a quantum computer?

Quantum devices

Stimulated emission

- Laser
- Atomic clock, GPS



Tunnelling

- Flash memory
- Scanning tunneling microscope

Magnetic resonance

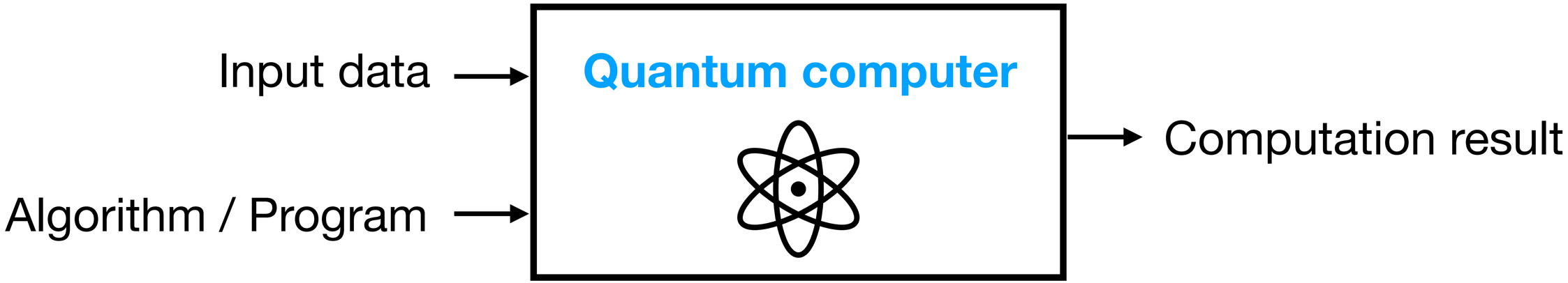
- Magnetic Resonance Imaging
- NMR spectroscopy

Photoelectric & Photovoltaic effect

- Solar panel
- CCD sensor

What is a quantum computer?

A physical device that exploits the laws of quantum mechanics to perform computations on data



Example of tasks: find integer solutions to $x^2 - 511y^2 = 1$, simulate the FeMoco molecule, find the prime decomposition of $2^{1550019073} - 1$

Why do we want quantum computers?

Properties predicted by mathematical models:

1. Faster algorithms for solving certain problems
2. New cryptographic tasks that are impossible to achieve with classical computers (q. key distribution, copy protection…)
3. Computer networks with enhanced properties (more secure communications, better distributed algorithms…)

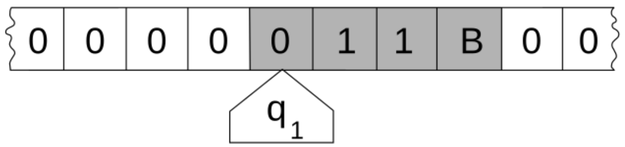
…

What quantum computers **will not do**?

- Speedup every task done by today's computers
  - Relatively few domains of applications in which QC are known to be superior
  - Forecasted as industry/research devices (same as supercomputers, GPU architectures…)
  - Overhead in implementation cost (error correction…) will cancel certain advantages of QC
- Try all solutions to a problem at once / in parallel
  - Properties of quantum mechanics (superposition, interferences,…) are more subtle than that
  - NP-hard problems are believed to remain hard for QC
- Break all existing encryption protocols
  - Post-quantum cryptography: study of quantum-safe protocols (ex: lattice-based crypto)
  - Current attacks (ex: breaking RSA with Shor's factoring) are out of reach of near term QC

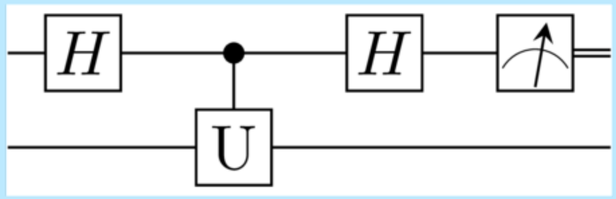
How to (mathematically) construct a quantum computer?

Several mathematical models describing how quantum computers are expected to behave:



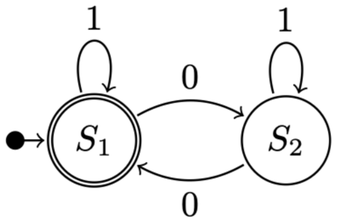
Q. Turing machine

(~1980-85)



Q. circuit

(~1989-93)



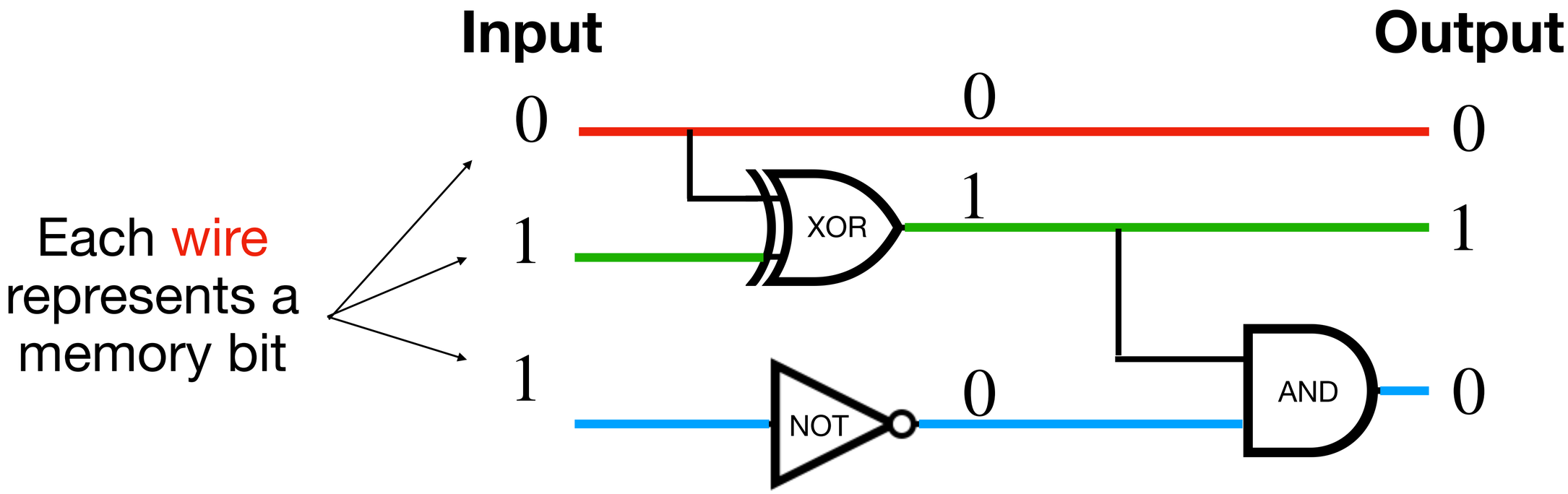
Q. finite automaton

(~1997-2000)

…

Classical computer

A classical computer can be modeled as a sequence of Boolean gates operating on memory bits

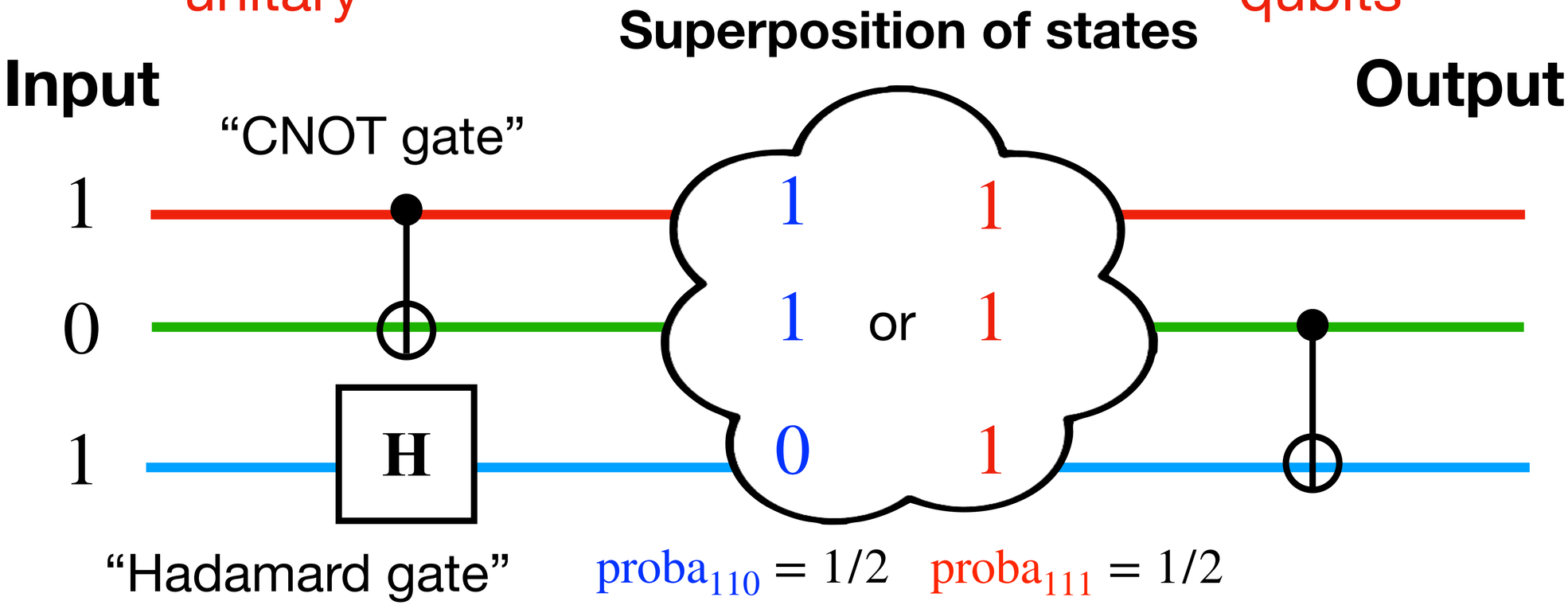


Quantum computer

quantum

A ~~classical~~ computer can be modeled as a sequence of

~~Boolean~~ unitary gates operating on memory ~~bits~~ qubits



Encoded as a vector: ("wave function")

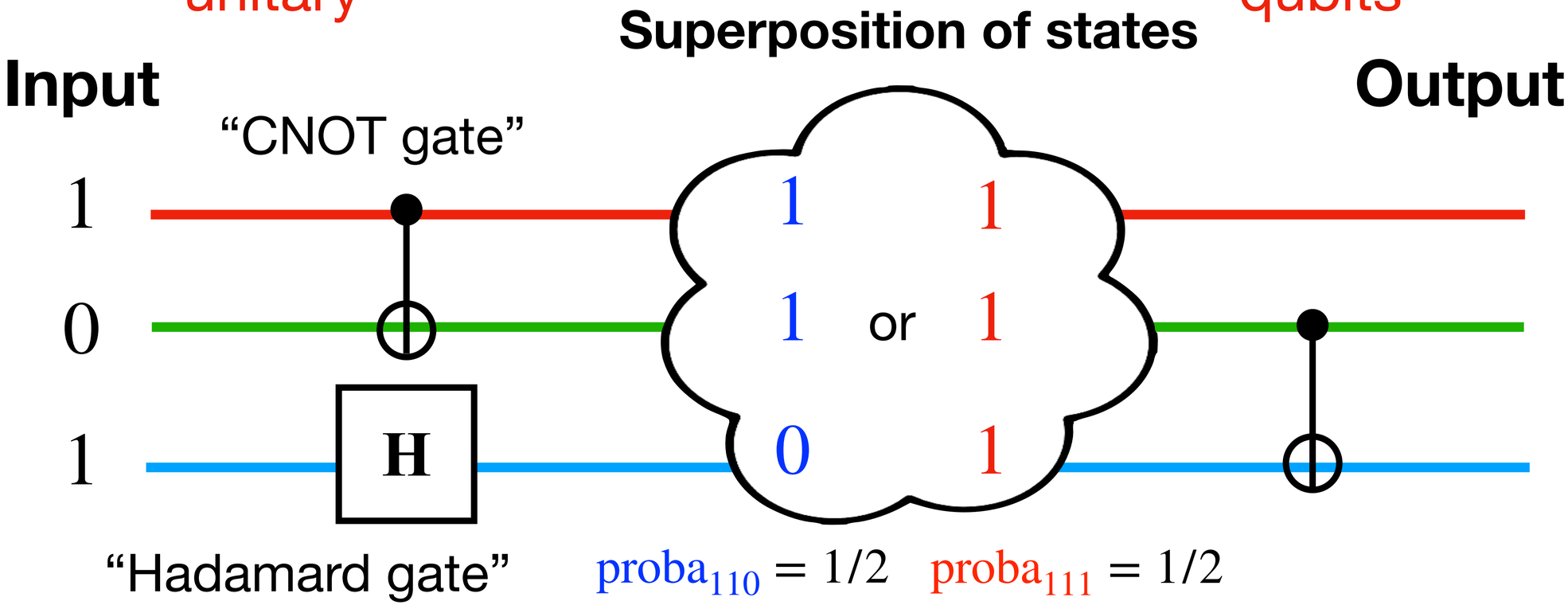
$$\vec{\psi} = \sqrt{1/2} \cdot \overrightarrow{110} - \sqrt{1/2} \cdot \overrightarrow{111}$$

Quantum computer

quantum

A ~~classical~~ computer can be modeled as a sequence of

~~Boolean~~ unitary gates operating on memory ~~bits~~ qubits



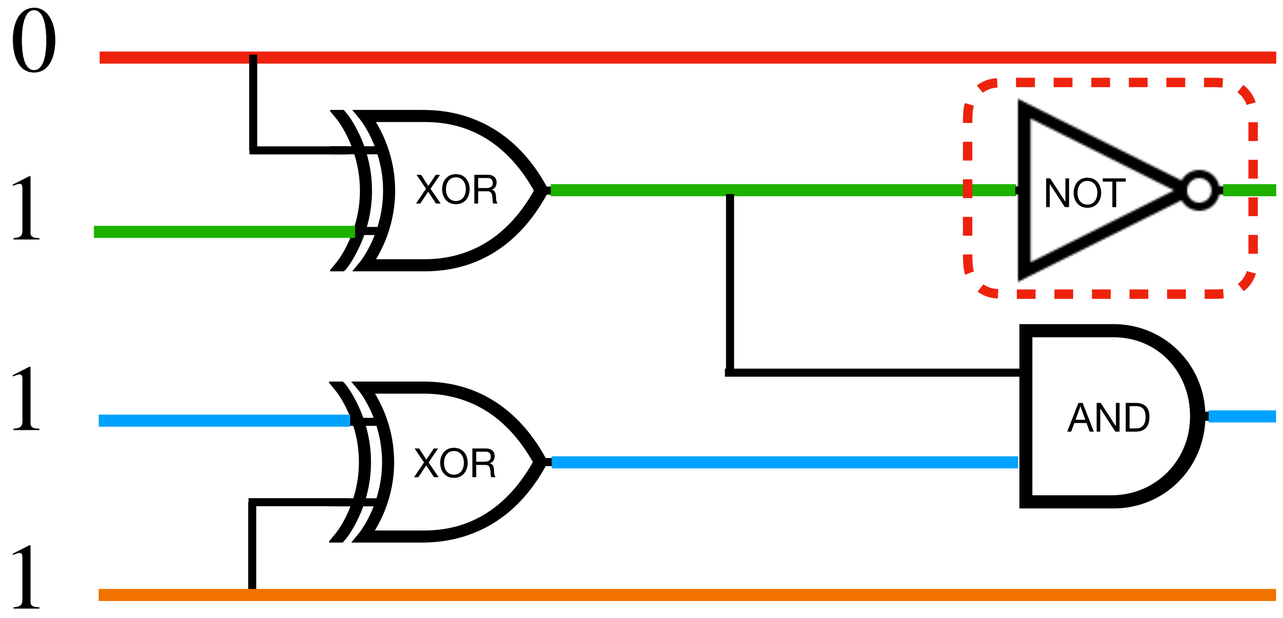
Encoded as a vector: ("wave function")

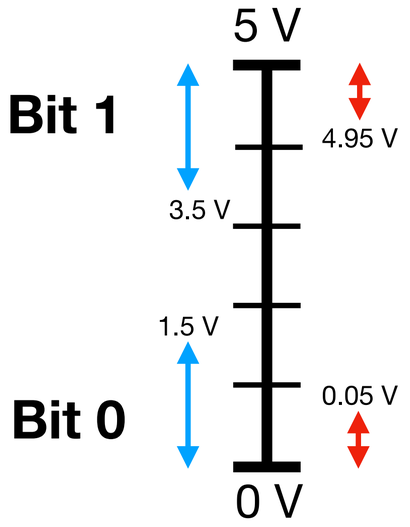
$$|\psi\rangle = \sqrt{1/2}\,|110\rangle - \sqrt{1/2}\,|111\rangle$$

How to (physically) construct a quantum computer?

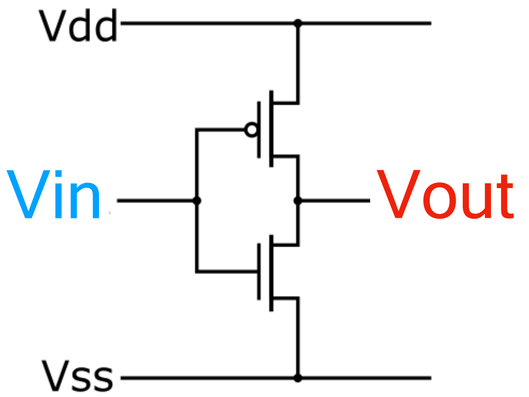
Classical computer

Lots of possible technologies (ex: transistors) that match very closely the mathematical model





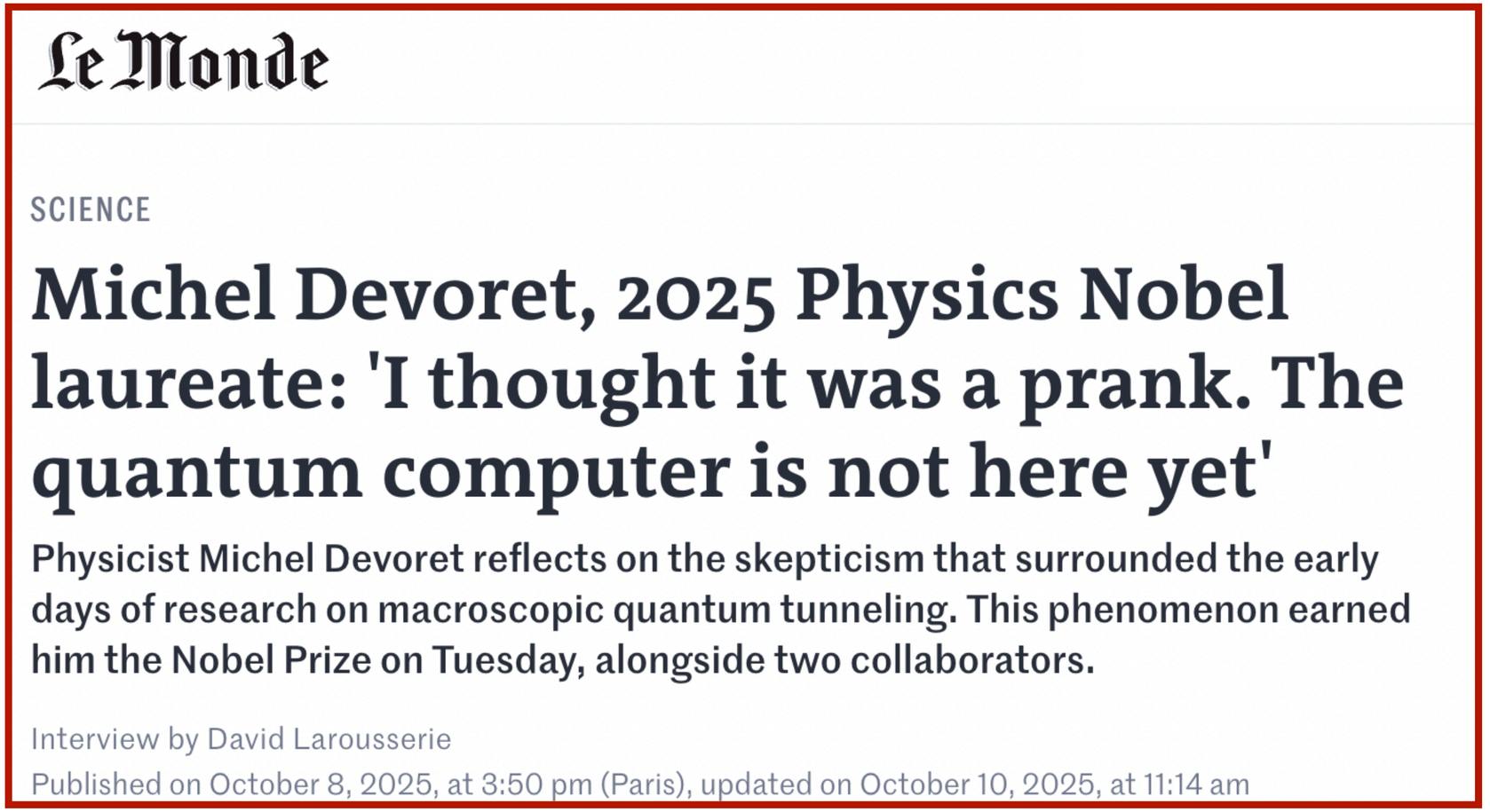
NOT gate



CMOS inverter

Quantum computer

We don’t have yet the technologies to construct large-scale quantum computers



Le Monde

SCIENCE

Michel Devoret, 2025 Physics Nobel laureate: 'I thought it was a prank. The quantum computer is not here yet'

Physicist Michel Devoret reflects on the skepticism that surrounded the early days of research on macroscopic quantum tunneling. This phenomenon earned him the Nobel Prize on Tuesday, alongside two collaborators.

Interview by David Larousserie

Published on October 8, 2025, at 3:50 pm (Paris), updated on October 10, 2025, at 11:14 am

Quantum computer

We don’t have yet the technologies to construct large-scale quantum computers

Some major challenges:

- Imperfections in qubits/gates implementations (noise accumulation)
- Decoherence effects (uncontrolled loss of quantum properties)

→ Both theoretical and engineering questions

(finding efficient quantum error correcting codes, constructing qubits and gates of good quality, …)

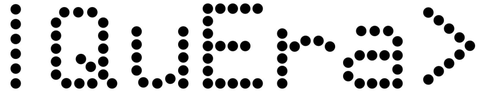
Candidates technologies for physical qubits

Superconductors

Trapped ions

Photons

Neutral atoms



...

When will quantum computers arrive?

Many companies have roadmaps aiming for the first fully functional, moderately large quantum computers by ~2030 … this looks very optimistic

Defense and security agencies warn of a quantum threat within 10-15 years

What we see at the moment:

- Steady progress in the technologies
- Sustained investments by public and private actors
- Push from manufacturers to start selling prototype quantum computers
- Broad industrial interest in potential quantum applications

PART 2

Some problems solved
by quantum algorithms

Area	Example
Simulation of quantum systems	Hamiltonian simulation
Cryptographic attacks	Factoring
Cryptographic protocols	Key distribution
Optimization	Linear programming
Learning	State tomography
...	...

Cryptographic attacks

Task 1: Factoring

Find the prime factors of an integer

$65535 = 3 \times 5 \times 17 \times 257$

Large fraction of crypto built on the assumption that Factoring is hard

Breakthrough in 1994 by Peter Shor: an efficient quantum algorithm

→ Factoring-based crypto (e.g. RSA) is not safe against quantum computers

→ Triggered a lot of research on quantum computing and cryptography

Part of a larger family of quantum attacks for Hidden Subgroup Problems

(discrete log, Simon's problem, Dihedral Coset Problem…)

Task 1: Factoring

Find the prime factors of an integer

$65535 = 3 \times 5 \times 17 \times 257$

A related task: finding the period of a function

In order to factor 65535 consider the function $f(x) = 2^x \mod 65535$

The function is periodic of period $s = 16$ (check that $f(x+s) = f(x)$)

Special product formula: $f(16) - f(0) = 2^{16} - 1 = (2^8 - 1)(2^8 + 1) = 0 \mod 65535$

Must have a factor in common!
(Easy to find with Euclidean algorithm)

Shor's algorithm is all about finding such periods

Task 2: Simon's problem

A toy problem invented in 1994 that displays an exponential quantum speedup and inspired Shor's algorithm

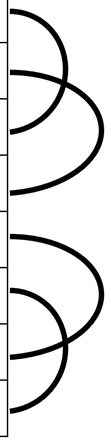
Find the secret period $s \in \{0,1\}^n$ of a function $f : \{0,1\}^n \to \{0,1\}^n$ that satisfies $f(x) = f(y)$ if and only if $y = x \oplus s$.

Example:

$n = 3$

$s = 010$

x	$f(x)$
000	011
001	111
010	011
011	111
100	110
101	000
110	110
111	000



Classical algorithm:

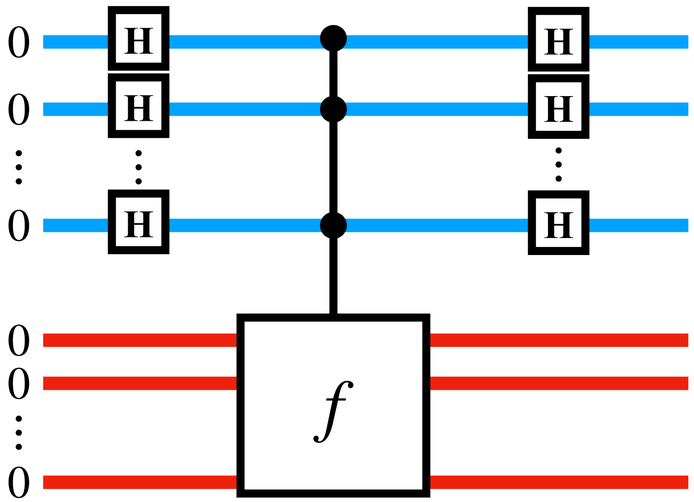
1/ Evaluate f on random $x_1, x_2, x_3 \ldots$ until finding a collision $f(x_i) = f(x_j)$

2/ Output $s = x_i \oplus x_j$

Birthday paradox: $\approx 2^{n/2}$ evaluations

Quantum algorithm: only $\approx n$ evaluations

Task 2: Simon's problem



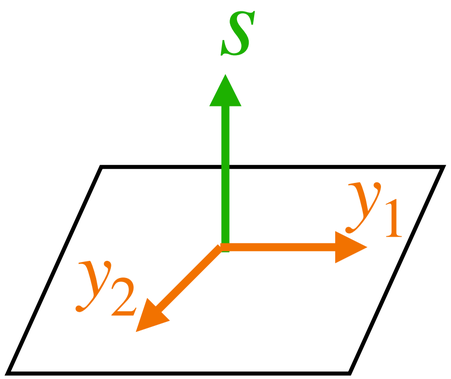
The quantum circuit

Key property

The output is a sample drawn from the uniform distribution on:

$$\{y \in \{0,1\}^n : \langle y, s \rangle = 0\}$$

Linear equation in s



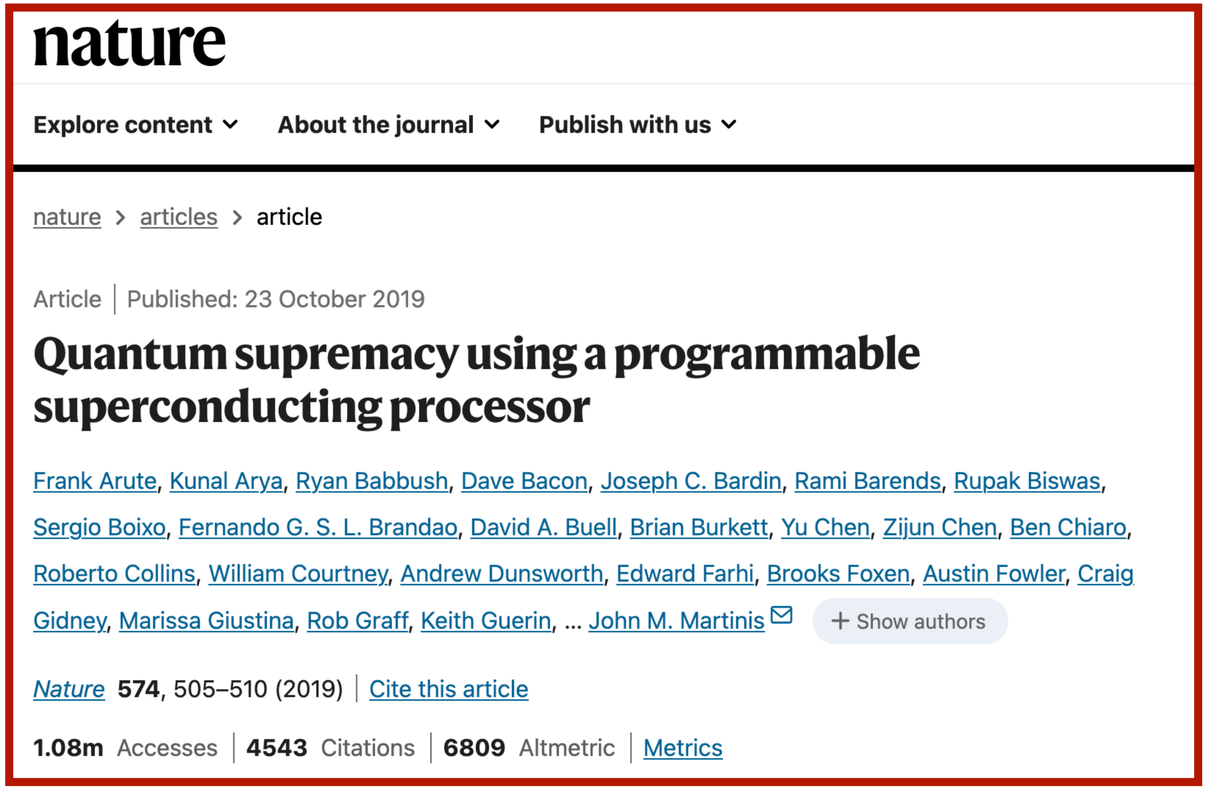
Overall quantum algorithm

Execute the quantum circuit $\approx n$ times to sample a system of n linear independent equations, and solve it by (classical) Gaussian elimination

Takeway Quantum algorithms excel at solving problems with periodic structures and at sampling from particular distributions.

First experimental confirmation of a quantum computing advantage:

? Sampling a distribution that is hard for classical computers but easy with current quantum technologies



nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature › articles › article

Article | Published: 23 October 2019

Quantum supremacy using a programmable superconducting processor

Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, ... John M. Martinis + Show authors

Nature **574**, 505–510 (2019) | Cite this article

1.08m Accesses | **4543** Citations | **6809** Altmetric | Metrics

(example: attempt by Google in 2019)

Simulation of quantum systems

Simulation of quantum systems

Simulating a system that evolves according to the laws of quantum mechanics and predicting its properties

The grand motivation for constructing a quantum computer:

"If you want to make a simulation of Nature,
you'd better make it quantum mechanical."

Feynman, 1981

Lots of use cases: chemistry (designing new drugs or battery materials…), condensed matter physics, high-energy physics…

Fundamental equations of time evolution

Classical mechanics

Classical state: $(\vec{x}(t), \vec{v}(t))$

Position Velocity

Newton's second law

$$\sum_k F_k = \frac{d\vec{v}(t)}{dt} \quad \text{and} \quad \vec{v}(t) = \frac{d\vec{x}(t)}{dt}$$

External forces

Quantum mechanics

Quantum state: $\vec{\psi}(t)$

Wave function: $\text{proba}_x = |\vec{\psi}(t)_x|^2$

Schrödinger equation

$$i\frac{d\vec{\psi}(t)}{dt} = H\ \vec{\psi}(t)$$

Hamiltonian of the system
(energy function)

Task: Hamiltonian simulation

Schrödinger equation

$$i\frac{d\vec{\psi}(t)}{dt} = H\ \vec{\psi}(t)$$

Solution

$$\vec{\psi}(t) = e^{-iHt}\ \vec{\psi}(0)$$

matrix

In general, classical computers need exponential time to compute a classical description of the solution (huge matrix exponentiation):

System of n particles $\Rightarrow$ $\vec{\psi}(t) \in \mathbb{C}^{2^n}$ $\Rightarrow$ H is a $2^n \times 2^n$ (Hermitian) matrix

For "well-behaved" Hamiltonians H, there are quantum circuits that can output the quantum state $\vec{\psi}(t) = e^{-iHt}\,\vec{\psi}(0)$ very fast (time $\mathrm{poly}(n,t)$)

Takeway

Quantum algorithms excel at solving problems that involve the simulation of quantum evolutions.

Another example: HHL algo

Construct the solution of a $N \times N$ linear system A in time $O(\log N)$ by exploiting a simulation of e^{iA}

Limitation: output is a quantum state, with restricted readout properties (collapsing, no-cloning, …)

Quantum Algorithm for Linear Systems of Equations

Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd
Phys. Rev. Lett. **103**, 150502 – Published 7 October 2009

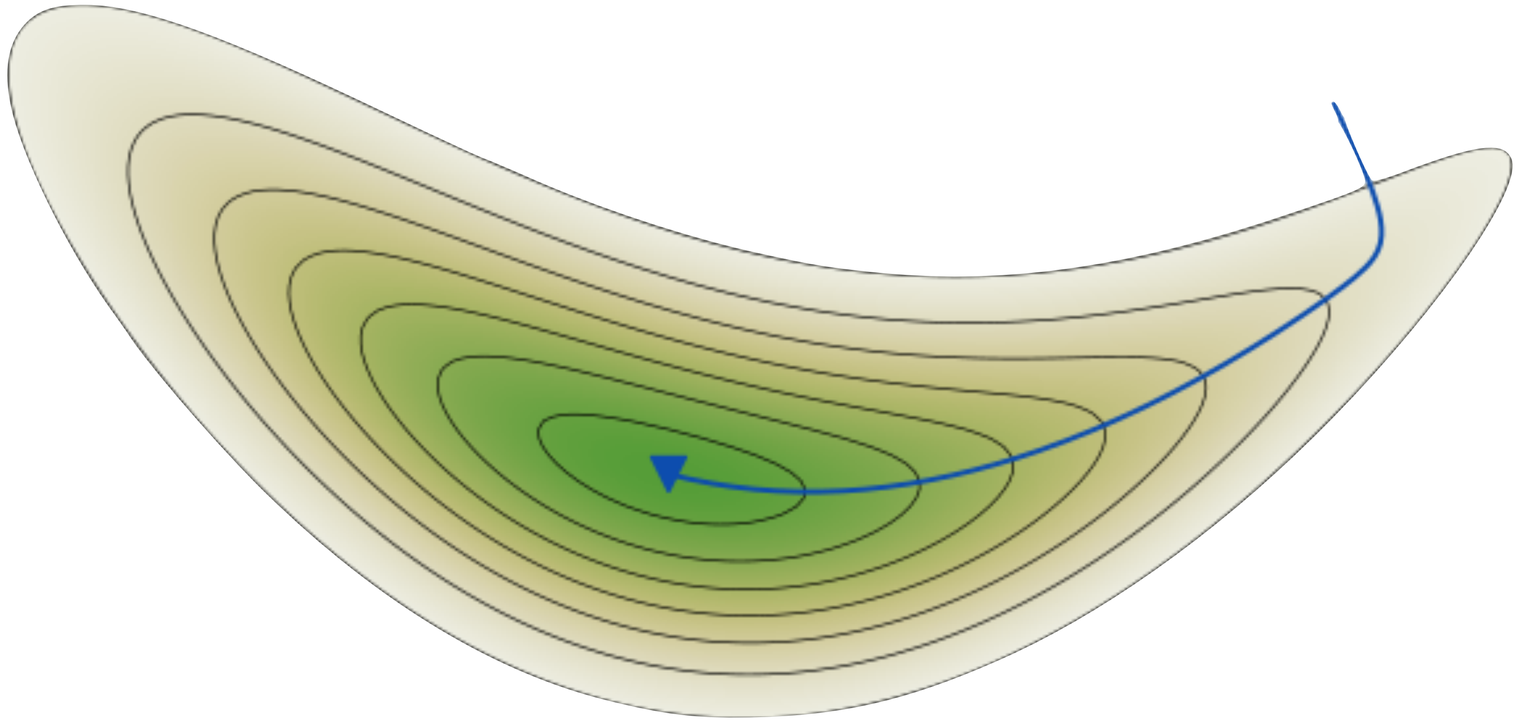
Physics See Synopsis: The quantum shortcut to a solution

ABSTRACT

Solving linear systems of equations is a common problem that arises both on its own and as a subroutine in more complex problems: given a matrix A and a vector $\vec{b}$, find a vector $\vec{x}$ such that $A\vec{x} = \vec{b}$. We consider the case where one does not need to know the solution $\vec{x}$ itself, but rather an approximation of the expectation value of some operator associated with $\vec{x}$, e.g., $\vec{x}^\dagger M\vec{x}$ for some matrix M. In this case, when A is sparse, $N \times N$ and has condition number κ, the fastest known classical algorithms can find $\vec{x}$ and estimate $\vec{x}^\dagger M\vec{x}$ in time scaling roughly as $N\sqrt{\kappa}$. Here, we exhibit a quantum algorithm for estimating $\vec{x}^\dagger M\vec{x}$ whose runtime is a polynomial of $\log(N)$ and κ. Indeed, for small values of κ [i.e., $\mathrm{poly}\log(N)$], we prove (using some common complexity-theoretic assumptions) that any classical algorithm for this problem generically requires exponentially more time than our quantum algorithm.

Optimization

Finding a value that minimizes a cost function



Spark a lot of interest from the industry for quantum computing

… but current evidence of such applications are rather limited

Quantum advantages for optimization

Provable results

- Finding the minimum of an unstructured set of numbers (Grover's algo.)
- Solving problems in convex optimization (ex: semidefinite programs, linear regression…)
- ….

Heuristics

- Solving problems in combinatorial optimization (ex., QUBO problem)

  (Variational quantum algorithms?)
- Finding the lowest energy state of a quantum system (ex., in chemistry)

  (Quantum annealing?)

Takeway

Lack of rigorous guarantees on the power of quantum algorithms for solving optimization problems

Further readings

Quantum computing 40 years later

John Preskill

Forty years ago, Richard Feynman proposed harnessing quantum physics to build a more powerful kind of computer. Realizing Feynman's vision is one of the grand challenges facing 21st century science and technology. In this article, we'll recall Feynman's contribution that launched the quest for a quantum computer, and assess where the field stands 40 years later.

https://arxiv.org/abs/2106.10522

Quantum Computing: Lecture Notes

Ronald de Wolf (QuSoft, CWI and University of Amsterdam)

This is a set of lecture notes suitable for a Master's course on quantum computation and information from the perspective of theoretical computer science. The first version was written in 2011, with many extensions and improvements in subsequent years. The first 10 chapters cover the circuit model and the main quantum algorithms (Deutsch-Jozsa, Simon, Shor, Hidden Subgroup Problem, Grover, quantum walks, Hamiltonian simulation and HHL). They are followed by 4 chapters about complexity, 4 chapters about distributed ("Alice and Bob") settings, a chapter about quantum machine learning, and a final chapter about quantum error correction. Appendices A and B give a brief introduction to the required linear algebra and some other mathematical and computer science background. All chapters come with exercises, with some hints provided in Appendix C.

https://arxiv.org/abs/1907.09415

Quantum algorithms: A survey of applications and end-to-end complexities

Alexander M. Dalzell, Sam McArdle, Mario Berta, Przemyslaw Bienias, Chi-Fang Chen, András Gilyén, Connor T. Hann, Michael J. Kastoryano, Emil T. Khabiboulline, Aleksander Kubica, Grant Salton, Samson Wang, Fernando G. S. L. Brandão

The anticipated applications of quantum computers span across science and industry, ranging from quantum chemistry and many-body physics to optimization, finance, and machine learning. Proposed quantum solutions in these areas typically combine multiple quantum algorithmic primitives into an overall quantum algorithm, which must then incorporate the methods of quantum error correction and fault tolerance to be implemented correctly on quantum hardware. As such, it can be difficult to assess how much a particular application benefits from quantum computing, as the various approaches are often sensitive to intricate technical details about the underlying primitives and their complexities. Here we present a survey of several potential application areas of quantum algorithms and their underlying algorithmic primitives, carefully considering technical caveats and subtleties. We outline the challenges and opportunities in each area in an "end-to-end" fashion by clearly defining the problem being solved alongside the input-output model, instantiating all "oracles," and spelling out all hidden costs. We also compare quantum solutions against state-of-the-art classical methods and complexity-theoretic limitations to evaluate possible quantum speedups.

https://arxiv.org/abs/2310.03011

Image credit: https://commons.wikimedia.org/

Slides: https://yassine-hamoudi.github.io/files/slides/Numerics.pdf