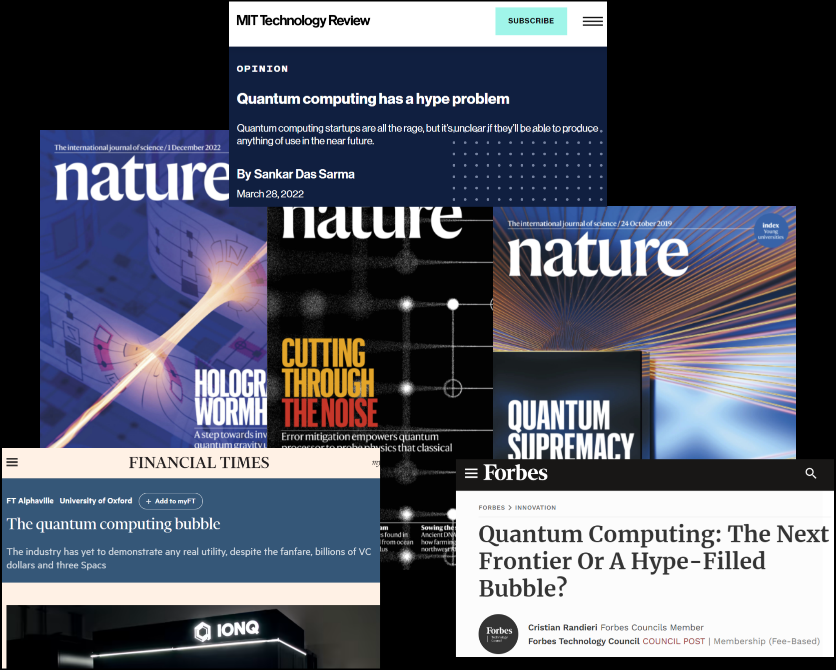
Quantum Computing: between the skepticism and the hype

Massimiliano Incudini

University of Verona

September 8, 2023

Departmental talk for CoSy.Bio,
Institute for Computational Systems Biology, Uni Hamburg


MIT Technology Review
SUBSCRIBE
OPINION
Quantum computing has a hype problem
Quantum computing startups are all the rage, but it's unclear if they'll be able to produce anything of use in the near future.
By Sankar Das Sarma
March 28, 2022
The international journal of science / 1 December 2022
nature
The international journal of science / 24 October 2019
nature
index Young universities
QUANTUM SUPREMACY
CUTTING THROUGH THE NOISE
Error mitigation empowers quantum
FINANCIAL TIMES
FT Alphaville University of Oxford + Add to myFT
The quantum computing bubble
The industry has yet to demonstrate any real utility, despite the fanfare, billions of VC dollars and three Spacs
IONQ
Forbes
FORBES › INNOVATION
Quantum Computing: The Next Frontier Or A Hype-Filled Bubble?
Cristian Randieri Forbes Councils Member
Forbes Technology Council COUNCIL POST | Membership (Fee-Based)

Enthusiasts

Skeptics

Today's topic: honest and critical introduction of the field

Agenda

1. From classical to quantum computing

Agenda

1. From classical to quantum computing
   - Technical part of the talk

Agenda

1. From classical to quantum computing
   - Technical part of the talk
2. Promises of quantum computing

Agenda

1. From classical to quantum computing
   - Technical part of the talk
2. Promises of quantum computing
   - Problems beyond the capabilities of classical computation

Agenda

1. From classical to quantum computing
   - Technical part of the talk
2. Promises of quantum computing
   - Problems beyond the capabilities of classical computation
3. Challenges of quantum computing

Agenda

1. From classical to quantum computing
   - Technical part of the talk
2. Promises of quantum computing
   - Problems beyond the capabilities of classical computation
3. Challenges of quantum computing
   - What we still miss to have a fully working device

From classical to quantum computing

Classical computation

$$\boxed{0} \qquad \boxed{1}$$

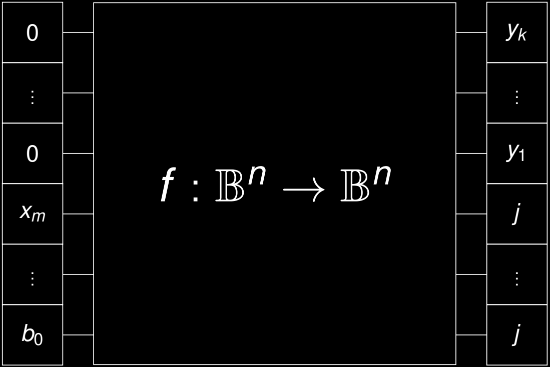
$$\mathbb{B}$$

Classical computation

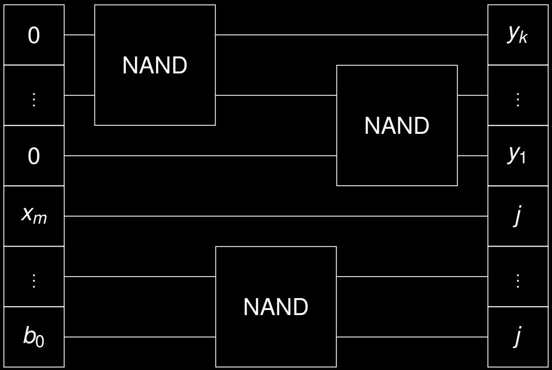
$$\begin{array}{|c|}\hline b_n \\ \hline \vdots \\ \hline \vdots \\ \hline b_1 \\ \hline b_0 \\ \hline \end{array}$$

$$\mathbb{B}^n$$

Classical computation



Classical computation



Classical computation

What if we express the computation model
in terms of linear algebra instead of Boolean algebra?

Classical computation (linear algebra)

$$\begin{bmatrix}1\\0\end{bmatrix} = |0\rangle \quad \begin{bmatrix}0\\1\end{bmatrix} = |1\rangle$$

$$\||b\rangle\|_0 = \#\text{nonzero} = 1$$

Classical computation (linear algebra)

An example of operation:

$$\mathrm{NOT} = \begin{array}{c} \\ \mathit{in} \\ 0 \\ 1 \end{array} \begin{array}{c} \mathit{out} \\ \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \end{array}$$

An example of computation:

$$\begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_0 \end{bmatrix}$$

Classical computation (linear algebra)

An example of a 2-bit operation:

$$\mathrm{CNOT} = \begin{array}{c c} & \begin{array}{cccc} 00 & 01 & 10 & 11 \end{array} \\ \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

The state is described by the vector:

$$\begin{bmatrix} b_{00} \\ b_{01} \\ b_{10} \\ b_{11} \end{bmatrix}$$

Classical computation (linear algebra)

For $f : \mathbb{B}^n \to \mathbb{B}^n$ we need $2^n \times 2^n$ Boolean matrices:

$$\begin{array}{c} \\ 00\cdots0 \\ 00\cdots1 \\ \vdots \\ 00\cdots1 \end{array} \begin{array}{c} \begin{array}{cccc} 00\cdots0 & 00\cdots1 & \cdots & 11\cdots1 \end{array} \\ \begin{pmatrix} m_{0,0} & m_{0,1} & \cdots & m_{0,2^n-1} \\ m_{1,0} & m_{1,1} & \cdots & m_{1,2^n-1} \\ \vdots & \vdots & \ddots & \vdots \\ m_{2^n-1,0} & m_{2^n-1,1} & \cdots & m_{2^n-1,2^n-1} \end{pmatrix} \end{array}$$

Classical computation (linear algebra)

The state of n bits is the tensor product of n spaces $\mathbb{B}^2$:

$$|b\rangle \otimes |c\rangle = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} \otimes \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} b_0 \begin{bmatrix} c_0 \\ c1 \end{bmatrix} \\ b_1 \begin{bmatrix} c_0 \\ c1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} b_0 c_0 \\ b_0 c_1 \\ b_1 c_0 \\ b_1 c_1 \end{bmatrix}$$

It holds that:

$$\| |b\rangle \otimes |c\rangle \|_0 = 1$$

Classical computation (linear algebra)

Given the state $|b\rangle = [\delta_{i,k}]_{i=0}^{2^n-1}$ (all 0 except k-th location),

the output is $b = \text{bin}(k) \in \mathbb{B}^n$.

Probabilistic computation

A p-bit is:

$$|p\rangle = \begin{bmatrix} p_0 \\ 1 - p_0 \end{bmatrix} \in \mathbb{R}^2,$$

constraint to

$$\| |p\rangle \|_1 = \sum_j |p_j| = 1$$

Probabilistic computation

The joint probability of the two p-bit is:

$$|p\rangle \otimes |q\rangle = \begin{bmatrix} p_0 \\ p_1 \end{bmatrix} \otimes \begin{bmatrix} q_0 \\ q_1 \end{bmatrix} = \begin{bmatrix} p_0 q_0 \\ p_0 q_1 \\ p_1 q_0 \\ p_1 q_1 \end{bmatrix}.$$

Probabilistic computation

The systems evolve accordingly to a stochastic matrix,

$$S \in \mathbb{R}_{\geq 0}^{2^n \times 2^n}$$

For example, the coin operation is:

$$S = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Probabilistic computation

The state of an n-p-bit system is:

$$|p\rangle = [p_k]_{k=0}^{2^n-1}$$

Its measurement leads to the output:

$$b = bin(k) \in \mathbb{B}^n \text{ with probability } p_k.$$

Quantum computation

Questions so far?

Quantum computation

Probabilistic computation with complex numbers.

Quantum computation

A qubit:

$$|\psi\rangle = \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \in \mathbb{C}^2$$

constraint to:

$$\||\psi\rangle\|_2 = \sum_j |\alpha_j|^2 = 1$$

A system of n qubits:

$$|\psi\rangle = \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{2^n-1} \end{bmatrix}$$

Quantum computation

The systems evolve accordingly to a unitary matrix,

$$U \in \mathbb{C}^{2^n \times 2^n}, \qquad U^\dagger U = UU^\dagger = I$$

For example, the Hadamard operation is:

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

Quantum computation

The state of an n qubit system is:

$$|\psi\rangle = [\alpha_k]_{k=0}^{2^n-1} \in \mathbb{C}^{2^n}$$

Its measurement leads to the output:

$$b = bin(k) \in \mathbb{B}^n \text{ with probability } |\alpha_k|^2 \in \mathbb{R}.$$

Quantum computation

The measurement is destructive.

The state $|\psi\rangle$ after the measurement collapses to

$$|k\rangle = [\delta_{i,k}]_{i=1}^{2^n-1}.$$

Quantum computation

Approach	**State**	**Evolution**	**Measurement**
Classical	$\mathbb{B}^n$	Boolean function	$\mathbb{B}^n$
Classical (lin)	$\mathbb{B}^{2^n}, \|\cdot\|_0 = 1$	Deterministic mat	$\mathbb{B}^n, p(k) = 1$
Probabilistic	$\mathbb{R}^{2^n}, \|\cdot\|_1 = 1$	Stochastic mat	$\mathbb{B}^n, p(k) = p_i$
Quantum	$\mathbb{C}^{2^n}, \|\cdot\|_2 = 1$	Unitary mat	$\mathbb{B}^n, p(k) = \lvert\alpha_i\rvert^2$ (destructive)

Quantum computation

Which features characterize quantum computing?

Quantum computation

Which features characterize quantum computing?

Superposition • Entanglement • Interference

Superposition

A particle can be in a superposition of multiple states.

The Hadamard operation allows the creation of uniform superposition of states:

$$|00\rangle \quad \xrightarrow{H \otimes H} \quad \sum_{i=0}^{3} |i\rangle \,,$$

or

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \xrightarrow{H \otimes H} \quad \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Entanglement

Given a group of particles, the system is entangled if the quantum state of each particle of the group cannot be described independently of the state of the others.

$$|00\rangle = |0\rangle \otimes |0\rangle \qquad \text{separable}$$

Entanglement

Given a group of particles, the system is entangled if the quantum state of each particle of the group cannot be described independently of the state of the others.

$$|00\rangle = |0\rangle \otimes |0\rangle \qquad \text{separable}$$

$$(H \otimes H)\,|00\rangle = \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \qquad \text{separable}$$

Entanglement

Given a group of particles, the system is entangled if the quantum state of each particle of the group cannot be described independently of the state of the others.

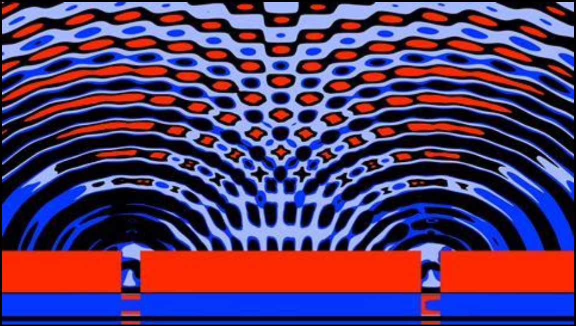
$$|00\rangle = |0\rangle \otimes |0\rangle \qquad \text{separable}$$

$$(H \otimes H)\,|00\rangle = \frac{1}{2}(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle) \qquad \text{separable}$$

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = ??? \otimes ??? \qquad \text{entangled}$$

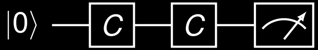
Interference

Interference is a phenomenon in which two coherent waves are combined by adding their intensities or displacements with due consideration for their phase difference.



Interference

Consider the probabilistic computation "flip-coin-twice":



1. The initial state is $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
2. The first coin flip lead to $\frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1\cdot 1+1\cdot 0 \\ 1\cdot 0+1\cdot 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
3. The second coin flip lead to $\frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1\cdot 1+1\cdot 1 \\ 1\cdot 1+1\cdot 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
4. I will see 50% times '0' and the other 50% '1'.

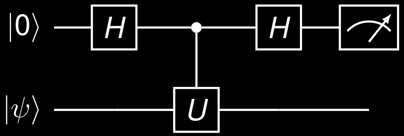
Interference

Consider the quantum computation "apply-Hadamard-twice":

1. The initial state is $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
2. $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1\cdot 1+1\cdot 0 \\ 1\cdot 0+1\cdot 1 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
3. $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1\cdot 1+1\cdot 1 \\ 1\cdot 1-1\cdot 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
4. I will always measure '0'.

Quantum Phase Estimation algorithm

Consider U such that $U|\psi\rangle = e^{i\theta}|\psi\rangle$



Output: estimation of the $\mathrm{Re}\{\theta\}$ (1 bit).

Applications of QPE

- Order finding, used in Factorization
  ($\psi =$ number to factorize, *U* $=$ modular exponentiation, $\theta \approx$ order)
- Quantum chemistry
  ($\psi =$ init. state, *U* $=$ system Hamiltonian, $\theta =$ energy)
- Topological data analysis
  ($\psi =$ simplex, *U* $=$ combinatorial Laplacian, $\theta =$ Betti number)

Promises of quantum computing

Motivation

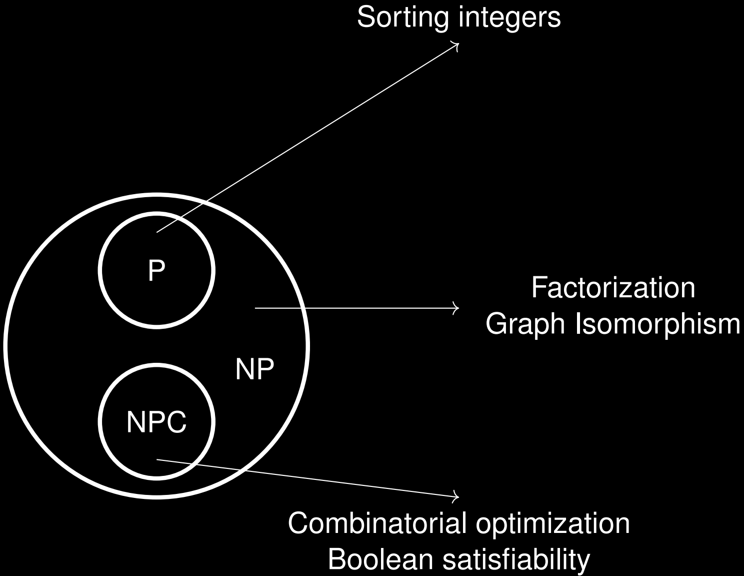
1. Theory of algorithms
2. Naturally quantum problems
3. Energetic consumption

Computational complexity

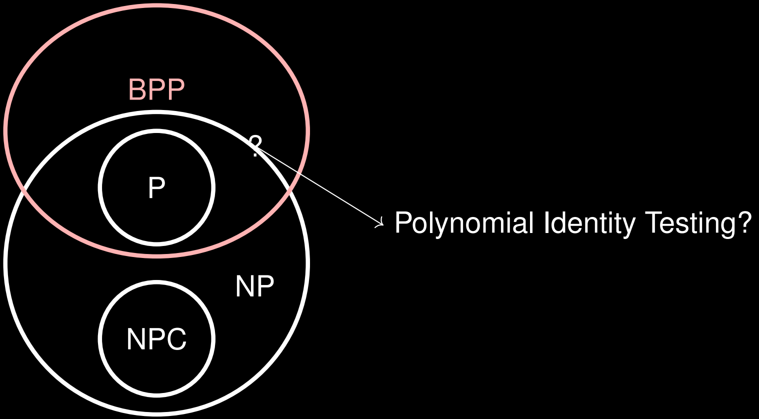
A probabilistic Turing machine can efficiently simulate any realistic model of computation.

Church-Turing thesis

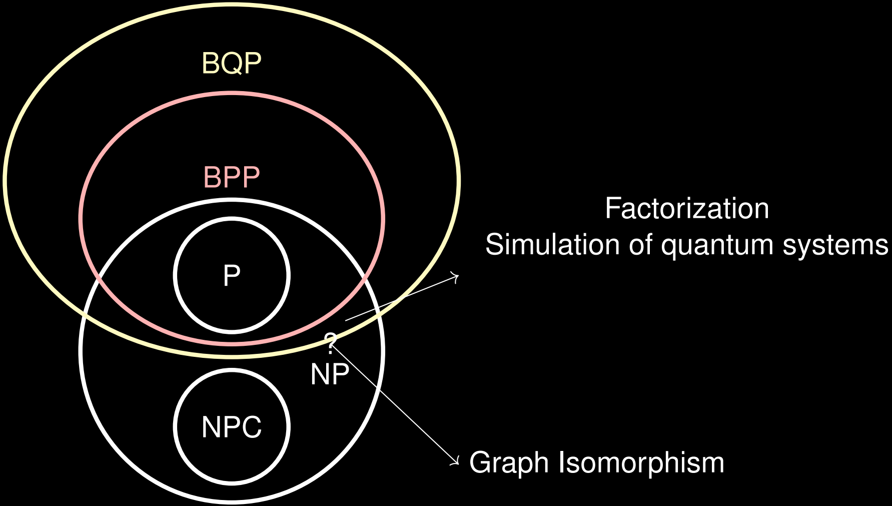
Computational complexity



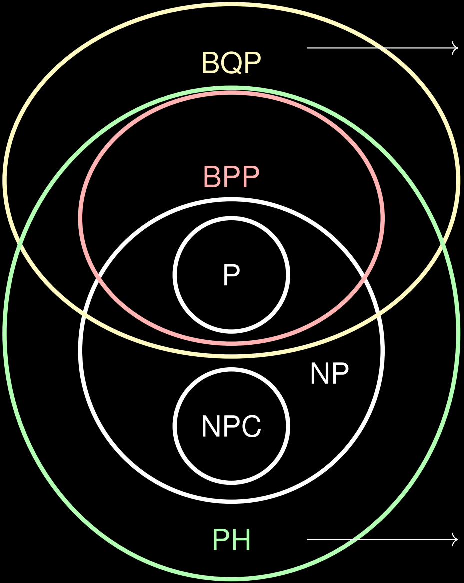
Computational complexity



Computational complexity



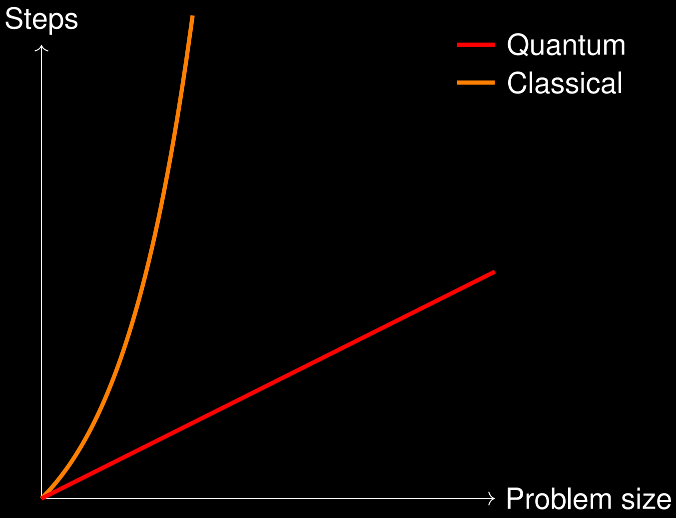
Computational complexity



approx Jones polynomial

$\forall x_1 \exists x_2 \cdots \forall x_k . \phi(x)$

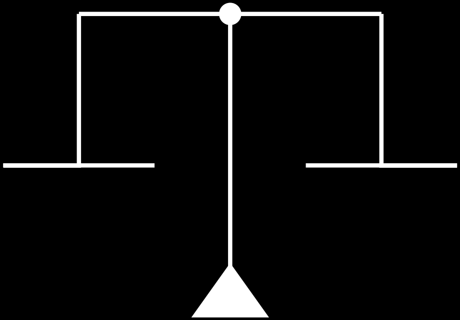
Speedup



(*) on some selected problems

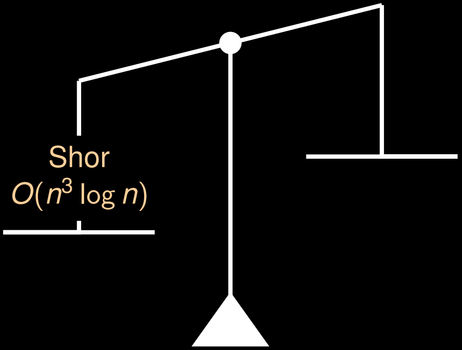
Query complexity and actual resource estimation

$$914411804139769320982514988529 = P \times Q$$



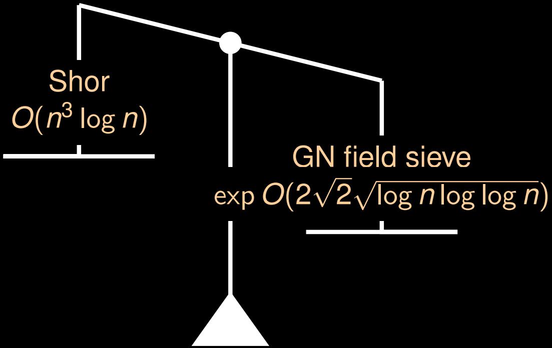
Query complexity and actual resource estimation

$$914411804139769320982514988529 = P \times Q$$



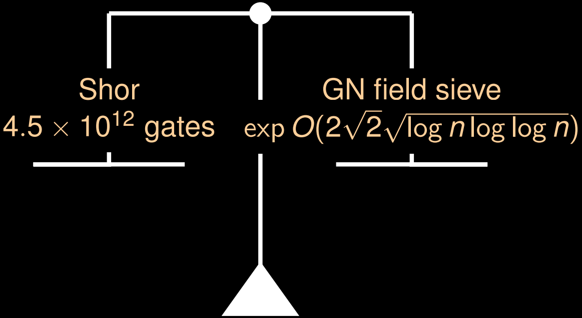
Query complexity and actual resource estimation

$$914411804139769320982514988529 = P \times Q$$



Query complexity and actual resource estimation

$$914411804139769320982514988529 = P \times Q$$



Killer applications

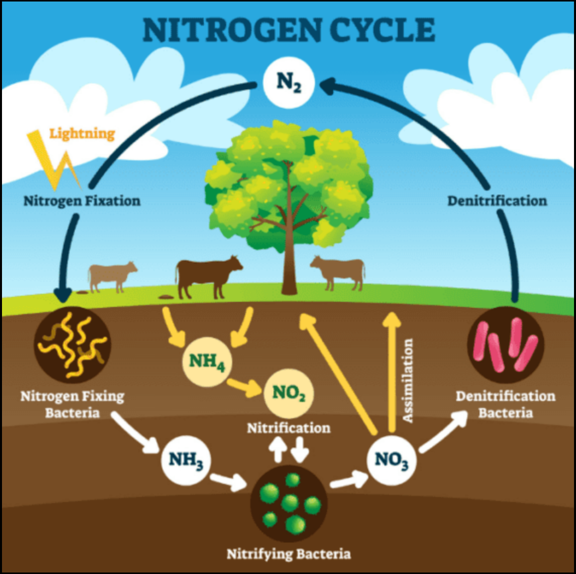
What are the killer applications of quantum computing?

Killer applications

What are the killer applications of quantum computing?

Simulation for physics / chemistry / material science

Fertilization



Fertilization

Synthetic fertilizers production (Haber-Bosch process)

Fertilization

Synthetic fertilizers production (Haber-Bosch process)

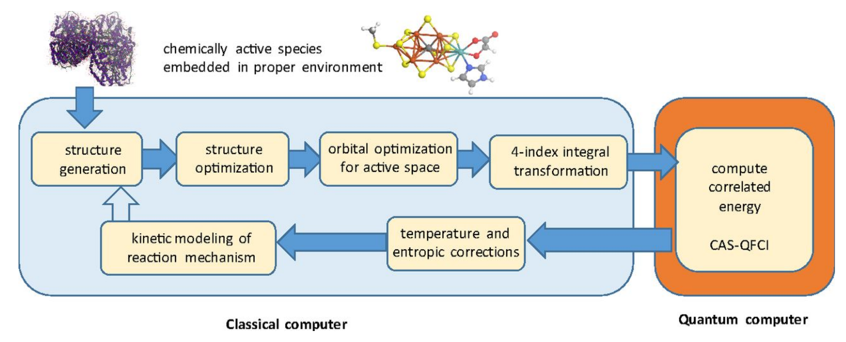
Abundance of food

Fertilization

Synthetic fertilizers production (Haber-Bosch process)

Abundance of food / 2% of the energy produced worldwide

Fertilization[1]



[1] Reiher, Markus, et al. "Elucidating reaction mechanisms on quantum computers." Proceedings of the national academy of sciences 114.29 (2017): 7555-7560.

Other promising use-cases

- Drug discovery
  Blunt, et al. ”Perspective on the current state-of-the-art of quantum computing for drug discovery applications.” Journal of Chemical Theory and Computation 18.12 (2022).
- Carbon capture
  Von Burg et al. ”Quantum computing enhanced computational catalysis.” Physical Review Research 3.3 (2021).
- Battery design
  Paudel, et al. ”Quantum computing and simulations for energy applications: Review and perspective.” ACS Engineering (2022).

Energy advantage

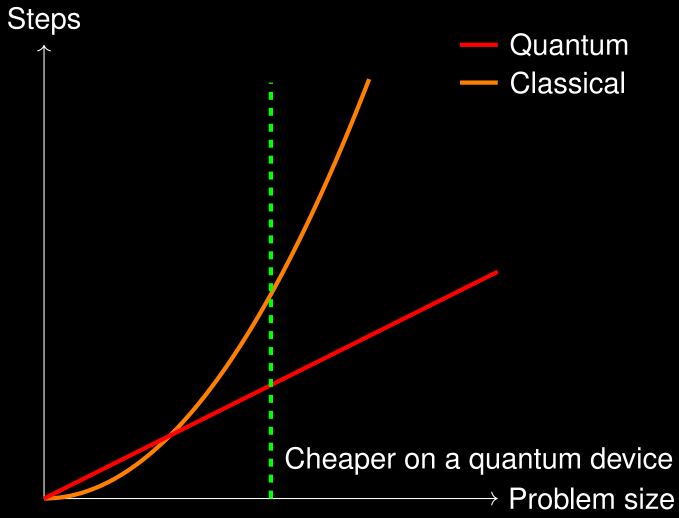
Data centers use 1.2% of the global electricity demand.

Energy advantage

Data centers use 1.2% of the global electricity demand.

What if quantum computers can provide an advantage in terms of energy consumption?

Energy advantage



(*) on some selected problems

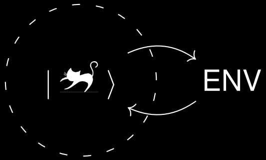
Energy advantage

- Via the Cournot competition model
  (Liu et al. arXiv:2308.08025)
- Compared to classical simulation of quantum systems
  (Jaschke et al. Quantum Sci. Technol. 8 025001)
- Inherited from an exponential time advantage
  (Meier et al. arXiv:2305.11212)
- For classical computation too
  (Moutinho et al. PRX Energy 2, 033002)

Challenges of quantum computing

Errors in quantum computing

Closed quantum system



Open quantum system

Interaction with the environment results in decoherence.

Errors in quantum computing

1. Sign / Phase flip errors
2. Non-unitary evolution
3. Control-related errors
4. Sampling errors

Quantum Fault-Tolerance Theorem

A quantum circuit with n qubits, m gates can be simulated with a probability of error $\leq \epsilon$ using $O(m \log^c m)$ gates on hardware whose components fail with a probability below a certain threshold.

(Such a threshold is hard to calculate in practice)

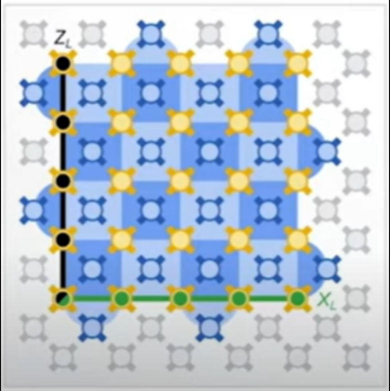
Quantum Error Correction

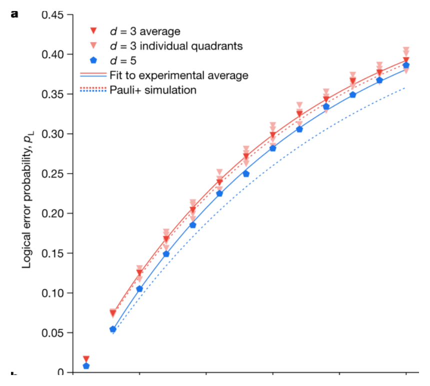
In contrast to classical error correction techniques, we struggle to introduce redundancy (*no cloning theorem*).

Although challenging, we can actually implement error correction scheme for quantum computers.

Grouping many physical qubits to a logical one increases or decreases the performances? Are we faster to correct errors than to introduce new ones?

Quantum Error Correction





72 physical qubits • 49 used • 2 logical [2]

[2] Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. Nature 614, 676–681 (2023).

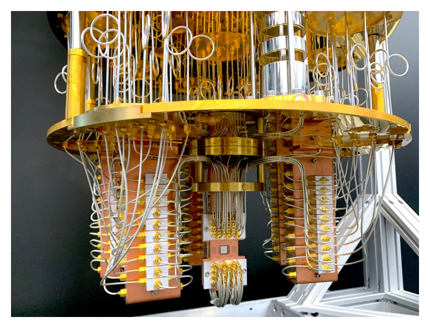
Hardware

- Superconducting qubits
- Photonic devices
- Trapped-ions
- Neural atoms
- Silicon-based
- Diamond nitrogen-vacancy

Each technology lead to completely different devices:

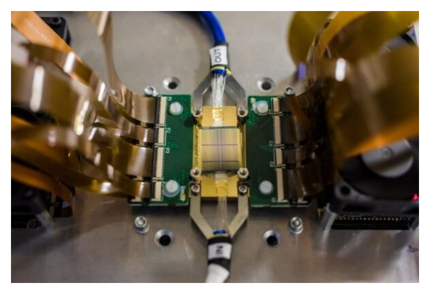
Scalability • Coherence time • Fidelity • Frequency •
Temperature • Size • Cost • Technological maturity

Hardware: superconducting qubits



- almost 500 qubits
- easy to scale
- limited connectivity
- low temperature required
- low fidelity

Hardware: photonics



- tens of qubits
- hard to scale
- full connectivity
- room temperature
- high fidelity

Realistic analysis of algorithms

The efficiency of quantum algorithms is often measured in terms of query complexity. What if we consider a more fine-grained analysis?

[3] Lemieux, et al. Efficient quantum walk circuits for Metropolis-Hastings algorithm. Quantum 4 (2020): 287.

[4] Layden, Mazzola, et al. Quantum-enhanced Markov chain Monte Carlo. Nature 619.7969 (2023): 282-287.

Realistic analysis of algorithms

The efficiency of quantum algorithms is often measured in terms of query complexity. What if we consider a more fine-grained analysis?

Arithmetic is very hard to implement on quantum devices (large pre-factors).

[3] Lemieux, et al. Efficient quantum walk circuits for Metropolis-Hastings algorithm. Quantum 4 (2020): 287.

[4] Layden, Mazzola, et al. Quantum-enhanced Markov chain Monte Carlo. Nature 619.7969 (2023): 282-287.

Realistic analysis of algorithms

The efficiency of quantum algorithms is often measured in terms of query complexity. What if we consider a more fine-grained analysis?

Arithmetic is very hard to implement on quantum devices (large pre-factors).

For combinatorial optimization, and targetting polynomial speedups, we need MHz-like logical gate frequency[3], which is realistic.

[3] Lemieux, et al. Efficient quantum walk circuits for Metropolis-Hastings algorithm. Quantum 4 (2020): 287.

[4] Layden, Mazzola, et al. Quantum-enhanced Markov chain Monte Carlo. Nature 619.7969 (2023): 282-287.

Realistic analysis of algorithms

The efficiency of quantum algorithms is often measured in terms of query complexity. What if we consider a more fine-grained analysis?

Arithmetic is very hard to implement on quantum devices (large pre-factors).

For combinatorial optimization, and targetting polynomial speedups, we need MHz-like logical gate frequency[3], which is realistic.

Recently shown how to bring this advantage to MCMC. [4]

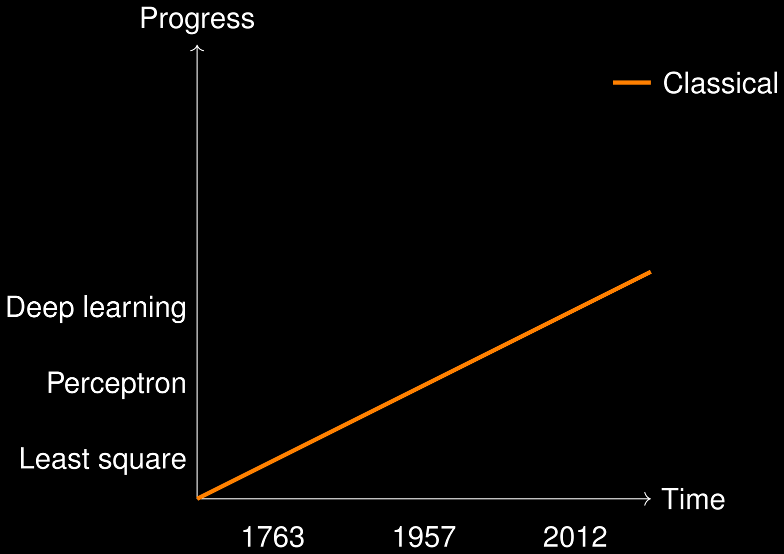
---

[3] Lemieux, et al. Efficient quantum walk circuits for Metropolis-Hastings algorithm. Quantum 4 (2020): 287.

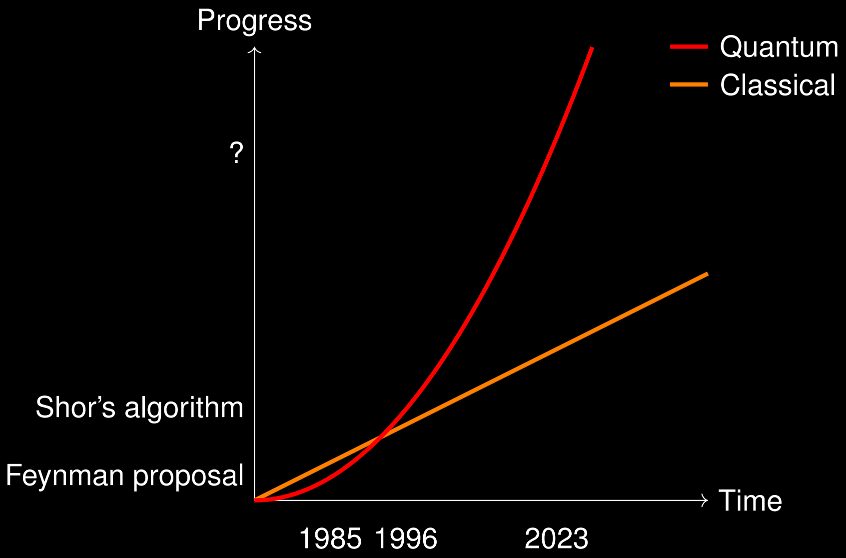
[4] Layden, Mazzola, et al. Quantum-enhanced Markov chain Monte Carlo. Nature 619.7969 (2023): 282-287.

Conclusions

Conclusions



Conclusions



Danke schön!

Any questions?

`massimiliano.incudini@univr.it`

`incud.github.io`