Lecture 12 – Quantum computers

TEK4500

04.11.2026

Håkon Jacobsen

hakon.jacobsen@its.uio.no

Shor's algorithm

1994

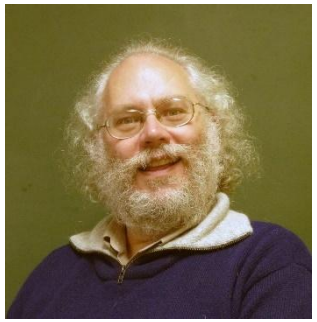
Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

Abstract

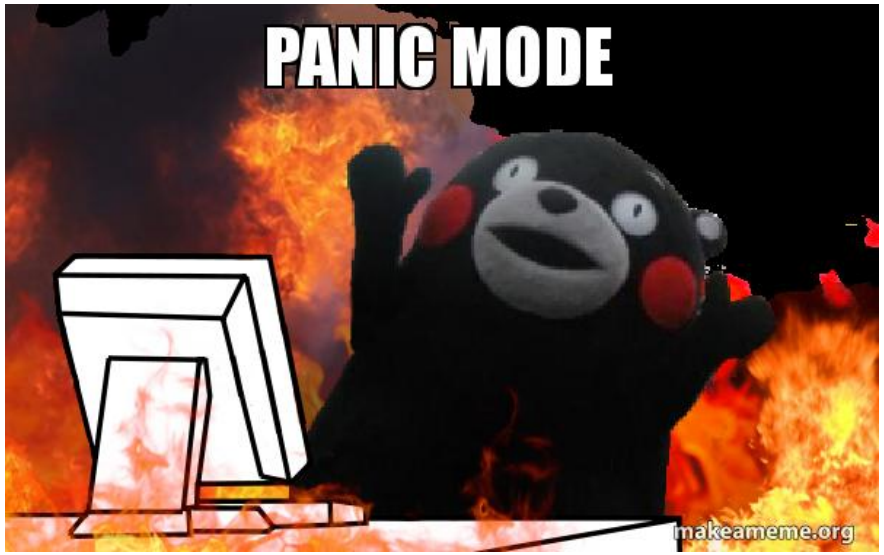
A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

Keywords: algorithmic number theory, prime factorization, discrete logarithms, Church's thesis, quantum computers, foundations of quantum mechanics, spin systems, Fourier transforms

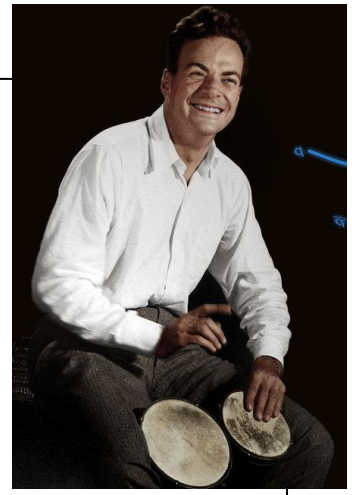


The quantum menace

- ~~Diffie-Hellman~~ – broken by Shor's algorithm
- ~~ECDSA~~ – broken by Shor's algorithm
- ~~RSA~~ – broken by Shor's algorithm
- …public-key cryptography is basically dead



Quantum computers – the beginning



International Journal of Theoretical Physics, Vol. 21, Nos. 6/7, 1982

Simulating Physics with Computers

Richard P. Feynman

Department of Physics, California Institute of Technology, Pasadena, California 91107

Received May 7, 1981

Qubits

- Classical bit:



- Qubit:

superposition **amplitudes**

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad \alpha, \beta \in \boldsymbol{C} \qquad |\alpha|^2 + |\beta|^2 = 1$$

We cannot observe α and β directly!

- Can **measure** $|\psi\rangle \Longrightarrow$ state collapses *randomly* to either $|0\rangle$ or $|1\rangle$ (with what probability?)

$$\Pr[|\psi\rangle \text{ collapses to } |0\rangle] = |\alpha|^2$$

$$\Pr[|\psi\rangle \text{ collapses to } |1\rangle] = |\beta|^2$$

$$|\psi\rangle = 0.6|0\rangle - 0.8|1\rangle$$

$$|0.6|^2 = 0.36 \qquad |-0.8|^2 = 0.64$$

Multiple qubits

$$|\psi\rangle = \alpha\,|00\rangle + \beta\,|01\rangle + \gamma\,|10\rangle + \delta\,|11\rangle$$

$$\alpha, \beta, \gamma, \delta \in \boldsymbol{C}$$

$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$$

- Can measure $|\psi\rangle \implies$ state collapses randomly to either $|00\rangle$, $|01\rangle$, $|10\rangle$ or $|11\rangle$ (with probability $|\alpha|^2$, $|\beta|^2$, …)
- Can also measure *just* the first qubit in $|\psi\rangle$
  - $\implies$ state collapses randomly to either $\alpha'|\mathbf{0}0\rangle + \beta'|\mathbf{0}1\rangle$ or to $\gamma'|\mathbf{1}0\rangle + \delta'|\mathbf{1}1\rangle$ (with probability $|\alpha|^2 + |\beta|^2$ or $|\gamma|^2 + |\delta|^2$)
- Can combine two independent qubits:

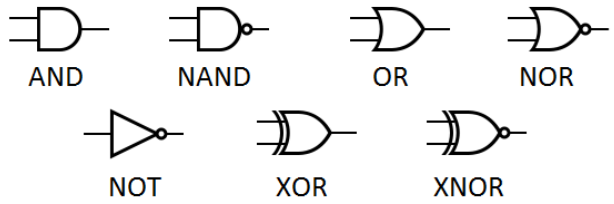
$$|\psi_0\rangle|\psi_1\rangle = (\alpha|0\rangle + \beta|1\rangle)(\gamma|0\rangle + \delta|1\rangle) = \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle$$

- $|\psi\rangle = \alpha_0|000\rangle + \alpha_1|001\rangle + \alpha_2|010\rangle + \alpha_3|011\rangle + \alpha_4|100\rangle + \alpha_5|101\rangle + \alpha_6|110\rangle + \alpha_7|111\rangle$
- N-qubit system: 2^N basis states

Quantum gates

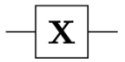
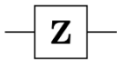
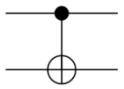
- Can change classical bits with logical gates

$$x \overset{\text{NOT}}{\longmapsto} \bar{x} \qquad x, y \overset{\text{AND}}{\longmapsto} x \wedge y$$



- Can change qubits with **quantum gates**

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad \overset{U}{\longmapsto} \quad |\psi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$$

Operator	Gate(s)
Pauli-X (X)	
Pauli-Z (Z)	
Hadamard (H)	
Controlled Not (CNOT, CX)	

- Constraints on U:
  - Normalization: $|\alpha'|^2 + |\beta'|^2 = 1$
  - Linearity: $U(\alpha|0\rangle + \beta|1\rangle) = \alpha \cdot U(|0\rangle) + \beta \cdot U(|1\rangle)$
  - $\Longrightarrow$ U must be *invertible*

(Quantum) NOT-gate

$$\boldsymbol{X}(|0\rangle) = |1\rangle$$

$$\boldsymbol{X}(|1\rangle) = |0\rangle$$

$$\boldsymbol{X}(|\psi\rangle) = \boldsymbol{X}(\alpha|0\rangle + \beta|1\rangle) = \alpha \cdot \boldsymbol{X}(|0\rangle) + \beta \cdot \boldsymbol{X}(|1\rangle) = \alpha|1\rangle + \beta|0\rangle$$

Hadamard gate

$$\boldsymbol{H}(|0\rangle) = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \overset{\text{def}}{=} |+\rangle$$

$$\boldsymbol{H}(|1\rangle) = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \overset{\text{def}}{=} |-\rangle$$

$$\boldsymbol{H}(|\psi\rangle) = \boldsymbol{H}(\alpha|0\rangle + \beta|1\rangle) = \alpha \cdot \boldsymbol{H}(|0\rangle) + \beta \cdot \boldsymbol{H}(|1\rangle) = \alpha|+\rangle + \beta|-\rangle$$

$$\Pr[|-\rangle \text{ collapses to } |0\rangle] = \left|\frac{+1}{\sqrt{2}}\right|^2 = 0.5$$

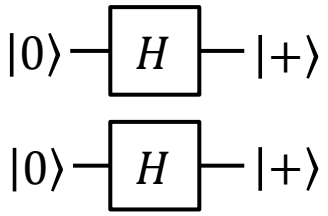
$$\Pr[|-\rangle \text{ collapses to } |1\rangle] = \left|\frac{-1}{\sqrt{2}}\right|^2 = 0.5$$

Hadamard gate

$$\boldsymbol{H}(|0\rangle) = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \overset{\text{def}}{=} |+\rangle$$

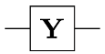
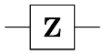
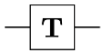
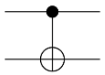
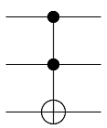
$$\boldsymbol{H}(|1\rangle) = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle \overset{\text{def}}{=} |-\rangle$$

$$\boldsymbol{H}(|\psi\rangle) = \boldsymbol{H}(\alpha|0\rangle + \beta|1\rangle) = \alpha \cdot \boldsymbol{H}(|0\rangle) + \beta \cdot \boldsymbol{H}(|1\rangle) = \alpha|+\rangle + \beta|-\rangle$$

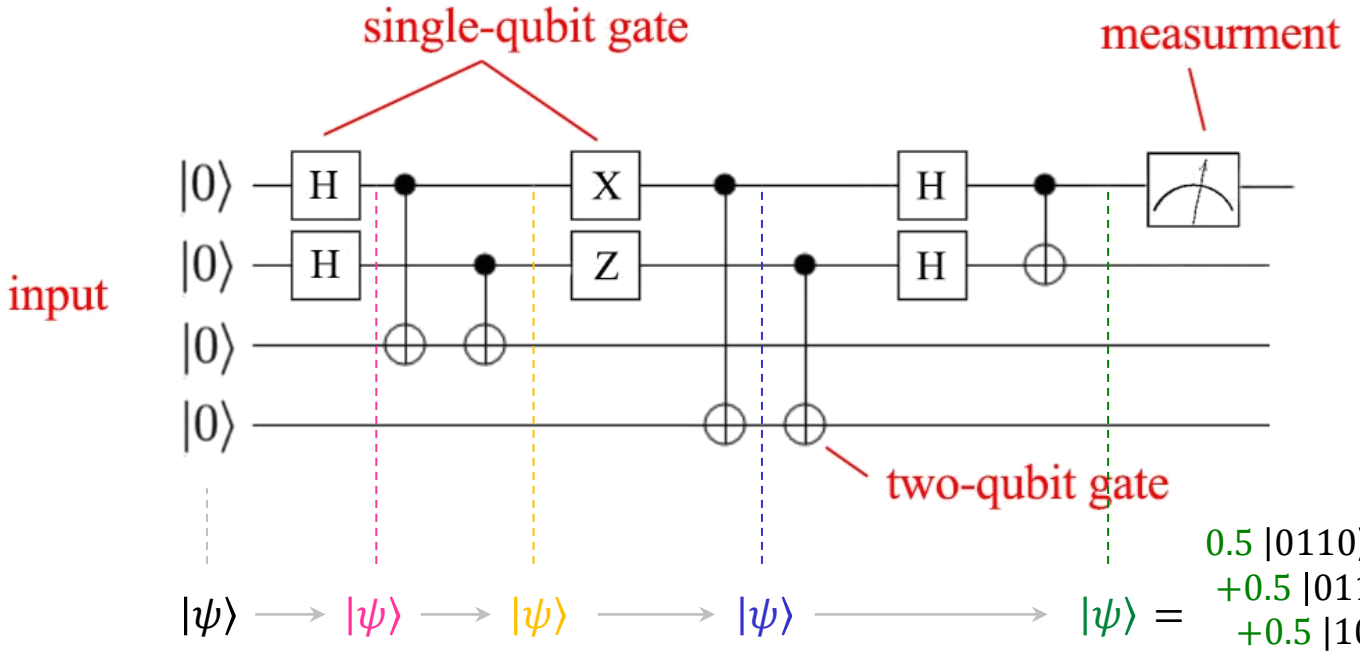


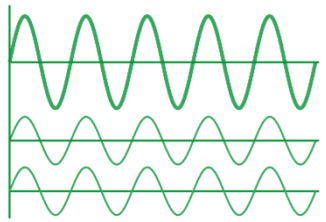
$$|+\rangle|+\rangle = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right)\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$$

(Infinitely) many more gates...

Operator	Gate(s)	Matrix
Pauli-X (X)		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Pauli-Y (Y)		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Pauli-Z (Z)		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Hadamard (H)		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$
Phase (S, P)		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$
$\pi/8$ (T)		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$
Controlled Not (CNOT, CX)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$
Controlled Z (CZ)		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Toffoli (CCNOT, CCX, TOFF)		$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Quantum computers

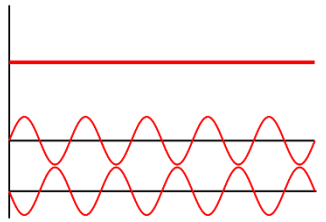




What makes quantum computation special?

Interference

$+0 \cdot |0000\rangle$
$+0 \cdot |0001\rangle$
$+0 \cdot |0010\rangle$
⋮



Deutsch–Jozsa's algorithm

- **Problem:** given only *oracle* access to f determine if it's **constant** or **balanced**

$\mathbf{x}$	$f(\mathbf{x})$
00	1
01	1
10	1
11	1

constant

$\mathbf{x}$	$f(\mathbf{x})$
00	0
01	1
10	1
11	0

balanced

Deutsch–Jozsa's algorithm

- **Problem:** given only *oracle* access to f determine if it's **constant** or **balanced**

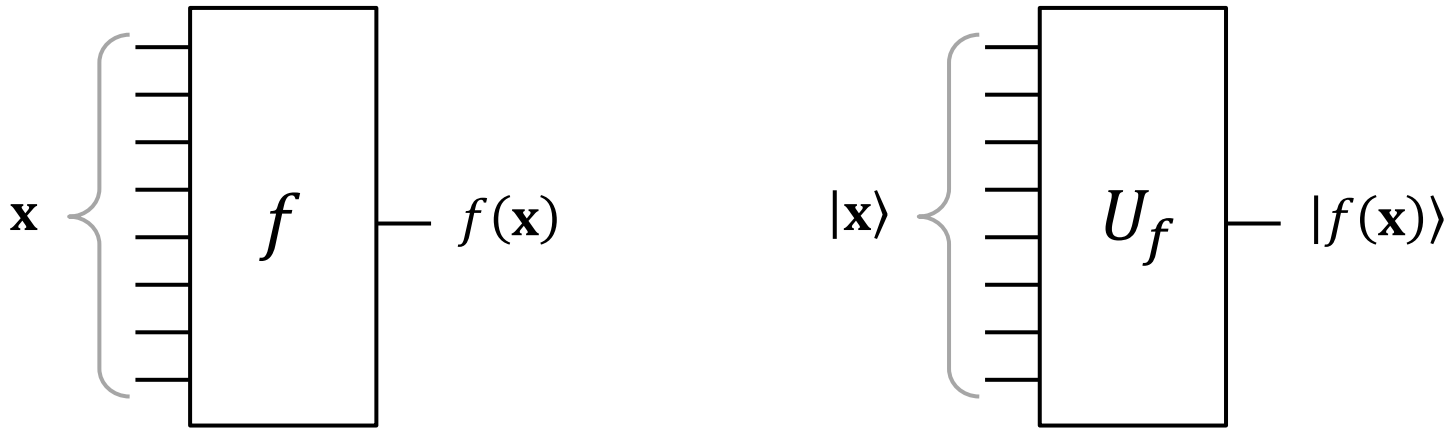
$\mathbf{x}$	$f(\mathbf{x})$
00	0
01	0
10	1
11	

balanced

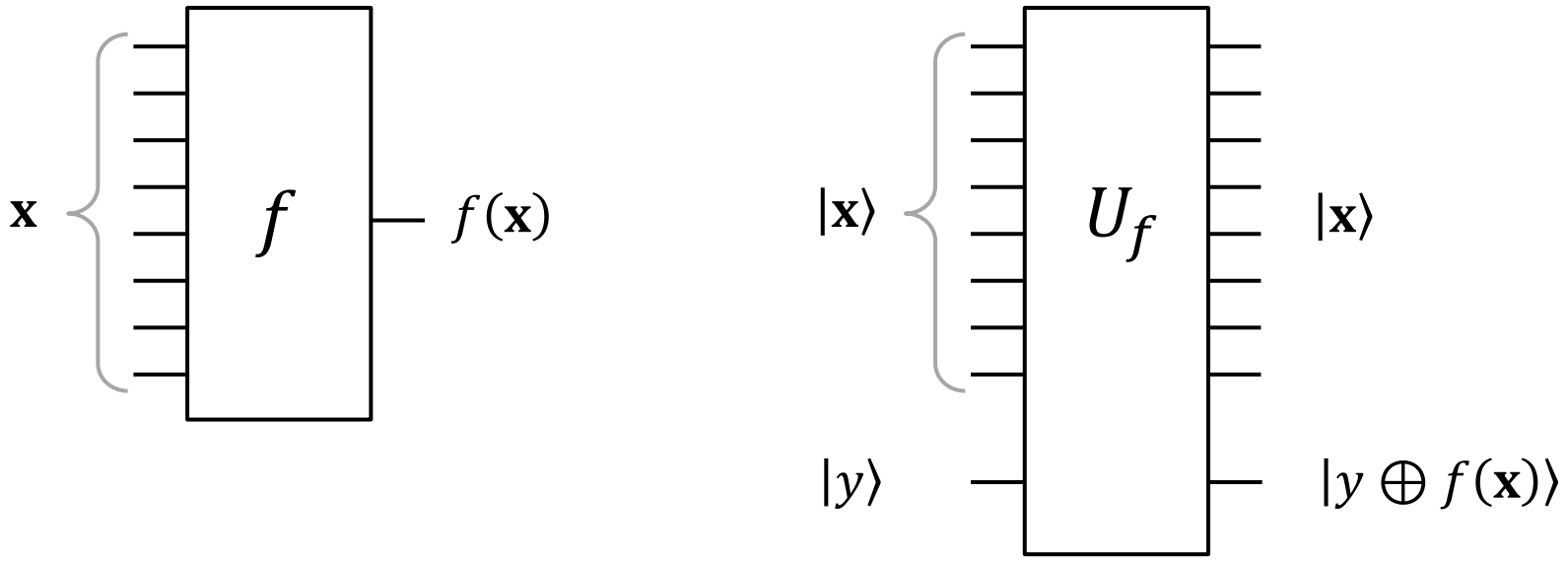
- Classic algorithm must test $2^n/2 + 1$ inputs in the worst case

- Can a quantum computer do better?

Deutsch–Jozsa's algorithm

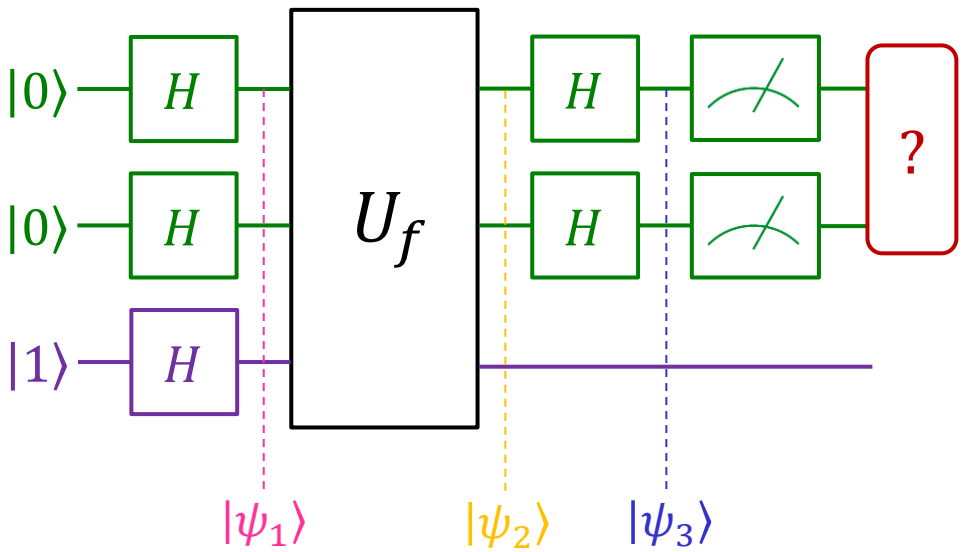


Deutsch–Jozsa's algorithm



Deutsch–Jozsa's algorithm

$$|\mathbf{x}\rangle|y\rangle \overset{U_f}{\longmapsto} |\mathbf{x}\rangle|y \oplus f(\mathbf{x})\rangle$$



$$|\psi_1\rangle = |+\rangle|+\rangle|-\rangle = (|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$$
$$= |00\rangle|0\rangle - |00\rangle|1\rangle + |01\rangle|0\rangle - |01\rangle|1\rangle + |10\rangle|0\rangle - |10\rangle|1\rangle + |11\rangle|0\rangle - |11\rangle|1\rangle$$

$$|\psi_2\rangle = |00\rangle|0 \oplus f(00)\rangle - |00\rangle|1 \oplus f(00)\rangle = (-1)^{f(00)}\,|00\rangle(|0\rangle - |1\rangle)$$
$$+ |01\rangle|0 \oplus f(01)\rangle - |01\rangle|1 \oplus f(01)\rangle = (-1)^{f(01)}\,|01\rangle(|0\rangle - |1\rangle)$$
$$+ |10\rangle|0 \oplus f(10)\rangle - |10\rangle|1 \oplus f(10)\rangle \quad \ddots$$
$$+ |11\rangle|0 \oplus f(11)\rangle - |11\rangle|1 \oplus f(11)\rangle$$

$$= \left((-1)^{f(00)}|00\rangle + (-1)^{f(01)}|01\rangle + (-1)^{f(10)}|10\rangle + (-1)^{f(11)}|11\rangle\right)(|0\rangle - |1\rangle)$$

$$|\psi_3\rangle = \begin{pmatrix} (-1)^{f(00)}|+\rangle|+\rangle \\ + \\ (-1)^{f(01)}|+\rangle|-\rangle \\ + \\ (-1)^{f(10)}|-\rangle|+\rangle \\ + \\ (-1)^{f(11)}|-\rangle|-\rangle \end{pmatrix}(|0\rangle - |1\rangle) = \begin{pmatrix} (-1)^{f(00)}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \\ + \\ (-1)^{f(01)}(|00\rangle - |01\rangle + |10\rangle - |11\rangle) \\ + \\ (-1)^{f(10)}(|00\rangle + |01\rangle - |10\rangle - |11\rangle) \\ + \\ (-1)^{f(11)}(|00\rangle - |01\rangle - |10\rangle + |11\rangle) \end{pmatrix}(|0\rangle - |1\rangle)$$

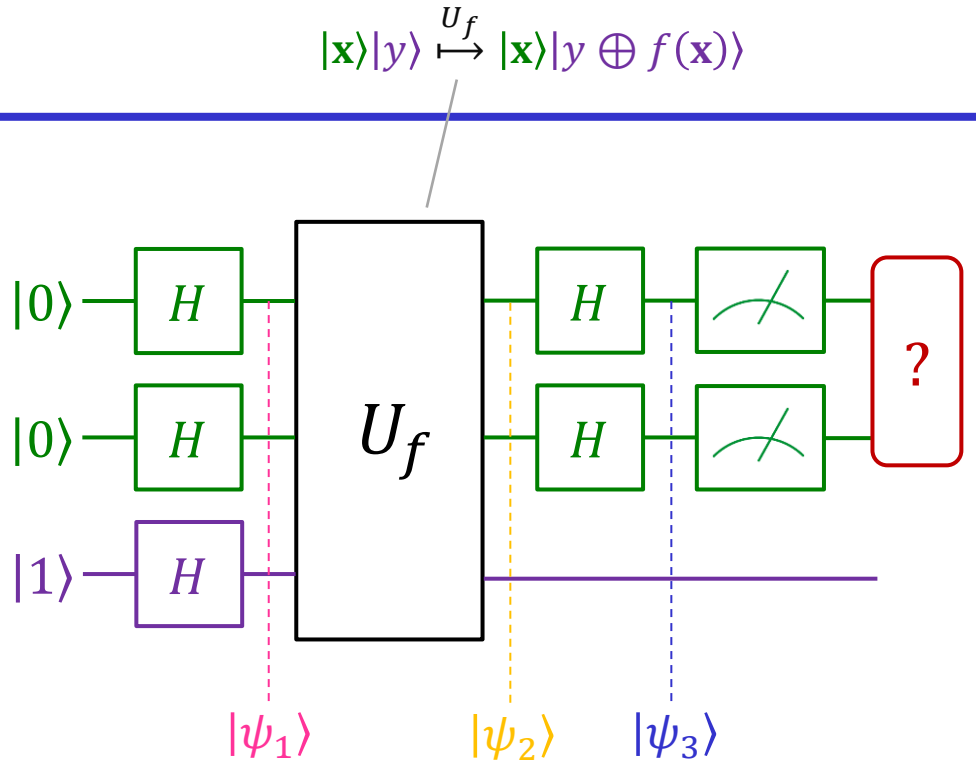
Deutsch–Jozsa's algorithm

$$|\psi_1\rangle = |+\rangle|+\rangle|-\rangle = (|0\rangle + |1\rangle)(|0\rangle + |1\rangle)(|0\rangle - |1\rangle)$$
$$= |00\rangle|0\rangle - |00\rangle|1\rangle + |01\rangle|0\rangle - |01\rangle|1\rangle + |10\rangle|0\rangle - |10\rangle|1\rangle + |11\rangle|0\rangle - |11\rangle|1\rangle$$

$$|\psi_2\rangle = |00\rangle|0 \oplus f(00)\rangle - |00\rangle|1 \oplus f(00)\rangle + |01\rangle|0 \oplus f(01)\rangle - |01\rangle|1 \oplus f(01)\rangle + |10\rangle|0 \oplus f(10)\rangle - |10\rangle|1 \oplus f(10)\rangle + |11\rangle|0 \oplus f(11)\rangle - |11\rangle|1 \oplus f(11)\rangle$$

$$= \left((-1)^{f(00)}|00\rangle + (-1)^{f(01)}|01\rangle + (-1)^{f(10)}|10\rangle + (-1)^{f(11)}|11\rangle\right)(|0\rangle - |1\rangle)$$

$$|\psi_3\rangle = \begin{pmatrix} (-1)^{f(00)}|+\rangle|+\rangle \\ + \\ (-1)^{f(01)}|+\rangle|-\rangle \\ + \\ (-1)^{f(10)}|-\rangle|+\rangle \\ + \\ (-1)^{f(11)}|-\rangle|-\rangle \end{pmatrix}(|0\rangle - |1\rangle) = \begin{pmatrix} (-1)^{f(00)}(|00\rangle \cancel{+ |01\rangle + |10\rangle + |11\rangle}) \\ + \\ (-1)^{f(01)}(|00\rangle \cancel{- |01\rangle + |10\rangle - |11\rangle}) \\ + \\ (-1)^{f(10)}(|00\rangle \cancel{+ |01\rangle - |10\rangle - |11\rangle}) \\ + \\ (-1)^{f(11)}(|00\rangle \cancel{- |01\rangle - |10\rangle + |11\rangle}) \end{pmatrix}(|0\rangle - |1\rangle)$$



Case 1: $f(\mathbf{x})$ constant

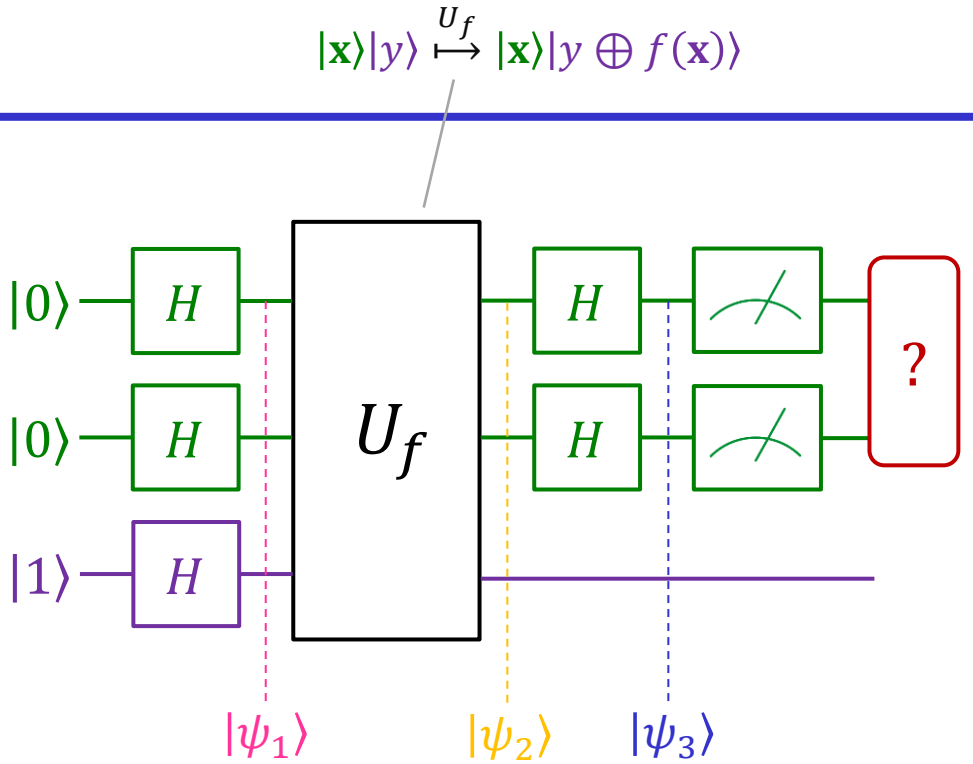
Deutsch–Jozsa's algorithm

$$|\psi_1\rangle = |+\rangle|+\rangle|-\rangle = (|0\rangle+|1\rangle)(|0\rangle+|1\rangle)(|0\rangle-|1\rangle)$$
$$= |00\rangle|0\rangle - |00\rangle|1\rangle + |01\rangle|0\rangle - |01\rangle|1\rangle + |10\rangle|0\rangle - |10\rangle|1\rangle + |11\rangle|0\rangle - |11\rangle|1\rangle$$

$$|\psi_2\rangle = |00\rangle|0\oplus f(00)\rangle - |00\rangle|1\oplus f(00)\rangle + |01\rangle|0\oplus f(01)\rangle - |01\rangle|1\oplus f(01)\rangle + |10\rangle|0\oplus f(10)\rangle - |10\rangle|1\oplus f(10)\rangle + |11\rangle|0\oplus f(11)\rangle - |11\rangle|1\oplus f(11)\rangle$$

$$= \left((-1)^{f(00)}|00\rangle + (-1)^{f(01)}|01\rangle + (-1)^{f(10)}|10\rangle + (-1)^{f(11)}|11\rangle\right)(|0\rangle-|1\rangle)$$

$$|\psi_3\rangle = \begin{pmatrix} (-1)^{f(00)}|+\rangle|+\rangle \\ + \\ (-1)^{f(01)}|+\rangle|-\rangle \\ + \\ (-1)^{f(10)}|-\rangle|+\rangle \\ + \\ (-1)^{f(11)}|-\rangle|-\rangle \end{pmatrix}(|0\rangle-|1\rangle) = \begin{pmatrix} (-1)^{f(00)}(\cancel{|00\rangle} + |01\rangle + |10\rangle + |11\rangle) \\ + \\ (-1)^{f(01)}(\cancel{|00\rangle} - |01\rangle + |10\rangle - |11\rangle) \\ + \\ (-1)^{f(10)}(\cancel{|00\rangle} + |01\rangle - |10\rangle - |11\rangle) \\ + \\ (-1)^{f(11)}(\cancel{|00\rangle} - |01\rangle - |10\rangle + |11\rangle) \end{pmatrix}(|0\rangle-|1\rangle)$$

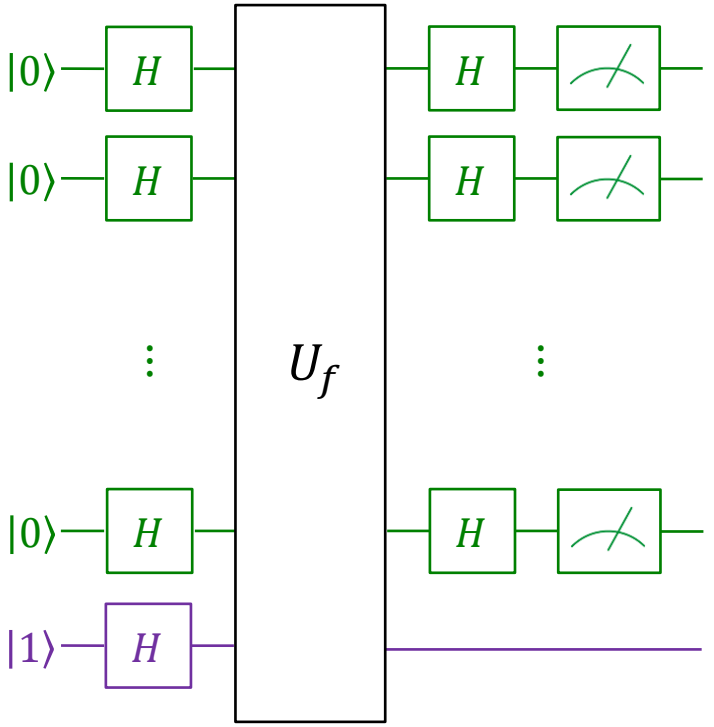


Case 2: $f(\mathbf{x})$ balanced

$\mathbf{x}$	$f(\mathbf{x})$
00	0
01	0
10	1
11	1

Deutsch–Jozsa's algorithm

- Easily generalized to functions of n bits
- Classical algorithm must make $2^n/2 + 1$ calls to f in the worst-case
- Deutsch–Jozsa's algorithm distinguishes constant and balanced f with **one** call to U_f



Shor's algorithm

1994

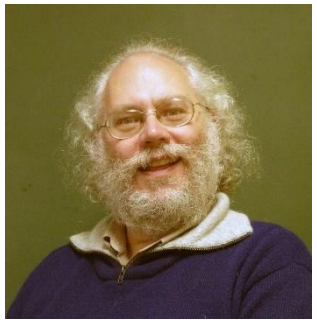
Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

Abstract

A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

Keywords: algorithmic number theory, prime factorization, discrete logarithms, Church's thesis, quantum computers, foundations of quantum mechanics, spin systems, Fourier transforms



DLOG to period finding

- **DLOG problem:** Given $A = g^a$ find a.

- Define a function $f : \mathbf{Z}_p \times \mathbf{Z}_p \to \boldsymbol{G}$ as

$$f(x, y) = g^x A^y$$

- **Observation:** function f is *periodic* with period $(a \cdot \omega, -\omega)$ for all $\omega \in \boldsymbol{Z}_p$:

$$f(x + a \cdot \omega, y - \omega) = g^{x+a\cdot\omega} A^{y-\omega} = g^{x+a\cdot\omega} g^{ay-a\omega} = g^x A^y = f(x, y)$$

- **Conclusion:** finding a nonzero period $(\overbrace{\varphi}^{a\cdot\omega}, -\omega)$ allows us to compute a:

$$a \leftarrow \varphi \cdot \omega^{-1} \pmod{p}$$

Shor's algorithm

Shor's algorithm

Input: $A = g^a$

Output: a

1. find a period $(\varphi, -\omega)$ of $f(x, y) = g^x A^y$
2. **if** $(\varphi, -\omega) = (0,0)$:
3. **goto** 1.
4. **return** $\varphi \cdot \omega^{-1} \pmod{p}$

where the quantum magic happens!

Shor’s algorithm

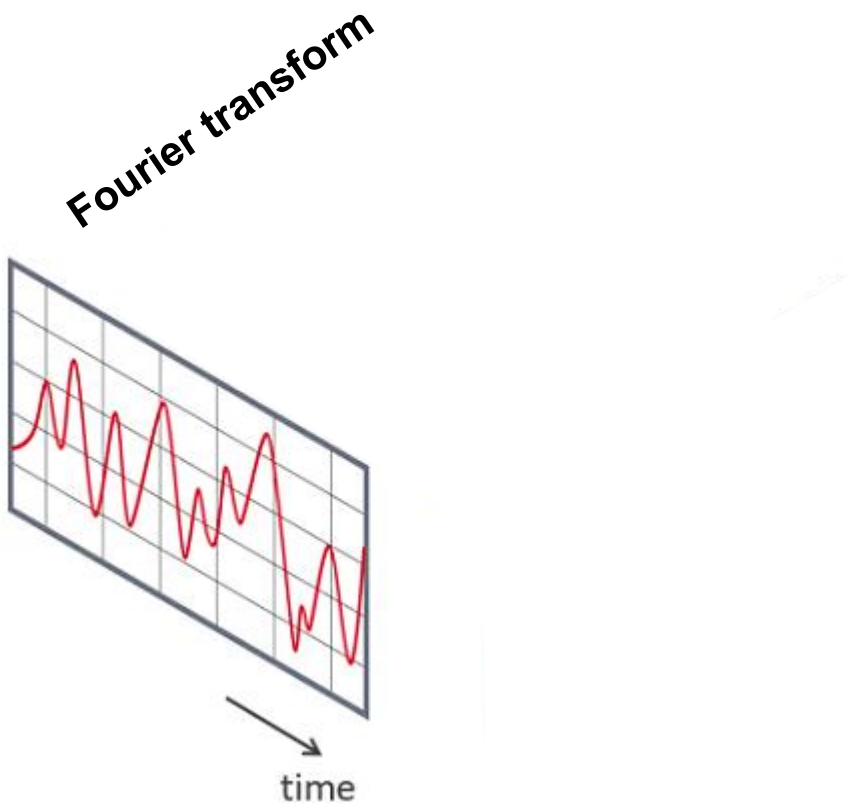
- Problem: period can be very large
  - Classical solutions take exponential time

- Observation:

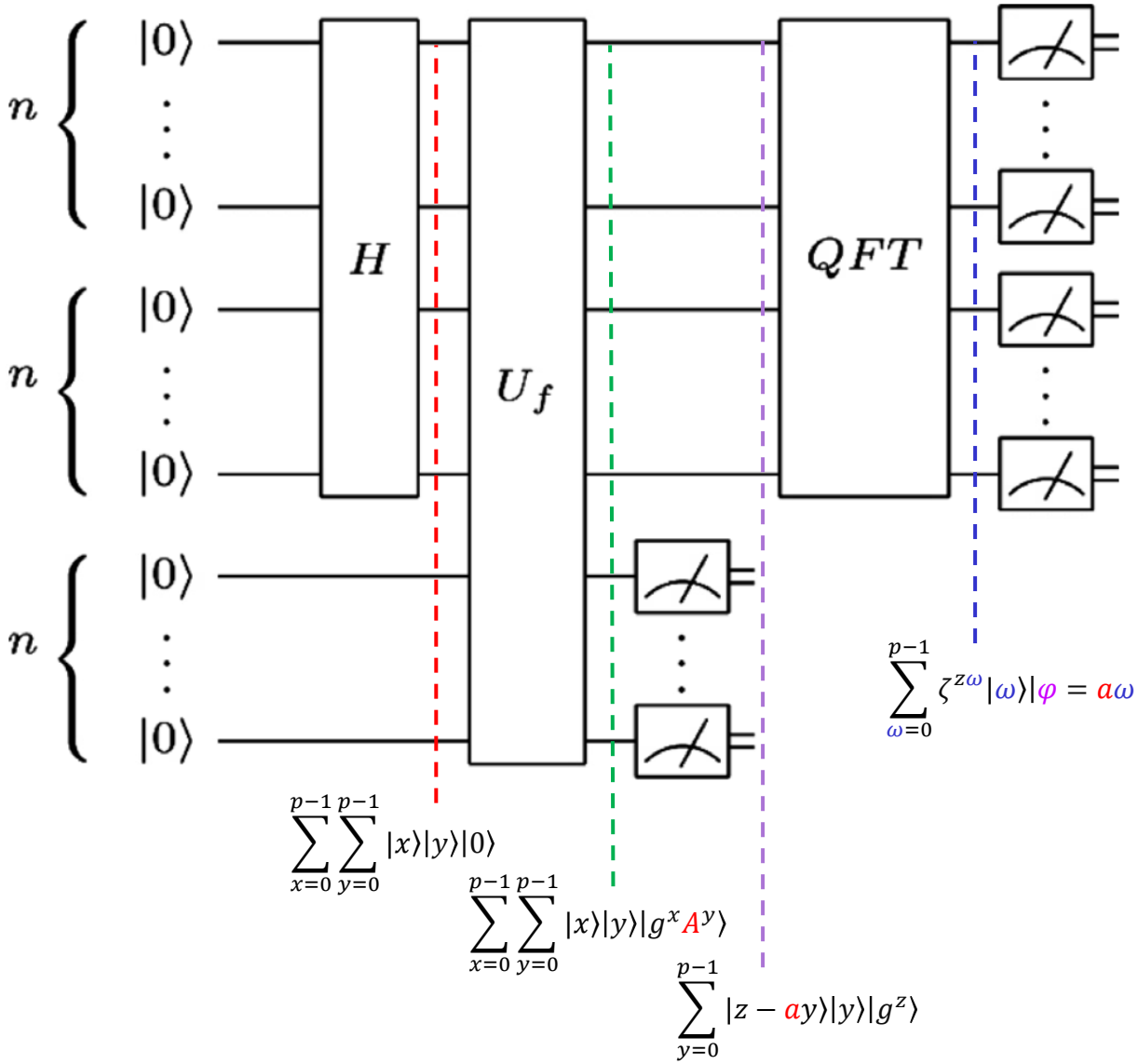
  finding signal period ⟺ finding signal frequencies

- Key ingredient of Shor's algorithm:

  Quantum Fourier Transform (QFT)



Shor's algorithm



Measure to get ω and $\varphi = a\omega \pmod p$

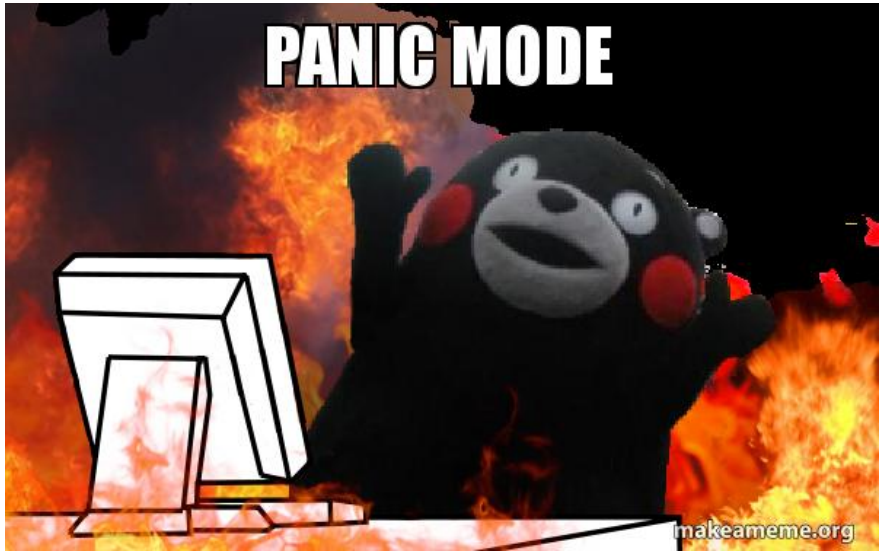
$\Rightarrow \quad a = \varphi \cdot \omega^{-1} \pmod p$

$\zeta = e^{2\pi i/p}$

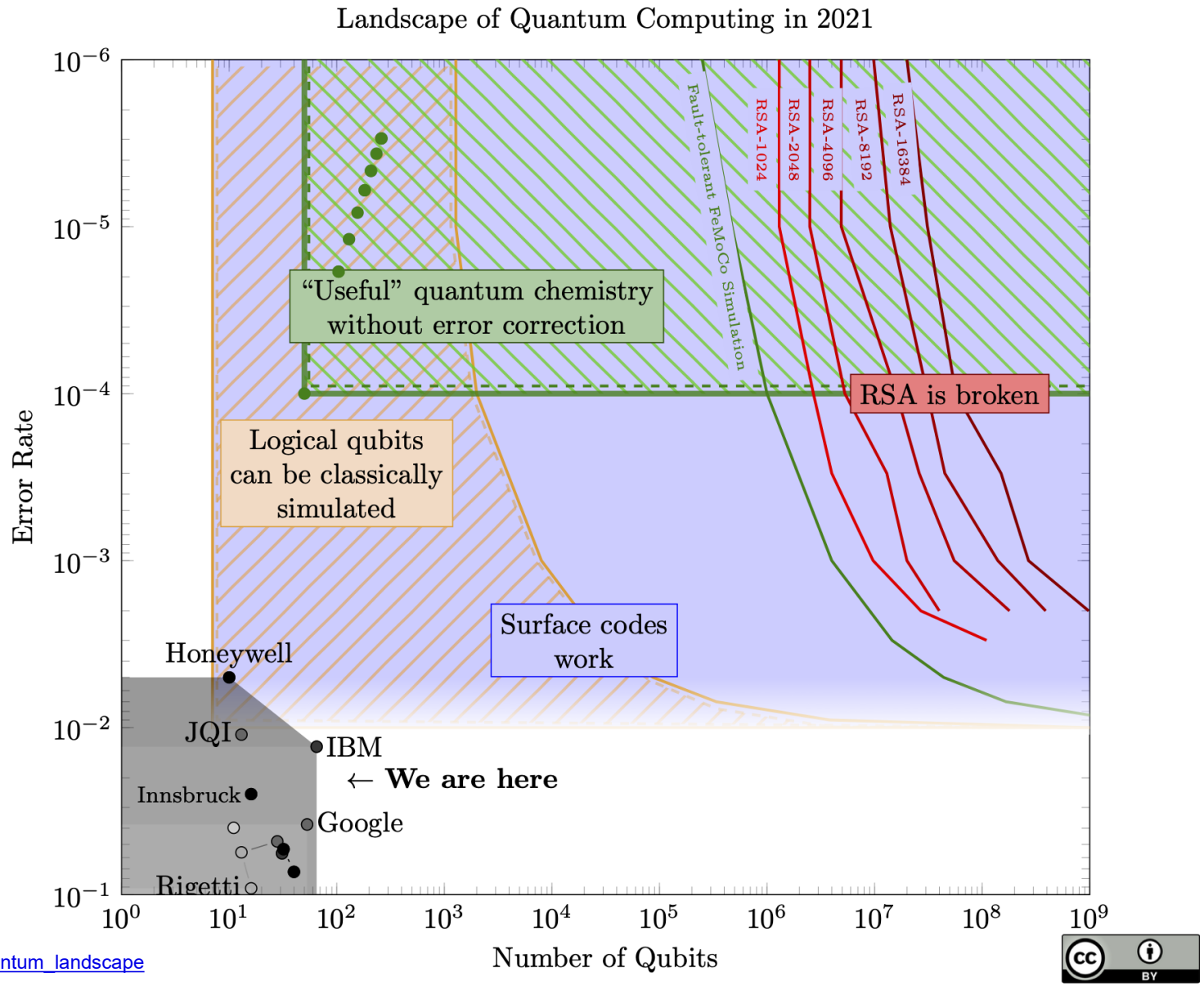
* simplified

The quantum menace

- ~~Diffie-Hellman~~ – broken by Shor's algorithm
- ~~ECDSA~~ – broken by Shor's algorithm
- ~~RSA~~ – broken by Shor's algorithm
- …public-key cryptography is basically dead



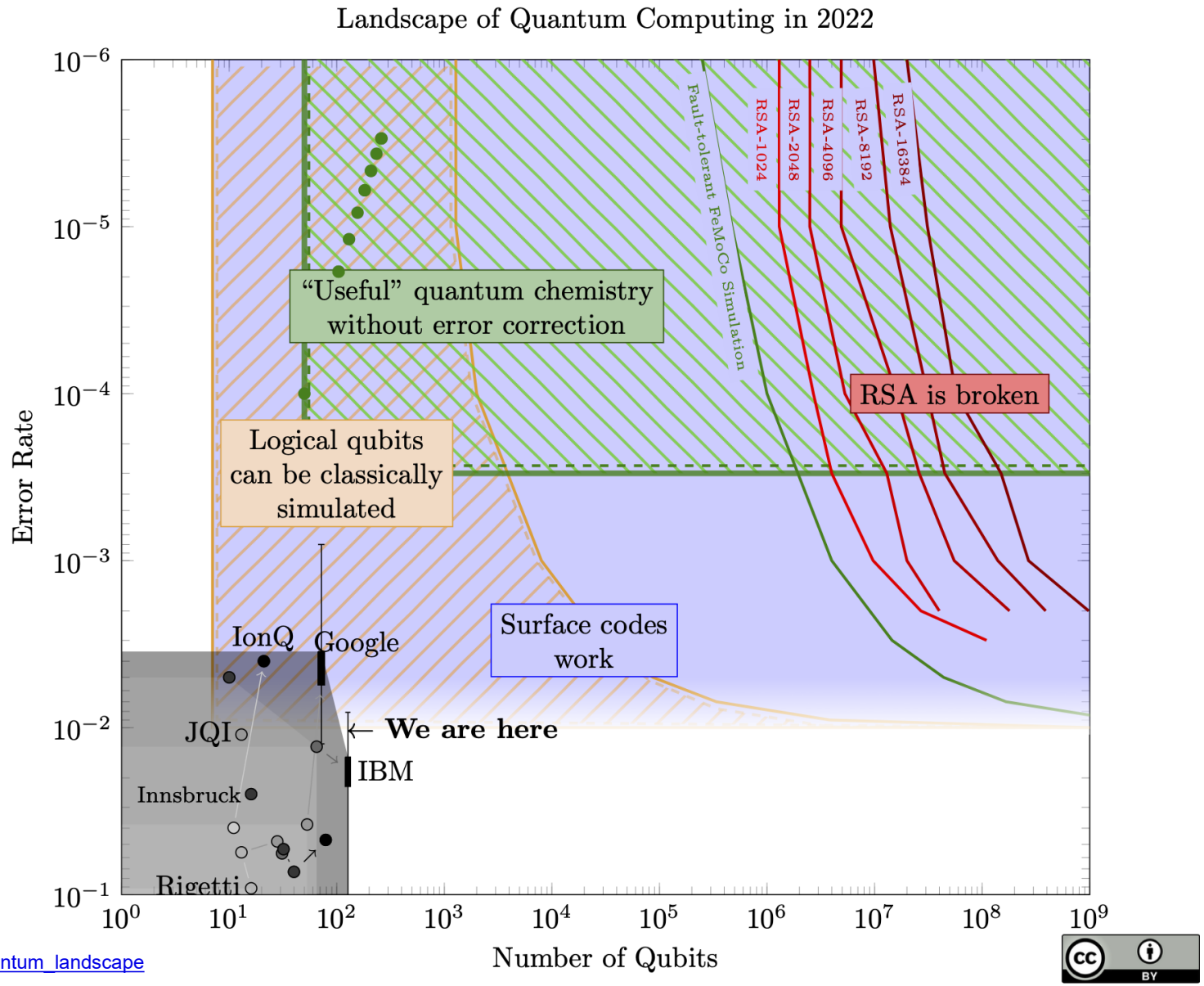
How far away are we?



https://sam-jaques.appspot.com/quantum_landscape

CC BY

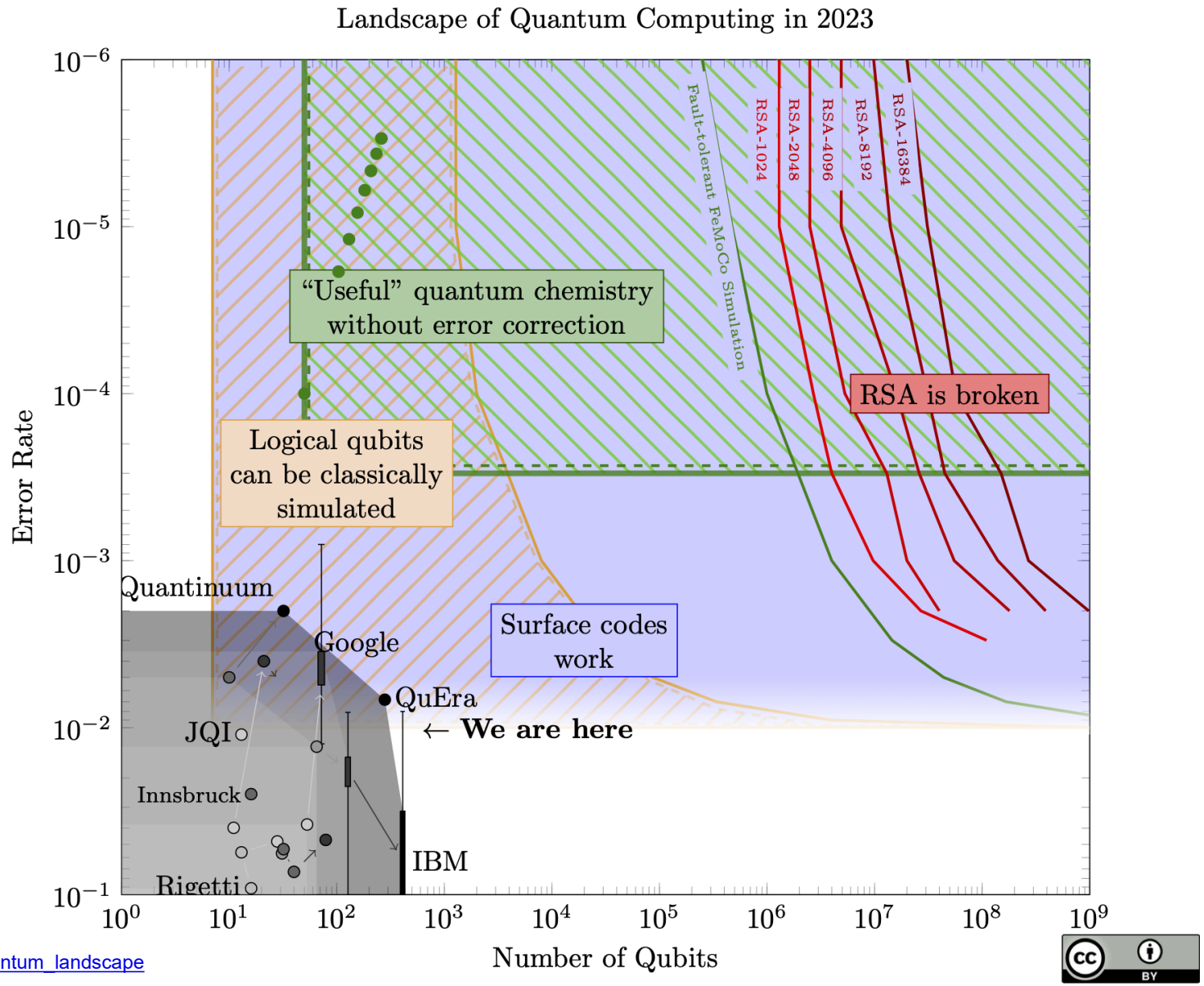
How far away are we?



https://sam-jaques.appspot.com/quantum_landscape

CC BY

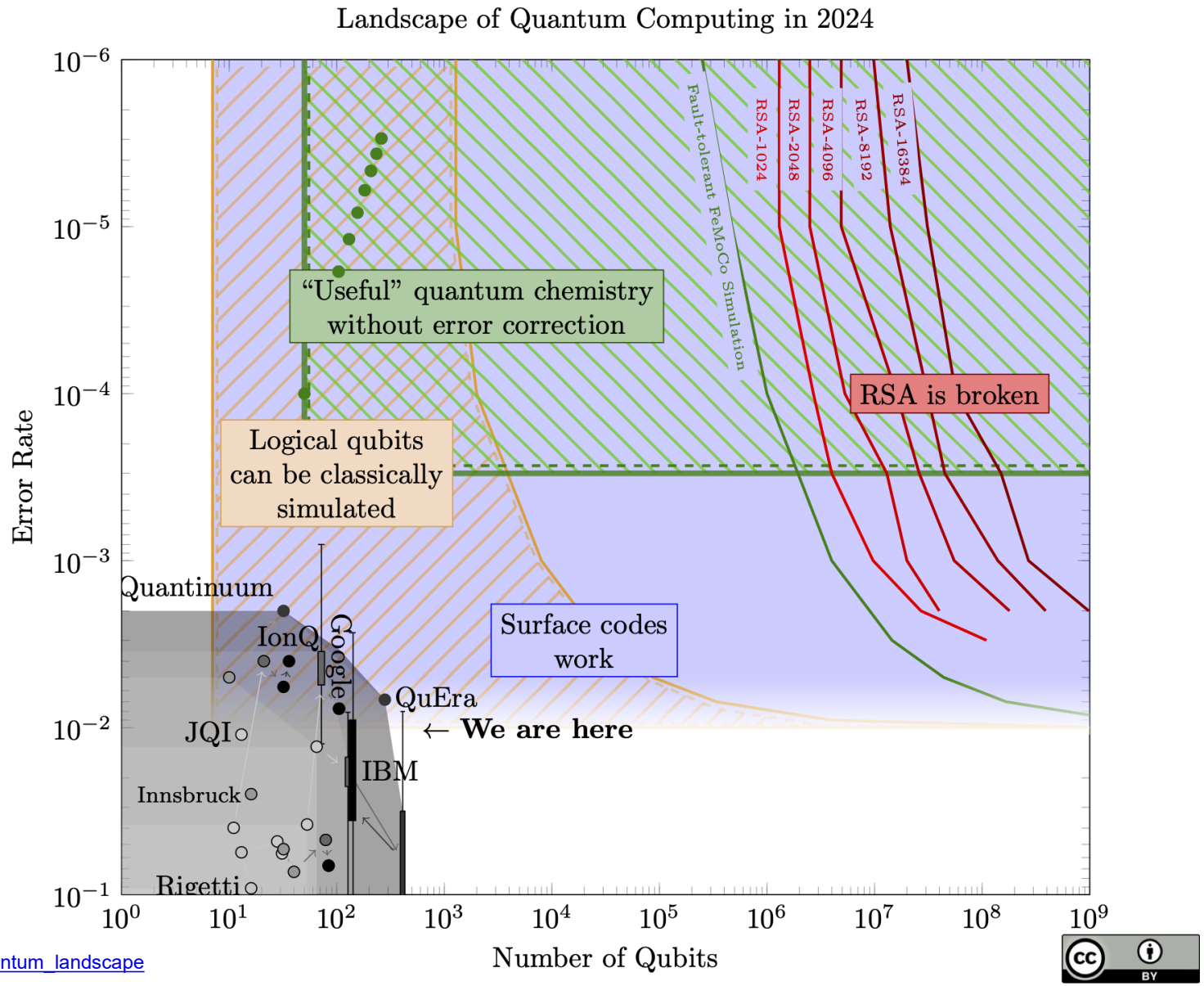
How far away are we?



https://sam-jaques.appspot.com/quantum_landscape

CC BY

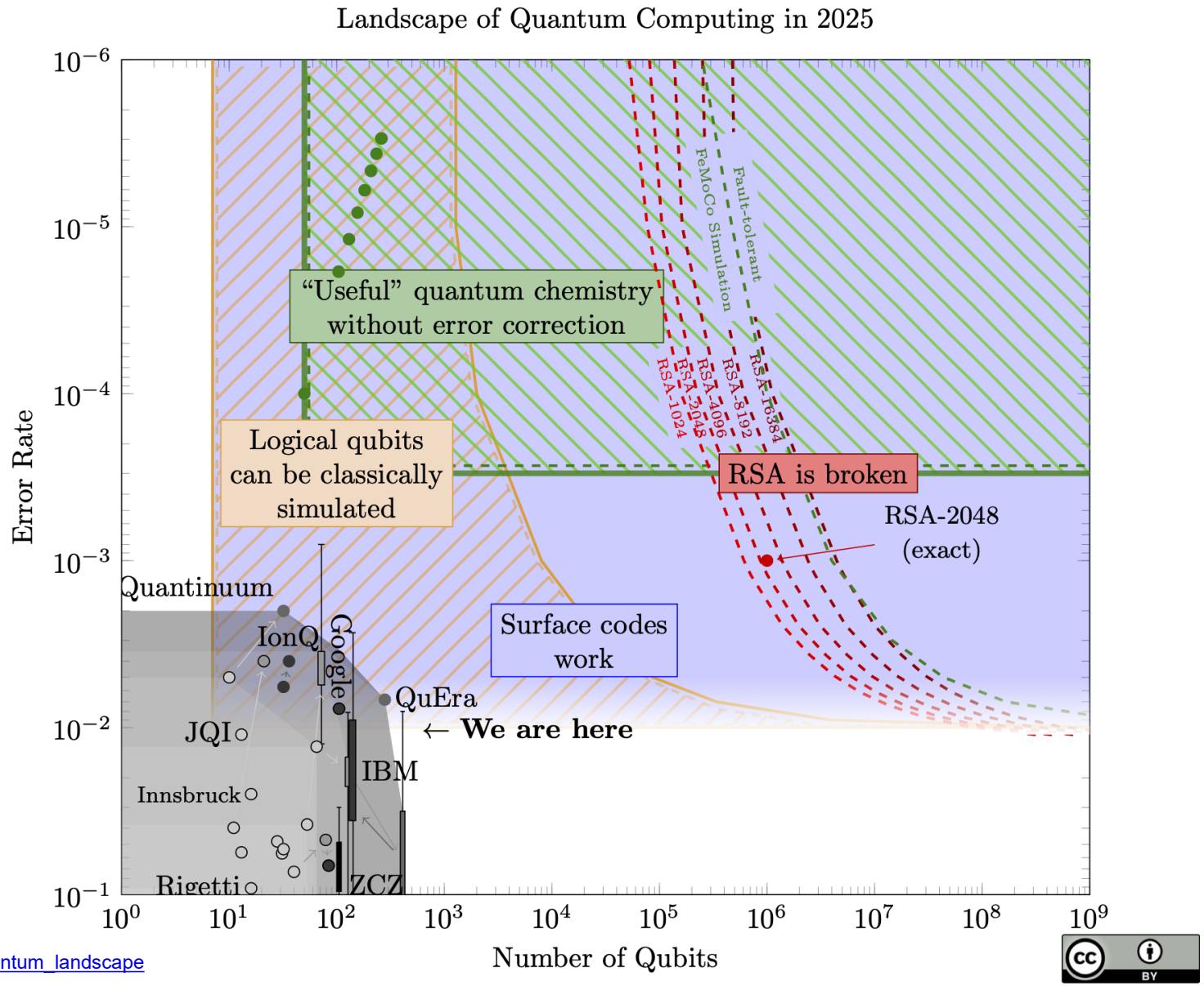
How far away are we?



https://sam-jaques.appspot.com/quantum_landscape

CC BY

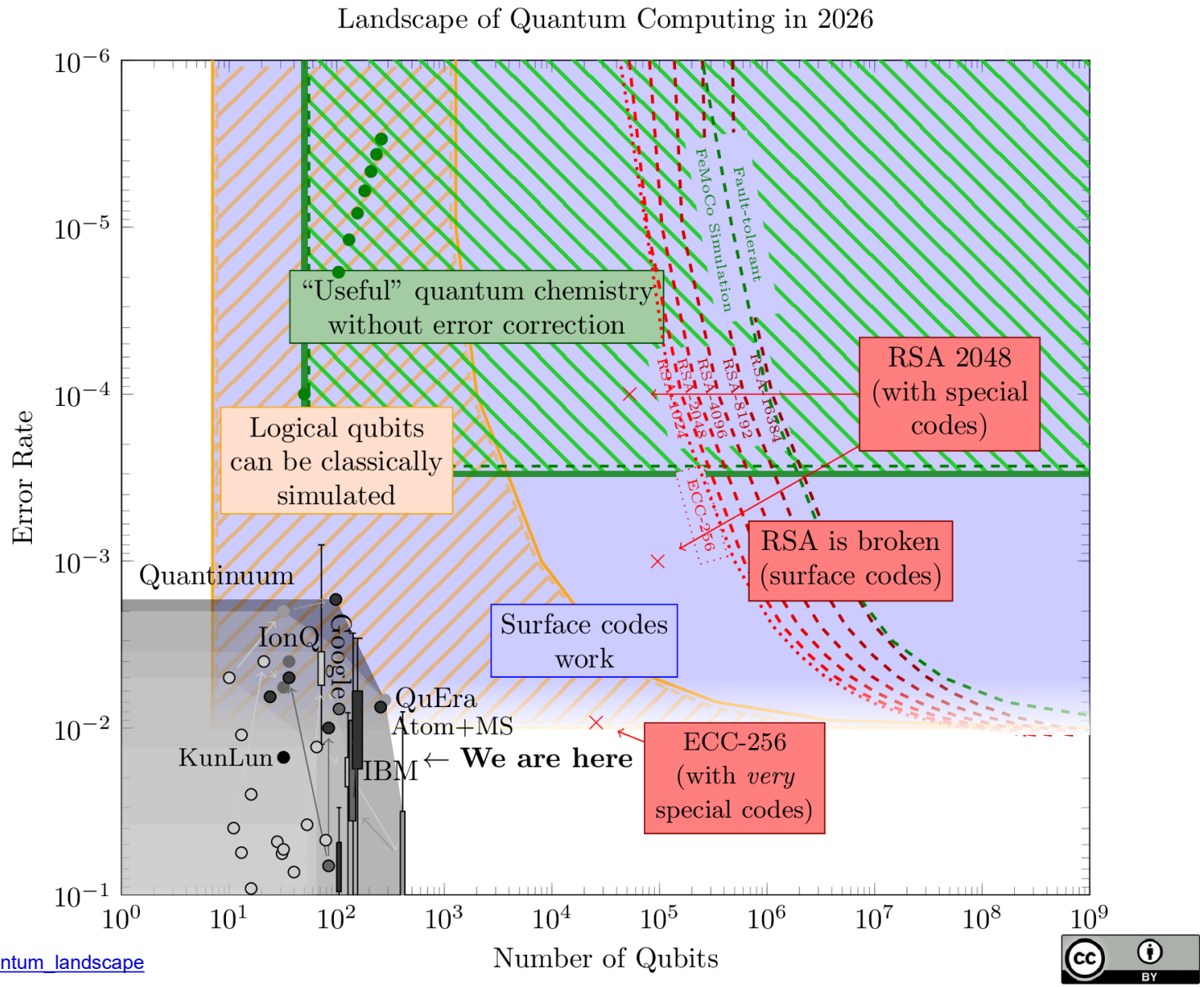
How far away are we?



https://sam-jaques.appspot.com/quantum_landscape

CC BY

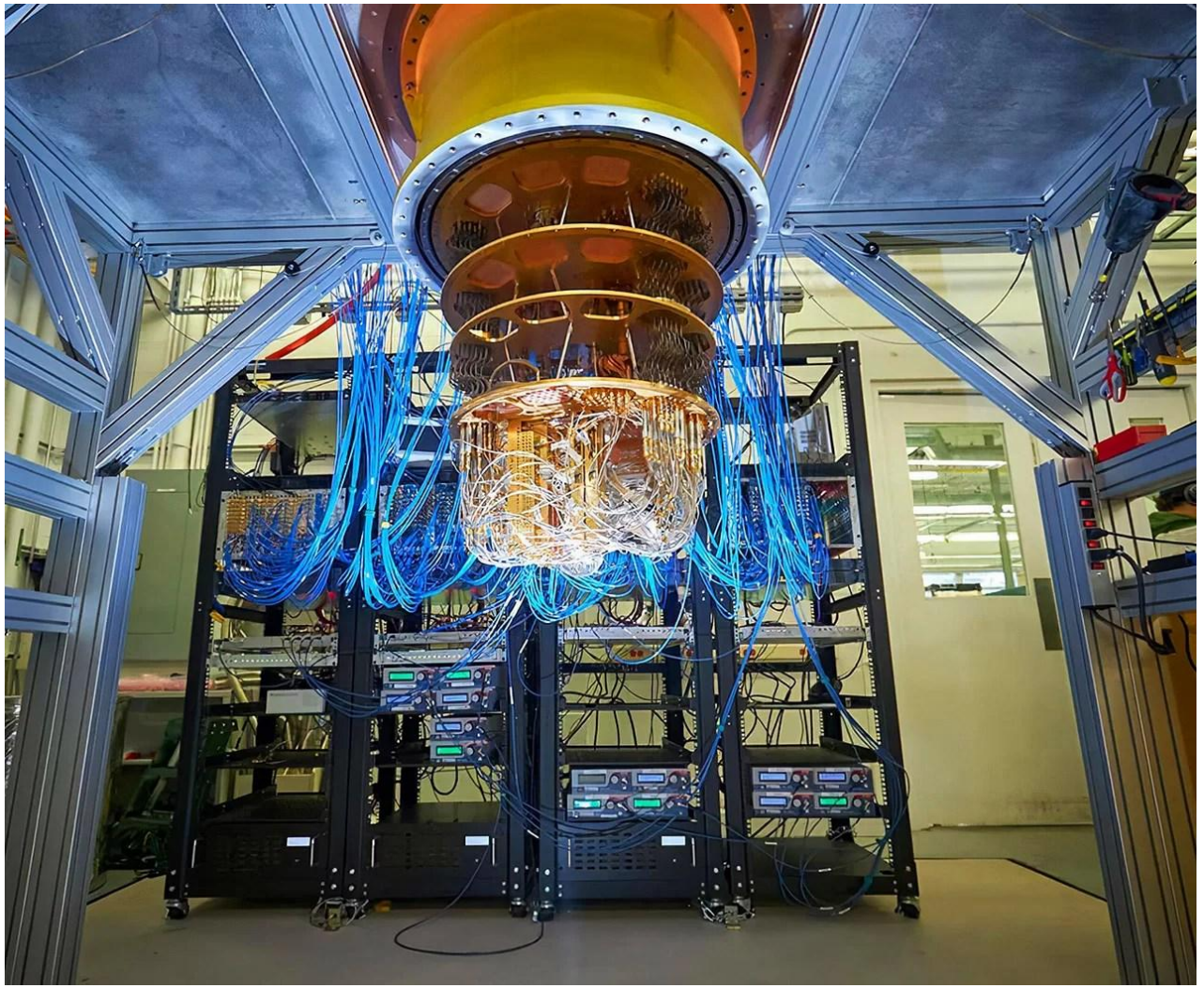
How far away are we?



https://sam-jaques.appspot.com/quantum_landscape

CC BY

Quantum computers



Dealing with quantum computers

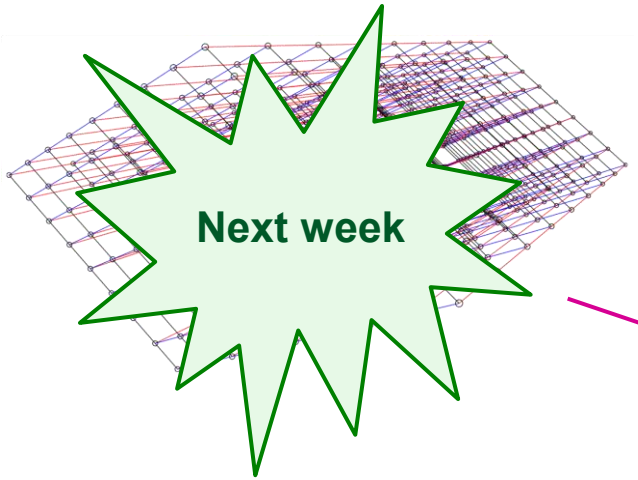
- **Symmetric cryptography**
  - Grover's algorithm: solves $\mathcal{O}(2^n)$ problems in $\mathcal{O}(2^{n/2})$ quantum steps
  - Inherently serial + huge constants
  - AES-128 is most likely safe; using AES-256 removes any doubts

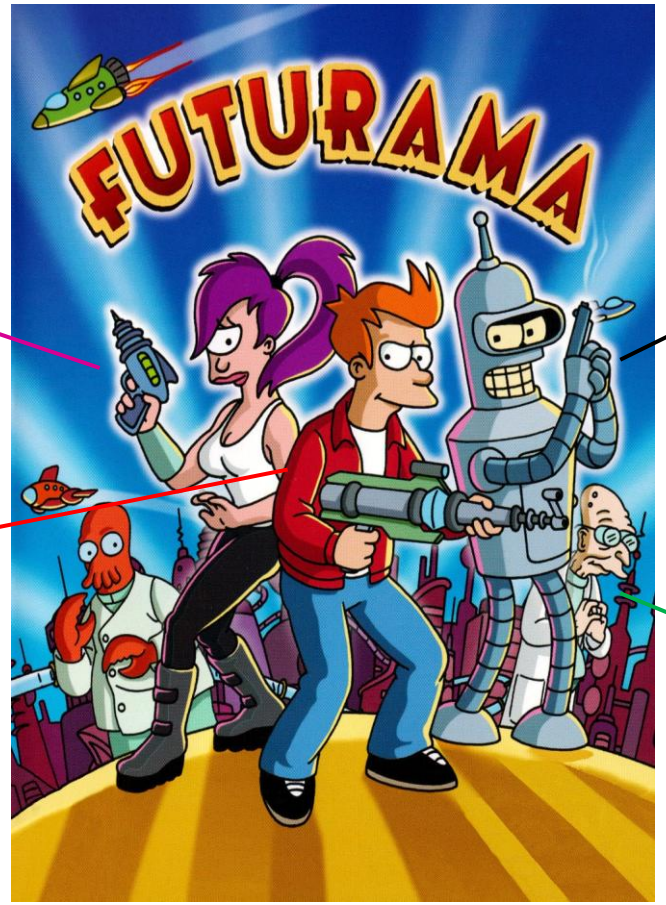
- **Quantum cryptography**
  - Use quantum mechanics to build cryptography
  - Requires specialized equipment
  - Only used for key distribution; does not solve authentication problem

- **Post-quantum cryptography**
  - Classical (asymmetric) algorithms believed to withstand quantum attacks

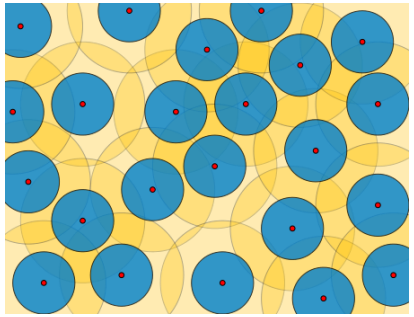
Post-quantum cryptography

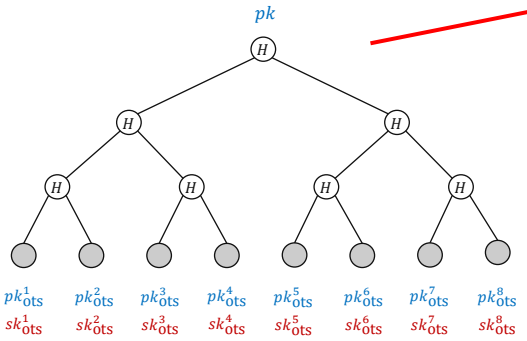
Lattice-based cryptography





Code-based encryption

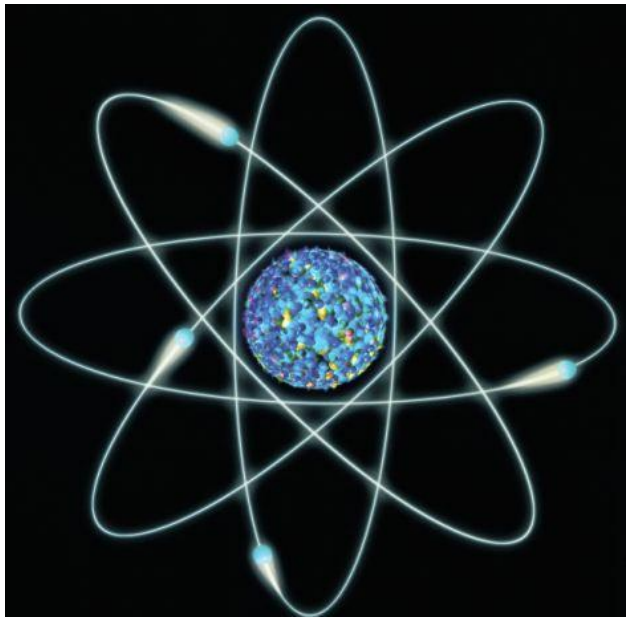




Hash-based signatures



Supersingular isogeny key exchange



Questions?