IQM

Introduction to Quantum Computing with IQM Resonance

IQM Summer Quantum School 2026

August 25, 2026

Dr. Danny Bulmash, Team Lead, Education and Community

Dr. Nadia Milazzo, Senior Quantum Education Specialist

Niklas Steinmann, Community Manager

www.iqm.tech

Outline of the day

In today‘s sessions **we** will ...

- ... learn what problems quantum computers can tackle.
- ... learn how quantum computers are different to classical computers.
- ... use **Qrisp** and **IQM Resonance** to perform our first quantum computations on real quantum computers

Your instructors

Our Mission

We build world leading quantum computers for the well-being of humankind, now and for the future

IQM by the numbers

400+
Employees

136
PHDs

$600M
Funding

2018
Founded

50+
Nationalities

150
Qubit chip in development

30+
Full systems built

26
Full systems sold

12,000+
Cloud users

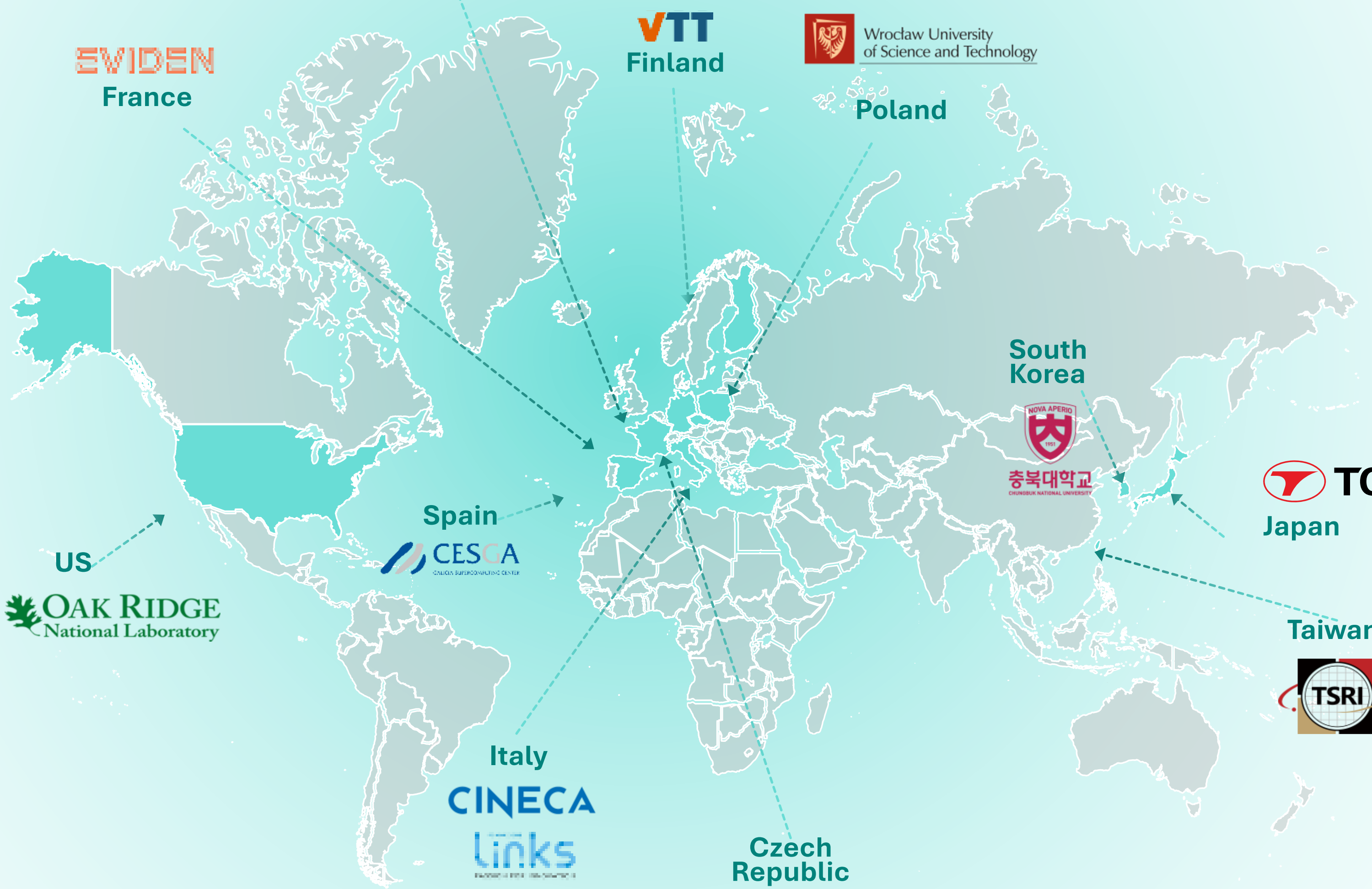
IQM Customers

30+
Full systems built

26
Full systems sold

12,000+
Cloud users

21%
Repeat customers*



IQM is public

First European quantum computing company listed on a major U.S. exchange

Nasdaq: IQMX

New York

Nasdaq Global Select
July 2, 2026

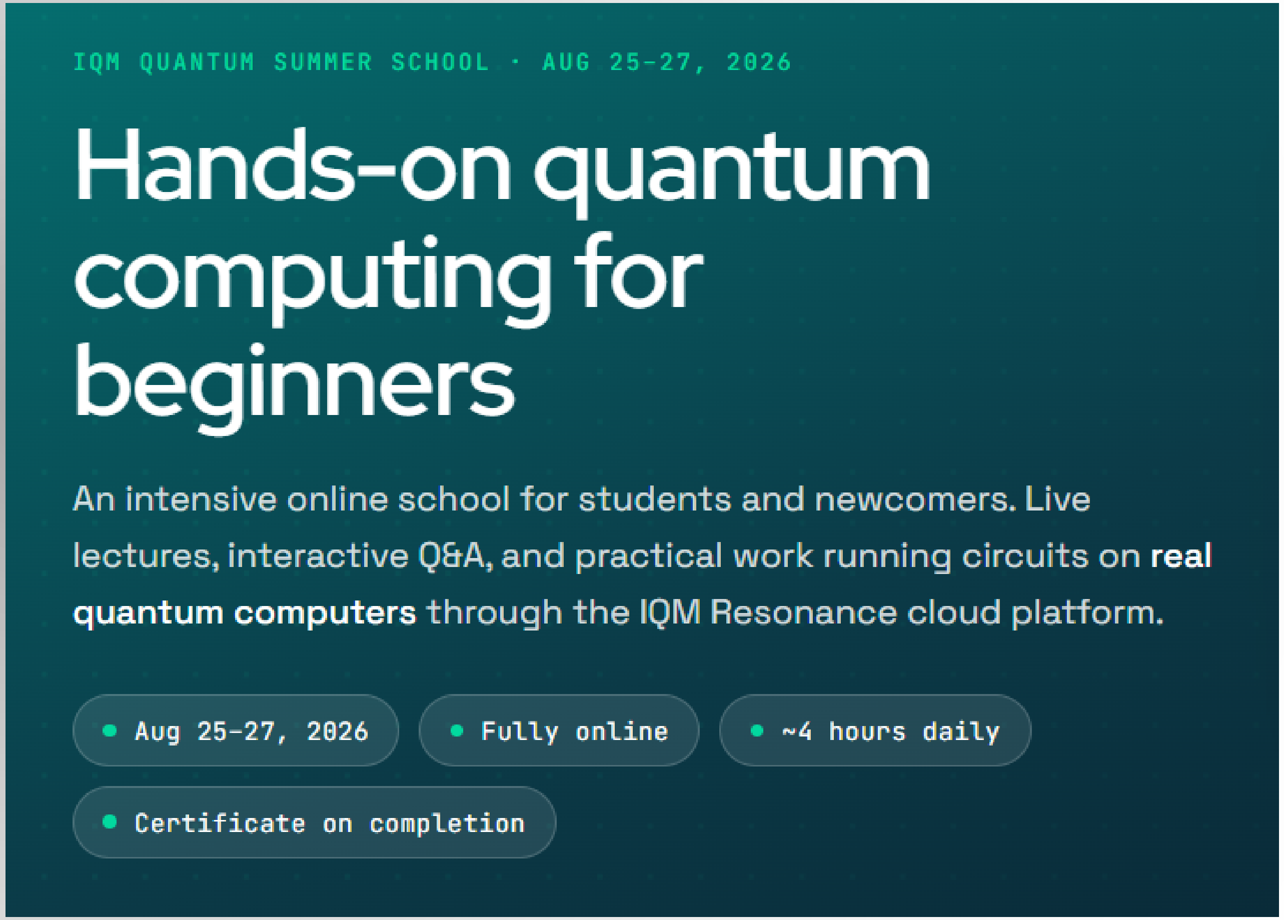
Helsinki

Nasdaq Helsinki
July 3, 2026

First-ever simultaneous listing on **Nasdaq U.S.** and **Nasdaq Helsinki**

About Quantum School

- Understand what quantum computing is about
- Get a feeling if quantum is something for you
- Be inspired and curious to learn and explore more
- You can expect more (and also deeper) events in the future
- We also do regular webinars, check out our website for these



Certificate


IQM Academy™

iqm.tech

IQM Quantum School

Certificate of Participation

This certifies that

Daniel Bulmash

has successfully participated in the IQM Quantum School, an intensive quantum computing educational event held on August 25–27, 2026.

The IQM Quantum School provided participants with comprehensive training in quantum computing fundamentals, hands-on experience with quantum algorithms, and insights into the latest developments in quantum technology. This program represents a significant commitment to advancing quantum literacy and practical quantum computing skills.

Granted by

Daniel Bulmash	Nadia Milazzo	Niklas Steinmann
Team Lead, Education and Community	Quantum Education Specialist	Community Specialist

Certificate ID: 8a917761-e0a7-4eb6-95d1-9f0e75b86303

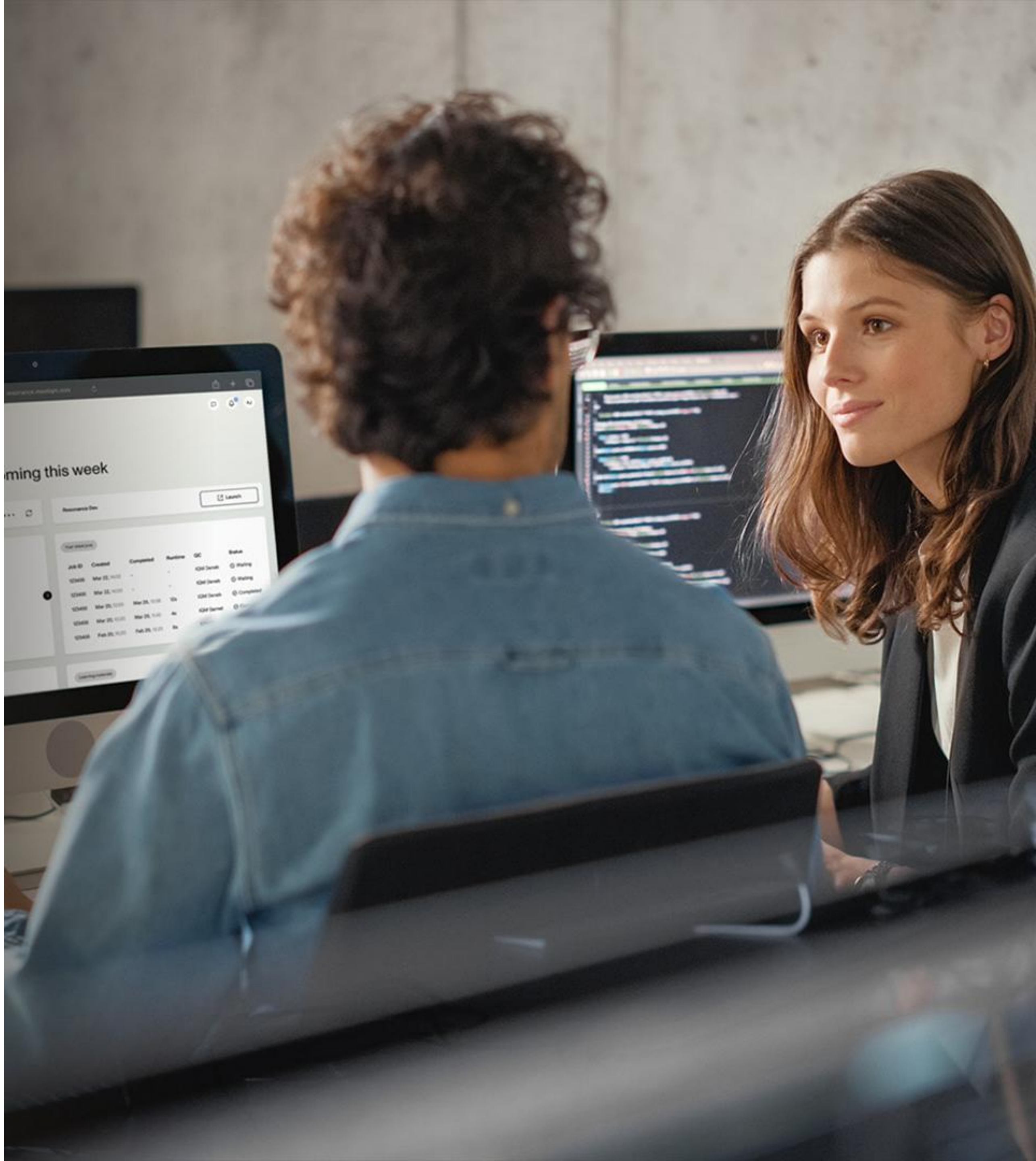
Issued by IQM Quantum Computers Oy | iqm.tech

Attendance link for today (we'll post again during the lab!)

https://tinyurl.com/3stwp7f7

IQM Academy

Your one-stop shop for online quantum education



Gain a thorough understanding of quantum computing and its applications.

- A free curriculum and learning platform for anyone
- Easy to pick up, no previous knowledge required
- Highly interactive and engaging

iqmacademy.com

IQM Quantum Computers, Built for Real Impact

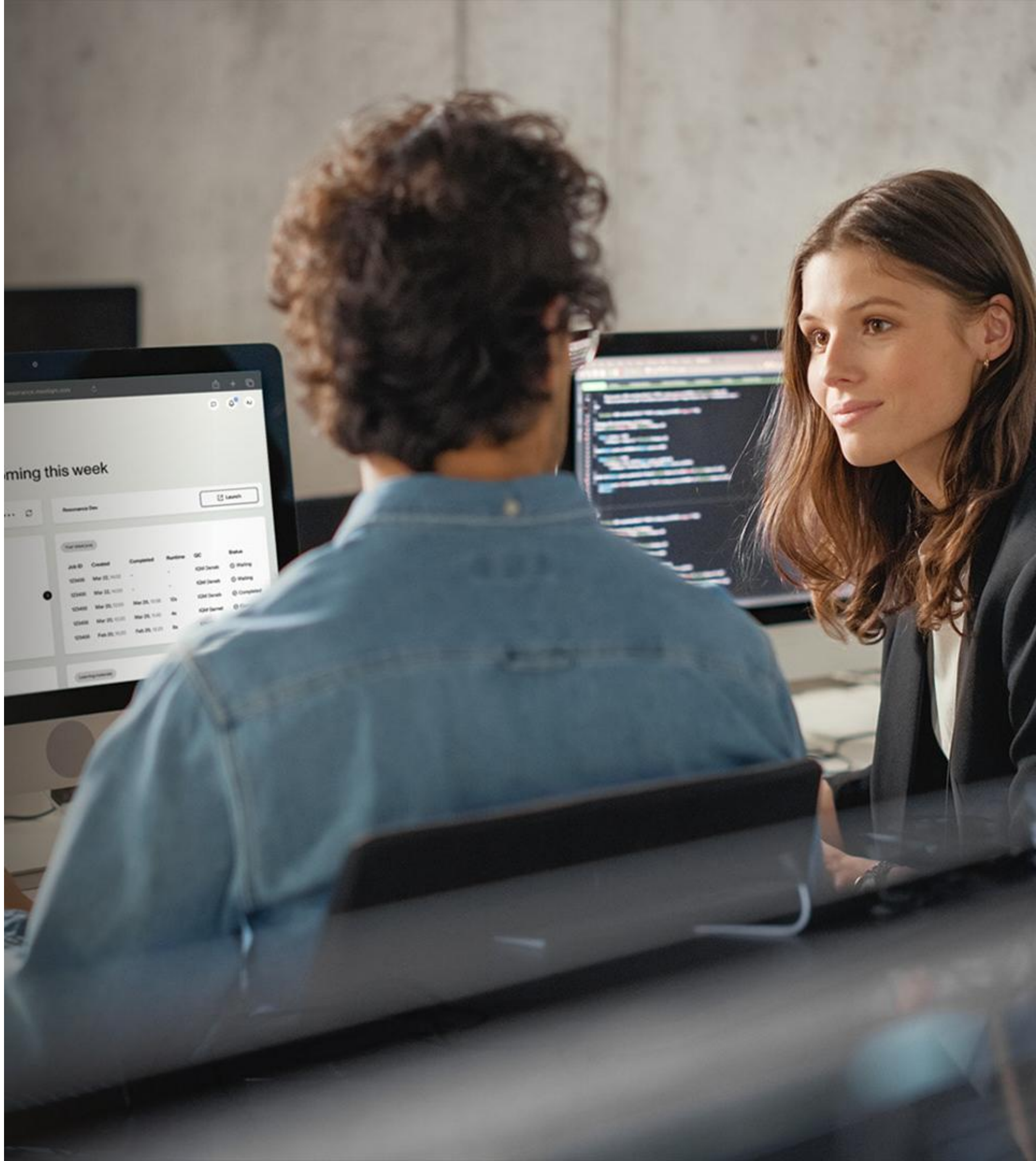
IQM Community Discord

Connect with the IQM community

Gain a thorough understanding of quantum computing and its applications.

- Chat about your latest project
- Get help from IQMers or other IQM users
- Stay in the loop about IQM events

https://discord.gg/F38bwqPYJ4



IQM

Quantum Computers,
Built for Real Impact

Why think about different computing paradigms?

Sorting a list of numbers

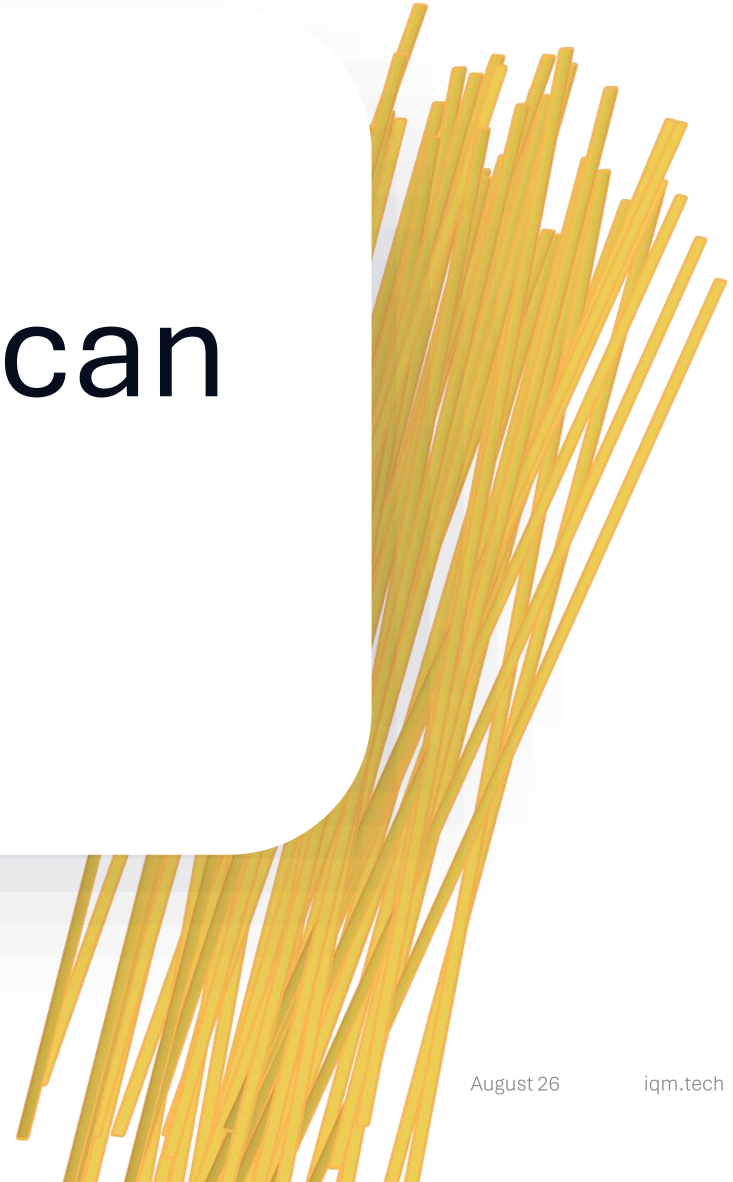
8	2	5	3

Sorting a list of numbers

2	8	5	3

Sorting a list of numbers

Thinking about new computing paradigms can lead to more efficient solutions




Spaghetti
N° 12

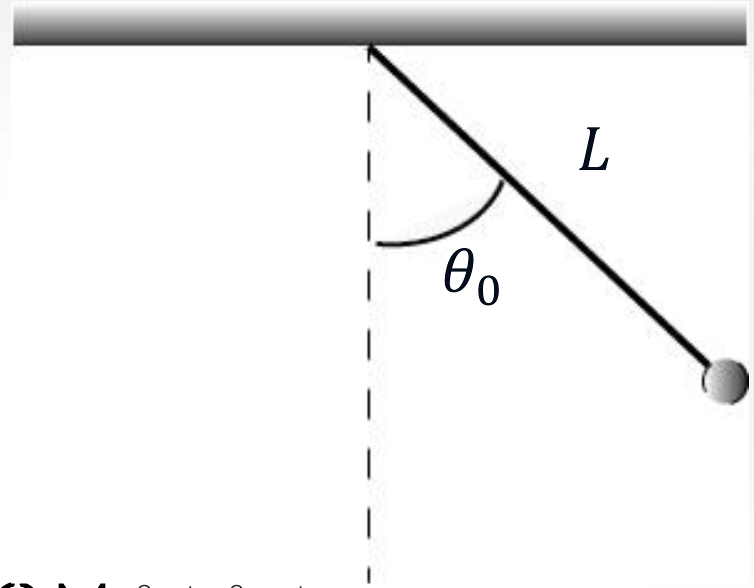
Information is Physical. (R Landauer, 1991)

Example: Evaluate the complete elliptic integral of the first kind

$$K(k) = \int_0^{\pi/2} \frac{1}{\sqrt{1 - k^2 \sin^2 \phi}} d\phi \qquad K(k) = \frac{\pi}{2} \sum_{n=0}^{\infty} [\frac{(2n-1)!!}{(2n)!!}]^2 k^{2n}.$$

Use a physical system as a (specialized) computer:

The period of a pendulum is $T = 4\sqrt{\frac{L}{g}} K(k), k = \sin\frac{\theta_0}{2}$.



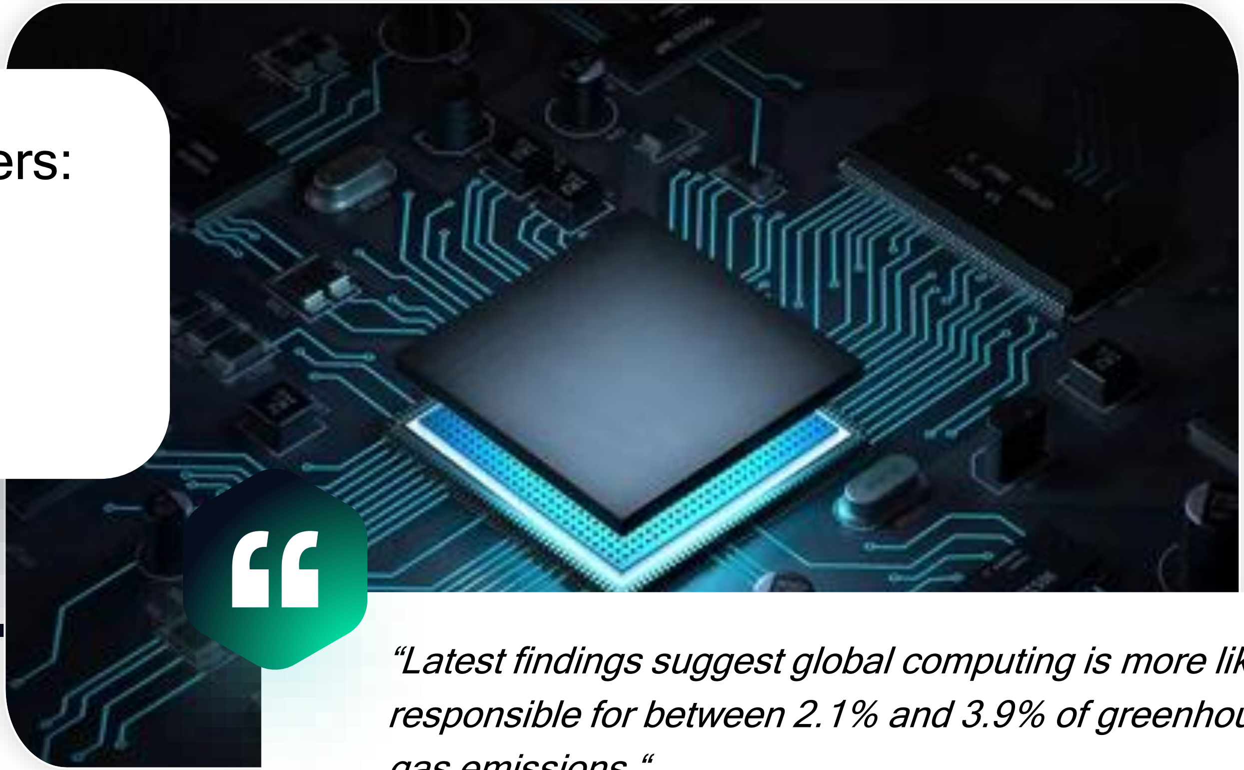
- It may take a few minutes to set up a pendulum. It is much shorter than the time required to build a digital computer from scratch.
- Pendulums are very effective at computing this integral!

There are problems a classical computer cannot (efficiently) solve

Problem with current computers:

- ♾️ Mathematical limits
- 💻 Physical limits
- ⏰ Time & power limits

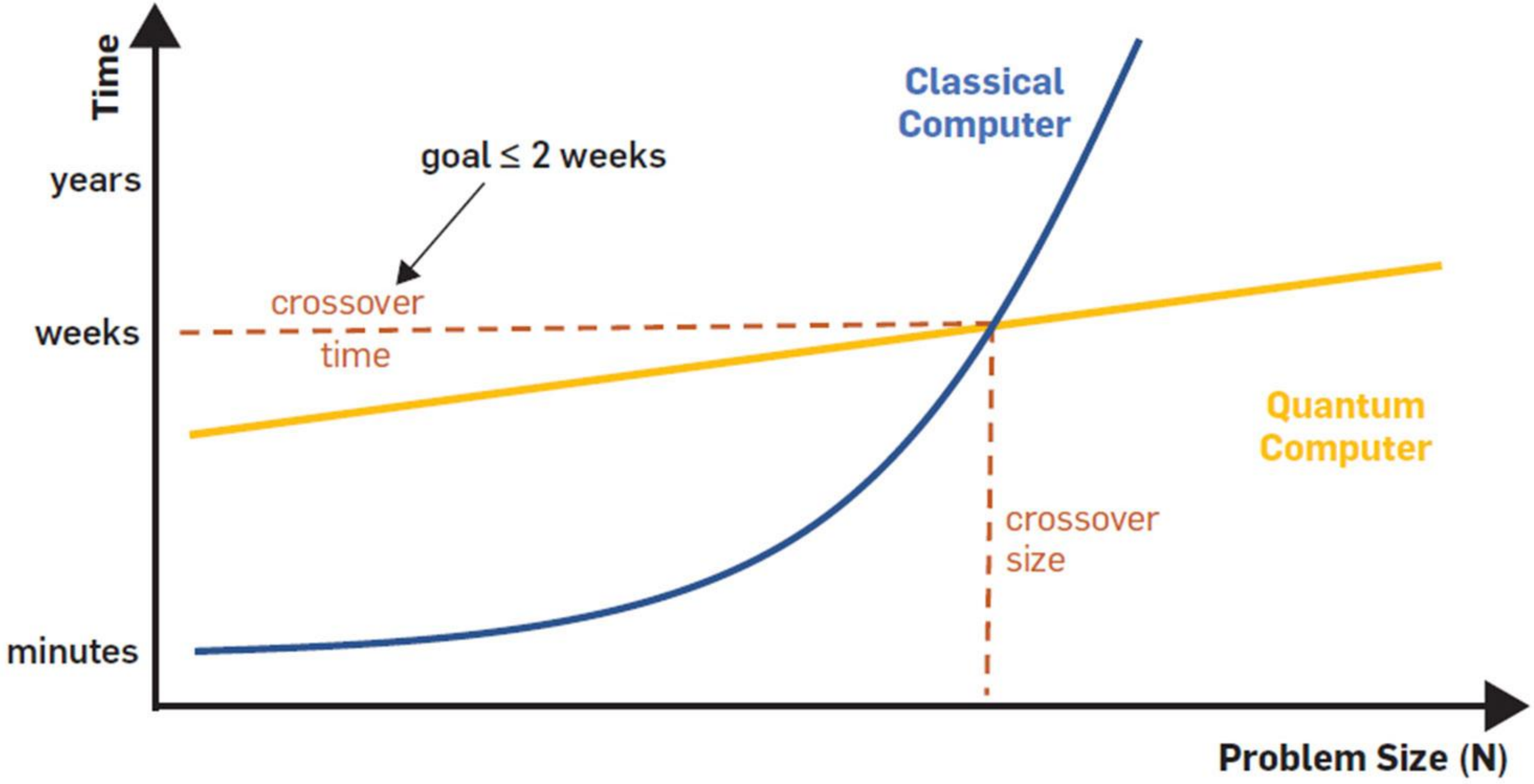
Current approach is not scalable!



"Latest findings suggest global computing is more likely responsible for between 2.1% and 3.9% of greenhouse gas emissions."

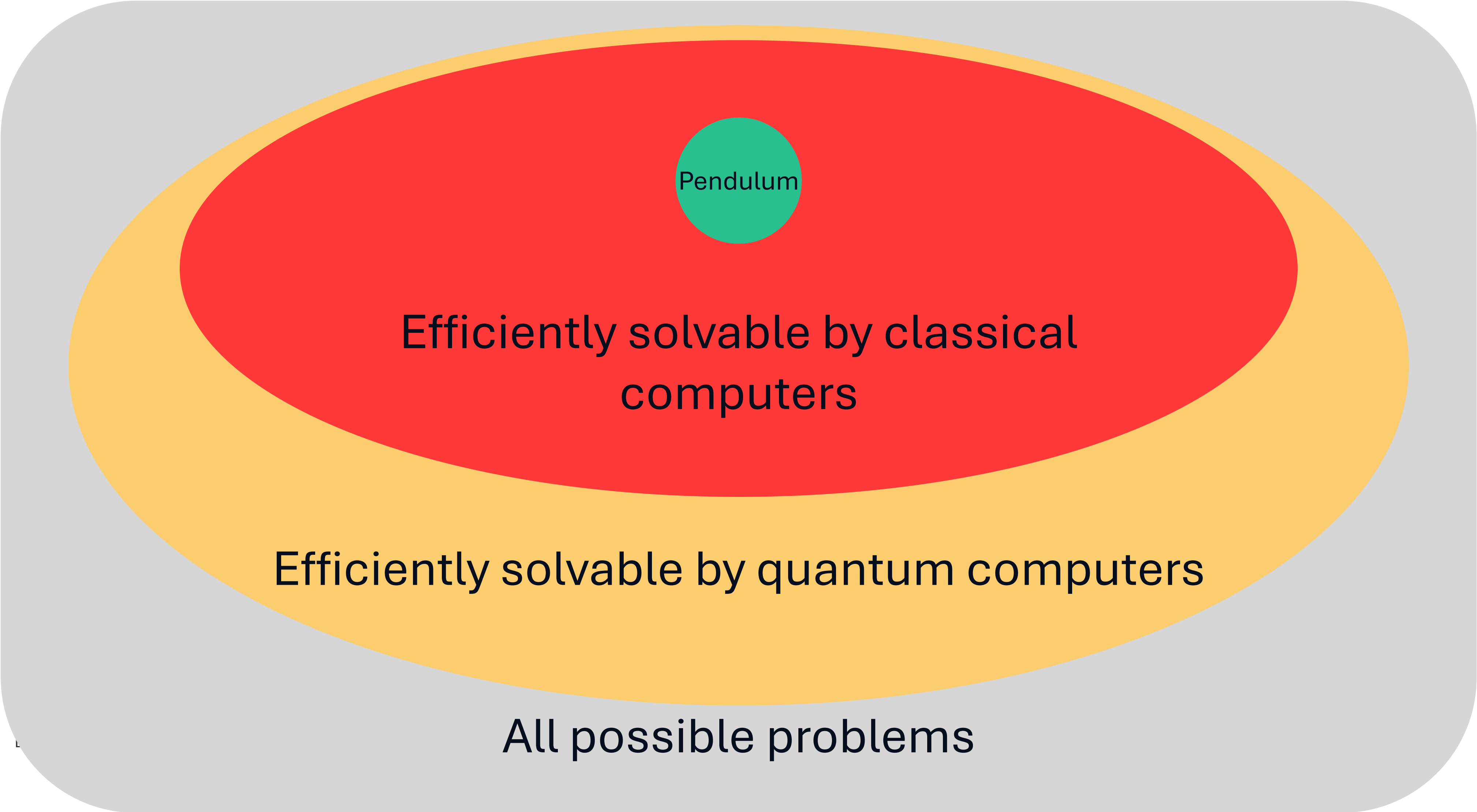
https://www.upi.com/Science_News/2021/09/10/communications-tech-carbon-emissions/8771631295350/

There are problems a classical computer cannot (efficiently) solve

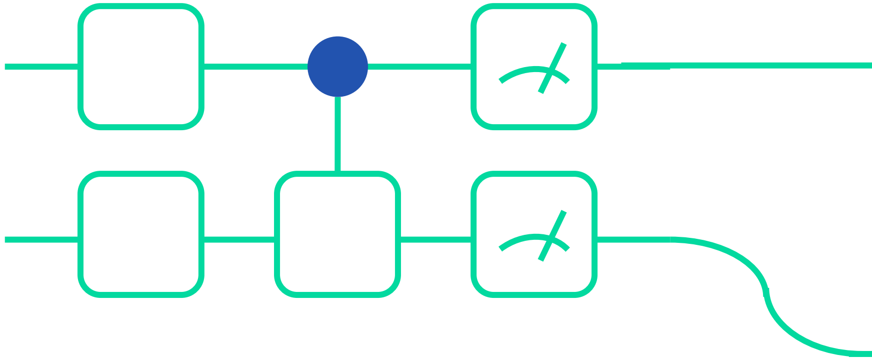


Graphics: ETH Zurich, Microsoft, ACM

Complexity depends on the device for processing



Quantum computers can unlock powerful applications in the future



Quantum simulation

(e.g. fertilizer production)

Optimization

(e.g. portfolio optimization)

Artificial intelligence / machine learning

(e.g. learning from quantum data)

Other

(e.g. factorization)

Solving problems that reduce carbon footprint: Battery optimization

Batteries with a higher lifetime

Batteries with faster charge - discharge cycles

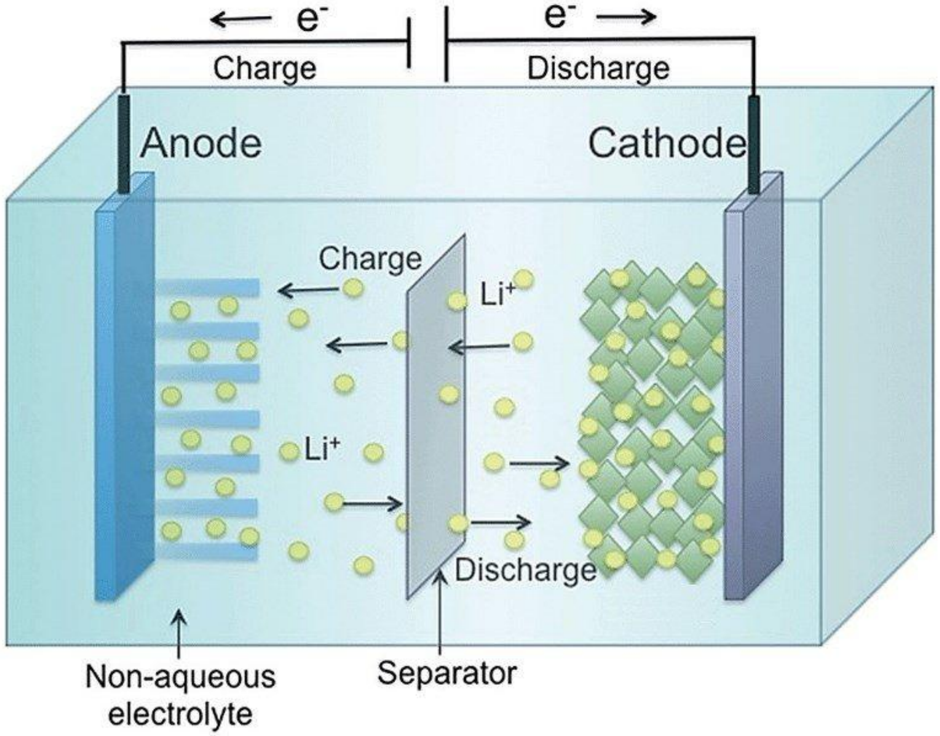
Batteries with higher energy densities

Battery materials that are easier to source

IQM x VOLKSWAGEN GROUP

Solving problems that reduce carbon footprint: Battery optimization

- → Understanding the chemical reactions of the electrolyte region is key to increase battery performance and longevity.
- → This region contains electrons that are strongly correlated (heavily entangled): very hard to compute classically.



Source: A Review of Lithium-Ion Battery Fire Suppression - Published in 2020

Lithium-air batteries have 5.0x theoretical energy density of Li-ion batteries

Towards simulation of next generation batteries

Battery materials are strongly-correlated

Approximations cannot capture the behavior of strongly-correlated systems, such as battery materials

Need to understand the reaction pathways

Computational cost too high for practical use

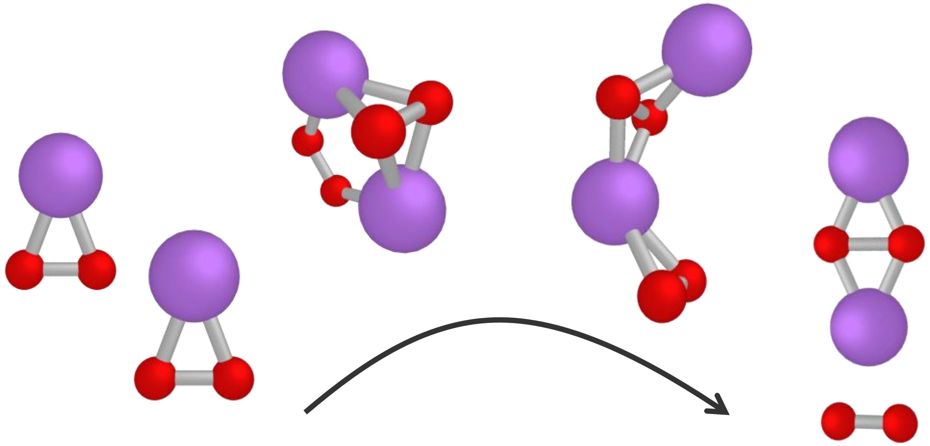
QC-AFQMC is a hybrid method that is a quantization of a promising classical method. While efficient, it is still too slow for any practical chemical system

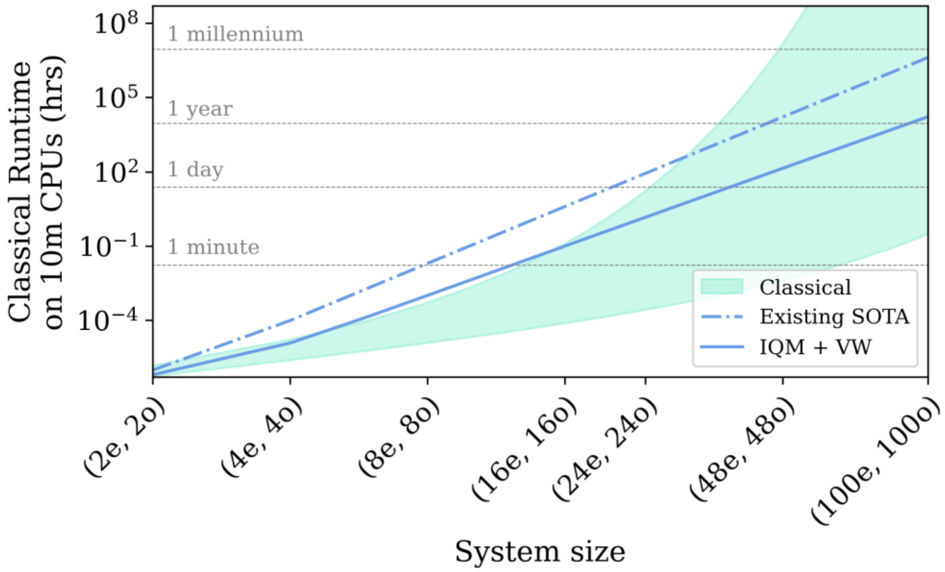
Improved SOTA runtime estimates by 248x using LUMI supercomputer

Combining algorithmic and implementation improvements, we bring the projected runtime for a 100-orbital system down to a couple years from half a millennium

Largest demonstration of this hybrid method

Contextualize the work with a reaction pathway for lithium-air batteries that fitted to 40 qubits





Fraud detection

Partner
Global insurance company

Problem

→ Endpoint detection and response based on training normal behavior and flagging anomalous behavior in operation

Solution

We use one-class support vector machine:

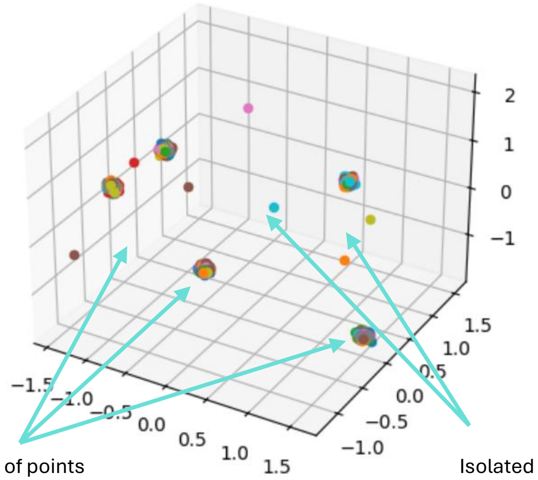
1. Using a quantum kernel method, we map data points into high-dimensional space where differences between normal and anomalous behaviour are easier to distinguish.
2. From the mapping we obtain a decision function to decide whether future data points are anomalies or not

Why Quantum

→ Quantum kernels produce mappings intractable to classical computers.

→ Quantum kernels can represent complex functions with fewer resources

→ In summary, quantum may be able to capture more difficult patterns, which could result in high accuracy fraud detection



Clusters of points represent normal activity

Isolated points represent anomalous activity

Power Plant Maintenance Scheduling

Partner: **Électricité de France SA (EDF)**

Problem

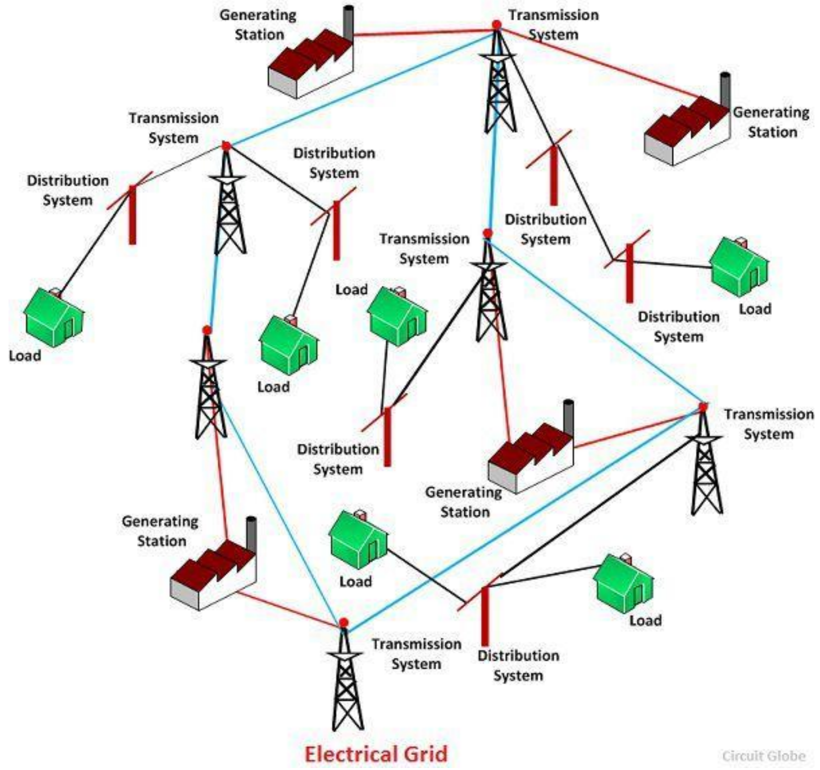
→ Planning the maintenance schedule of power plants
→ Satisfying constraints with regular maintenance intervals
→ Prevent power grid failure because of multiple power plants being maintained at the same time, and therefore not providing power

Solution

1. Represent the problem as graph (nodes = individual schedules, edges = incompatibility)
2. Define a Maximum independent set (MIS) problem (NP complete) on the above defined graph
3. Solve the above MIS problem with tailored QAOA problem

Why Quantum

Quantum computers leverage quantum superposition for parallel exploration of the solution space, as well as entanglement for complex correlations. Once the HW is ready in terms of precision and qubit number, they may potentially outperform classical approaches both in accuracy and time-to-solution

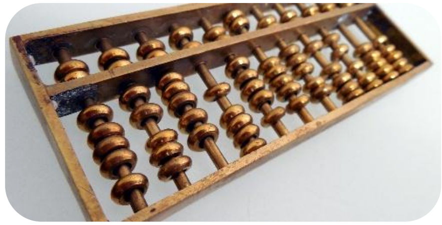


Overview

- Some problems 📈 scale too fast for ordinary digital computers, meaning digital computers need a lot of time and/or memory to solve them.
- Different ⚛️ physical systems may offer different computational complexity
- Quantum information processing can reduce computational complexity for some tasks in
  - Optimization
  - Simulation
  - and machine learning.

Classical Computer vs. Quantum Computer

Classical computer



Quantum computer

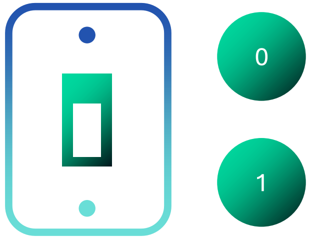
Ion Trap By Coldsmokerider - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=140354004

Bits and Qubits

Computers encode information in many *bits*

Classical bit

Ordinary physical object with 2 distinct states



switch on/off

Measuring the state does not change the state

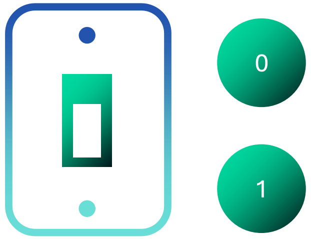
We can invent codes that store information (or messages) as bits, then manipulate that message.

Message	Bits
I	01001001
Q	01010001
M	01001101
IQM	01001001 01010001 01001101

Classical Computer vs. Quantum Computer

Classical bit

Ordinary physical object with 2 distinct states

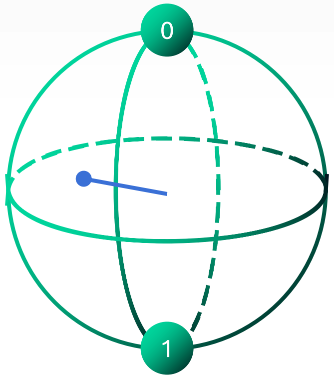


switch on/off

Measuring the state does not change the state

Quantum bit

Physical object which can be in a “quantum superposition” of 2 states



Point on the surface of a sphere

Measurement of the state will cause it to **irreversibly change** to 0 or 1

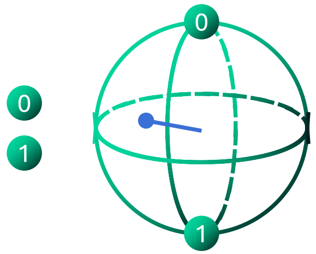
What is quantum computing?

Classical vs quantum computer

Quantum computers follow different rules than classical computers – those of quantum physics.

Bit vs qubit (quantum bit)

A bit can be either 0 or 1, while a qubit can be in a **superposition:** qubit can be both 0 and 1 at the same time. Measurement will yield either 0 or 1.



Two qubits can be **entangled**. Changing one directly impacts the other.

→ 1 qubit can be in a superposition of 2 basis states

→ 1 bit can be in just one state

1Qubit

Basis states:

- 0
- 1

What is quantum computing?

Classical vs quantum computer

Quantum computers follow different rules than classical computers – those of quantum physics.

→ 2 qubits can be in a superposition of all 4 basis states

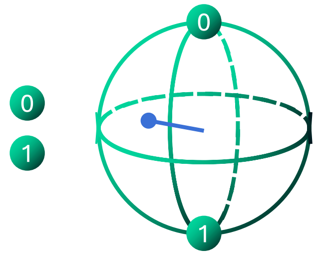
→ 2 bits can be in just one state

Basis states:

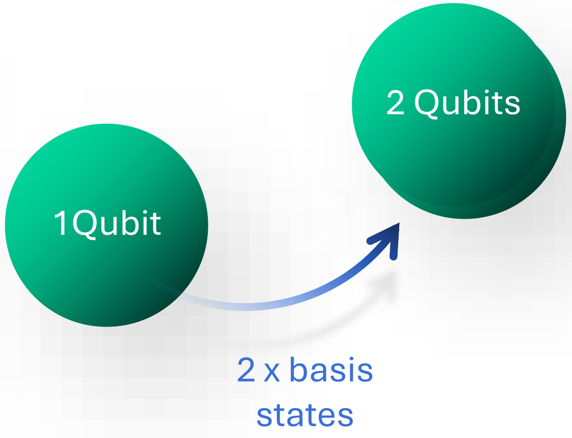
- 00
- 01
- 10
- 11

Bit vs qubit (quantum bit)

A bit can be either 0 or 1, while a qubit can be in a **superposition:** qubit can be both 0 and 1 at the same time. Measurement will yield either 0 or 1.



Two qubits can be **entangled**. Changing one directly impacts the other.



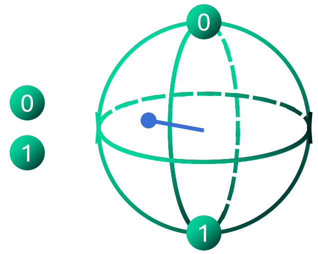
What is quantum computing?

Classical vs quantum computer

Quantum computers follow different rules than classical computers – those of quantum physics.

Bit vs qubit (quantum bit)

A bit can be either 0 or 1, while a qubit can be in a **superposition:** qubit can be both 0 and 1 at the same time. Measurement will yield either 0 or 1.

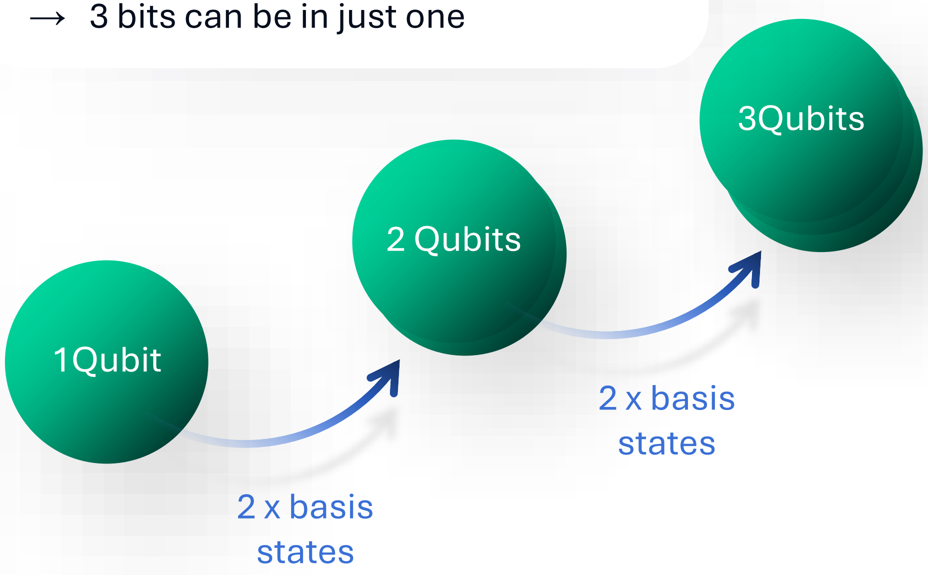


Two qubits can be **entangled**. Changing one directly impacts the other.

000	001	010	011
100	101	110	111

→ 3 qubits can be in a superposition of all $2^3 = 8$ basis states

→ 3 bits can be in just one



Basis states:

- 000
- 100
- 010
- 110
- 001
- 101
- 011
- 111

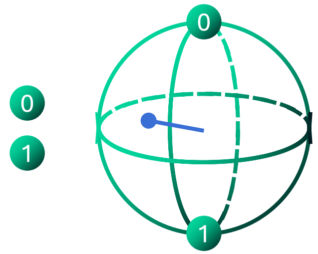
What is quantum computing?

Classical vs quantum computer

Quantum computers follow different rules than classical computers – those of quantum physics.

Bit vs qubit (quantum bit)

A bit can be either 0 or 1, while a qubit can be in a **superposition:** qubit can be both 0 and 1 at the same time. Measurement will yield either 0 or 1.

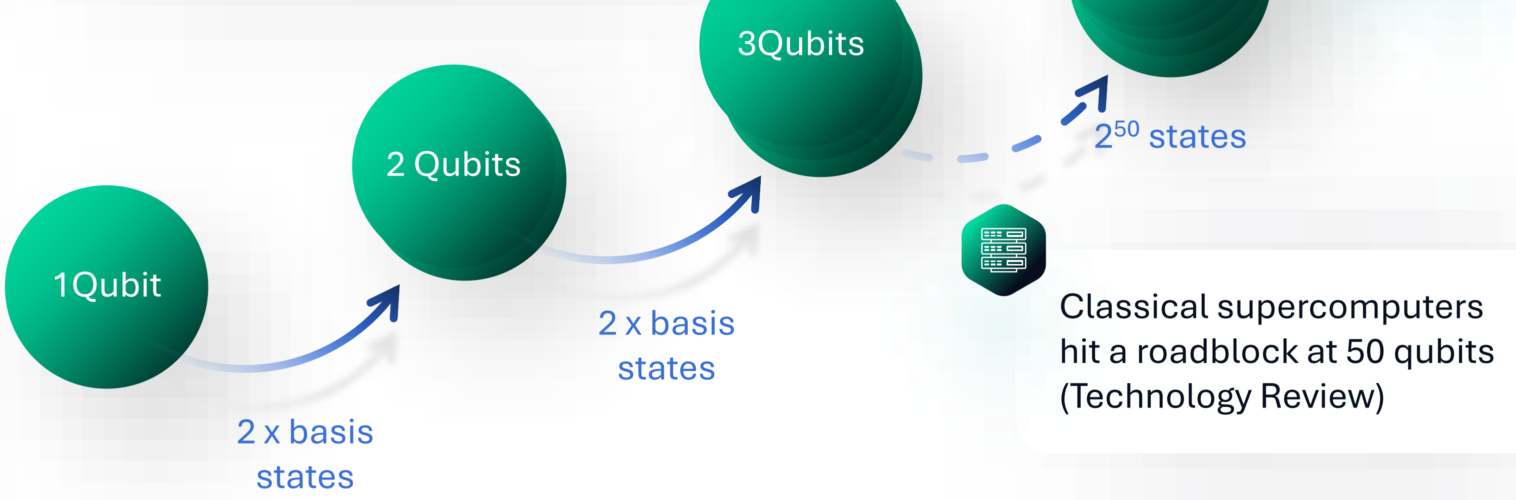


Two qubits can be **entangled**. Changing one directly impacts the other.

000 001 010 011
100 101 110 111

→ 3 qubits can be in a superposition of all $2^3 = 8$ basis states

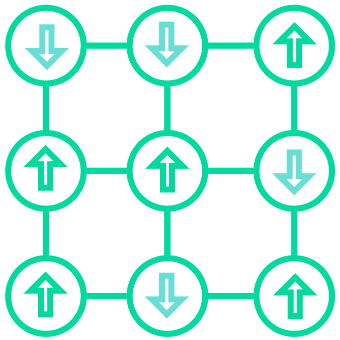
→ 3 bits can be in just one



Classical supercomputers hit a roadblock at 50 qubits (Technology Review)

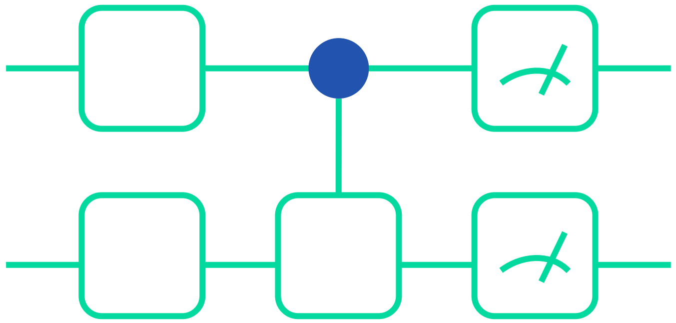
How to work with qubits?

Different ways to work with qubits



Quantum annealing

Designed for particular optimization problems



Gate-based quantum computing

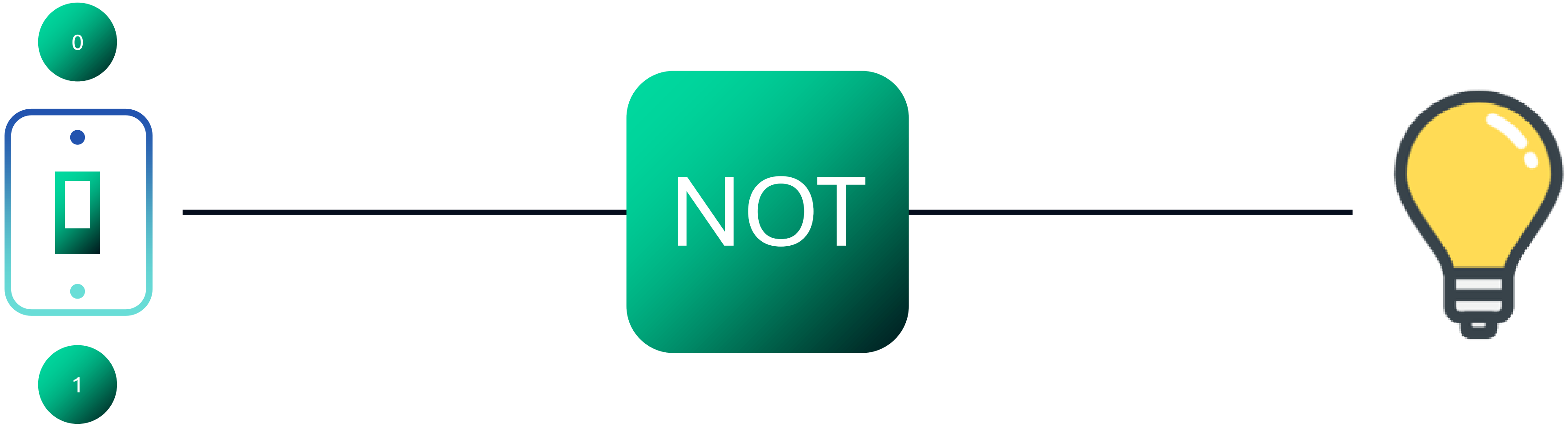
Designed for universal (quantum) computing

Conventional computer: bits

Bits are the foundation of conventional computers

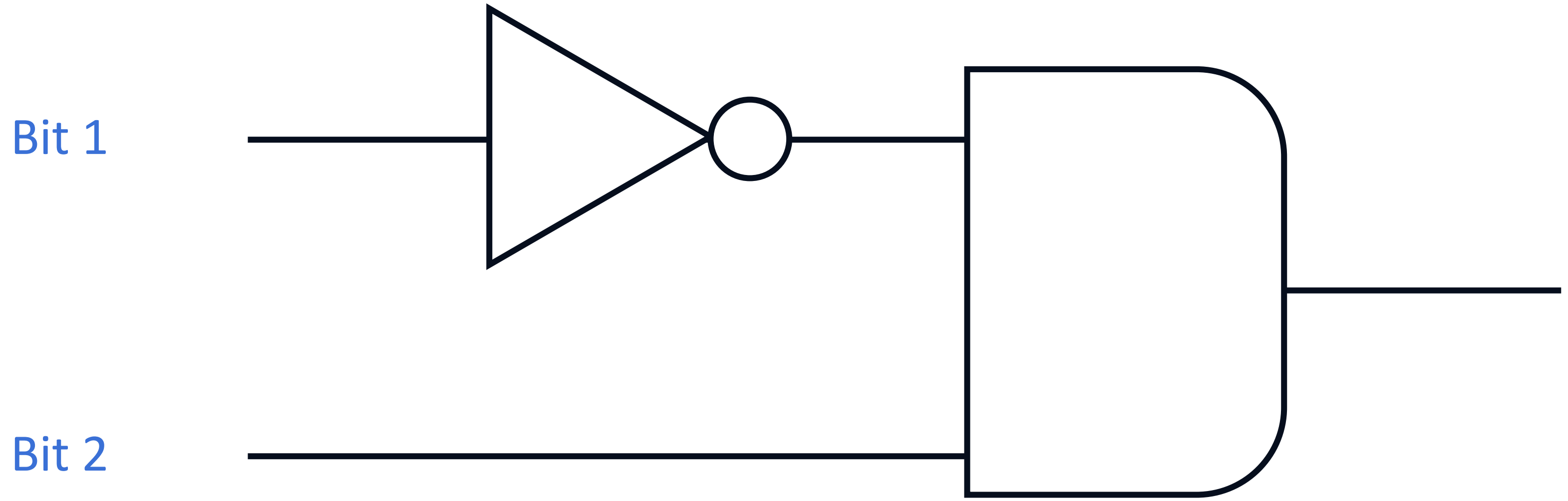
Conventional computer: bits

Information processing is performed with the help of logical gates



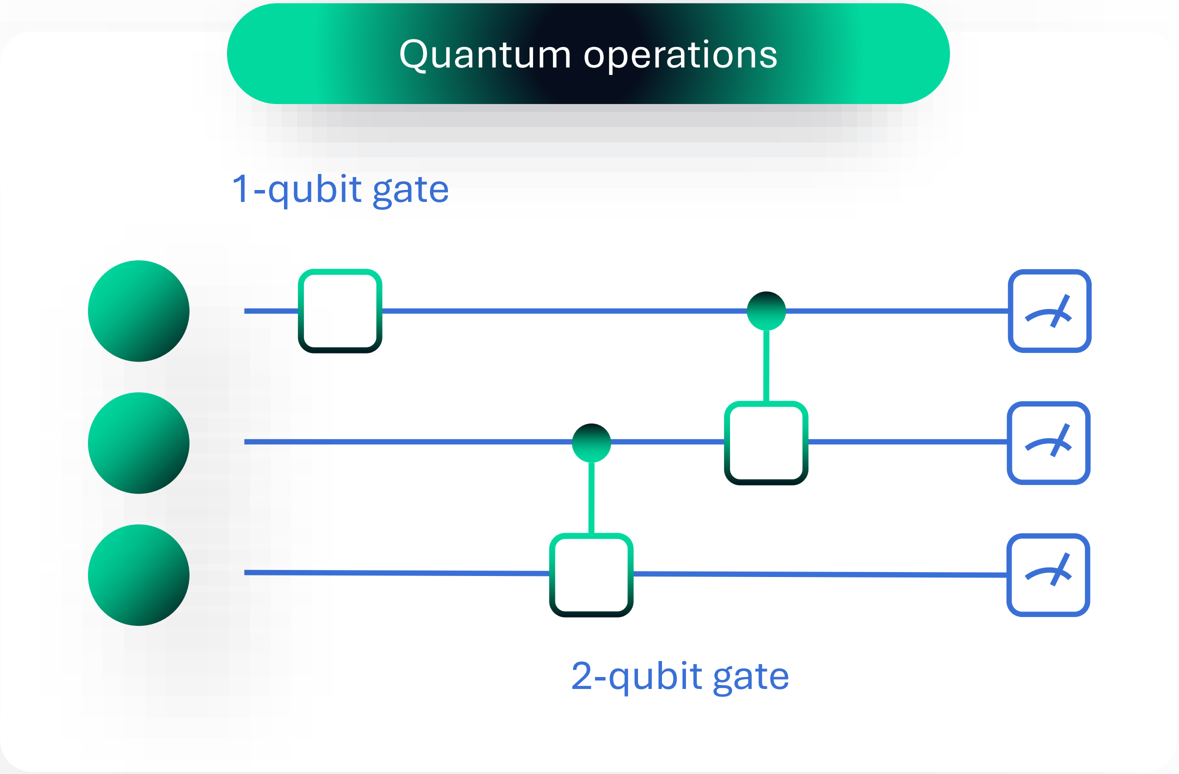
Conventional computer: algorithms

- Multiple gates form a circuit
- An algorithm is implemented in a sequence of gates
- Circuits are represented as a time sequence like musical notes

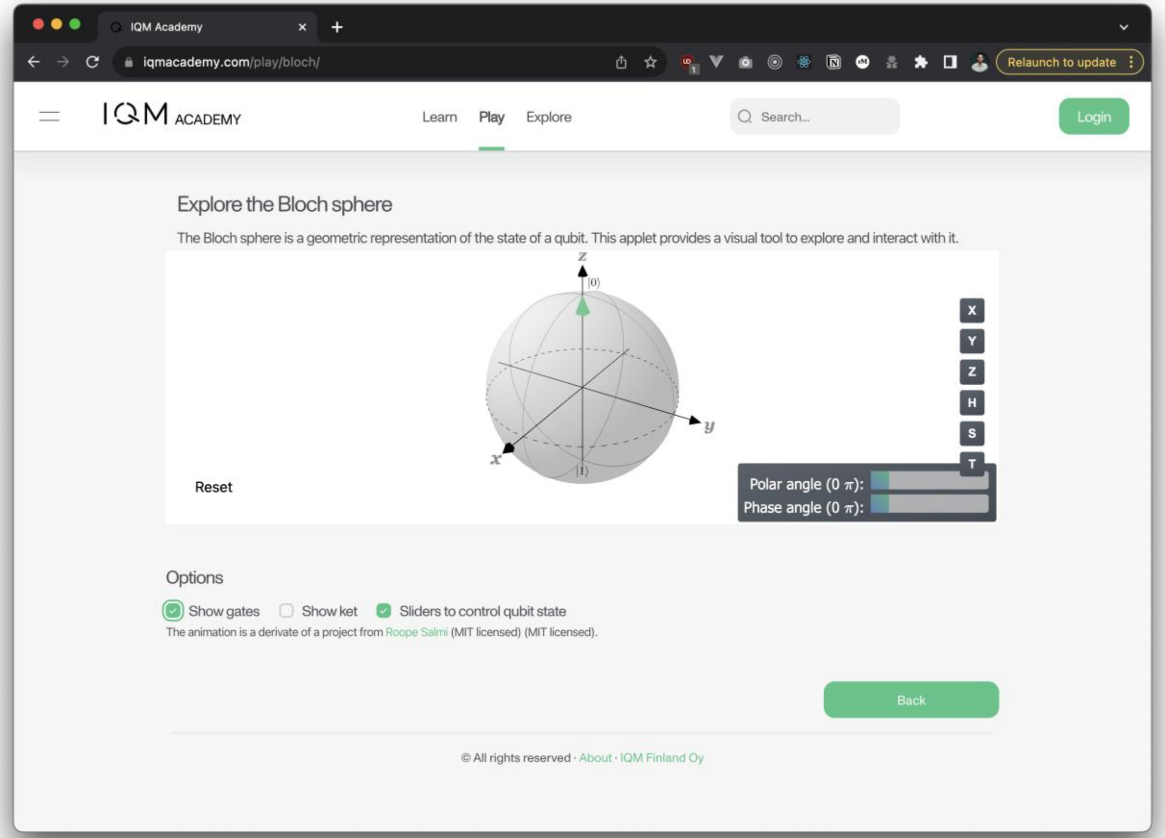


Classical circuits are irreversible: Two inputs, but just one output in above case

Quantum computing: algorithms



Activity 1: Bloch Sphere



Open https://tinyurl.com/iqm-bloch-sphere and investigate the effects of gate H and gate X ! See what happens if you apply the H gate twice.

Activity 1: Description of a quantum state

- Classical information processing uses a bit as a unit of information. A bit takes a value 0 or 1.

- n bits take one of 2^n different values from 000…0 to 111…1.

- A qubit is a unit of quantum information. It encodes probabilities of measuring 0 and 1. In this sense, a qubit can be 0 and 1 simultaneously.

Activity 1: Description of a quantum state

A single-qubit quantum state $|\psi\rangle$ can be described mathematically as

This is called a ket, Dirac notation for vectors

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

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

Activity 1: Description of a quantum state

A single-qubit quantum state $|\psi\rangle$ can be described mathematically as

This is called a ket, Dirac notation for vectors

$$|\psi\rangle = a\begin{pmatrix}1\\0\end{pmatrix} + b\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}a\\b\end{pmatrix} \text{ with } |a|^2 + |b|^2 = 1$$

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

If you measure the qubit, it will always change randomly into either the state $|0\rangle$ or the state $|1\rangle$.

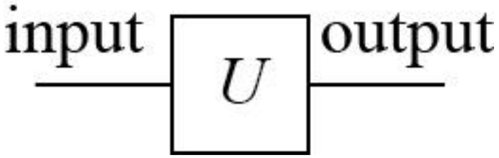
- $|a|^2$ tells you the probability it turns into $|0\rangle$.
- $|b|^2$ tells you the probability it turns into $|1\rangle$.

$|\psi\rangle$ represents a qubit that is in both the $|0\rangle$ and $|1\rangle$ states simultaneously.

One-Qubit Gates

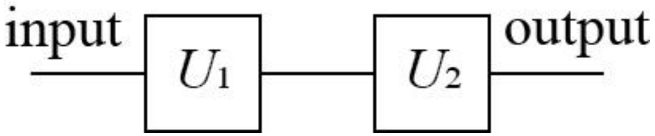
A quantum gate that acts on a single qubit is called a 1-qubit gate.

$|\psi_{\text{in}}\rangle$



$|\psi_{\text{out}}\rangle = U|\psi_{\text{in}}\rangle$

$|\psi_{\text{in}}\rangle$



$|\psi_{\text{out}}\rangle = U_2 U_1 |\psi_{\text{in}}\rangle$

Example: the X (NOT) gate

This is how it looked in the activity

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

This is how it looks in matrix form

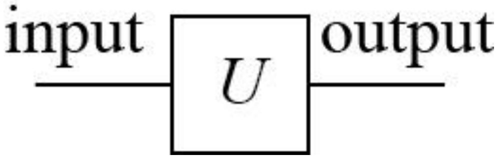
$$X|0\rangle = |1\rangle \;\&\; X|1\rangle = |0\rangle$$

This is how it acts on a quantum state

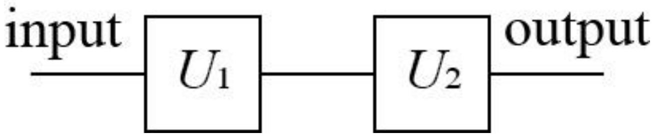
X flips the qubit states

One-Qubit Gates

A quantum gate that acts on a single qubit is called a 1-qubit gate.

$|\psi_{\text{in}}\rangle$ 

$|\psi_{\text{out}}\rangle = U|\psi_{\text{in}}\rangle$

$|\psi_{\text{in}}\rangle$ 

$|\psi_{\text{out}}\rangle = U_2 U_1 |\psi_{\text{in}}\rangle$

Example: the H (Hadamard) gate

This is how it looked in the activity

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

This is how it looks in matrix form

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

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

This is how it acts on a quantum state

H maps $|0\rangle$ and $|1\rangle$ to superposition states

Multi-qubit Systems and Entanglement

- Consider a system made of 2 qubits: if the state of the first qubit is $|0\rangle$ and the second qubit is $|1\rangle$, the total state is written as:

$$|0\rangle|1\rangle \quad \text{or} \quad |01\rangle \quad \text{or} \quad |0\rangle \otimes |1\rangle$$

- Now consider the 2-qubit state $|\Psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle + |1\rangle|0\rangle)$:

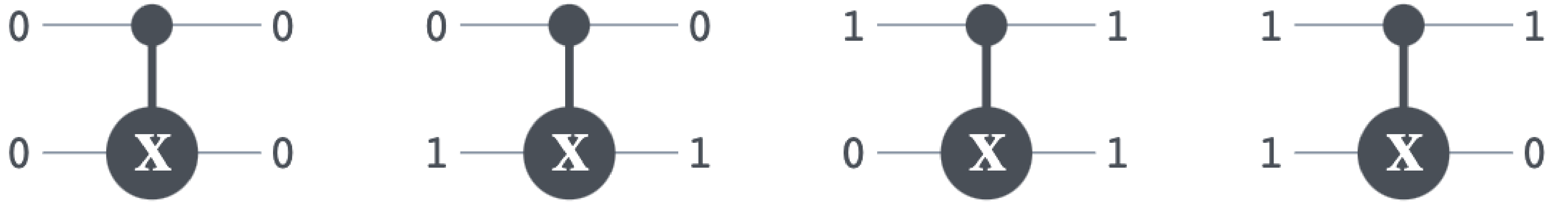
It can be written as $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$. It is a tensor product state or a product state.

If we measure the first qubit → $50\%\ |0\rangle, \quad 50\%\ |1\rangle$.
If we measure the second qubit → $100\%\ |0\rangle$ independently of the outcome of the first qubit

They are not correlated

Activity 2: A 2-qubit gate

The gate below is called **CNOT** (Controlled NOT)
With two qubits, the gate flips the second (target qubit) if the first (control qubit) is in state $|1\rangle$
If the control qubit is in the $|0\rangle$ state, nothing happens to the target qubit.



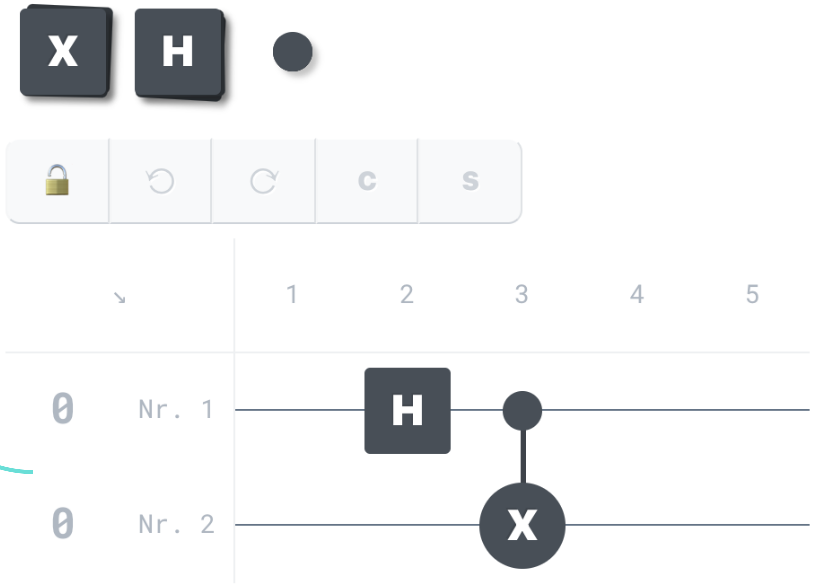
What happens, if the control qubit is in a superposition state of $|0\rangle$ and $|1\rangle$?

Open https://tinyurl.com/iqm-entanglement to check it out!

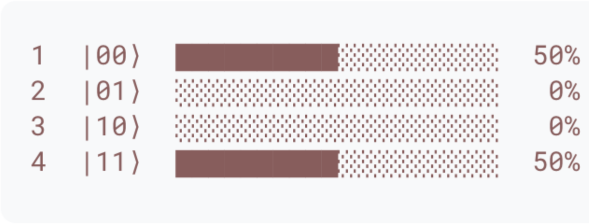
Activity 2 - Discussion

- If the **control qubit** is in a superposition, the measurement result of the target qubit depends on that of the control qubit.
- This combination of gates "**entangles**" the two qubits.

- If the state of one of the qubits is then measured, the state of the other qubit is automatically known.
- Viewed as a whole, the two qubits have a well-defined state, but the individual qubits cannot be assigned their own well-defined state.



1000 measurements after...



Activity 2: description of entanglement

The state we generated is: $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$

Let's assume that we can write it as the **product** of two **independent** single-qubit states:

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \stackrel{?}{=} (a|0\rangle + b|1\rangle) \otimes (\mathrm{c}|0\rangle + d|1\rangle)$$

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \stackrel{?}{=} ac|00\rangle + ad|01\rangle + \mathrm{bc}|10\rangle + bd|11\rangle$$

$$\frac{1}{\sqrt{2}} = ac, 0 = ad, 0 = bc, \frac{1}{\sqrt{2}} = bd$$

There is **no choice of a, b, c, and d** that can satisfy all four equations at the same time!

The qubits are correlated: they are **ENTANGLED**

Conclusion: Foundations

- Qubits can *not only* be in the two states $|0\rangle$ and $|1\rangle$, but *also* in a **superposition** of $|0\rangle$ and $|1\rangle$. In this case, they have a certain probability of being measured as $|0\rangle$ or $|1\rangle$. However, a measurement destroys the superposition.

- Quantum computers use **quantum gates** to manipulate the state of qubits.

- Two qubits can be entangled with each other. If you then measure the state of one of the qubits, you automatically know the state of the other qubit.

- Superposition and entanglement are the key ingredients of quantum algorithms (more on that tomorrow!)

Today's Lab

https://tinyurl.com/QS20206-Day1-LAB-participant

qrisp.eu

IQM Academy .com

IQM Resonance

resonance.iqm.tech

GitHub github.com/eclipse-qrisp

IQM

Introduction to Qrisp with IQM Resonance

IQM QuantumSchool 2026

Authors: Niklas Steinmann

Last Updated 8/2026

www.meetiqm.com

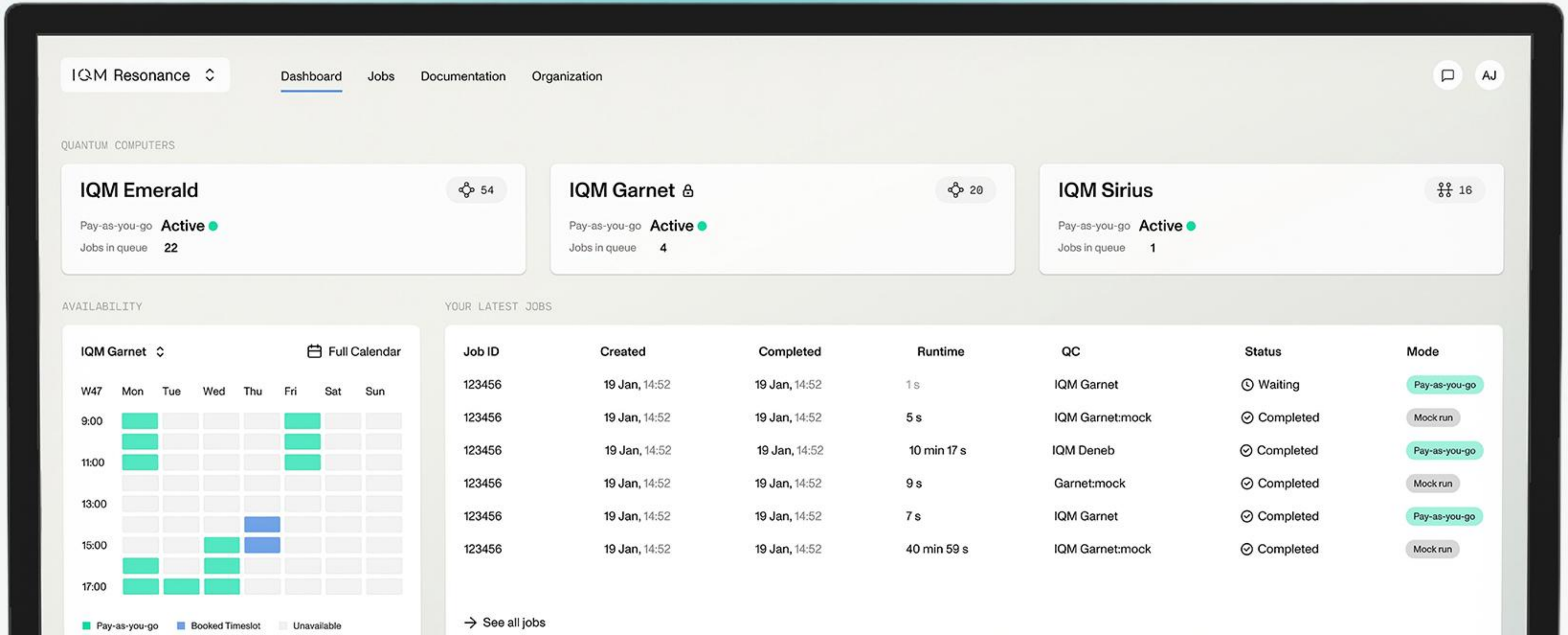
The tools

IQM Resonance™

www.iqm.tech

IQM Resonance™

https://resonance.iqm.tech/



IQM Resonance Intro

qrisp

- qrisp is a high level programming framework for working with quantum computing programs
- It’s pythonic

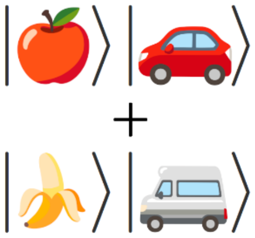
pip install „qrisp[iqm]“

- It’s open source

https://github.com/eclipse-qrisp/Qrisp

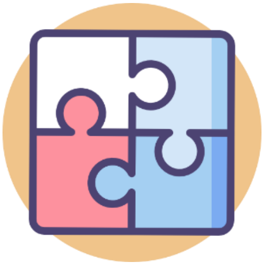
- Created at Fraunhofer Fokus, IQM actively contributing to development

Features of qrisp

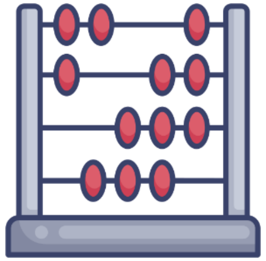


Typed quantum variables

Automatic uncomputation,
i.e. garbage collection

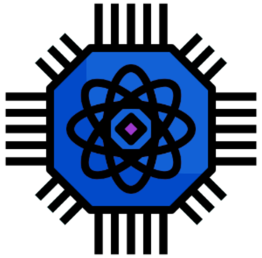


Modularity



Arithmetics

Compatibility



Full support for
IQM hardware,
High-perfomance simulator

There are different quantum computing frameworks, but qrisp will enable you to learn what matters

qrisp | **Qiskit** | **PENNYLANE**

```python
from qrisp import QuantumFloat
n = 6
a = QuantumFloat(n)
b = QuantumFloat(n)
a[:] = 3
b[:] = 4
res = a*b
print(res)
#Yields: {12: 1.0}
```

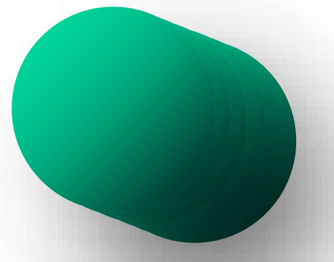
```python
from qiskit import (QuantumCircuit, QuantumRegister,
ClassicalRegister, Aer, execute)
from qiskit.circuit.library import RGQFTMultiplier
n = 6
a = QuantumRegister(n)
b = QuantumRegister(n)
res = QuantumRegister(2*n)
cl_res = ClassicalRegister(2*n)
qc = QuantumCircuit(a, b, res, cl_res)
for i in range(len(a)):
    if 3 & 1<<i: qc.x(a[i])
for i in range(len(b)):
    if 4 & 1<<i: qc.x(b[i])
qc.append(RGQFTMultiplier(n, 2*n),
list(a) + list(b) + list(res))
qc.measure(res, cl_res)
backend = Aer.get_backend('qasm_simulator')
counts_dic = execute(qc, backend).result().get_counts()
print({int(k, 2) : v for k, v in counts_dic.items()})
#Yields: {12: 1024}
```

```python
import pennylane as qml
import numpy as np
w_m = [0, 1, 2]
w_k = [3, 4, 5]
w_sol = [6, 7, 8, 9]
dev = qml.device("default.qubit",wires=w_m + w_k + w_sol,shots=1)
n_wires = len(dev.wires)
def add(k, wires):
    for j in range(len(wires)):
        qml.RZ(k * np.pi / (2**j), wires=wires[j])
def multiplication(w_m, w_k, w_sol):
    qml.QFT(wires=w_sol)
    for i in range(len(w_k)):
        for j in range(len(w_m)):
            coeff = 2 ** (len(w_m) + len(w_k) - i - j - 2)
            qml.ctrl(add,control=[w_k[i], w_m[j]])(coeff, w_sol)
    qml.adjoint(qml.QFT)(wires=w_sol)
@qml.qnode(dev)
def mul(m, k):
    qml.BasisEmbedding(m, wires=w_m)
    qml.BasisEmbedding(k, wires=w_k)
    multiplication(w_m, w_k, w_sol)
    return qml.sample(wires=w_sol)
print(f"The ket representation of 3*7 is {mul(3,4)}")
#Yields: The ket representation of 3*7 is [10101]
```

QuantumVariables

- The **QuantumVariable** is the quantum equivalent of a **regular variable** in classical programming languages.

```
class QuantumVariable(size, qs=None, name=None)
```



N Qubits
=size

Where each qubit i **can be**

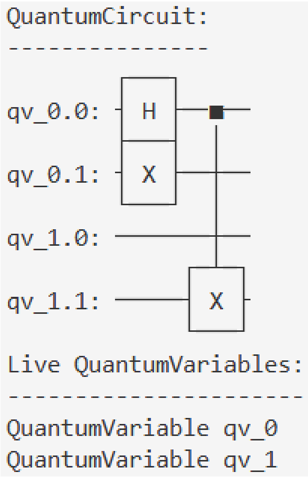
- $a_i|0\rangle + b_i|1\rangle$ **OR**
- **entangled with other qubits j**

➢ N qubits can be in a superposition of all 2^N basis states

➢ Instantiated in $|0\rangle^{\otimes N}$

Quantumsession
qs

- Look at the **statevector**
- retrieve **QuantumVariable** and **circuit info** (**display**, cnot_count, depth)




```
sqrt(2)*(|01>*|00> + |11>*|01>)/2
```

QuantumVariables

- You can apply gates to the **QuantumVariable** as a whole, or to singular qubits

```python
from qrisp import QuantumVariable, h, x, z, cx
qv = QuantumVariable(2)
h(qv[0])
cx(qv[0], qv[1])
print(qv)
```

```
{'00': 0.5, '11': 0.5}
```

QuantumVariables

- You can apply gates to the **QuantumVariable** as a whole, or to singular qubits

- They are smoothly integrated into the **typing infrastructure of Python**

➢ Create **functions** with correct call signatures and apply them!

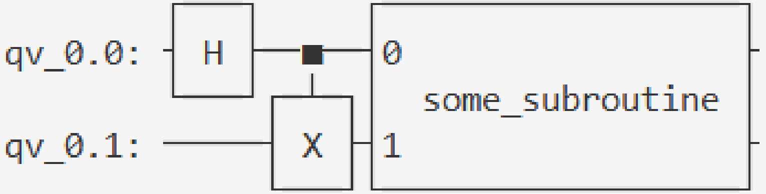
```python
from qrisp import QuantumVariable, h, x, z, cx
qv = QuantumVariable(2)
h(qv[0])
cx(qv[0], qv[1])
print(qv)
```

```
{'00': 0.5, '11': 0.5}
```

```python
@gate_wrap
def some_subroutine(qv):
    # as complex as you like
    h(qv)
some_subroutine(qv)

print(qv.qs)
```

```
QuantumCircuit:
---------------
```



Important gates

1-qubit gates

$|0\rangle = \binom{1}{0}$ and $|1\rangle = \binom{0}{1}$

- $H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$, $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.
- $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. $X|0\rangle = |1\rangle$ & $X|1\rangle = |0\rangle$ NOT.
- $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. $Z|0\rangle = |0\rangle$ & $Z|1\rangle = -|1\rangle$.

2-qubit gates

- U_{CNOT}

$U_{\mathrm{CNOT}}|x\rangle|y\rangle = |x \oplus y\rangle|y\rangle$. U_{CNOT} flips $|x\rangle$ if $|y\rangle = |1\rangle$. Does nothing to $|x\rangle$ if $|y\rangle = |0\rangle$. Similar to XOR.

Lab-time: QuantumVariables & Gates

https://tinyurl.com/QS20206-Day1-LAB-participant

qrisp.eu

IQM Academy.com

IQM Resonance

resonance.iqm.tech

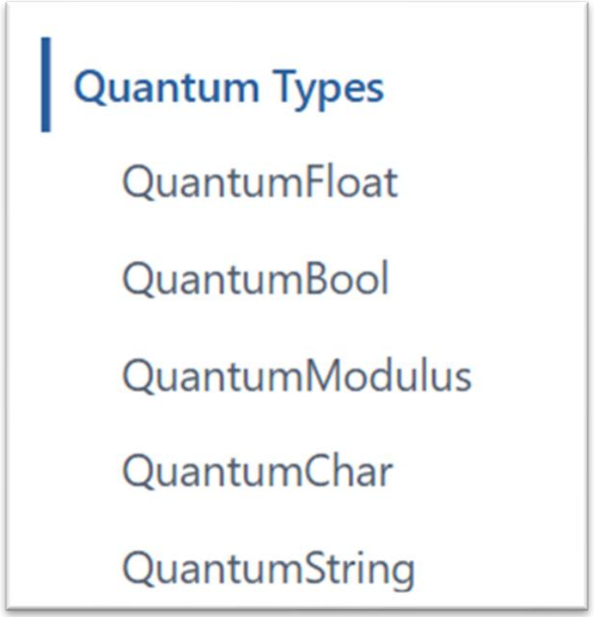
GitHub github.com/eclipse-qrisp

QuantumTypes

- **QuantumVariables** allow for many automatizations and quality of life improvements such as **hidden qubit management, de/encoding to human readable labels** or –

Typing

- All **QuantumTypes inherit** from the QuantumVariable class.
- User can create **custom types**

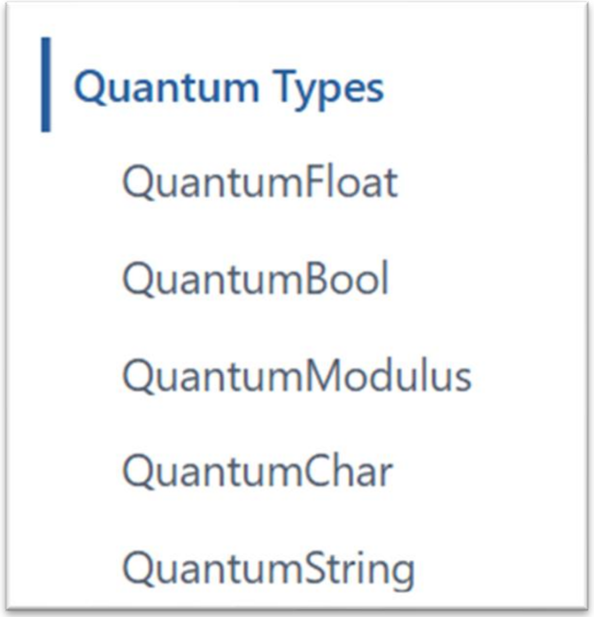


QuantumTypes

- **QuantumVariables** allow for many automatizations and quality of life improvements such as **hidden qubit management, de/encoding to human readable labels** or –

<u>**Typing**</u>

- All **QuantumTypes inherit** from the QuantumVariable class.
- User can create **custom types**



QuantumFloat

From binary to numbers:

- Each **bitstring** has its **associated numerical value**

Classical Floats (integers)

$$001 \Longrightarrow 1 \quad 010 \Longrightarrow 2$$

$$011 \Longrightarrow 3$$

$$111 \Longrightarrow 7$$

From Quantum binary to numbers:

- Each **bitstring** represent one possible **Quantum state**, which can be in a superposition
- Each **Quantum state** has its **associated numerical value**

Quantum Floats (integers)

$$|001\rangle \Longrightarrow |1\rangle$$

$$|011\rangle \Longrightarrow |3\rangle$$

$$|111\rangle \Longrightarrow |7\rangle$$

QuantumFloats can be in a superposition!

$$|001\rangle + |011\rangle + |111\rangle$$

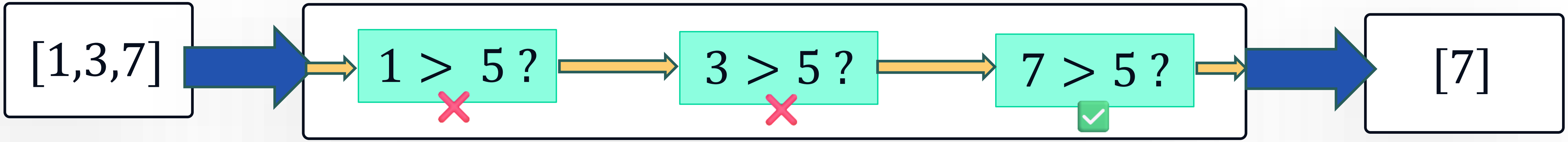
$$=$$

$$|1\rangle + |3\rangle + |7\rangle$$

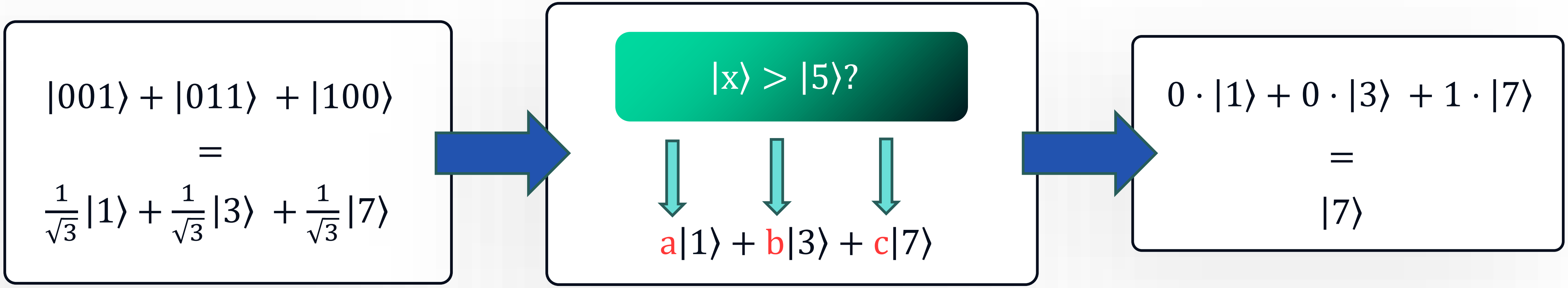
QuantumFloat

GOAL: Find all elements >5 !

Classical Algorithm



Quantum Algorithm



QuantumFloat

```
class QuantumFloat(msize, exponent=0, qs=None, name=None, signed=False)
```

```
1    from qrisp import QuantumFloat
2    a = QuantumFloat(3, -1, signed = False)
```

Number of values stored

$2^{3} = 8$

With precision

$2^{-1} = 0.5$

```
4    for i in range(2**a.size): print(a.decoder(i))
```

```
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
```

`QuantumFloat(4, -1)`

Number of values stored

$2^4 = 16$

With precision

$2^{-1} = 0.5$

```
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
```

`QuantumFloat(4, -2)`

Number of values stored

$2^4 = 16$

With precision

$2^{-2} = 0.25$

```
0.0
0.25
0.5
0.75
1.0
1.25
1.5
1.75
2.0
2.25
2.5
2.75
3.0
3.25
3.5
3.75
```

QuantumFloat

```
class QuantumFloat(msize, exponent=0, qs=None, name=None, signed=False)
```

```
from qrisp import QuantumFloat
a = QuantumFloat(3, -1, signed = False)
```

Number of values stored

$$2^{3} = 8$$

With precision

$$2^{-1} = 0.5$$

```
for i in range(2**a.size): print(a.decoder(i))
```

```
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
```

```
from qrisp import QuantumFloat, h
a = QuantumFloat(2)
h(a[0])
print(a)
```

```
{0: 0.5, 1: 0.5}
```

```
b = QuantumFloat(3)
b[:] = 4
print(b)
```

```
{4: 1.0}
```

```
c = a+b
print(c)
```

```
{4: 0.5, 5: 0.5}
```

Lab-time: QuantumTypes

https://tinyurl.com/QS20206-Day1-LAB-participant

qrisp.eu

IQM Academy .com

IQM Resonance

resonance.iqm.tech

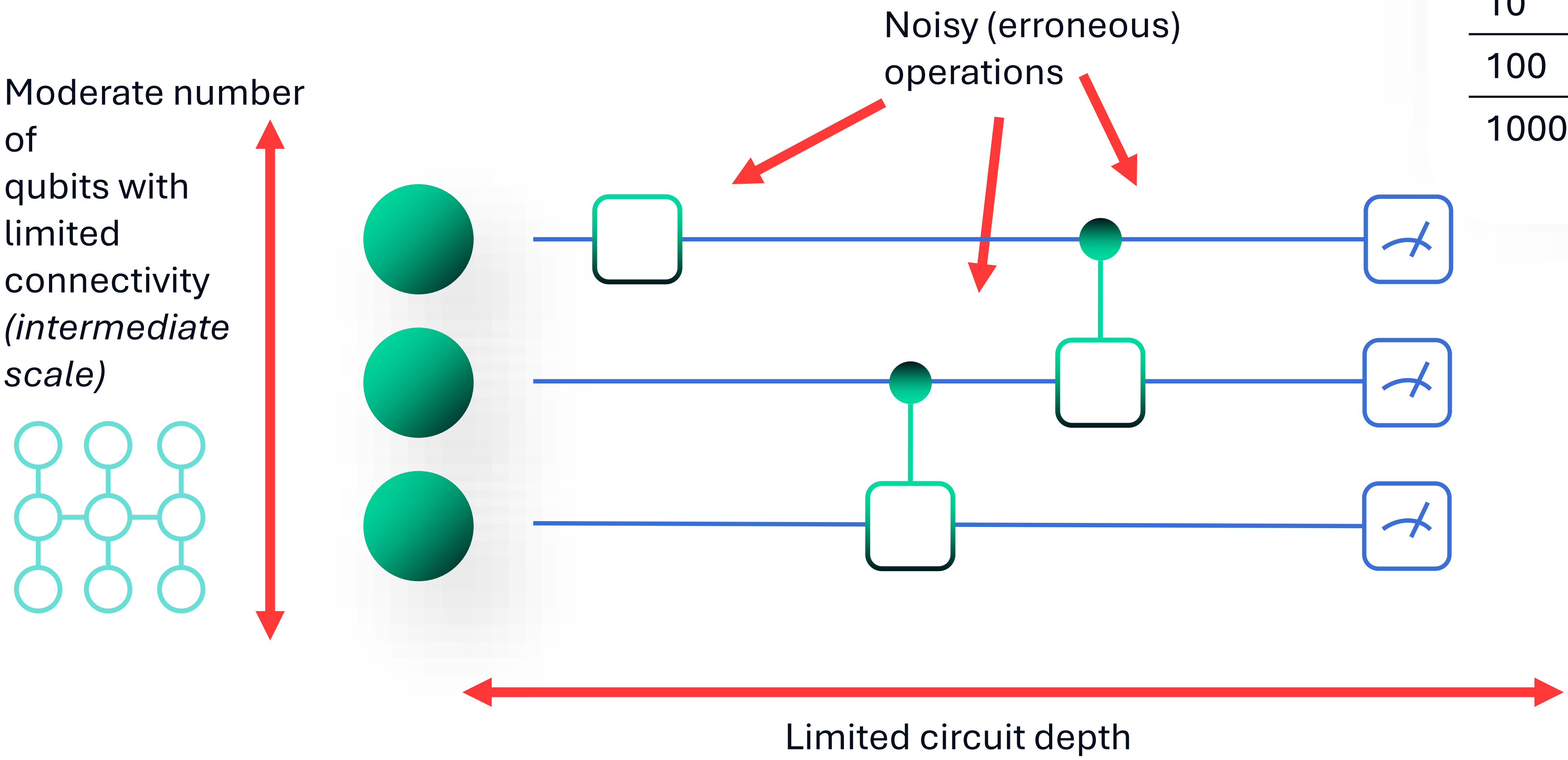
GitHub github.com/eclipse-qrisp

Quantum Computing Eras

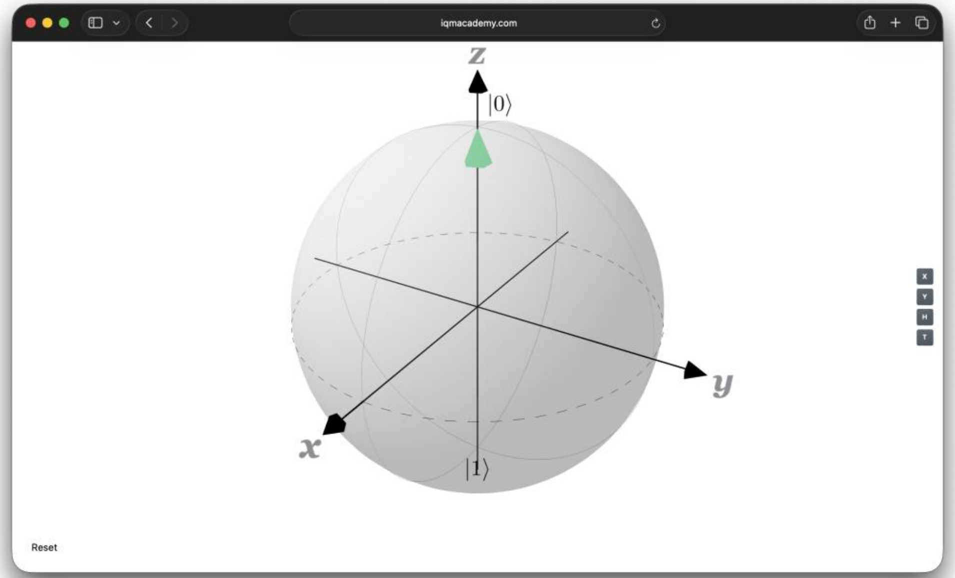
Quantum computers: NISQ era

# gates in algorithm	Algorithm fidelity
1	99.9%
10	99.0%
100	90.5%
1000	36.8%

Error rate: 1/1000 = 0.1%



One type of noise: slightly imperfect gates



If we apply a gate such as the X gate multiple times in a row, we see the error stacking up

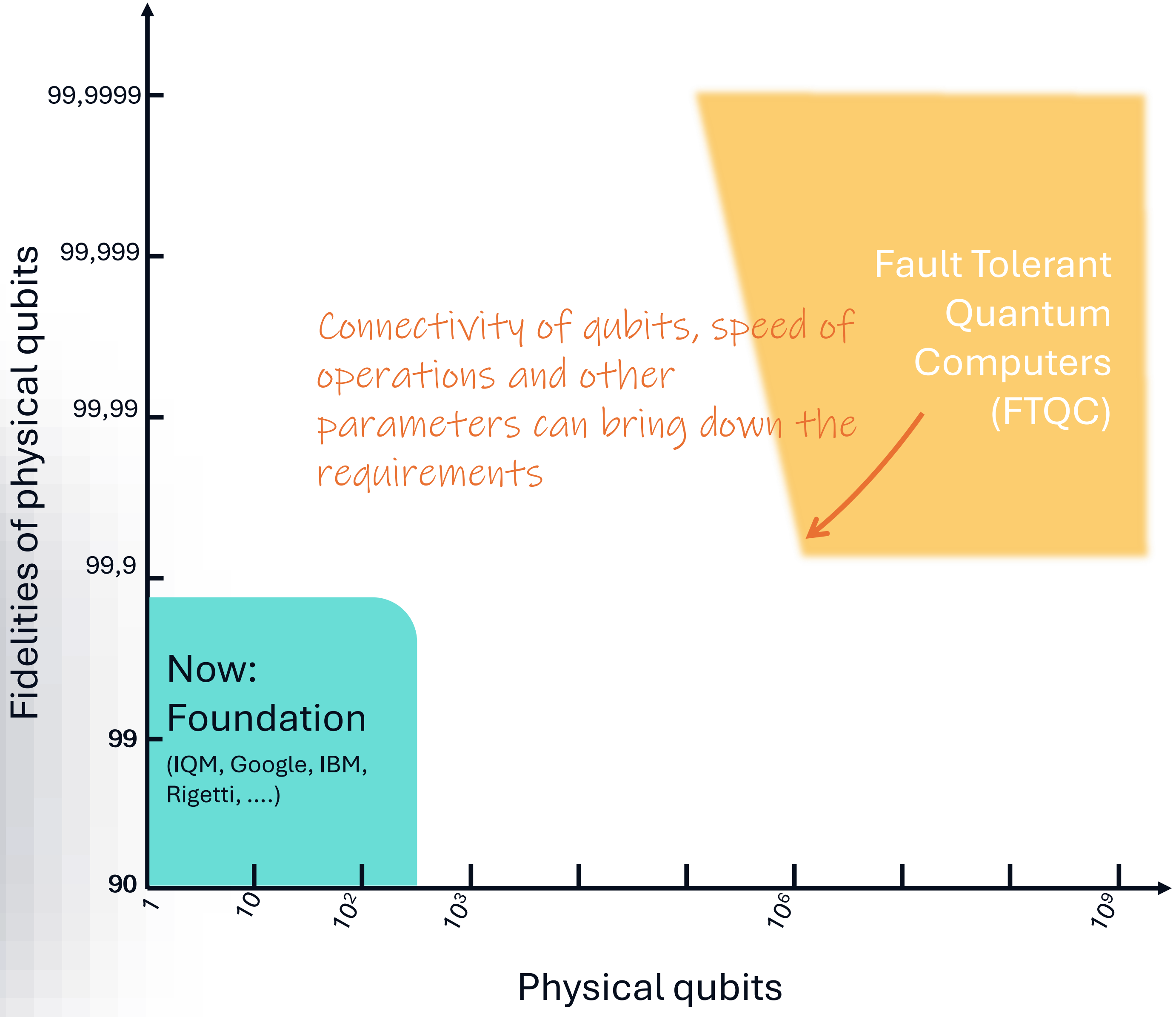
https://tinyurl.com/iqm-NoisyGates

NISQ and FTQC

- Horizon: Fault Tolerant (FT) area

Why so complex?

- Error correction is based on redundancy
- 1 single fault tolerant qubit requires multiple physical qubits
- Noise needs to be reduced below a certain threshold

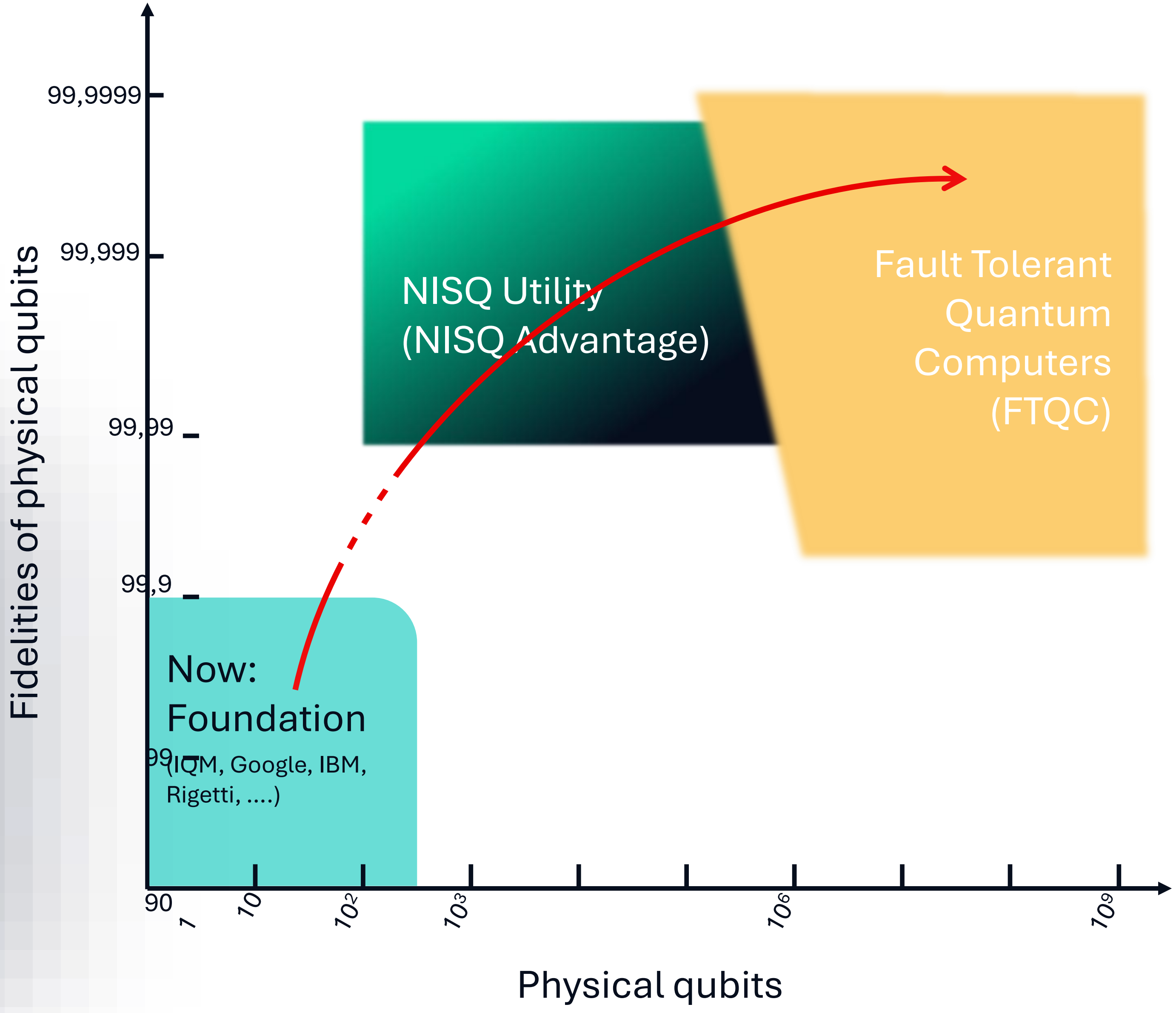


NISQ and FTQC

- Horizon: Fault Tolerant (FT) area

But before that: NISQ Utility

But before that: NISQ Utility (NISQ Advantage)
- Hybrid approaches
- Potential for NISQ quantum utility
- Solving tasks
  - faster,
  - better,
  - or using less energy



Thank you! We hope you learned a lot!

https://tinyurl.com/QS20206-Day1-LAB-participant

Contact
Niklas Steinmann
Niklas.Steinmann@iqm.tech

qrisp.eu

IQM Academy.com

IQM Resonance

resonance.iqm.tech

GitHub github.com/eclipse-qrisp