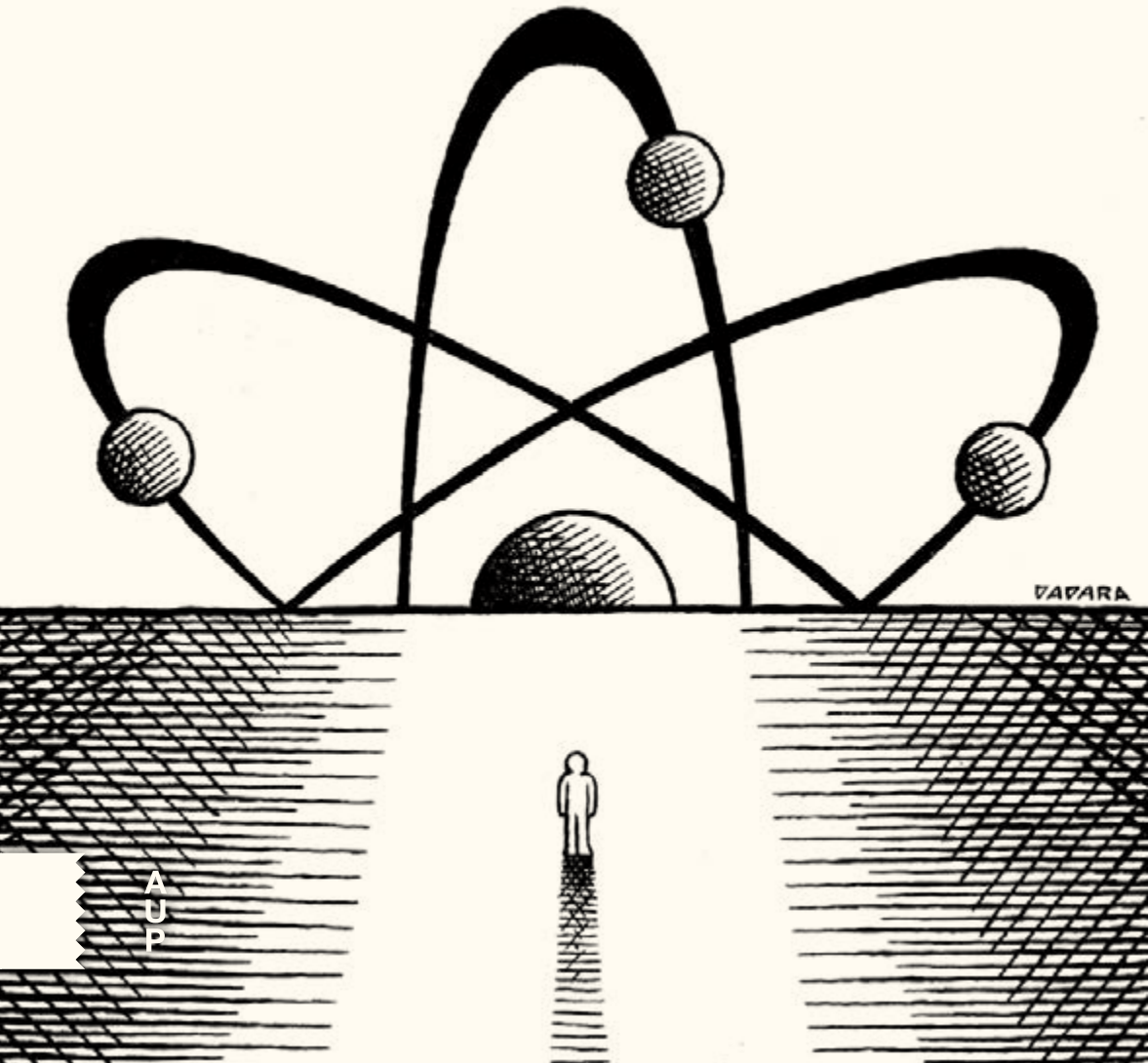
KOEN GROENLAND

INTRODUCTION TO QUANTUM COMPUTING FOR BUSINESS



AUP

Introduction to Quantum Computing for Business

Introduction to Quantum Computing for Business

Koen Groenland

Amsterdam University Press

Cover illustration: © Dadara

Cover design: Mijke Wondergem
Lay-out: Crius Group, Hulshout
Illustrations: © Dadara

ISBN 978 90 4856 898 7
e-ISBN 978 90 4856 899 4 (pdf)
DOI 10.5117/9789048568987
NUR 120

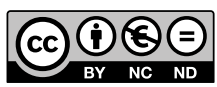


Table of Contents

Part 1 The essentials

Part 3 The hardware and strategic actions

Part 4 The final bits

Part 1

The essentials

Preface: Why this book?

'Quantum computing will change everything', the man in front of me said. Standing tall and confident, he took another sip of his drink before continuing, 'It will be the biggest revolution since the invention of the transistor. Imagine a world where we can cure any disease with personalised medicine. A world where new energy sources will free us from our dependence on fossil fuels. Not to mention that…'.

'Well…', I tried to interrupt, but the man rattled on, passionately.

'It will finally enable us to build general Artificial Intelligence that can take over our tedious everyday jobs, so 95% of our population no longer has to work!'

'You know that quantum computers are still quite some years away, right?', I countered. He leaned in, eyes gleaming with excitement.

'That's what most people think. But the reality is, we're closer than ever. Quantum supremacy has already been achieved. Google did it in 2019. Since then, progress has been exponential. Did you see the presentation by that guy from Goldman Sachs? Their investments are already seeing higher returns than ever since their new Monte Carlo algorithm'.

The above conversation captures a feeling that many seasoned experts in quantum computing have. A group of enthusiasts presents 'quantum' as a revolutionary technology with unprecedented capabilities. Plentiful reputable sources report how next-generation devices are key in tackling climate change, revolutionising AI, and building unhackable networks.

Experts who are actually *building* quantum computers are much, much more reticent. At an academic conference, you will hear a completely different story. Scientists ridicule the absurd claims that some consultants and startups make. They will point out that the applications of quantum computers are still highly uncertain and that we're still searching for convincing use cases.

The quantum scene seems divided into two distinct worlds. One is the business world, eager to reach out to anybody who will listen to the game-changing capabilities of quantum computers. The other is a more cautious community of scientists and technical experts, quietly working to make quantum computers a reality, sharing their results in specialised papers that require a PhD to understand.

I was fascinated by this paradoxical situation. Who is right? How powerful are these quantum computers really, and how do they compare to existing technologies? In what year will we have a large-scale quantum computer, and what will it look like? These are billion-dollar questions, but the answers vary wildly, depending on who you ask.

After searching for these answers for a decade, I find myself in a unique position to address most of these questions. As a former academic researcher, I acquired a detailed understanding of quantum computers and their algorithms. For the past four years, I have had the privilege of forming R&D collaborations with startups, enterprises, and governments while having countless meetings with CEOs, research leads, and policymakers. I've seen the perspectives from both worlds and can cut through dishonest and deceptive claims. Additionally, after training many new colleagues and setting up professional learning programmes, I have developed a good intuition about what newcomers *want* to know about quantum technology and how to explain it in an accessible way.

However, the decisive factor that led me to write this book is my unease about other sources. Like many others in this field, I'm unhappy with the many hyped and unbalanced articles that populate the top entries in Google search results (or even the *New York Times* best-selling books[1]). There is a clear need for a neutral source of information that others can reference when disagreeing about facts or debunking myths, and I'm very happy that it's finally complete.

That doesn't mean that this book contains only confirmed facts – not at all! Writing about a computer of the future comes with mountains of uncertainty. In 2005, nobody could have predicted that, a mere five years ahead, everyone would be playing games and consuming the internet on their smartphones. In 2015, nobody could have predicted the impact of Large Language Models like ChatGPT. And, indeed, today's best predictions of a future quantum revolution will prove not to be entirely accurate either.

Even worse, experts wildly disagree in several cases. For example, the usefulness of quantum AI and optimisation is vigorously disputed, and the rate at which hardware will progress depends on many yet-to-discover breakthroughs. The best I can do is describe various perspectives on these matters and highlight the most salient arguments from both sides.

My colleagues and I had many discussions and disagreements, without which I wouldn't have been able to gather the facts and opinions in this book. And it shouldn't stop there. I continue to welcome criticism, opinions, and feedback about these complex topics, aiming to refine these texts even more in future updates.

Even though much remains uncertain, I think that a reliable indication of the prospects of quantum computing is more important than ever. Quantum startups are acquiring huge investments, allowing them to hire managers, software developers, salesmen, and marketers. Governments need informed policymakers, and journalists should cover quantum breakthroughs. Pretty much every organisation that deals with IT will want to keep a close eye on the impact that 'quantum' will have on them.

This book is for precisely these people, who don't need to understand all the technical details but who still need to talk, read, and write about quantum technologies. We will not dwell on the underlying maths or physics, but rather focus on the *functionality* of a quantum computer: the opportunities, threats, and concrete actions that organisations can take.

How should you read this book? I chose to split the content into three parts. The first part contains the essentials that we recommend everyone should read. This is an efficient way to learn all the background that you need – it will prime you for understanding other sources and give you some depth in professional discussions or meetings. Going into more detail, Part Two and Three contain more information about the (software) applications and the (hardware) devices, respectively. A final, fourth part is reserved for additional resources that can be useful or fun when continuing your quantum journey.

Note

1. I am referring to Michio Kaku's book *Quantum Supremacy*, but before you even consider reading it, you might like to see the book review by a professor in quantum computer science at https://scottaaronson.blog/?p=7321.

1 An introduction to the quantum world

At a glance

You don't need to understand quantum mechanics to understand the *functionality* of quantum computers. But if you insist, quantum mechanics describes the behaviour of the smallest particles. It leads to many counter-intuitive phenomena: computer memory can store multiple pieces of data simultaneously, but, when measured, nature selects just a single piece and throws away all the others.

If you want to drive a car, do you need to understand how its engine works? Of course, you don't! In a similar vein, you don't need to know the details of quantum physics to read the rest of this book. So, feel free to skip this chapter.

Nevertheless, we know that most people *want* to have some conceptual intuition about what quantum mechanics really is. It is not natural to leave one of the most used words in this book as an abstract concept, and it might be hard for the human brain to proceed without at least seeing some examples.

Here is my best attempt to explain quantum mechanics in accessible terms. Proceed with caution, as things will almost certainly get confusing from here.

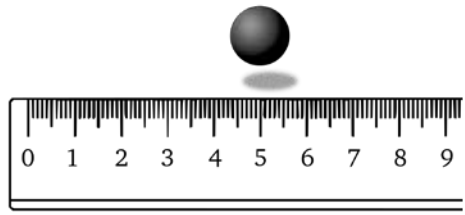
1.1 What is quantum?

Quantum physics or **quantum mechanics** is the theory that describes the tiniest particles, such as electrons, atoms, and small molecules. The theory is meant to describe the fundamental laws of nature using a set of mathematical equations, allowing us to predict cause and effect at the scale of nanometres. It answers questions like 'What happens when I bring two electrons close together?' or 'Will these two substances undergo a chemical reaction?'. You can contrast quantum mechanics to Newton's classical physics, which we learned in high school. The classical theory works great for objects the size of a building or a football but becomes inaccurate at much smaller scales. Quantum is, in a sense, a *refinement* of classical physics: the theories are effectively identical when applied to a coffee mug, but the more difficult quantum theory is needed to describe very small things.

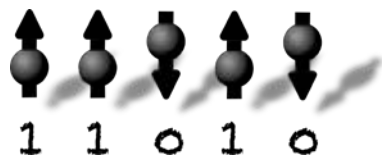
Some examples of systems where quantum could play a role are:

– Atoms and the electrons that orbit around them.
– Flows of electricity in microscopic (nano-scale) wires and chips.
– Photons, the particles out of which light is made.

We are going to need some physics jargon to proceed. We like to use the word '**state**', which is a complete description of all the physical properties of the world at one instance: the locations of all the different particles, their velocities, how much they rotate, etc. Usually, the entire universe is too big to study, so we often simplify our world to a single, isolated particle or to a limited piece of computer memory. Let's imagine a bare particle in an otherwise empty world. We may be interested in its location, which we'll call x. For example, the world might look something like the image below, which can be described by a very simple state: $x = 5$ (the ruler is just virtual).



In the spirit of computing, we might look at a 'bit' that stores information. Think of it as a tiny magnet that can either point 'up' (1) or 'down' (0). The state of a piece of memory is easy to describe, simply by expressing the bit values one by one. For example: 11010.



Importantly, the state of the world can change over time. We will often care about the state of the world at a certain moment, for example, at the beginning of a computation or at the end of it.

1.2 Four surprising phenomena

The most iconic quantum phenomenon is **superposition**. Think about any property that we can (classically) measure, such as the position of a particle or the value of a bit on a hard drive (0 or 1). In quantum mechanics, many different measurement outcomes can be somewhat 'true' at the same time: a particle can be in multiple positions at once, or a bit could be 0 and 1 simultaneously. When we say 'at the same time' we mean that, to predict

any cause and effect, we need to keep track of all these possibilities. To illustrate a superposition, I sometimes picture a quantum particle splitting into many opaque copies of itself, spread out over space, where the degree of transparency determines how likely the particle is to be found there: the darker it is, the more presence it has at that location.

To throw in some more examples of superpositions: an electron can move at a velocity of 10 m/s and 100 m/s at the same time (which obviously also leads to a superposition in its location). More relevant for us: a computer memory might store the numbers 5 and 11 'simultaneously' or even 46 different Microsoft Excel spreadsheets 'at once'. An important building block to make this all work is the **qubit**, which is any kind of hardware that can store bit values 0 and 1, and any possible superposition of these two. If we have a bunch of qubits together, we'll call it a quantum memory.

Let us illustrate the weirdness of superpositions with an example where the 46 spreadsheets each take 1 megabit (Mb) to store. A regular, classical hard drive would allocate the first Mb to a first spreadsheet, then another Mb to store the second, and so forth. In total, it would use 46 Mb. The quantum memory has an *additional* option to store the spreadsheets in superposition: using the qubit-equivalent of just 1Mb (one million qubits) it would encode all the data in just that limited amount of memory. Whereas 1 Mb of classical memory can fit just one spreadsheet, a quantum memory of 1 Mb can represent several of them, all thanks to the unique properties of quantum physics. However, as we'll see later, there is a catch to storing all that data so compactly.

How can you possibly describe a world where particles and computer memories are in superposition? For now, let's focus on an isolated particle. We specify its state using a lengthy list, where for each possible position, we store a number called the *amplitude,* which is related to how likely the particle is to be found at that location. In other words, the state describes precisely to what extent a particle is at position $x = 0$, to what extent at position $x = 1$, and so forth, for every possible location that the particle can be at. And indeed, this list could be infinitely long! Luckily, when dealing with computers, we work with simpler objects. A quantum bit

needs just two amplitudes, which denote the extent to which the bit is '0' or '1', respectively.

The amplitudes used to describe quantum states feel somewhat analogous to probabilities, which can similarly tell us the likelihood that, for example, a particle can be found at a particular location. However, there is a fundamental difference. Probabilities in the classical world help us deal with information we don't have: surely, the particle is already at some location, but perhaps we just don't know which location yet. Quantum mechanics is different. Even if we know every tiny detail about the location of a particle, we still need to describe it as a superposition. Fundamentally, the location is *not determined* yet. Hence, there is literally no better way to describe the particle than by tracking this convoluted superposition. Amplitudes are also more finicky to deal with than probabilities because these numbers can become negative (and for math experts, they can even be complex numbers).

The second weird phenomenon is how **quantum measurements** work. Why do we never observe an electron at two places at the same time? Why do I never find a car both moving and standing still? In quantum mechanics, as soon as we measure the location of a particle, it instantly jumps to a single location at random – making its location fully determined. Similarly, when we measure a qubit, it jumps to either '0' or '1'. When we measure the data in a quantum memory, we may find any one of the 46 spreadsheets that were stored. A measurement essentially changes a system into a normal, classical state.

The effect of a measurement is intrinsically random (and hence, our world is not deterministic!). But this doesn't imply that we cannot understand quantum mechanics. We can calculate the *probabilities* of measurement outcomes with incredible precision as long as we know the state before the measurement.

It is important to note that we cannot learn anything about the world without measuring – it is our only way to obtain data about physical objects. Any observation, even a slight peek at our system, is a measurement in quantum mechanics. Additionally, measurements are destructive in the sense that they change the state of the world. We fundamentally cannot 'look' at a particle without disturbing it. In fact, measurements delete all the rich data encoded in a superposition! If a particle was initially at position $x = 0$, $x = 3$ and $x = 10$, all simultaneously, then upon measurement, it jumps to one of these three options. To give you a bit of jargon, we call this instantaneous change a 'collapse.' From that moment, it is 100% at a fixed location: if, at first, we measure the particle to be at $x = 3$, then any

subsequent measurement will give the same result, until some other force moves it again. In the context of a quantum computation, this means that we should carefully choose when we perform any measurements – we cannot just peek at the data at any moment we like, or we risk disturbing a superposition.

This also means that a single piece of quantum memory cannot store an immense number of spreadsheets at the same time – at least, you wouldn't be able to retrieve each of them. To store 15 Mb worth of classical data, we need 15 Mb worth of qubits. Hence, quantum computers are not particularly useful for storing classical data.

The fact that a measurement changes the state of the world poses a serious problem for the engineers who are building quantum computers. No matter what material we construct our qubits from, they will surely interact with other nearby particles, and some of these interactions could act like destructive measurements. We call this effect **decoherence**, and, as we will see later, this forms one of the core challenges to large-scale quantum computation.

At this point, quantum data doesn't seem particularly useful. Why would we want to deal with superpositions if they lead to all this uncertainty? The important advantage stems from the way in which a quantum computer can process quantum data. Using quantum mechanics, a device can manipulate data in ways that a classical computer could never do.

That leads us to the third unique phenomenon. A quantum computer can manipulate the data it stores using so-called **quantum gates**, or simply 'gates' for short. These are rapid bursts of some physical forces that change the state of one or more qubits. They can turn a classical-looking state into a quantum superposition or vice versa. They can act like logical operations, like the AND and OR gates that are used in classical electronics, but also like new quantum logic that has no classical counterpart.

From a functional perspective, a quantum gate takes one or more qubits as input, changes their internal state, and then outputs the same number of qubits (with their altered states). In other words, the number of physical objects remains unchanged, but the overall state changes. As an example, you may think of our prototypical magnet that was initially pointing 'up', but a quantum gate might flip this to 'down'. There are many such gates possible, each having a different effect on their input. We like to give them names in capital letters, such as X, Z, H, and CX. Importantly, a quantum gate is *deterministic,* meaning that its input-output behaviour is always the same, as opposed to the quantum measurements we saw earlier.

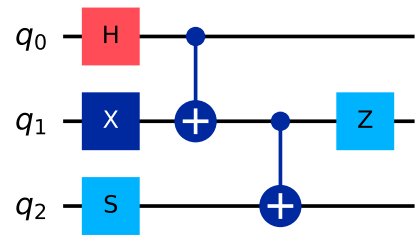
The canonical way to describe a quantum computer program is by defining a sequence of quantum gates, where for each gate, we also indicate what qubits are supposed to be the gate's input. At the end of the computation, we measure all qubits. An example of such a program, using the standard Quantum Assembly (QASM) language, is given below.

OpenQASM code

```
qreg q[3];

h q[0];
x q[1];
s q[2];
cx q[0], q[1];
cx q[1], q[2];
z q[1];
```

Resulting circuit



Together, these steps can be graphically displayed in a **quantum circuit**, as shown here on the right. Quantum circuits represent each qubit with a horizontal line and indicate time flowing from left to right. Whenever a box with a letter is displayed over a qubit line, then the corresponding gate should be applied. This isn't unlike the way we read sheet music! You may notice that sometimes, two or more gates can be performed in parallel as long as they act on different qubits.

When we run a circuit on an actual quantum computer, the final measurements lead to probabilistic outcomes. We get to see a bunch of ones and zeroes: one classical bit for each qubit. If the circuit is a good quantum algorithm, then, with high probability, these classical bits will tell us the answer we are looking for. But even then, we might need to redo the computation a few times and take (for example) the most common result as our final answer.

If you are completely confused at this point, you are not alone. The whole business of quantum superposition and quantum operations is incredibly complex and is not something you could possibly master after reading a few pages. Scientists who have studied the subject for many years are still

frequently baffled by deceptive paradoxes and counter-intuitive phenomena. On the other hand, we hope that the *functionality* of quantum circuits makes some sense: we define a list of instructions and feed them into a machine that can execute them. We don't have to know precisely what's going on under the hood!

There is one remaining quantum phenomenon to cover – one that comes with a mysterious flair surrounding it. We're talking about quantum **entanglement**, which we'll describe using the following example.

Imagine that we have two qubits, which we can transport independently from each other without disturbing the data they store. Together, the qubits can represent the states 00, 01, 10, or 11, or any superposition of these. According to quantum mechanics, we can create a very specific state where the pair of qubits is simultaneously 00 and 11. Now, imagine that computer scientist Alice grabs one of the qubits, takes it on her rocket ship, and flies it all the way to the dwarf planet Pluto. The other qubit remains on Earth in the hands of physicist Bob. Upon arriving on Pluto, Alice measures her qubit and finds outcome '1'. A deep question is: what do we now know about Bob's qubit?

Since the only possible measurement outcomes were 00 and 11, the other qubit can only be measured as '1' from now onwards. It essentially collapses to be 100% in the state '1'. But how could the Earth-based qubit possibly *know* that a measurement occurred on Pluto? What mechanism made it collapse? According to Einstein's theory of relativity, information cannot travel faster than the speed of light, which translates into a few hours between Earth and Pluto. Nevertheless, measuring the qubits in two faraway locations will always give a consistent result, even when the two qubits are measured at exactly the same time.

This paradox reveals, once again, how confusing quantum mechanics can be. However, the story above is perfectly consistent with both quantum mechanics and the theory of relativity. The core principle is that *no information can be sent faster than light between Alice and Bob*. For example, can you see why Bob has no way of detecting when Alice performs her measurement just by looking at his entangled qubit? In the most common interpretation of quantum mechanics, the Earth qubit does indeed change its state instantaneously when Alice measures her qubit, although there is no way to exploit this effect for fast messaging.

More generally, entanglement is the phenomenon where two or more faraway qubits can have *correlated* measurement outcomes that are classically impossible. There is a fascinating further discussion about the philosophy behind entanglement, but we'll leave that to other sources. What matters

to us is that entanglement leads to new functionalities that we can exploit. We will discover what these are in the chapter on quantum networks.

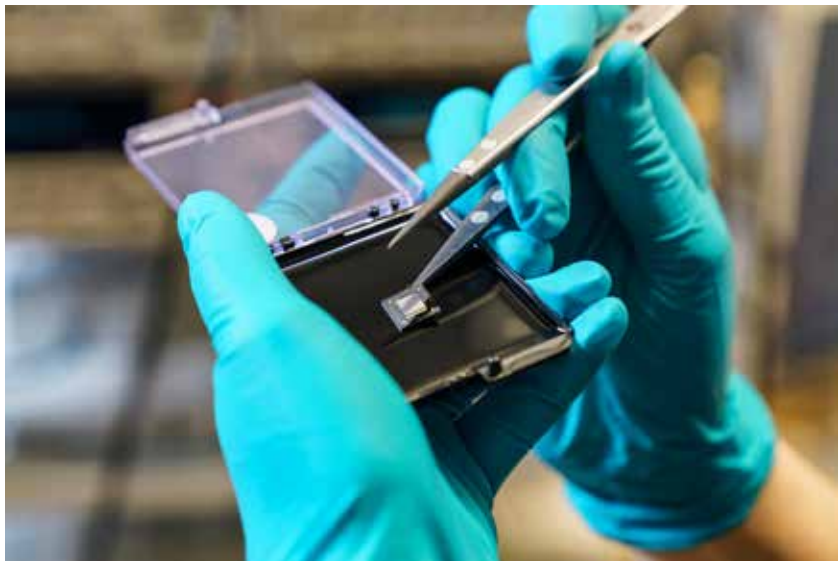
So, there you have it: four surprising phenomena you may hear frequently in quantum technology conversations. To summarise:

- Superposition: the phenomenon where a qubit is both 0 and 1 at the same time.
- Quantum measurement: measuring a quantum memory destroys superposition. The result we obtain is probabilistic.
- Quantum gates: deterministic changes to the state of qubits, which generalise classical logic gates like OR, AND, NOT. A list of several quantum gates (together with the qubits they act on) forms a quantum circuit.
- Entanglement: qubits separated over a long distance can still share unique properties.

1.3 What does a quantum computer look like?

Most large-scale computing today happens in data centres, where we don't care much about the specifics of the devices that do our calculations. We also expect that future quantum computers will mostly be tucked away in the 'cloud', making their appearance and inner workings largely irrelevant to most users. However, for this optional chapter, we can take the opportunity to view what today's cutting-edge hardware looks like. There are many different ways to build a quantum computer, each based on distinct physical systems and principles. Here, we describe the example of so-called superconducting qubits, a relatively mature platform used by companies like IBM, Google, and Rigetti and several academic institutes. Research institute QuTech in Delft, the Netherlands, was kind enough to provide photos that allow us to look inside their labs. We will see that only a tiny part of the computer is actually 'quantum', whereas most of the machine consists of classical machinery that's required to keep the computer working.

The real quantum magic happens on a chip, not unlike the computer chips used in your laptop or phone. The qubits are formed by tiny electronic circuits where the flow of electrical current is restricted to just one out of two states: the 'bit' states 0 and 1. Since this is a quantum system, the current can also be in a superposition – picture all the electrons in the wire participating both in flow '0' and flow '1' simultaneously! This only works when the chip is cooled down to unimaginably low temperatures, down to around 10 millikelvin – a hundredth of a degree above absolute zero. At these temperatures, the electronic circuits become superconducting, such



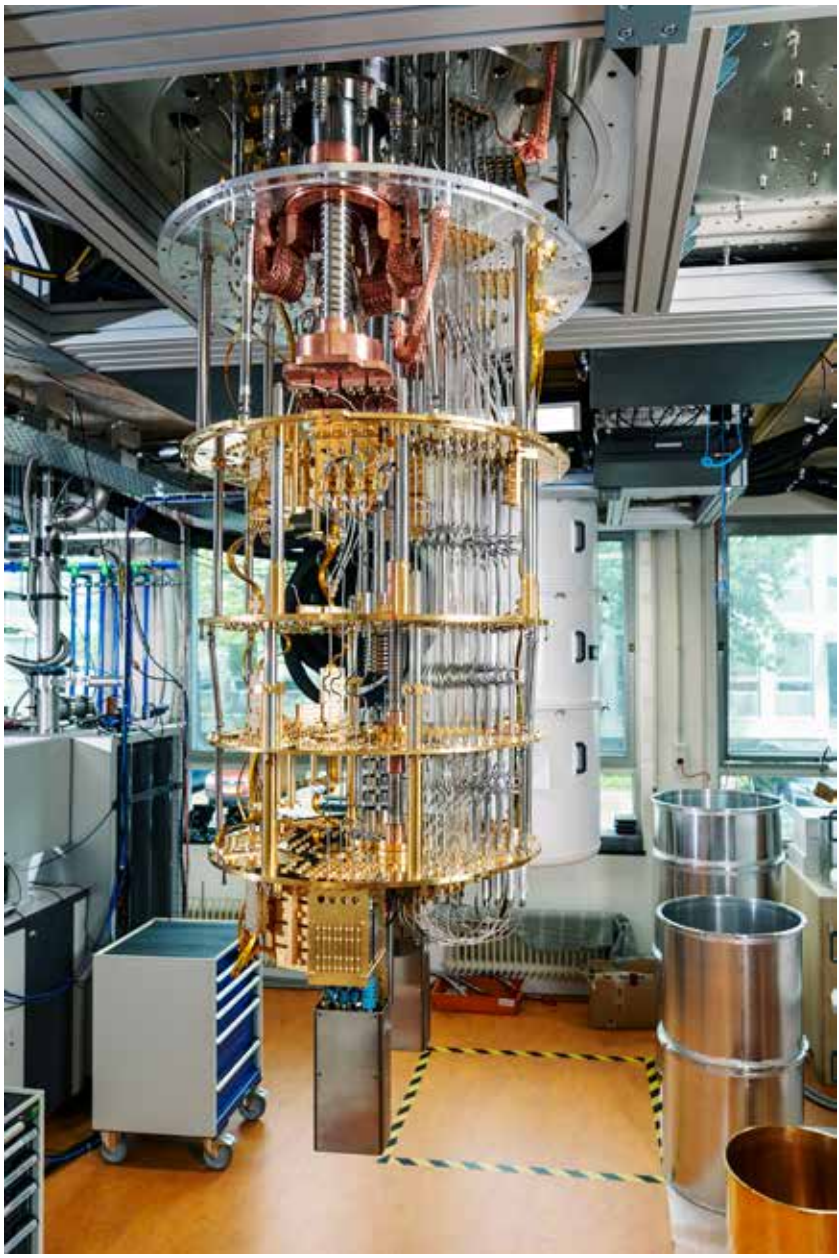
A quantum chip. Photo credits: Marc Blommaert for QuTech.

that an initial current can flow indefinitely. This is important because any damping of the current would cause unwanted disturbance to the qubit state.

The temperature constraint is why the quantum chip is placed in a massive dilution refrigerator, a cylinder of about half a metre in diameter and over a metre tall, which specialises in keeping the quantum chip cool. In the future, larger quantum computers may need even bigger fridges or combine several of these close together. Deeper parts of the fridge have increasingly low temperatures, allowing us to cool in stages. An example could be to cool a first environment to 35 Kelvin (-283 °Celsius or -396.7 °Fahrenheit), followed by subsequent stages to ~3K, 900mK, 100mK, until the final stage of ~10mK is reached.

Engineers typically suspend the fridge on the ceiling so that the higher temperatures are on top, and the ultracold quantum chip is placed at the very bottom. The internals are shaped accordingly: several layers of gold disks are hung below one another, one disk for each temperature zone. A large number of wires run between the disks, transporting signals between the ceiling and the lowermost areas. The whole structure forms the iconic metal chandelier that you often see in images, although it would all be covered by a boring metal case when the fridge is in operation.

To make the qubits do something useful, like executing a quantum gate or performing a measurement, we need to send signals into the chip. Just like with classical computers, a 'signal' is a voltage difference between



The interior of a dilution fridge, as used for superconducting quantum computers. Photo credits: Marc Blommaert for QuTech.

A stack of classical control electronics used to generate and measure electronic signals. Photo credits: Marc Blommaert for QuTech.

two or more wires. Some voltages remain constant over time, others oscillate at microwave frequencies. Having a larger number of wires can lead to more precise quantum gates, but extensive wiring also leads to two fundamental challenges. Firstly, we currently need around 2–4 wires to control a single qubit, which is problematic when we scale to millions of qubits – it's impossible to connect that many wires to a tiny chip. We'll need to find multiplexing solutions, where a single wire can serve multiple qubits at once. Secondly, wires connect the ultracold chip to other hardware that sits at room temperature, forming a channel for heat and noise to enter. The dilution fridge circumvents this by incrementally cooling and damping the signals as they travel through the different layers of the fridge, but it can only handle so many cables.

Besides the large chandelier, an array of specialised control electronics is needed to produce the necessary electronic pulses and to carefully read out the tiny signals that qubits produce when we measure them. These devices sit in one or multiple electronics racks, each half a metre wide and nearly two metres tall, similar to the ones you'll find in a typical data centre. Ironically, the actual quantum software can be written on a simple laptop, from where the instructions are passed to the control electronics to run a quantum circuit.

The state of today's quantum hardware is reminiscent of early computers in the 1940s and 1950s, which similarly occupied entire rooms and required several engineers for all kinds of laborious manual maintenance tasks. Moreover, the dilution fridges are particularly noisy – to the extent that those who operate them ideally do this from a different room – and they are fairly power-hungry. The quantum computer described above consumes around 25 kW, comparable to driving an electric car. Fortunately, we have good reasons to believe that, over the comings decades, quantum computers will become increasingly compact, efficient, powerful, and dependable, much like their classical cousins did.

1.4 Further reading

If you'd like to know more about the physics and math behind qubits, we recommend the following sources:

Quantum Country – a great online textbook about Quantum Computing by Andy Matuschak and Michael Nielsen.

QuTech Academy's *School of Quantum* explains a broad range of quantum topics using short videos.

(YouTube) A video tour that looks inside IBM's superconducting quantum computer.

2 The background: Why are we so enthusiastic about quantum technology?

At a glance

Quantum technology is an umbrella term for devices that exploit quantum phenomena such as superposition and entanglement. The most notable innovations are expected in computers, networks, sensors, and simulators.

Quantum computers can have a speed advantage thanks to their ability to run quantum algorithms, which solve specific problems in much fewer steps than conventional methods. However, quantum computers are expected to have relatively low clock speeds, so the algorithmic advantage must be significant before a practical speedup manifests itself.

2.1 What is quantum technology?

Quantum physics, the rules that dictate the behaviour of the tiniest particles, has already proven itself as an invaluable basis for new technologies. Without this scientific theory, many invaluable tools like LED lighting, MRI scanners and solar cells may not have been invented. And it's still relevant to push the limits of innovation, with nano-size vehicles that consist of just a few atoms or ever-smaller transistors on computer chips on the horizon.

Just ahead of us is a new paradigm, which we'll call **quantum technology**. The distinguishing factor is that it goes beyond merely building stuff from small particles. Quantum technology is about devices that perform certain processes in a fundamentally different way. That is, the data (or operations) we work with can have special properties unique to quantum physics, such as superposition and entanglement.

In our jargon, we will refer to 'classical' technology for devices that don't carefully exploit the possibilities of quantum physics – they are based on 'classical' physics that we're used to from high school. Your laptop and phone are examples of classical computers, and they're connected to the classical internet. The internal transistors and electrical circuits might be so tiny that quantum physics is relevant there, but the fundamental point is that the information that they process is purely classical. Whereas classical computers work with 'bits', quantum technology will need a different type

of information carrier that itself can be controlled at a quantum-mechanical level. We'll call these objects 'quantum bits', or 'qubits' for short.

Within the field of quantum technology, we distinguish four categories:

- **Quantum computers** are devices that use quantum physics to perform automatic calculations to solve a problem. Computing is considered the most impactful application for most organisations, hence it's the main focus of this book.
- **Quantum networks** are connections between quantum devices over which *qubits* (or similar forms of quantum data) can be transmitted. The most relevant use case is to strengthen the cryptography used by classical computers, but there are many more applications.
- **Quantum sensors** are devices that exploit the effects of quantum physics to accurately measure certain quantities, such as a magnetic field or the strength of the Earth's gravity. Quantum clocks also fall into this category.
- **Quantum simulators** are devices similar to quantum computers, except that they specialise in solving a limited set of problems. Typically, they are built to reproduce the behaviour of atoms and electrons in a specific molecule or a piece of material, allowing us to measure properties like energies and reaction rates.

Each of these categories accomplishes a different goal or functionality. For now, we'll remain agnostic about how they are built – it will be a task for hardware engineers to figure out how our desired functionality is best implemented. Since all these devices have to deal with quantum-mechanical processes under the hood, it is not uncommon that they use similar building blocks. In this book, we mainly focus on **computers** and **networks** because these seem to have the biggest impact on typical (business) users.

2.2 The importance of high-performance computing

The abundance of cheap computational power has given humanity incredible wealth. We automated the most tedious tasks to free up time for leisure and to solve other urgent problems. It allowed us to scale factories, supply chains, and logistics to unprecedented sizes, allowing us to transport resources around the globe at minimal costs. Thanks to computer-aided design, the performance of computer chips, aeroplane wings, heart monitors, and LCD screens has improved with every generation.

Today, our computers are already incredibly fast. In fact, for many applications, there is little economic gain in making these computers even faster.

Decades-old machines can successfully oversee factory operations, and writing a text document or scheduling a meeting with eight busy colleagues is not limited by the speed of your computer in any way.

However, this book is specifically about the applications where we are still hungry for more computational power. For example, by feeding more data into weather models, forecasts can become more accurate. If staff rostering would take less time, we could take more last-minute changes into account. Accurate predictions of drug reactivity in the human body could save on costly medical trials and reduce the time to market. Machine learning models like ChatGPT are still demanding more training hours to produce more sophisticated results.

It should be clear that we're not talking about computations that happen on your laptop. We're thinking of problems where somehow there's value in investing in the fastest possible computers on Earth. This is the domain of **high-performance computing (HPC)**, colloquially called *supercomputers*. Merely looking at the market, there seems to be incredible value in computing stuff: companies and academics spend tens of billions of dollars on them,[1] and hardware suppliers like Nvidia have rapidly grown to become among the most valuable companies. We should keep a close eye on this field because the kinds of problems that are now being crunched in HPC are likely to be the ones where radically new computational tools like quantum computers can have the biggest commercial impact.

2.3 Why can quantum computers have an advantage?

A naive view of quantum computers is that they're simply *faster* than their conventional cousins. Or perhaps one may naively point at Moore's law: with transistors reaching atomic scales, we run into quantum effects, so quantum physics may help us make better chips. However, none of these are our core motivations for looking at quantum computers.

When we talk about a computer's speed, most people will refer to its clock speed: the number of basic computational steps that a single processor core can complete in one second. Unfortunately, it seems unlikely that quantum computers will catch up with classical machines in terms of raw clock speed any time soon, partly because the speed of a modern CPU is already spectacular. A modern desktop processor, or even the one in your phone, works at a rate of several GHz, that is, several billions of steps per second. In each of these steps, a broad palette of operations can be applied to astronomically large numbers – modern chips work with 64-bit values,

meaning that numbers up to 18,446,744,073,709,551,615 can be processed. Each of these elementary steps can be something like addition, multiplication, a comparison, etc., and we have powerful tools to weave these basic operations together to form efficient software.

Now, quantum computers are supposed to be even faster, right? Well, it's not hard to find support for that claim:

News headers by Techradar[2] and IFLScience[3].

You may be disappointed to hear that, as of 2024, quantum computers cannot even add or multiply numbers of more than 3 or 4 bits. And even if they could, their rate of operation would by no means reach several GHz, but more likely several MHz (a few *million* operations per second) at best. In other words, they're more than a thousand times *slower*. To make things worse, the information in quantum computers is extremely fragile and needs to be constantly checked and corrected using so-called **error correction**. This is a form of overhead that could make quantum computers another several orders of magnitude slower. Even in the far future, when quantum computers are more mature and more reliable, we still expect them to be much slower than the classical chips at that time.

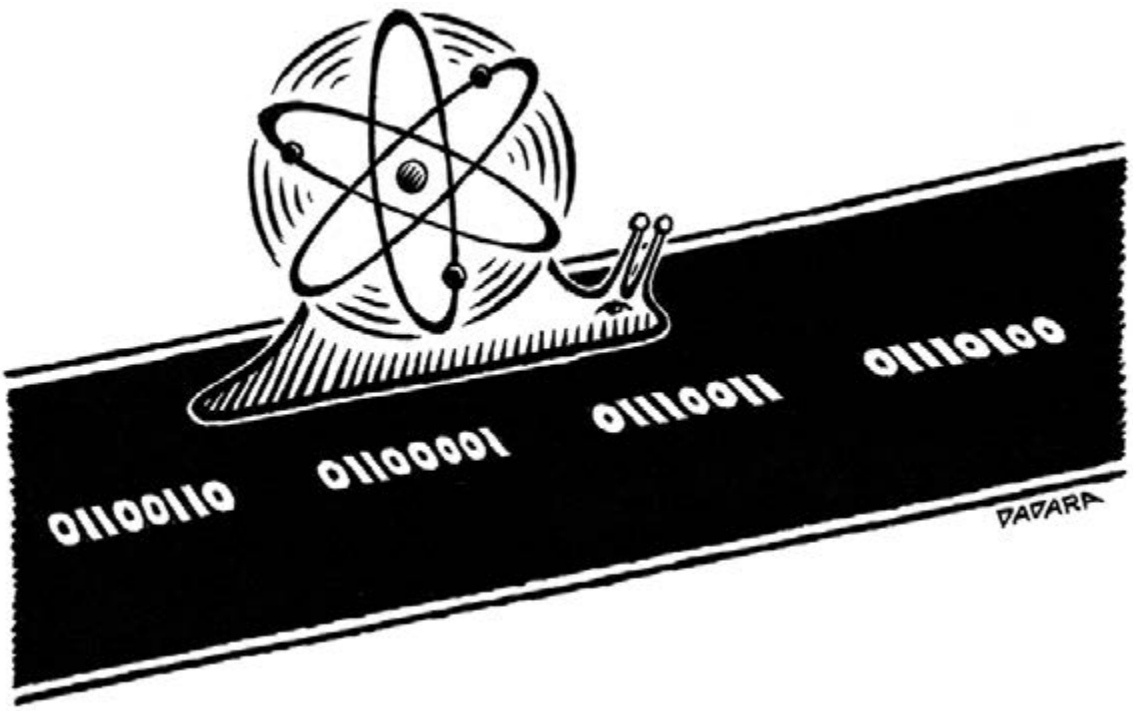
How does this rhyme with the news about ever-faster quantum computers? And why are we still interested in these slow machines? As we claimed before, we hope to do certain computations in a **fundamentally different way**. Let's look at a beautiful analogy that Andy Matuschak and Michael Nielsen bring up in their online course Quantum Country[4].

Imagine that you'd like to travel from Morocco to Spain, which are separated by a small piece of sea called the Strait of Gibraltar. If your technology does not allow you to cross the sea, then you'd need to take a large detour, all the way through North Africa, past the Arabian Peninsula, and through Europe, before you can reach your destination. This represents the steps taken by a classical computer. In the same analogy, a quantum computer grants you the ability to traverse both land and sea (much like a hovercraft) so that you can take a much more direct route.

The beauty of quantum computation is that we have a fundamentally different way to travel (do computations), which can sometimes bring us to our destination using a shorter route (doing fewer computational steps). Even with a much slower vehicle (computer), one may arrive at the destination sooner. In fact, the quantum advantage often grows as problems become larger and more complicated.

The analogy also shows that quantum computers do not always have an advantage: you would not want to travel from Amsterdam to Berlin by hovercraft. Unfortunately, in many cases, we don't yet know what the fastest means of transportation is. It is still an active area of research to completely map out the landscape over which quantum and classical computers can travel and to determine which problems allow a speedup, and which don't.

For this reason, we don't expect that classical computers will be replaced any time soon. Instead, classical and quantum processors will live side by side, and programmers will pick whichever tool is better suited to solve a certain problem. The situation could be similar to how we use graphical processing units (GPUs) today, which offer tremendous


01100110
01100001
01110011
01101010
DAOARA

speedups for the training of artificial intelligence models but are not made to replace regular classical processors (CPUs). Perhaps we should even give quantum computers a similar abbreviation, like 'QPU' for Quantum Processing Unit.

In the analogy with the Strait of Gibraltar, the precise route that you travel denotes the chosen **algorithm**. In the field of computer science, an algorithm is a step-by-step list of instructions that describes how a computational problem should be solved. The 'steps' here should be sufficiently simple so that it is completely unambiguous how to do them. They could be operations such as adding, multiplying, or comparing two numbers. Needless to say, the fewer steps the algorithm requires, the better.

By exploiting quantum mechanics, a quantum computer introduces new basic steps that are impossible to perform on a classical computer. For example, the previous chapter introduced quantum logic gates that generalise operations like AND and OR. Using these building blocks, we can formulate quantum algorithms that take much fewer steps than the best classical algorithm ever could!

In the end, the time needed to solve a problem can be very roughly calculated as:

"Time to solve a problem" = "time per step" × "number of steps required"

The 'time per step' is a property of the hardware that you use. Clearly, a faster CPU will lead to faster solutions. The 'number of steps required' is dictated by the algorithm. The latter is precisely how quantum computers can offer spectacular speedups. As long as the improvement in the 'number of steps required' compensates for the disadvantage in 'time per step', a quantum computer can help us solve problems in less time!

A recurring theme in this book is the search for industrially relevant quantum algorithms. This turns out to be more challenging than it seems at first sight. Quantum algorithms are built on deep and complex mathematics, rely on counter-intuitive quantum phenomena, and require inventive new methods to tackle a problem. Simple tweaks to existing classical algorithms are rarely sufficient. In fact, for most problems, no quantum speedups have been identified at all, despite the best attempts by scientists worldwide. We might go as far as to say that, even if we had a large-scale quantum computer today, its value would be limited. For this reason, the ongoing development of novel algorithms is exceedingly important.

2.4 From algorithm to software

In the end, simply finding a good algorithm is not enough: it has to be turned into software, a piece of language that explicitly tells a computer how to execute the step-by-step instructions.

The difference between 'algorithms' and 'software' is subtle. An algorithm is a purely mathematical description that describes precisely how numbers should be manipulated. It could tell which two numbers must be multiplied, what function must be evaluated, or how an image must be transformed. However, different computers can use different types of processors and memory, and an algorithm does not describe how these operations are done *on a specific computer*. This is where software comes into play. It describes precisely what hardware operation must be called, where each number is stored in memory, and how an image is represented in binary.

As an analogy, you may think of the algorithm as a recipe to bake the perfect chocolate cookie. The algorithm should unambiguously describe what should happen to the ingredients: in what order they should be mixed, how long they should be heated at what temperature, etc. However, to build a factory that produces these cookies, you need to be even more specific: Where is the sugar stored? Out of what pipe does the dough flow? How are cookies laid next to each other in the oven?

Fundamentally, core scientific breakthroughs come from finding new algorithms. Once a new algorithm is found, it can be re-used many different times on any capable machine (assuming a good software developer will turn it into appropriate code!).

In this book, we care less about quantum software and more about quantum algorithms. Firstly, the algorithms tell us precisely the functionality that quantum computers can offer. Moreover, we don't yet know how a mature quantum computer will be programmed or how quantum hardware and software will change in the following years. On the other hand, once a new algorithm is found, it can be cherished forever.

Now that we have come to appreciate algorithms, it is natural to ask *which* quantum algorithms we know of. What problems do quantum computers solve well? And how do these algorithms compare to their classical equivalents? This will be the topic of the next chapter.

2.5 Further reading

The Map of Quantum Computing (YouTube) – A 30-minute overview video by Domain of Science that forms a great supplement to this book.

Chris Ferrie's book *What You Shouldn't Know About Quantum Computers* debunks several myths about quantum computers, presented in an accessible way.

Are you looking for a much more extensive and technical source that covers pretty much everything there is to know about quantum computers? French consultant Olivier Ezratty has written a 1500+ page book, *Understanding Quantum Technologies*.

2.6 Notes

1. See e.g. https://www.marketsandmarkets.com/Market-Reports/Quantum-High-Performance-Computing-Market-631.html and https://www.mordorintelligence.com/industry-reports/cloud-high-performance-computing-hpc-market.
2. Wyciślik-Wilson, S.E. (2019) '*Google creates quantum chip millions of times faster than the fastest sup*ercomputer', *TechRadar*. https://www.techradar.com/news/google-creates-quantum-chip-millions-of-times-faster-than-the-fastest-supercomputer.
3. Dunhill, J. (2021) 'Chinese Scientists Create Quantum Processor 60,000 Times Faster Than Current Supercomputers', *IFLScience*. https://www.iflscience.com/chinese-scientists-create-quantum-processor-60000-times-faster-than-current-supercomputers-61475.
4. Matuschak, A. and Nielsen, M. (2019) '*Quantum Country*'. https://quantum.country.

3 The applications: What problems will we solve with quantum computers?

At a glance

The most important application areas are:

1. the simulation of material properties and chemical processes;
2. cracking cryptography;
3. using quantum networks to distribute cryptographic keys; and
4. solving large-scale optimisation and AI problems.

Getting utility out of a quantum computer is not straightforward. It requires an algorithm that beats all other known methods (even those that run on very fast classical computers), and it must tackle a problem with real-world relevance. Especially in optimisation and AI, we have not found a convincing 'killer application' yet.

In the previous chapter, we saw that quantum algorithms can solve certain problems in fewer steps, allowing a large-scale quantum computer to complete specific tasks much faster than any classical computer could. However, the precise speedup depends strongly on the task at hand. Therefore, the most important question in this field is: for which problems do quantum computers offer a meaningful advantage?

The Quantum Algorithm Zoo[1] lists pretty much all known quantum algorithms. It has become an impressive list that cites over 400 papers. Unfortunately, upon closer inspection, it's hard to extract precisely the useful business applications, for a few reasons. Some algorithms solve highly artificial problems for which no real business use cases are known. Others may make unrealistic assumptions or may only offer a speedup when dealing with an outrageously large amount of data (that we never encounter in the real world). Nevertheless, scrolling through it is definitely recommended.

For this book, we take a different approach. We focus specifically on algorithms with plausible business applications. To assess their advantage, we split our main question into two parts:

– What applications offer a quantum speedup?
– How large is this speedup in practice?

3.1 What applications offer a quantum speedup?

We foresee four major families of use cases where quantum computing can make a real impact on society. We briefly discuss each of them here. For more details, we dedicate a more in-depth chapter to each application family in Part 2.

1. **Simulation of other quantum systems: Molecules, materials, and chemical processes**

Most materials can be accurately simulated on classical computers. However, in some specific situations, the locations of atoms and electrons become notoriously hard to describe, sometimes requiring quantum mechanics to make useful predictions. Such problems are the prototypical examples of where a quantum computer can offer a great advantage. Realistic applications could be in designing new chemical processes (leading to cheaper and more energy-efficient factories), estimating the effects of new medicine, or working towards materials with desirable properties (like superconductors or semiconductors). Of course, scientists will also be excited to simulate the physics that occur in exotic circumstances, like at the Large Hadron Collider or in black holes.

Simulation is, however, not a silver bullet, and quantum computers will not be spitting out recipes for new pharmaceuticals by themselves. Breakthroughs in chemistry and material science will still require a mix of theory, lab testing, computation, and, most of all, the hard work of smart scientists and engineers. From this perspective, quantum computers have the potential to become a valued new tool for R&D departments.

2. **Cracking a certain type of cryptography**

The security of today's internet communication relies heavily on a cryptographic protocol invented by Rivest, Shamir, and Adleman (RSA) in the late 70s. The protocol helps distribute secret encryption keys (so that nobody else can read messages in transit) and guarantees the origin of files and webpages (so that you know that the latest Windows update actually came from Microsoft, and not from some evil cybercriminal). RSA works thanks to an ingenious mathematical trick: honest users can set up their encryption using relatively few computational steps, whereas 'spying' on others would require one to solve an extremely hard problem. For the RSA cryptosystem, that problem is *prime factorisation,* where the goal is to

decompose a very large number (for illustration purposes, let's think of 15) into its prime factors (here: 3 and 5). As far as we know, for sufficiently large numbers, this task takes such an incredibly long time that nobody would ever succeed in breaking a relevant code – at least on a classical computer. This all changed in 1994 when computer scientist Peter Shor discovered that quantum computers happen to be quite good at factoring.

The quantum algorithm by Shor can crack RSA (and also its cousin called elliptic curve cryptography, abbreviated to ECC) in a relatively efficient way using a quantum computer. To be more concrete, according to a recent paper,[2] a plausible quantum computer could factor the required 2048-bit number in roughly eight hours (and using approximately twenty million imperfect qubits). Note that future breakthroughs may further reduce the stated time and qubit requirements.

Fortunately, not all cryptography is broken as easily by a quantum computer. RSA and ECC fall into the category of *public key cryptography,* which delivers a certain range of functionalities. A different class of protocols is *symmetric key cryptography,* which is reasonably safe against quantum computers but doesn't provide the same rich functionality as *public key* crypto. The most sensible approach is replacing RSA and ECC with so-called post-quantum cryptography (PQC): public key cryptosystems resilient to attackers with a large-scale quantum computer. Interestingly, PQC does *not* require honest users (that's you) to have a quantum computer: it will work perfectly fine on today's PCs, laptops, and servers.

At the time of writing, a complex migration lies ahead of pretty much every large organisation in the world, which comes in addition to many existing cybersecurity threats. The foundations have been laid: thanks to the American National Institute of Standards and Technology (NIST), cryptographers from around the globe came together to select the best quantum-safe alternatives, culminating in the publication of the first standards in August 2024. These are the new algorithms that the vast majority of users will adopt.

Unfortunately, many governments and enterprises run a great amount of legacy software that is hard to update, making this a complex IT migration that could easily take 5–15 years, depending on the organisation. There's a serious threat that quantum computers will be able to run Shor's algorithm within such a timeframe, so organisations are encouraged to start migrating as early as possible.

A new type of cryptography comes with its own additional risks: the new standards have not yet been tested as thoroughly as the nearly fifty-year-old RSA algorithm. Ideally, new implementations will be *hybrid*, meaning that

they combine the security of a conventional and a post-quantum algorithm. Moreover, organisations are encouraged to adopt *cryptographic agility*, meaning that cryptosystems can be easily changed or updated if the need arises.

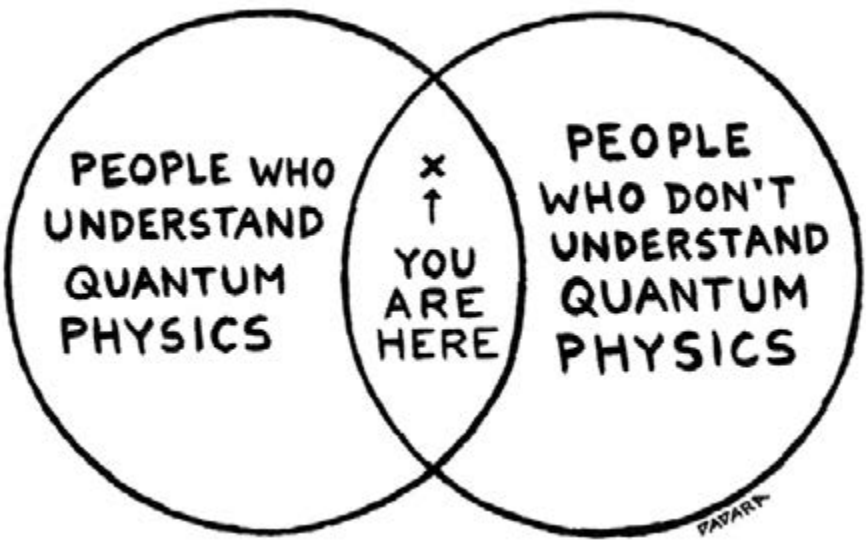
3. Quantum Key Distribution to strengthen cryptography

Out of all the applications for quantum networks, Quantum Key Distribution (QKD) is the one to watch. It allows two parties to generate secure cryptographic keys together, which can then be used for everyday needs like encryption and authentication. It requires a quantum network connection that transports photons in fragile quantum states. Such connections can currently reach a few hundred kilometres, and there is a clear roadmap for expanding to a much wider internet. The most likely usage will be as an 'add-on' for high-security purposes (such as military communication or data exchange between data centres), in addition to standard post-quantum cryptography.

Unfortunately, we often see media articles suggesting that QKD is a solution to the threat of Shor's algorithm and that it would form an 'unbreakable internet'. Both claims are highly inaccurate. Firstly, QKD does not offer the wide range of functionality that public key cryptography offers, so it is not a complete replacement for the cryptosystems broken by Shor. Secondly, there will almost certainly be ways to hack a QKD system (just like with any other security system). So, why bother with QKD? The advantage of QKD is based on one important selling point: contrary to most other forms of cryptography, it does not rely on assumptions about the computational power of a hacker. This can be an essential factor when someone is highly paranoid about their cryptography or when data has to remain confidential for an extremely long period of time.

As of 2024, pretty much every national security agency discourages the use of QKD simply because the available products are far from mature, and because PQC should be prioritised. It is unclear how successful QKD could be in the future – we will discuss this in-depth in the dedicated chapter on quantum networks.

We firmly warn that other security products with the word 'quantum' in the name do not necessarily offer protection against Shor's algorithm. In particular, quantum random number generators (QRNGs) are sometimes promoted as a saviour against the quantum threat, which is nonsense. These devices serve a completely different purpose: they compete with existing hardware to generate unpredictable secret keys, which find a use (for example) in hardware security modules in data centres.


PEOPLE WHO UNDERSTAND QUANTUM PHYSICS
YOU ARE HERE
PEOPLE WHO DON'T UNDERSTAND QUANTUM PHYSICS
PAPART

4. Optimisation and machine learning

This is the part where most enterprises get excited. Can we combine the success of artificial intelligence (AI) and machine learning with the radically new capabilities of quantum computers? Can we create a superpowered version of ChatGPT or DALL-E, or at least speed up the demanding training process?

In this section, we'll take a closer look at the known applications for quantum computers on 'non-quantum problems' other than cryptography. We focus specifically on the harder optimisation problems that currently take up large amounts of classical resources. Under the hood, all such applications are based on concrete mathematical problems such as binary optimisation, differential equations, classification, optimal planning, and so forth. For conciseness, we will use the word 'optimisation' as a catch-all term for all these problems, including things like machine learning and AI.

Unfortunately, the amount of value that 'quantum' can add to optimisation tasks is a highly disputed topic. The situation here is very subtle: many promising quantum algorithms exist, but, as we'll see, each comes with important caveats that might limit their practical usefulness. To start, we can classify the known algorithms into the following three categories.

Rigorous but slow algorithms

Many quantum optimisation algorithms have a well-proven *quantum speedup:* there is no dispute that these require *fewer computational steps* than any classical algorithm. For instance, a famous quantum algorithm invented by Lov Grover (with extensions by Dürr and Høyer) finds the maximum of a function in fewer steps than a conventional brute-force search. Similarly, quantum speedups were found for popular computational methods such as backtracking, gradient descent, linear programming, lasso, and for solving differential equations.

The key question is whether this also means that the quantum computer requires less *time*! All of the above optimisation algorithms offer a so-called *polynomial speedup* (in the case of Grover, this is sometimes further specified to be a *quadratic speedup*). As we will soon see, it is not entirely clear if these speedups are sufficient to compensate for the slowness of a realistic quantum computer – at least in the foreseeable future.

Heuristic algorithms

Some algorithms claim much larger speedups, but there is no undisputed evidence to back this up. Often, these algorithms are tested on small datasets using the limited quantum computers available today – which are still so tiny

that not much can be concluded about larger-scale problems. Nonetheless, these 'high risk, high reward' approaches typically make the bold claims that receive media attention. The most noteworthy variants are the following.

- Variational quantum circuits (VQC) are relatively short quantum programs that a classical computer can incrementally change. In jargon, these are quantum circuits that rely on a set of free parameters. The classical computer will run these programs many times, trying different parameters until the quantum program behaves as desired (for example, it might output efficient train schedules or accurately describe a complex molecule). The philosophy is that we squeeze as much as possible out of small quantum computers with short-lived qubits: the (fast) classical computer takes care of most of the computation, whereas the quantum computer runs just long enough to sprinkle some quantum magic into the solution.
  Although its usefulness is disputed, this algorithm is highly flexible, leading to quantum variants of classifiers, neural networks, and support vector machines. Variants of this algorithm may be found under different names, such as Quantum Approximate optimisation Algorithm (QAOA), Variational Quantum Eigensolver (VQE), and quantum neural networks.
- Quantum annealing solves a particular subclass of optimisation problems. Instead of using the conventional 'quantum gates', it uses the native physical forces that act on a set of qubits in a more analogue way.
  Annealing itself is a mature classical algorithm. The advantage of a 'quantum' approach is not immediately apparent, although there are claims that hard-to-find solutions are more easily reached thanks to 'quantum fluctuations' or 'tunnelling'. Quantum annealing was popularised by the Canadian company D-Wave, which builds dedicated hardware with up to 5000 qubits and offers a cloud service that handles relatively large optimisation problems.

Fast algorithms in search of a use case

Finally, there are algorithms with large speedups, for which we are still looking for applications with any scientific or economic relevance. These are classic cases of solutions in search of a problem. The most notable example is the quantum algorithm that solves systems of linear equations[3] with an exponential advantage. This problem is ubiquitous in engineering and optimisation, but, unfortunately, there are so many caveats that no convincing practical uses have been found.[4]

Recently, much attention has gone to the algorithm for topological data analysis (a method to assess certain global features of a dataset), which promises an exponential advantage under certain assumptions. Again, scientists are still searching for a convincing application.

Similarly, a quantum version of a classical machine learning algorithm called Support Vector Machines was found to have an exponential advantage over classical methods.[5] Unfortunately, this only works with a very specific dataset based on the factoring problem that Shor's algorithm is well known for. No rigorous advantage is known for more general datasets.

A fourth class: Quantum-inspired algorithms
Some impressive speedups that were recently found have been 'dequantised': these algorithms were found to work on classical computers too! There's a beautiful story behind this process, where Ewin Tang, an undergraduate student at the time, made one of the most unexpected algorithmic breakthroughs of the decade. A great report by Robert Davis can be found on *Medium*.[6]

What's left?
Unfortunately, a quantum optimisation algorithm with undisputed economic value does not yet exist; all of them come with serious caveats. This perspective is perhaps a bit disappointing, especially in a context where quantum computing is often presented as a disruptive innovation. Our main takeaway is that quantum optimisation (especially quantum machine learning!) is rather over-hyped.

That doesn't mean that there's no hope for quantum optimisation. Firstly, there are good reasons to believe that *new* algorithms and applications will be found. Secondly, the usefulness of the 'slower' quantum optimisation algorithms ultimately depends on the speed of a future quantum computer compared to the speed of a future classical computer. To better understand the differences in computational speeds, we will need to quantify the amount of 'quantum advantage' that different algorithms have.

3.2 How can we compare different types of speedups?

When looking at the applications of quantum computers, one should always keep in mind: are these actual improvements over our current state-of-the-art? Anyone can claim that their algorithm *can* solve a problem, but what we really care about is whether it solves it *faster*. Classical computers are already extremely fast, so quantum algorithms should offer a substantial speedup before they become competitive.

The fairest way to compare algorithms is by running them on actual hardware in a setting similar to how you would use the algorithm in practice.

In the future, we expect such **benchmarks** to be the main tool to compare quantum and classical approaches. However, mature quantum hardware is not available yet, so we resort to a more theoretical comparison tool: the asymptotic runtime of an algorithm.

What does asymptotic runtime mean?

An important figure of merit of an algorithm is its so-called **asymptotic complexity** or **asymptotic runtime**, which describes how much longer a computation takes as the problem becomes 'bigger' or more complicated. The term 'asymptotic' refers to the problem's size, which gets (asymptotically) larger, theoretically all the way to infinity.

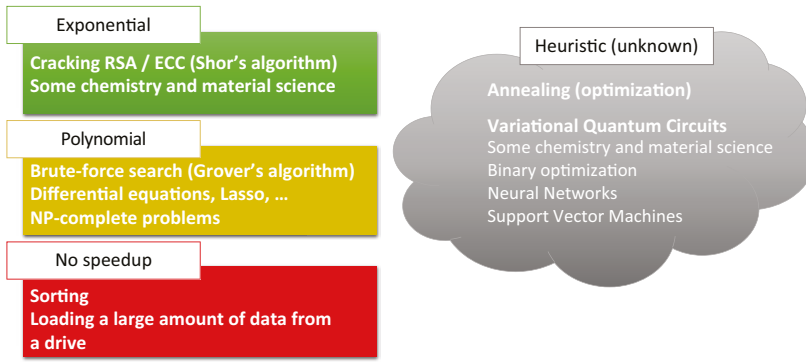
Size turns out to be a very relevant parameter. For example, computing 54 x 12 is much easier than 231,423 x 971,321, even though in technical jargon, they are *instances* of the same problem of multiplication, and we'd use the very same long multiplication algorithm that we learned in elementary school to tackle them. Similarly, creating a work schedule for a team of 5 is simpler than dealing with 10,000 employees. We typically use the letter n to denote the problem size. You can see n as the number of digits in a multiplication (like 2 or 6 above) or the number of employees involved in a schedule.

For some very hard problems, the time to solution takes the form of an exponential, something like $T \sim 2^n$ or $T \sim 10^n$, where T is the number of steps (or time) taken.[7] Exponential scaling is typically a bad thing, as such functions become incredibly large even for moderate values of n. For example, brute-force guessing a pin code of n digits takes roughly $T \sim 10^n$.

There are also problems for which the number of steps scales like a polynomial, such as $T \sim n^3$ or $T \sim n$. Polynomials grow much slower than exponentials, allowing use to solve large problems in a reasonable amount of time. Whenever a new algorithm can bring an exponential scaling down to a polynomial, we may call this an '**exponential speedup**'. Such speedups are a computer scientist's dream because they have a tremendous impact on practical runtimes. For example, quantum computers can factor large numbers in time roughly $T \sim n^3$ (thanks to Shor's algorithm[8]), whereas the best classical algorithm requires close to exponentially many steps.[9]

Often, we deal with 'merely' a **polynomial speedup**, which happens when we obtain a smaller polynomial (for example, going from $T \sim n^2$ towards $T \sim n$ or perhaps even a 'smaller' exponential function (like $T \sim 2^n$ towards $T \sim 2^{n/2}$). Reducing the exponent by a factor of two (like $n^2 \rightarrow n$) is also sometimes called a **quadratic speedup**, which is precisely what Grover's algorithm gives us.

Here is a rough overview of quantum speedups as we understand them today, categorised by their type of asymptotic speedup:



- The '**exponential**' box is the most interesting one, featuring applications where quantum computers seem to have a groundbreaking benefit over classical computers. It contains **Shor's algorithm** for factoring, explaining the towering advantage that quantum computers have in codebreaking. We also believe it contains some applications in **chemistry and material science**, especially those relating to dynamics (studying how molecules and materials change over time).
- The '**polynomial**' box is still interesting, but its applicability is unclear. Recall that a quantum computer would need much more time *per step* – and, moreover, it will have considerable overhead due to error correction. Does a polynomial reduction in the number of steps overcome this slowness? According to a recent paper,[10] small polynomial speedups (as achieved by **Grover's algorithm**) will not cut it, at least not in the foreseeable future.
- For some computations, a quantum computer offers **no speedup.** Examples include sorting a list or loading large amounts of data.
  If this were the complete story, then most people would agree that quantum computing is a bit disappointing. It would be a niche product for hackers and a tiny community of physicists and chemists who study quantum mechanics itself.
- Fortunately, there is yet another category: many of the most exciting claims come from the **heuristic** algorithms. This term is used when an algorithm might give a suboptimal solution (which could still be useful) or when we cannot rigorously quantify the runtime. Such algorithms are common on classical computers: neural networks fall in this category, and these caused a significant revolution in AI. Unfortunately, it is unclear what the impact of currently known heuristic quantum algorithms will be.

In summary, the potential for economic value varies greatly across quantum algorithms. The case of factoring has a clear and convincing speedup, but is only useful for codebreaking (where we hope that impact is limited thanks to the adoption of quantum-safe cryptography). In contrast, machine learning and optimisation do tackle a broad palette of relevant problems, but the speed advantage of a quantum computer remains uncertain in this field. The applications of chemistry and material science fall somewhere in the middle, with some relevant areas of applicability and concrete indications of a practical speed advantage.

3.3 Where is the killer application?

Is there hope that we'll find new quantum algorithms with a large commercial or societal value? For a quantum algorithm to be truly impactful, we require two properties:

1. [Useful] The algorithm solves a problem with real-world significance (for example, because organisations can work more efficiently or because it helps answer a scientific question).
2. [Better/faster] Using this particular algorithm is the most sensible* choice from a technical perspective,** even when compared to all other possible methods.

Throughout this book, we will use the term **quantum utility** when both properties are convincingly satisfied.

The precise definition can be a bit finicky, so before we start searching for utility, we need to get some technical details out of the way.

* What is 'sensible' (2) depends strongly on the context of the real-world problem (1). In most cases, we care about how fast a problem is solved, but one should also take into account the total cost of developing the software, the cost of leasing the hardware, the energy consumption, the probability of errors, and so forth. For example, a high-frequency trader might be happy with a 2% faster algorithm even if the costs are sky-high and there's a decent chance of failure, whereas a hospital could dismiss a 200x faster quantum approach if the costs don't outweigh the benefits. Indeed, what is 'sensible' is highly subjective. In practice, we can relax this requirement somewhat and focus primarily on speed, which is a sufficiently complex figure of merit on its own. Ideally, the quantum algorithm should enjoy an *exponential* speedup or at least a large polynomial speedup.

** We explicitly look for *technical* perspectives. Otherwise, one might also say that using a quantum algorithm is commercially the best option because it creates good PR or because it keeps the workforce excited. Then, perhaps, the first utility has already been reached! However, this is not the computational revolution that we're looking for, so we explicitly exclude such non-technical reasons in property (2). Similarly, we don't want to worry too much about legal issues ('it doesn't comply with regulations') because it feels somewhat artificial to dismiss a quantum algorithm for such reasons.

Supremacy, advantage, utility

Around 2019 and 2020, the terms **quantum supremacy** and **quantum advantage** were popularly used when quantum computers did, for the first time, beat the best supercomputers in terms of speed (property 2).[11,12] This involved an algorithm that was cherry-picked to perform well on a relatively small and noisy quantum computer whilst being as challenging as possible for a conventional supercomputer. Quantum advantage was mostly a man-on-the-moon-type scientific achievement, showcasing the rapid progress in hardware engineering and silencing the sceptics who still thought quantum computing wouldn't work. There was no attempt to have any practical value (1).

As a natural next step, the race is on to be the first to run something *useful* whilst leaving classical supercomputers in the dust. This led IBM to coin the term **quantum utility**,[13] which we adapted above. In the following years, we can expect the leading hardware and software manufacturers to maximise the amount of 'utility' that they could possibly squeeze out of medium-sized quantum computers, whilst competitors will use their best classical simulations to dispute these claims. The first battles have already been fought: in June 2023, IBM claimed to simulate certain material science models better than classically possible,[14] quickly followed by two scientific responses that showed how easy it was to simulate the same experiment on a laptop.[15][16]

It seems to us that such healthy competition is good for the field overall. It should lead to increasingly convincing and rigorous quantum utility, from which the end-users will eventually profit!

In parallel, there is a rapidly expanding number of press releases by startups and enterprises that claim to create business value by solving industrial problems on today's hardware, often without sharing many details. These approaches typically start with a relevant problem in mind and hence score well on usefulness (1). However, it is questionable whether

quantum algorithms were indeed the best option (2), and most reports we've seen hardly bother to show any argumentation in this direction. Such claims should only be taken seriously if a rigorous benchmark against state-of-the-art classical techniques is included.

Do known algorithms provide utility?

With the quantum utility criteria in mind, we can revisit the algorithms that were discussed before.

	(1) Useful	(2) Better than classical
Optimisation: Rigorous but slow algorithms	✓	?
Optimisation: Fast algorithms in search of a use cases	?	✓
Optimisation: Heuristic algorithms	✓	?
Simulation of molecules and materials	✓	?
Breaking RSA	✓	✓

Several 'rigorous but slow' algorithms, most notably Grover's algorithm, have an extensive range of industrial applicability. However, it seems that, in practice, other (classical) approaches solve such problems faster. The quadratic speedup will be insufficient in the near term, and it's unclear if it will be in the future.

Then, we have several exponential speedups, like the algorithm for topological data analysis, for which no practical uses have been found (despite many scientific and industrial efforts).

Most optimistic outlooks focus on heuristic algorithms, for which the speed advantage will become clear with maturing hardware. Nevertheless, we judge that no optimisation algorithms can tick both boxes for quantum utility yet.

Even for simulation of molecules and materials, it is not straightfoward to pinpoint precisely where we can find utility. Classical computers are already incredibly fast, and excellent classical algorithmic techniques have been developed. Scientist Garnet Chan even gives talks that are suggestively titled 'Is There Evidence of Exponential Quantum Advantage in Quantum Chemistry?'.[17] The case for quantum simulation is subtle, and we elaborate on this matter in the chapter on applications in chemistry and material science.

To the best of our knowledge, codebreaking (Shor's algorithm) is the only impactful algorithm that has little competition from classical methods. Hopefully, most critical cryptography will be updated well before a quantum computer arrives, making large-scale deployment of Shor's algorithm

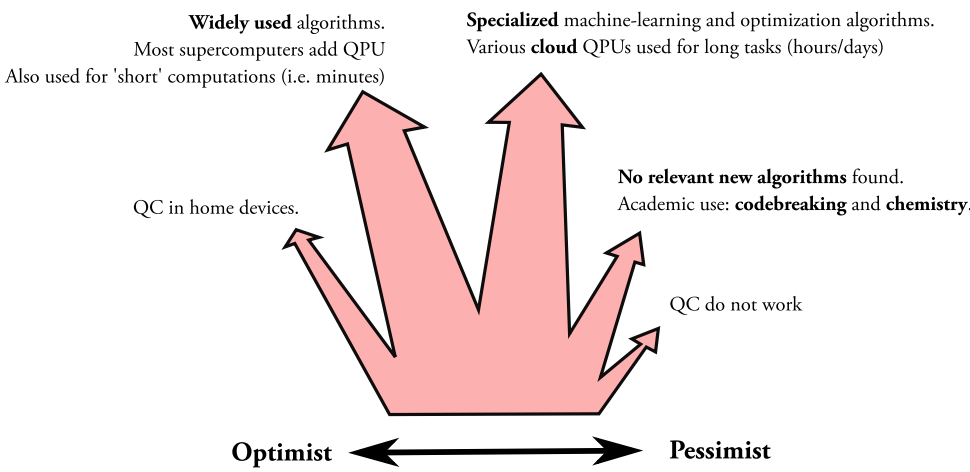
relatively uninteresting. Either way, the application of codebreaking is not quite the positive innovation that quantum enthusiasts are looking for.

Could the nature of quantum mechanics be such that exponential speedups are only found in codebreaking, chemistry, and a bunch of highly artificial toy problems, but nowhere else in the broad spectrum of practically relevant challenges? Most people would argue that such a scenario is unlikely. There are still high hopes that either some of the caveats with existing algorithms will be addressed or that new breakthrough algorithms will be discovered.

How optimistic you are about quantum computing should depend on (at least) the following questions:

– How impactful will heuristic and to-be-discovered algorithms be compared to classical algorithms? In other words, what is the algorithmic potential of quantum computing?
– How will quantum hardware develop relative to classical hardware?

Ultimately, the commercial success of quantum computers depends strongly on these questions. If we allow ourselves to do some more hypothetical dreaming, we imagine that the following future scenarios could be possible, on a spectrum of optimism versus pessimism:



Starting on the pessimistic side, if one believes that optimisation algorithms turn out to be lacklustre, then quantum computing might remain a niche for academics. However, depending on the utility of more widely applicable algorithms, one might predict that quantum computers will be installed in special-purpose computing facilities or, even more optimistically, that they

become increasingly common additions to data centres (much like GPUs today). Where would you place yourself in this figure?

3.4 Further reading

'The Potential Impact of Quantum Computers on Society'[18] (Ronald de Wolf, 2017) is an accessible overview of known algorithms, together with an assessment of how we can ensure a mostly positive net effect on society.

'Quantum Algorithms: An Overview'[19] (Ashley Montanaro, 2016) is a more technical overview paper that describes a selection of impactful algorithms in greater detail.

Professor Scott Aaronson warns us to *'Read The Fine Print'* of optimisation algorithms. [Appeared in *Nature* Physics, with paywall]

Professor Sanker Das Sarma warns of hype within the field of quantum optimisation and machine learning.

(Technical) A quantitative analysis of Grover's runtime compared to today's supercomputers.

(Scientific paper) Amazon researchers lay out a comprehensive list of end-to-end complexities of nearly every known quantum algorithm.

3.5 Notes

1. Jordan, S. (2024) *Quantum Algorithm Zoo*. https://quantumalgorithmzoo.org.
2. Gidney, C. and M. Ekerå (2021) 'How to Factor 2048 Bit RSA Integers in 8 Hours Using 20 Million Noisy Qubits', *Quantum*, 5, p. 433. https://doi.org/10.22331/q-2021-04-15-433.
3. Harrow, Aram W, Avinatan Hassidim, and Seth Lloyd (2008). 'Quantum Algorithm for Linear Systems of Equations'. *Physical Review Letters*, 103 (15) 150502. https://doi.org/10.1103/PhysRevLett.103.150502
4. Aaronson, S. (2015). Read the fine print. *Nature Physics*, *11*(4), 291–293. https://doi.org/10.1038/nphys3272. Open access: https://www.scottaaronson.com/papers/qml.pdf.
5. Liu, Y., Arunachalam, S., & Temme, K. (2021). A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, *17*(9), 1013–1017. https://doi.org/10.1038/s41567-021-01287-z
6. Qiskit. 'How Ewin Tang's Dequantized Algorithms Are Helping Quantum Algorithm Researchers'. *Qiskit* (blog), 15 March 2023. https://medium.com/qiskit/how-ewin-tangs-dequantized-algorithms-are-helping-quantum-algorithm-researchers-3821d3e29c65.
7. With the symbol ~ we mean 'roughly proportional to'. It allows us to write down an approximation of a function, making them easier to read, throwing away some details are not important here.
8. You may find even sources stating that Shor's algorithm takes a time proportional to $n^2 \log(n)$. Such scaling is theoretically possible but relies on asymptotic optimisations that are unlikely to be used in practice.
9. Technically, the best algorithms for factoring, like the general number field sieve, have a scaling behaviour that lies between polynomial and exponential. Hence, the speedup of Shor's algorithm is technically a bit less than 'exponential' – a more correct term would be 'superpolynomial'. Still, this book (and many other sources) continue to use the term 'exponential speedup' to emphasise the enormous scaling advantage over polynomial speedups.
10. Babbush, R. *et al.* (2021) 'Focus beyond Quadratic Speedups for Error-Corrected Quantum Advantage', *PRX Quantum*, 2(1), p. 010103. https://doi.org/10.1103/PRXQuantum.2.010103.
11. Zhong, H.-S. *et al.* (2020) 'Quantum computational advantage using photons', *Science*, 370(6523), pp. 1460–1463. https://doi.org/10.1126/science.abe8770.
12. Arute, F. *et al.* (2019) 'Quantum supremacy using a programmable superconducting processor', *Nature*, 574(7779), pp. 505–510. https://doi.org/10.1038/s41586-019-1666-5.
13. Technically, IBM has a subtly different interpretation. In a blog post (see https://www.ibm.com/quantum/blog/what-is-quantum-utlity), they define 'utility' as: *'Quantum computation that provides reliable, accurate solutions to problems that are beyond the reach of brute force classical computing methods, and which are otherwise only accessible to classical approximation methods'*. In other words: a quantum computer doesn't have to outperform any classical algorithm, it merely has to compete with the silly approach of brute-force search – which is almost never the best algorithm in practise. This definition seems heavily focused on claiming utility as soon as possible. Nevertheless, if we look at the big picture, we seem to have a similar notion of 'advantage for end-users' in mind, so I'm happy to adopt the term 'utility' anyway.
14. Kim, Y. *et al.* (2023) 'Evidence for the utility of quantum computing before fault tolerance', *Nature*, 618(7965), pp. 500–505. https://doi.org/10.1038/s41586-023-06096-3.
15. Begušić, T. and Chan, G.K.-L. (2023) 'Fast classical simulation of evidence for the utility of quantum computing before fault tolerance'. arXiv. https://doi.org/10.48550/arXiv.2306.16372.

16. Tindall, J. *et al.* (2024) 'Efficient Tensor Network Simulation of IBM's Eagle Kicked Ising Experiment', *PRX Quantum*, 5(1), p. 010308. https://doi.org/10.1103/PRXQuantum.5.010308.
17. Chan, G. (2022) 'Is There Evidence of Exponential Quantum Advantage in Quantum Chemistry?' *Berkeley Quantum Colloquium*, 12 April. https://www.youtube.com/watch?v=DZPH7ENcRLU.
18. De Wolf, R. (2017) 'The Potential Impact of Quantum Computers on Society', *Ethics and Information Technology*, 19(4), pp. 271–276. https://doi.org/10.1007/s10676-017-9439-z (open access: https://arxiv.org/abs/1712.05380).
19. Montanaro, A. (2016) 'Quantum Algorithms: An Overview', *npj Quantum Information*, 2(1), pp. 1–8. https://doi.org/10.1038/npjqi.2015.23.

4 Timelines: When can we expect a useful quantum computer?

At a glance

The earliest commercial quantum applications will need several million qubits, according to the most rigorous studies.

Assuming an exponential growth similar to Moore's law, we predict that the first applications could be within reach around 2035–2040.

The billion-dollar question in our field is:

When will quantum computers outperform conventional computers on relevant problems?

In the previous chapter, we defined the requirements more precisely and coined the term 'utility' for such an achievement.

Unfortunately, nobody can confidently answer this question today, and past predictions often proved inaccurate. Moreover, a relevant quantum computer won't just appear from one day to the next: there's a continuous evolution where these devices will become increasingly capable. In this chapter, we will show how we can make a rough prediction about future timelines and discuss what will happen on the path towards large-scale quantum computation.

Note

As an important disclaimer, this chapter is highly subjective. It's not hard to arrive at different conclusions simply by choosing other sources and making different assumptions. We did our utmost best to rely on the most up-to-date information, combining the views of the most widely accepted papers, and making assumptions that align with the view of most experts to present a balanced perspective.

4.1 What parameters are relevant?

Compared to currently available technology, we'd require a fundamental improvement to these specifications:

– **Number of qubits**

– **Accuracy of elementary operations (gates).** This means that quantum computers have the ability to perform long computations without making mistakes.

Quite a few other parameters matter, such as the **connectivity**, the **available set of gates**, the **speed of operations**, and so forth. In this chapter, we choose to simplify matters by assuming that all of these other parameters are not a bottleneck, allowing us to focus only on the number of qubits and gate accuracies.

The relevance of accuracy is often overlooked, perhaps because this hardly plays a role for classical computers anymore. The problem is as follows. A computation consists of many small, discrete steps called **quantum gates**. Unfortunately, even the most precisely engineered quantum computers are imperfect, and every gate has a slight chance of introducing an error. You can picture this intuitively as a qubit accidentally flipping from '0' to '1' or vice versa.[1] The probability that a gate introduces such an error on today's hardware is around 0.1% to 1%. Sometimes, the term 'accuracy' or 'fidelity' is used for the probability of not making an error, translating into numbers like 99.9%.

Now, a serious computation can easily use billions of gates. You can hopefully see the issue here: for long calculations on current hardware, the output is almost certainly garbled by errors. In fact, given a certain error gate fidelity, there is a ballpark maximum number of steps that can be reasonably performed. With a 1:1000 probability of error, we can do roughly a thousand steps, and if the error is one in a million, we can do approximately a million gates. To solve increasingly complex problems, we not only need to increase the number of qubits, but we *also* need to reduce the likelihood of errors.

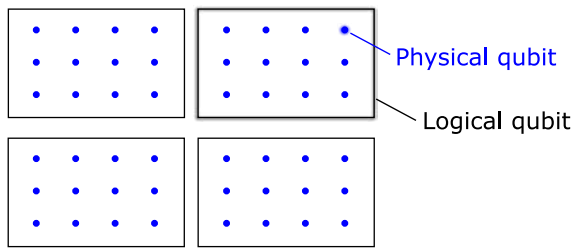
We should take a moment to appreciate the enormous challenge ahead of us. It took decades of engineering to minimise errors to about one in a thousand. Now, we should bring this rate down to one in *billions.* That's a huge gap that likely cannot be covered by hardware improvements alone – even a breakthrough that reduces errors by 100x wouldn't cut it.

Balancing qubits and accuracies

Fortunately, a technique exists that shrinks the probability of mistakes *by any desired amount*: **error correction.** It works roughly as follows. For every qubit that an algorithm requires, we don't just build a single hardware qubit, rather, we dedicate a large number of qubits, like a hundred or a thousand. We use the term **physical qubits** for the actual qubits present in

the hardware, whereas the virtual error-mitigated ones are named **logical qubits**.

For example, suppose we have a device with a million physical qubits. In that case, we might group a hundred of these to form a more error-resilient logical qubit, leaving a programmer with just 10,000 logical qubits to use. The image below shows a similar situation with a ratio of 1:12 between logical and physical qubits.



For error correction to work, we need to make several assumptions. For example, depending on the precise error correction protocol, gate fidelities need to be quite good to start with – numbers like 99.99% are often mentioned. This means that, as of 2024, the world's best devices would still need to improve gate fidelities by more or less a factor of 10. Moreover, qubits need to be routinely measured and reset, and large amounts of classical processing are needed to deduce precisely how to repair a given error. These are significant engineering challenges, but experts are optimistic that this can be achieved. We discuss many more details in a separate chapter on error correction.

For now, let's take for granted that we can somehow reach any desired accuracy (or any desired computation length) by simply adding sufficiently many physical qubits. Then, we can greatly simplify our analysis! For each application, we will forget about errors altogether and only count the number of physical qubits needed.

This leads to an interesting situation. To solve larger, more complex problems, we'll need more qubits for two reasons: to store more data *and* to reduce the probability of errors so that the computation can run longer.

Isn't the focus on just qubits a bit short-sighted? Doesn't this create a perverse incentive for manufacturers to focus only on qubit numbers, forgetting about all the other parameters? Well, we would certainly be worried that some companies can make headlines with unusable computers that happen to have a record qubit number. Fortunately, most manufacturers seem dedicated to making the most 'useful' computers, and customers will

surely judge their products by the capabilities of their logical qubits. We're obviously making a coarse simplification here, but making predictions about the future is hard enough as it is.

Back to the main question: When can we expect a large quantum computer? Now that we're only counting qubits, we can break our billion-dollar question into two parts:

- How many qubits are needed?
- In what year will that many qubits be available?

4.2 How many qubits are needed?

In the previous chapter, we discussed the three main applications of quantum computers: quantum simulation, breaking cryptography, and optimisation.

The most concrete numbers can be given for **Shor's algorithm** (breaking cryptography), where we have a very clear problem to tackle: obtain a private (secret) key from a widely used cryptosystem, like the RSA-2048 protocol. This is the perfect benchmark because there can be no discussion about whether the problem is solved: one either obtains the correct key or one doesn't. Moreover, we're quite convinced that even the best classical computers can't solve the problem (or else you shouldn't use internet banking or trust software updates).

A recent estimate finds that a plausible quantum computer would require roughly **20 million** 'reasonably good' physical qubits to factor a 2048-bit number. The whole computation would take about eight hours.[2] Such estimates require several assumptions on what a future quantum computer would look like. In this case, the authors assume qubits are built using superconducting circuits, which are laid out in a square grid. Error correction is assumed to be done using the so-called surface code, assuming the best-known methods for error correction in 2020. Note that future breakthroughs could reduce the required time and number of qubits even further.

For **chemistry and material simulation**, it's a lot harder to make such estimates because there is not just a single problem to tackle here: one typically uses computers to gradually improve our understanding of a complex structure or chemical reaction. This should be combined with theoretical reasoning and practical experiments. Moreover, classical computers can often perform the same computations that the quantum computer would make at the cost of making certain assumptions or simplifications. There's a fuzzy region between 'classically tractable' and 'quantum advantage'.

The most concrete task in quantum simulation is to compute the energy of certain molecular configurations. The benchmark is to obtain energies more accurately than done in conventional experiments; one canonically takes the 'chemical accuracy' of roughly 1 kcal/mol as the precision to beat. Then, we should focus on molecules where classical computers cannot already achieve such accuracies.

Note that the accuracy of a chemical energy should not be confused with the accuracy of a quantum gate, which is a whole different number.

A highly promising and well-studied benchmark problem is the simulation of the so-called FeMo cofactor of the nitrogenase enzyme, in short, FeMoco. This active site is relevant when bacteria produce Ammonia (NH_3), a compound that is of great relevance to a plant's root system. A better understanding of this process could help us reduce the ridiculously large carbon emissions now associated with the production of artificial fertiliser. We give more details in a separate chapter.

Simulating FeMoco is believed to require around **4 million qubits**[3] (and around 4 days of computation time). The hardware and error correction assumptions are similar to those of Shor's algorithm: the estimate is based on a square grid of superconducting qubits, using surface code to correct errors.

For a different enzyme, namely cytochrome P450, it has been estimated that around **5 million** qubits are needed[4] (again taking roughly four days of computation). Altogether, we conclude that a few million qubits (of sufficiently high quality) can make quantum computers relevant for R&D in chemistry.

Some tasks that are mainly of interest for scientific purposes, such as simulating models of quantum magnets, can be achieved with fewer resources. Under similar assumptions, simulating a 2D transverse field Ising model is estimated to take just under 1 million qubits.[5]

For many **optimisation problems**, it's practically impossible to give reasonable estimates. As we saw previously, a true killer algorithm for optimisation problems is not known yet. The algorithms that are presented as the most promising are often *heuristic*, meaning that it's hard to predict how accurate their results will be compared to conventional methods. We'll need to test them in rigorous benchmarking once larger quantum computers become available.

Our perspective starkly contrasts with some other sources claiming that quantum computers are already solving practical problems today. But don't be fooled: these articles state that quantum computers *can* indeed solve relatively simple problems but often fail to mention that there are *different* approaches by which classical computers can solve the same problems much, much faster.

Moreover, many of these algorithms involve optimisation problems that have a plethora of potential solutions, but the goal is to find the *optimal* solution (say, the one that incurs the least costs or gets you to your destination the fastest). The solution space is often so large that we don't even know if we hit this optimal solution, but we're okay with finding one that's *pretty close*. Several papers claim that a quantum computer already finds solutions *faster*, but in all cases, worrying sacrifices were made in the optimality of the solutions for more complex problems.

What about D-Wave's quantum annealer?
A particularly difficult case is the approach taken by D-Wave. This Canadian scale-up manufactures a quantum computer that is purpose-built to execute a specific optimisation algorithm called quantum annealing. With around 5000 qubits, it can handle reasonably large problems. The bare hardware alone doesn't seem to perform that well, but D-Wave cleverly combines it with classical high-performance computing in what they call a 'hybrid' solver. Comparisons and benchmarks of the hybrid solution report results ranging from 'much worse' to 'very competitive' relative to classical optimisation solutions. Because it is unclear to what extent the hybrid solver actually exploits quantum phenomena and little is known about D-Wave's future plans, we don't dare to make any future predictions about annealing.

We can summarise our conclusions in the table below.

Application	How well can we estimate qubit requirements?	Use case example	Physical qubits needed	Gate error assumed
Breaking cryptography	Good	Cracking RSA-2048	~ 20 million	~ 0.1%
Chemistry	Reasonable	Simulation of FeMoco	~ 4 million	~ 0.1%
		Simulation of P450	~ 5 million	~ 0.1 %
Optimisation / AI	Bad	?	?	

What about future improvements?
It seems almost inevitable that the above methodologies will improve. Unfortunately, it's impossible to estimate by how much. Will we reduce

the number of qubits required by a few per cent? Or by a factor of ten? By a factor of one thousand?

Some sources actually try to extrapolate the reduction in required qubits over time (like YouTube science educator Veritasium[6] and a report by McKinsey[7]), but this is such a wonky extrapolation over a handful of data points that we will not follow this strategy.

On the other hand, it would also be naive to stick to the numbers above without assuming some margin for improvements. In error correction techniques alone, there appears to be steady progress to improve the ratio between logical and physical qubits. Based on discussions with scientists, lowering the qubit requirements by a factor of 3 to 10 seems plausible. Hence, for optimistic readers, we can set another target at around 400,000 qubits. Interestingly, this number is similar to the qubit requirements for the simulation of models that are especially of scientific interest.

Application	How well can we estimate qubit requirements?	Use case example	Qubits needed?	Gate error assumed?
Chemistry (Optimistic)	Reasonable	Simulation of FeMoco (with 10x improved methods)	~ 400,000	~ 0.1 %
Science	Reasonable	2D Transverse field Ising model	~ 900,000	~ 0.1 %

Can noisy algorithms be good enough?

Current quantum computers have a limited number of qubits and are not yet capable of large-scale error correction; they are Noisy Intermediate-Scale Quantum (NISQ) devices. An important question is: can we already achieve any utility with such noisy devices before the era of large-scale error correction? That is one of the most disputed topics in our field, and therefore it deserves some attention.

A growing community of scientists, startups, and enterprises are searching for such near-term applications. If successful, this would massively increase the overall usefulness of quantum computers. Some experts seem optimistic that this is possible, but a larger and more authoritative group remains highly sceptical about NISQ's utility.

In the past decades, when NISQ devices with just a handful of qubits were just on the horizon, several consultants made ridiculous claims about how

such tiny machines would bring an exponential advantage over enormous supercomputers. Now that the field is coming of age, many are becoming more careful. To illustrate, when looking back at a 2021 report, consultancy firm BCG chivalrously admits:[8]

> Our assumptions for near-term value creation in the NISQ era, however, have proved optimistic and must be revised.

The most serious recent claim about NISQ utility comes from the IBM team in a paper titled 'Evidence for the utility of quantum computing before fault-tolerance,'[9] in which a quantum simulation of a specific physical system was performed using 127 noisy qubits. However, their arguments were quickly refuted by further studies that simulated IBM's impressive quantum experiment on a conventional laptop.[10]

Maryland-based professor Sankar Das Darma expresses the view of many academics in his opinion article 'Quantum computing has a hype problem'.[11] He stresses that 'the commercialisation potential [of NISQ] is far from clear', pointing out that claims of speedups in finance, machine learning and drug discovery have so far come with highly unsatisfying evidence.

That certainly doesn't mean that NISQ utility is ruled out. Most experts seem to keep an eye on the developments of NISQ applications but will agree that, as yet, no utility for NISQ machines has been found. To illustrate, an overview article about pharmaceutical applications[12] has a careful but suggestive message:

> Most NISQ algorithms [...] rely heavily on classical optimisation heuristics, and the actual run time is difficult to estimate. Furthermore, recent results suggest that in NISQ approaches, the number of measurements required to achieve a given error scales exponentially with the depth of the circuit. For these reasons, here we focus our discussion exclusively on fault-tolerant quantum computers.

Similarly, a recent overview[13] of quantum chemistry seems to remain agnostic with regard to NISQ advantage while pointing out that fault-tolerance has a higher chance of succeeding:

> [I]t is difficult to predict when or if algorithms on near-term noisy intermediate-scale quantum devices will outperform classical computers for useful tasks. But it is likely that, at some point, the achievement of large-scale quantum error correction will enable the deployment of a host of so-called error-corrected quantum algorithms.

In this book, we choose to follow the view of most scientists and stick to the well-understood use cases for early fault-tolerant quantum computers that we discussed previously. Nobody can rule out new breakthroughs that allow NISQ utility, but it seems unwise to count on these. A potential scientific leap could completely stir up our fragile prediction – but so would unexpected backlashes in hardware development or even unforeseen funding stops.

4.3 How long until we have million-qubit machines?

Now that we've set our target to roughly a million qubits, we'd like to estimate when such hardware will be available. We highlight the following sources:

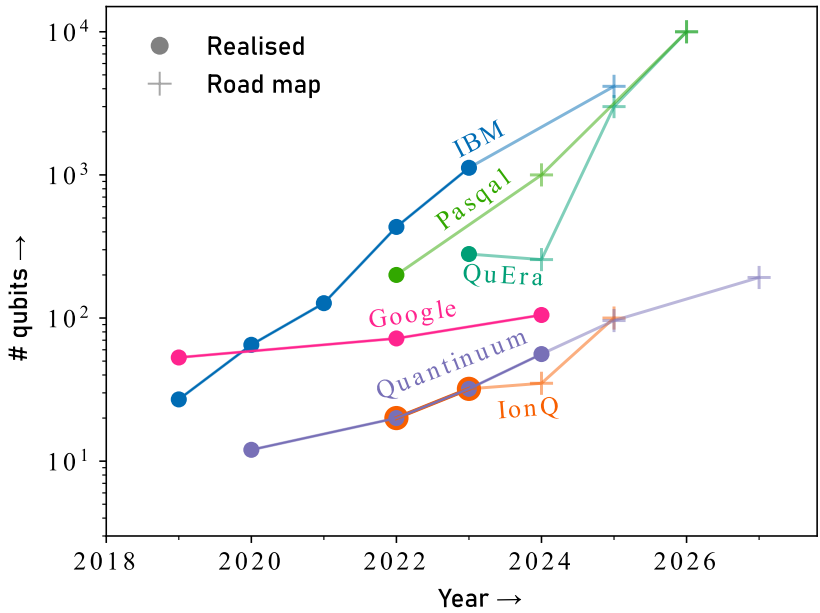
1. Road maps and claims of hardware manufacturers;
2. Surveys to experts;
3. Extrapolation of Moore's law.

What do manufacturers say?

Below, we see the qubit numbers that several manufacturers have already realised (solid disks) and what they will produce in the future according to their public road maps (opaque plusses). Note that the vertical axis is logarithmic, displaying a broad range from around 10 to 10,000 qubits. A lower number of qubits by no means indicates that these computers are worse. In fact, the machines with the lower numbers of qubits on this graph have an important edge in other parameters, such as gate accuracies and qubit connectivity.

Besides their road maps, companies sometimes make more daring claims in media interviews or at presentations at large events. Based on the application targets above, it should come as no surprise that manufacturers aim for around a million qubits as a 'moonshot' accomplishment. Back in 2020, IBM claimed that it would reach the 1 million qubit target by 2030.[14] Around the same time, journalists interpreted Google's pronouncements as meaning that it would do this even faster (around 2029[15]). The start-up PsiQuantum, which made waves thanks to record-high investments of over a billion dollars for their photonic quantum chips, went as far as claiming that it would have a million qubits by 2025.[16, 17]

It seems that these claims were too ambitious. In 2024, with only a year to go and no publicly presented product progression, PsiQuantum shifted its 1 million qubit road map to 2027.[18] IBM took an even more conservative

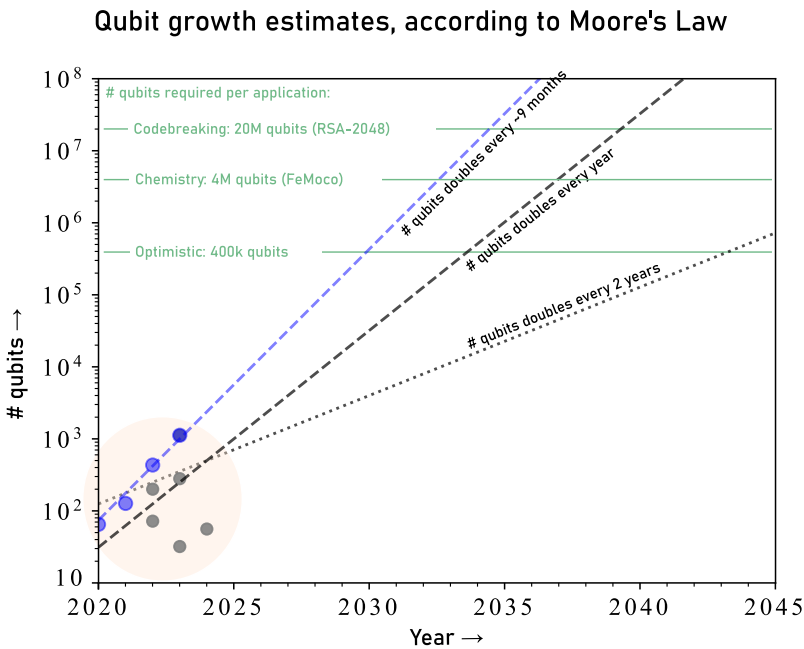


The largest number of qubits demonstrated by a selection of hardware manufacturers, shown for different years. Opaque plusses indicate manufacturers' road maps. Data taken from publicly available sources up until August 2024.

step, and it's now claiming that it will have just 100,000 qubits in 2033[19] (although this machine should meet the error correction capabilities that we assumed in the previous sections). Although this delay sounds disappointing, hardware manufacturers are still making impressive progress, not least because the number of available qubits grows faster than one would predict according to Moore's law for classical chips!

Trapped-ion machines tend to have fewer qubits but higher gate accuracies. Perhaps this is why IonQ displays its road map in a different format: they aim to achieve 1024 so-called algorithmic qubits by 2028.[20] This means that IonQ will have *at least* this number of qubits, but it also guarantees sufficient gate accuracy to run reasonably long circuits. It's unclear whether error correction will be used for this. Competitor Quantinuum recently announced a more concrete road map,[21] predicting around 100 logical qubits in 2027. These should bring the effective gate errors down by roughly a factor of 10. Looking ahead to 2029, Quantinuum projects thousands of physical qubits that form hundreds of logical qubits. This might not be enough to run the algorithms discussed earlier, but it's not too far off either.

What does Moore's law say?



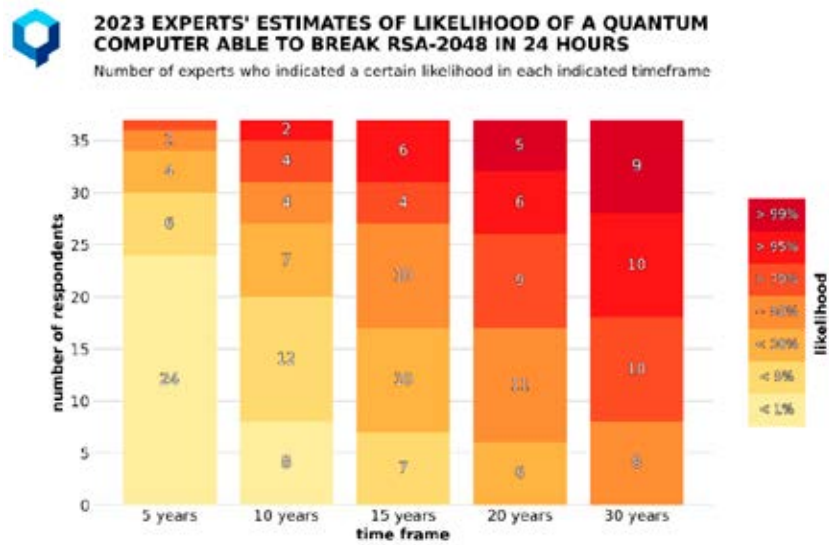
One could assume that quantum computers will 'grow' at a similar rate as classical computers. Moore's law states that the number of transistors in a dense integrated circuit grows exponentially: the number doubles roughly every two years. This has been a surprisingly accurate predictor for the development of classical IT. If we apply Moore's law to quantum, then boosting qubit numbers from around a thousand to one million would take around twenty years – predicting that the one million qubit mark won't be passed until 2044. Clearly, most hardware manufacturers are more optimistic. If we assume the number of qubits doubles each year, then one would predict that one million qubits will be available in ten years. While doubling a quantum computer's size each year is already a daunting challenge, companies like IBM, Pasqal, and QuEra set the bar even higher for themselves, hoping to double every 7–9 months.

What do experts say?

The Global Risk Institute conducts annual surveys asking experts to state the *likelihood* that quantum computers will pose a significant threat to public key cryptography 5 years from now. Similarly, respondents also estimate the likeliness 10, 15, 20, and 30 years away.

This essentially boils down to the question*: when will a quantum computer run Shor's algorithm to crack RSA-2048?* We previously saw that around 20 million qubits would be needed for this (although experts may take into account that this number can still be lowered).

We consider this an important source because many important authorities in the field (like professors and corporate leaders) take part in this study. The results from December 2023,[22] gathered from 37 participants, are displayed below.



Results of the December 2023 expert survey by Global Risk Institute. Figure credits: M. Mosca, M Piani, www.globalriskinstitute.org.

How to read this graph?

Let's look at the column labelled '5 years'. A total of 24 correspondents indicate that there is less than 1% probability that quantum computers pose a security threat in the next five years. A single person is quite pessimistic and assigns a >70% chance that this will happen. On average, experts say that there's a fairly small likelihood that quantum computers will pose a threat to cryptography in the next five years.

Further to the right, the ratios shift. Looking at 20 years from now, the majority of experts believe that quantum computers pose a serious threat, with over half of them assigning a likelihood of 70% or more.

It appears that the majority of experts believe that the tipping point is between 10–20 years from now. Somewhere between 15 and 20 years away, there's a point where the median participant assigned roughly 50% chance to see a quantum computer capable of breaking cryptographic codes. However, we should take into account a significant uncertainty: even experts make wildly varying estimates, so there's no obvious conclusion from this data.

These experts are almost certainly aware of hardware manufacturer's road maps, as we shall see below.

4.4 Putting it all together

The graph on the next page sums up our earlier findings.

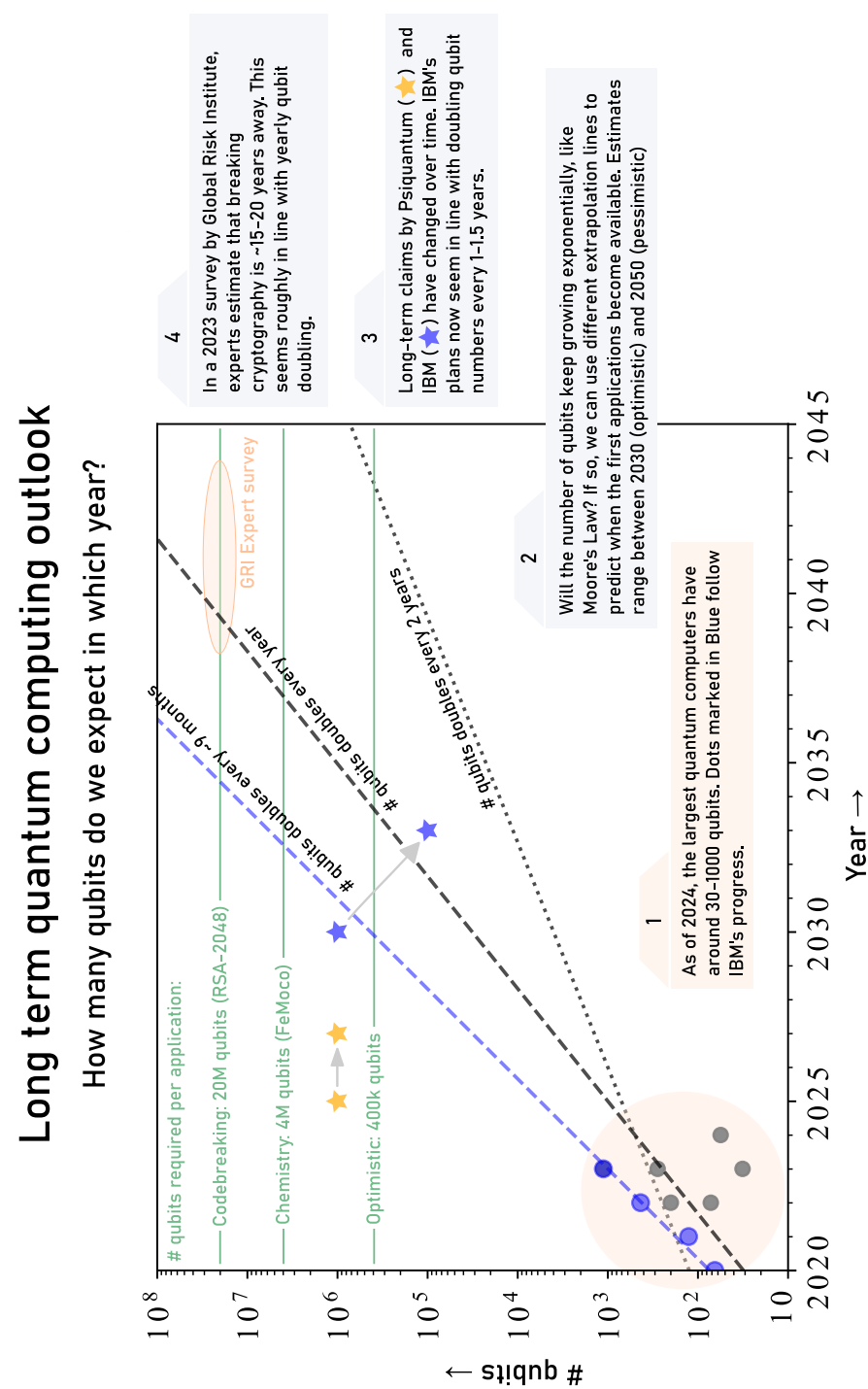
Assuming that qubit numbers will grow exponentially (and that all other parameters will keep up accordingly), we can consider several scenarios. A pessimistic scenario would be that the number of qubits 'merely' follows the classical version of Moore's law, and qubit numbers double only once every two years (dotted line). Then, we would have to wait until well past 2040 to reach 100,000 qubits. An even worse scenario would be if we cannot achieve exponential growth, which would stretch the timelines even further.

An extremely optimistic outlook would follow the blue dashed line (which extrapolates the progress by IBM, doubling their qubits every ~9 months). If one also believes in practical applications with much less than a million qubits, then these could be available by 2030.

An intermediate perspective is to assume that the number of qubits doubles annually. Interestingly, this seems to approximately align with IBM's latest claims and the typical expert opinion. Depending on the application, it would mean that quantum chemistry simulation and codebreaking can be within reach between ~2033 and 2040.

To conclude, our estimates strongly depend on the assumptions that you're willing to accept (who would've thought!). Do you believe that improving algorithms and error correction techniques will allow for applications with much less than a million qubits? How quickly do you believe that the hardware will improve? If you were to force me to make a prediction, I'd say the first applications will arise around 2035, with the understanding that there's a considerable margin for error.

As a final remark, a full utility-scale quantum computer requires much more than just some number of qubits. To reach the first useful applications, we likely require simultaneous progress in algorithmics, software, gate accuracies, error correction techniques, fridges, lasers, and many other


Long term quantum computing outlook
How many qubits do we expect in which year?
qubits required per application:
Codebreaking: 20M qubits (RSA-2048)
Chemistry: 4M qubits (FeMoco)
Optimistic: 400k qubits
GRI Expert survey
qubits doubles every ~9 months
qubits doubles every year
qubits doubles every 2 years
qubits
Year
1
As of 2024, the largest quantum computers have around 30–1000 qubits. Dots marked in Blue follow IBM's progress.
2
Will the number of qubits keep growing exponentially, like Moore's Law? If so, we can use different extrapolation lines to predict when the first applications become available. Estimates range between 2030 (optimistic) and 2050 (pessimistic)
3
Long-term claims by Psiquantum (★) and IBM (★) have changed over time. IBM's plans now seem in line with doubling qubit numbers every 1-1.5 years.
4
In a 2023 survey by Global Risk Institute, experts estimate that breaking cryptography is ~15-20 years away. This seems roughly in line with yearly qubit doubling.

important subfields of quantum computing. Hopefully, all these disciplines will find the required breakthroughs that will sustain the exponential growth of quantum computing hardware.

4.5 Further reading

Scientist Samuel Jaques (Waterloo) makes insightful graphs that combine the number of qubits and the error rates, and puts them in the perspective of applications requirements.

4.6 Notes

1. Technically, quantum gates are continuous operations, so numbers like fidelity are defined slightly differently. Still, the picture of discrete bit flips is not too far off and will lead to the same conclusions, so we prefer this more accessible explanation.
2. Gidney, C. and Ekerå, M. (2021) 'How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits', *Quantum*, 5, p. 433. https://doi.org/10.22331/q-2021-04-15-433.
3. Lee, J. *et al.* (2021) 'Even More Efficient Quantum Computations of Chemistry Through Tensor Hypercontraction', *PRX Quantum*, 2(3), p. 030305. https://doi.org/10.1103/PRXQuantum.2.030305.
4. Goings, J.J. *et al.* (2022) 'Reliably assessing the electronic structure of cytochrome P450 on today's classical computers and tomorrow's quantum computers', *Proceedings of the National Academy of Sciences*, 119(38), p. e2203533119. https://doi.org/10.1073/pnas.2203533119.
5. Beverland, M.E. *et al.* (2022) 'Assessing Requirements to Scale to Practical Quantum Advantage'. *arXiv*. https://doi.org/10.48550/arXiv.2211.07629.
6. See https://www.youtube.com/watch?v=-UrdExQWocs&t=1024s, starting at 17:04.
7. McKinsey Digital (2024) 'Quantum Technology Monitor'. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/steady-progress-in-approaching-the-quantum-advantage.
8. Bobier, J.-F. *et al.* (2024) *The Long-Term Forecast for Quantum Computing Still Looks Bright*, *BCG Global*. https://www.bcg.com/publications/2024/long-term-forecast-for-quantum-computing-still-looks-bright.
9. Kim, Y. *et al.* (2023) 'Evidence for the utility of quantum computing before fault tolerance', *Nature*, 618(7965), pp. 500–505. https://doi.org/10.1038/s41586-023-06096-3.
10. Begušić, T. and Chan, G.K.-L. (2023) 'Fast classical simulation of evidence for the utility of quantum computing before fault tolerance'. *arXiv*. https://doi.org/10.48550/arXiv.2306.16372.
11. Das Sarma, S. (2022) 'Quantum computing has a hype problem'. https://www.technologyreview.com/2022/03/28/1048355/quantum-computing-has-a-hype-problem/.

12. Santagati, R. *et al.* (2024) 'Drug design on quantum computers', *Nature Physics*, 20(4), pp. 549–557. https://doi.org/10.1038/s41567-024-02411-5.
13. Cao, Y. *et al.* (2019) 'Quantum Chemistry in the Age of Quantum Computing', *Chemical Reviews*, 119(19), pp. 10856–10915. https://doi.org/10.1021/acs.chemrev.8b00803.
14. Hackett, R. (2020) *IBM plans a huge leap in superfast quantum computing by 2023*, *Fortune*. https://fortune.com/2020/09/15/ibm-quantum-computer-1-million-qubits-by-2030/.
15. Finke, D. (2020) 'Google Goal: Build an Error Corrected Computer with 1 Million Physical Qubits by the End of the Decade', *Quantum Computing Report*, 5 September. https://quantumcomputingreport.com/google-goal-error-corrected-computer-with-1-million-physical-qubits-by-the-end-of-the-decade/.
16. Wang, B. (2020) 'PsiQuantum Targets Million Silicon Photonic Qubits by 2025', 23 April. https://www.nextbigfuture.com/2020/04/psiquantum-targets-million-silicon-photonic-qubits-by-2025.html.
17. *What will million-qubit computers look like in a few years?* (2022) *ICV TAnK-icv*. https://www.icvtank.com/newsinfo/629365.html.
18. Finke, D. (2024) 'PsiQuantum Receives $940 Million AUD ($620M USD) to Install a 1 Million Qubit Machine in Australia by 2027', *Quantum Computing Report*, 30 April. https://quantumcomputingreport.com/psiquantum-receives-940-million-aud-620m-usd-to-install-a-1-million-qubit-machine-in-australia-by-2027/.
19. Baker, B. (2023) *IBM Details Road to 100,000 Qubits by 2033*, *IoT World Today*. https://www.iotworldtoday.com/industry/ibm-details-road-to-100-000-qubits-by-2033.
20. Chapman, P. (2020) 'Scaling IonQ's Quantum Computers: The Roadmap', *IonQ*, 9 December. https://ionq.com/posts/december-09-2020-scaling-quantum-computer-roadmap.
21. *Quantinuum accelerates the path to Universal Fully Fault-Tolerant Quantum Computing* (2024) *Quantinuum*. https://www.quantinuum.com/blog/quantinuum-accelerates-the-path-to-universal-fault-tolerant-quantum-computing-supports-microsofts-ai-and-quantum-powered-compute-platform-and-the-path-to-a-quantum-supercomputer.
22. Mosca, M. and Piani, M. (2023) *Quantum Threat Timeline Report 2023*. https://globalriskinstitute.org/publication/2023-quantum-threat-timeline-report/.

5 Four myths about quantum computing

This chapter relies on a bit of quantum physics jargon. See the chapter 'An introduction to the quantum world' for a quick introduction.

5.1 Myth 1: Quantum computers find all solutions at once

This myth is likely the most technical, and builds on a misinterpretation of the concept of superposition. A single qubit can be in two states at the same time (0 and 1), two qubits can represent four states (00, 01, 10, 11), and three qubits are potentially in eight unique configurations simultaneously. As we increase the number of qubits, this number of coexisting states scales exponentially!

This means that a mere 1000 qubits can effectively 'store' 2^{1000} unique values, all at the same time. That's an incomprehensibly large number, much more than there are atoms in the visible universe. Even the fastest computers in the world couldn't loop through all these states in a lifetime. Each of these states can be interpreted like a file on a computer, be it an Excel spreadsheet, a web page, a CAD drawing, or whatever kind of data we choose to work with.

A smart computer scientist can also devise a way to make 1000 bits represent 'solutions' to a problem. For example, imagine that we want to find an optimal aeroplane wing that generates incredible lift while requiring as few materials as possible. Using quantum superposition, we might represent 2^{1000} such wings simultaneously.

We picked the example of aeroplane wings because simulating their aerodynamic properties requires a pretty hefty computation. Let's assume that we have written such a computer program that accurately simulates any wing. Let's call that program *f*. It will output 1 if the wing works well (according to whatever metric), and 0 otherwise. Surely, the program takes a very large number of computation steps, which we'll call T. The program will need some input, denoted by x, which is a 1000-bit description of all the relevant properties of a hypothetical aeroplane wing. In other words, the computer program computes $f(x) = 1$ if x is a fantastic wing, and $f(x) = 0$ if it's rubbish.

Now, a quantum computer should be able to execute any classical function, right? We should be able to run *f* on a quantum computer, but now we have the unique feature that the 1000-qubit input can represent a humongous number of potential aeroplane wings at the same time. By doing a mere T computational steps, we can check the properties of 2^{1000} solutions!

If this actually worked, quantum computers would have an astonishing power. They could straightforwardly find mathematical proofs that humans haven't been able to solve in centuries, simply by trying all possible proofs in parallel. They would rapidly produce the perfect train and bus schedules, discover new drugs, and straightforwardly hack encryption systems. They would solve problems in the complexity class NP, which is widely believed to be impossible with machines in our universe, owing to the famous P ≠ NP conjecture.

So, what's the catch? For those who read the introduction to quantum physics, we shouldn't forget about the postulate of quantum measurement. The output of the computation would be a *superposition* over 2^{1000} outcomes. If we want to learn anything about this output, we'd perform a quantum measurement that collapses this superposition. Instead of looking at 2^{1000} different solutions simultaneously, we only get to see one outcome – corresponding to the performance of just a random aeroplane wing. In this case, there is no advantage compared to a classical computer because we could've just as well picked a random wing at first, and then spent the same T steps on a (much faster) classical machine.

Although this 'quantum parallelism' is too good to be true, quantum computers can use the above idea to a lesser extent. Using Grover's algorithm, we can find desirable solutions (the x for which $f(x) = 1$) in roughly the *square root* of the number of values that x can take. In the above example, the number of required steps is reduced to $\sqrt{2^{1000}}\ T = 2^{500}\ T$. This is an incredible reduction, but we're still looking at a number of steps larger than the number of atoms in the universe – finding solutions with this brute-force method remains far from efficient.

5.2 Myth 2: Qubits can store much more data than the same number of classical bits

This myth is similar to the previous one: can't N qubits represent 2^N different numbers at the same time? Or aren't they perhaps even more powerful, because for each of the 2^N different numbers, there is a complex *amplitude*, which can have as many decimal digits as we like?

Again, by the rules of quantum measurement, this is too good to be true. It's impossible to store much information in a qubit because it collapses to a classical 0 or 1 when we measure it. The problem is really in *retrieving* the information, as we have very limited capabilities to do so. For the same reason, when sending a classical message over a long distance, there's little value in using qubits as information carriers.

As a side note, there is a fascinating related protocol called 'superdense coding', which you may want to look up out of theoretical interest. Also, when your data itself represents something quantum (for example, the state of electrons in a molecule), then storing this data in qubits does have a potentially huge advantage.

5.3 Myth 3: Entanglement allows you to send information faster than light or to influence objects at a distance

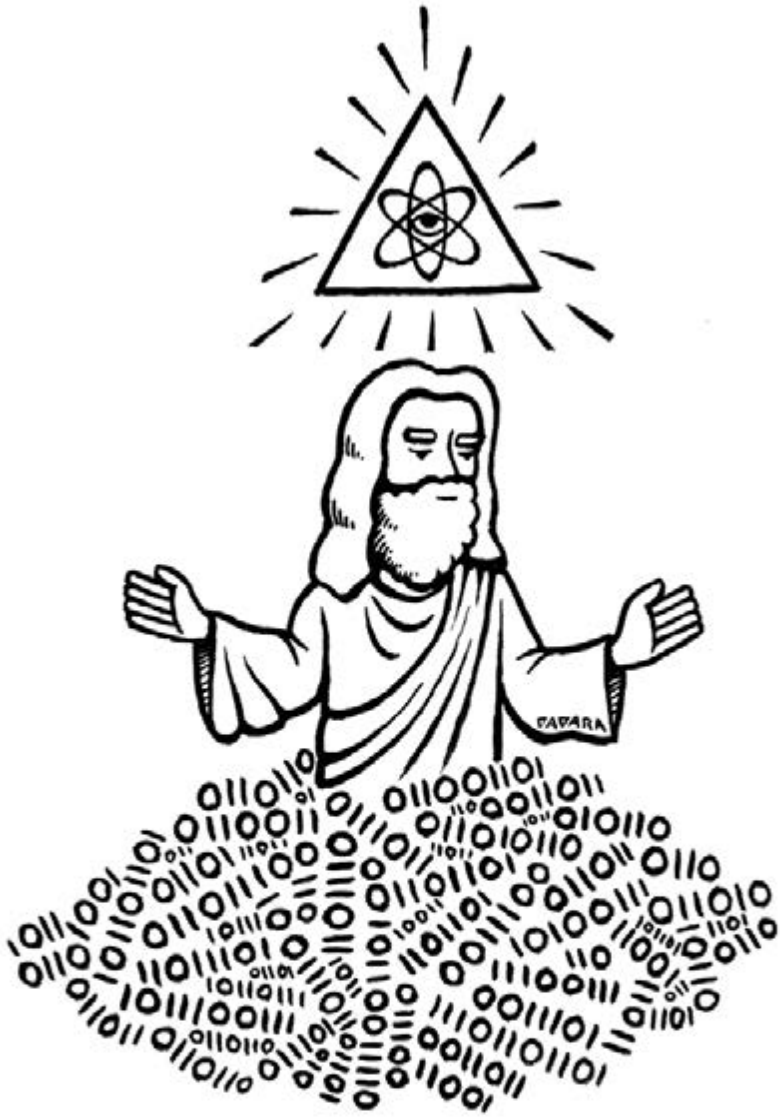
Entanglement is an incredibly confusing phenomenon. In particular, our most common interpretation of quantum mechanics states that whenever we measure one qubit, the state of another distant qubit can drastically change. Whilst this picture is helpful for physicists when performing computations, it tricks our intuition.

Imagine that, in the faraway future, we want to protect our solar system against an alien invasion. We have installed sentinels on distant outposts, which should alert Earth to any approaching dangers. Alice is one of these noble guards stationed on a remote asteroid in the icy Kuiper Belt. She brought with her a qubit labelled A, which is entangled with another qubit B that's safely kept on Earth by her colleague Bob. Whilst it takes light signals around five hours to travel between them, isn't there a way for Alice to alarm Bob any faster, possibly by doing some special operations on her qubit? Perhaps she could even give some clues about the type of looming threat?

Unfortunately, Alice cannot remotely change any *measurable quantity* of Bob's qubit. Bob's measurements will always have the same outcome probabilities, no matter what Alice does to her qubit. Using more qubits or employing different quantum objects won't help either. Fundamentally, there is no way to signal any information faster than the speed of light.

There is a subtle difference between 'changing measurable quantities' and 'knowing something' about the state of a particle. To illustrate, assume that we start with a particular entangled state: measuring qubits A and B will result either in both qubits being '0' or both qubits being '1', let's say with 50% probability each. Measuring something like A= '0' and B= '1' is impossible.

When Alice measures her qubit and reads the outcome '0', she immediately knows the outcome of a future measurement made by Bob: she knows this will be '0' with 100% probability. However, this knowledge is not accessible to Bob. He doesn't even know whether Alice measured or not! Even if they agreed in advance that Alice would measure at a set time, Bob doesn't know her outcome. From his perspective, '0' or '1' are still equally likely.



WITH QUANTUMCOMPUTERS
GOD WOULD HAVE CREATED
THE WORLD IN ONE DAY

Something interesting happens when Alice sends a message to Bob to inform him that her measurement returned '0'. With this updated knowledge, Bob suddenly knows precisely what the state of his qubit is: it must have collapsed to '0', and he can perfectly predict the outcome of a subsequent measurement. In a way, this did indeed change the state of the qubit from Bob's perspective, but it was only possible after some (classical) communication took place between Alice to Bob, a process that is limited by the speed of light.

What is quantum entanglement good for, then? Some potential applications include:

- Creating certifiably secure encryption keys at remote locations;
- Creating certifiable randomness;
- Forming connections between separate quantum computers, allowing them to send quantum data to each other using teleportation. For this to work, devices also need to transfer some classical data, so qubit transmission is never faster than the speed of light. Teleportation is an intriguing method for scaling up quantum computers when a limited number of qubits can fit on a single chip or within a single fridge.

5.4 Myth 4: Quantum computers are always ten years away.

This statement is a playful reference to the situation around nuclear fusion, where predictions of its realisation being just thirty years in the future have repeatedly been postponed. Scientists have been working on fusion for decades, but it's still far from a mature energy source.

Similarly, we've heard several overly optimistic claims about quantum computers being made in the past ten years, often claiming that quantum computers are somewhere between three to ten years away. An article in *TechCrunch*[1] boldly paraphrases Dario Gil (IBM) and Chad Rigetti (founder of Rigetti Computing) saying that 'the moment that a quantum computer will be able to perform operations better than a classical computer is only three years away'; this article was published back in 2018. For reference, the 127-qubit Eagle chip was announced by IBM at the end of 2021, but several years later, it's still primarily used for testing and education. In 2019, consulting firm Gartner published '*The CIO's Guide to Quantum Computing*', which indicates that 100–200 qubits are sufficient for 'key potential applications' in chemistry. They also predicted that 'by 2023, 20% of organisations will be budgeting for quantum computing projects'. Clearly, these predictions were overly optimistic.

Similarly, Microsoft made claims in 2018 that their cloud platform Azure would feature quantum computing in five years,[2] which is technically true. However, they have repeatedly hinted at doing this with fault-tolerant topological qubits, which currently remain elusive. Startup PsiQuantum famously claimed that it would have a million photonic qubits by 2025,[3] and consultants at BCG advised that quantum computers would also 'generate business value' in that same year.[4] Again, it remains to be seen if this holds true.

Fortunately, if you're reading this book, you will have noticed that not all experts share the same vision. Most scientists have warned for a long time that quantum computing is a long-term effort.

Nevertheless, the thesis that 'quantum computing is always X years away' is hard to defend, thanks to convincing evidence that we are steadily progressing towards a clear goal. Every year, quantum hardware sees major improvements in the number of qubits, their stability, and the level of control that is demonstrated. Most experts even expect an exponential scaling of the number of qubits, similar to Moore's law, and manufacturers have clear roadmaps that underline these predictions. Moreover, theorists have set clear targets for when the hardware is good enough – and we'd sooner see the requirements drop with new breakthroughs than become more stringent. Building a quantum computer is a marathon, not a sprint. It's impossible to predict when 'quantum' will become commercially relevant, but the rapid rate of progress is undeniable.

5.5 Further reading

(YouTube) Veritasium explains Entanglement

(YouTube, technical!) Minute Physics explains Teleportation

Chris Ferrie debunks more myths in his free book *What You Shouldn't Know about Quantum Computers*

Scott Aaronson shares a transcript of a public talk, explaining why he is optimistic about the steady progress towards large-scale quantum computers.

5.6 Notes

1. Shieber, J. (2018) *The reality of quantum computing could be just three years away*, *TechCrunch*. https://techcrunch.com/2018/09/07/the-reality-of-quantum-computing-could-be-just-three-years-away/.
2. Saran, C. (2018) *Microsoft predicts five-year wait for quantum computing in Azure*, *ComputerWeekly.com*. https://www.computerweekly.com/news/252440763/Microsoft-predicts-five-year-wait-for-quantum-computing-in-Azure.
3. Cookson, C. (2021) 'PsiQuantum Expects Commercial Quantum Computer by 2025', 13 March. https://www.ft.com/content/a5af3039-abbf-4b25-92e2-c40e5957c8cd.
4. Matt Langione *et al.* (2023) *Quantum Computing Is Becoming Business Ready*, *BCG Global*. https://www.bcg.com/publications/2023/enterprise-grade-quantum-computing-almost-ready.

Part 2

More about the applications

6 Applications in chemistry and material science

Perhaps the most credible application of quantum computers is to study quantum physics itself. This deepens our understanding of microscopic systems like molecules, atoms, or even sub-atomic particles, ultimately leading to the discovery of new drugs, materials, and chemical production methods. At first sight, there seems to be a significant advantage compared to conventional computers, which struggle to store the complex quantum state of systems with many particles. As far back as 1981, physicist Richard Feynman ended a conference talk with a famous quote, hinting at the opportunities of quantum computing:[1]

> I'm not happy with all the analyses that go with just the classical theory, because nature isn't classical, dammit, and if you want to make a simulation of nature, you'd better make it quantum mechanical.

Since then, scientists have become increasingly adept at accurately controlling quantum systems. Today, universities boast a wide spectrum of analogue quantum experiments that help us understand nature under exotic circumstances. We're now lining up our tools to take these simulations to the next level: studying nature with digital quantum machines.

In this chapter, we will assess how quantum computers can impact the fields of chemistry and material science. That makes this chapter more technical, and we'll assume some (very) basic background in chemistry and physics. We discuss the most relevant algorithms, evaluate claims about quantum computing's benefits in the fight against climate change, and analyse why the nitrogenase enzyme receives such widespread attention.

6.1 What problems in chemistry and material science will we solve?

The computational problems that chemists care about typically come in two flavours: static and dynamic problems. The most studied problem is the static variant, where the goal is to find the arrangement(s) of particles with the lowest possible energy. We call such an arrangement the *ground state*. These states are relevant because we usually find systems in (or close to) their

lowest energy states in nature. In the context of molecules, the atomic nuclei are relatively heavy, while the lightweight electrons move much faster and are more prone to be entangled or in a quantum superposition. Therefore, chemists tend to make approximations that allow them to focus primarily on the positions and spins of the electrons: the *electronic structure problem.*

The other main problem is about dynamics: given some initial configuration of particles, how do they reconfigure themselves after a certain amount of time? This is often referred to as a system's (*time*) *evolution*. Both problems are informally referred to as **quantum simulation**.

We often receive the question of why it's so hard to simulate quantum mechanics on a *classical* computer. Intuitively, this hardness arises when we deal with many particles that exhibit large amounts of superposition and entanglement, such that the location of one particle is heavily dependent on the (undecided) position of many other particles. We call such states *strongly correlated*. Classical computers struggle because they need to keep track of all the possible locations that particle A can be, but also all the locations of particle B, and the same for particle C, etc. As the number of particles grows, the number of possible configurations of these particles increases *exponentially*. This means that the number of relevant *amplitudes* (see the chapter on quantum physics) that a classical computer needs to process grows very quickly. Even with a mere one hundred particles, brute-force simulation is far beyond the capabilities of the world's best supercomputers.

It is a common misconception that quantum computers straightforwardly offer an exponential advantage compared to classical computers for all chemistry problems. An influential recent paper reports[2]:

> [W]e conclude that evidence for such an exponential advantage across chemical space has yet to be found. While quantum computers may still prove useful for ground-state quantum chemistry through polynomial speedups, it may be prudent to assume exponential speedups are not generically available for this problem.

Note that this comment is specifically about finding *ground states,* which, arguably, remains the most relevant problem in chemistry. There is still ample evidence that quantum computers offer an exponential speedup for *time evolutions.*

There is more bad news for quantum computers. Over the years, computational chemists have found brilliant approximations, hacks, and optimisations to work around the classical computer's bottlenecks, raising a high bar before a quantum computer can meaningfully compete. For nearly

every problem in chemistry, there appears to be a clever trick to solve it somewhat efficiently on a classical machine.

For a killer application, we likely need to search in a fairly specific niche, right at the sweet spot where classical methods struggle while a quantum computer excels. It is not entirely clear how large this niche is, and it is an active research area to identify more systems where classical methods fall short. One promising area involves multi-metal systems, where multiple metal ions are close together. Such systems are present in biologically relevant enzymes such as P450 and FeMoco.[3] Another is in heterogeneous catalysis, where the catalyst and reagents/products are in a different phase of matter.[4]

The first practical users of quantum simulation algorithms will most likely be scientists who study the fundamentals of quantum systems. Physicists are already employing devices that are similar to early quantum computers to mimic certain classes of materials. We wouldn't call these devices computers yet, but rather analogue simulators. One of the first actual applications of a fully digital quantum computer could be to analyse theoretical models of quantum materials, such as the famous Hubbard model.[5]

The first error-corrected quantum computers will hopefully find their place in industrial R&D settings. One of the first application areas could be to better understand the aforementioned multi-metal systems, which are relevant in the calculations of ligand binding affinities in drugs and in understanding the mechanism behind the biological production of ammonia. We address the latter example at the end of this chapter. Another exciting area could be to explore the mechanism behind Type-II superconductivity and to search for materials that become superconducting at even higher temperatures.[6] It is hard to say what the impact of quantum computers will be beyond such niche areas, as this will depend strongly on the usefulness of small polynomial speedups and unpredictable breakthroughs in quantum algorithms. We see a broad palette of other impactful applications that have been proposed, such as photocatalytic reactions (for example, efficiently splitting water to produce hydrogen fuel),[7] carbon capture mechanisms,[8] the study of efficient solar cells,[9] and the development of higher-capacity batteries.[10]

6.2 Algorithms for quantum chemistry

We describe three of the most important quantum simulation algorithms. The first is the **Trotter-Suzuki** method, sometimes called 'Trotterisation',

which simulates time evolution. In this case, we assume that some correct initial state of the world is encoded in the qubits of some quantum computer. The Trotter-Suzuki method is guaranteed to return a good approximation of the state at a later time, again encoded in the qubit registers.

The second algorithm is **quantum phase estimation (QPE)**, which reports the energy of a certain quantum state and can be used to produce a system's ground state. As a subroutine, it requires some time evolution method, like Trotter-Suzuki. Unfortunately, QPE can only provide information about a certain state if it receives an input that is already a reasonable approximation to this state. Especially in the context of describing low-energy configurations, this shifts the problem to producing good candidate ground states.

The most popular algorithm for creating states with certain properties (like very low energies) is the **variational quantum eigensolver (VQE)**. This is an example of a variational quantum circuit: a series of gates that can be gradually changed until the output matches certain requirements. Just like other variational approaches, it is a heuristic algorithm, lacking rigorous guarantees that it will produce the desired output in a reasonable time. However, it is a popular method today thanks to its ease of use and the ability to work with small, noisy computers.

Creating a good approximation to a ground state is, in general, NP-hard. This means that it is extremely unlikely that a rigorous algorithm exists that can find the ground state of *any* quantum system. On the other hand, there is good hope that more *heuristic* methods (just like VQE) will be found that work well on certain subsets of systems. In fact, such heuristic methods already form the workhorse of classical computational chemistry, with tools such as Density functional theory (DFT), Configuration Interaction (CI) and Quantum Monte Carlo (QMC). These work for small systems but are often too slow to study large systems such as proteins or drugs.[11] A workaround is to apply these methods to just a small part of the target system, employing faster but less accurate methods to oversee the larger whole.

An example of a basic workflow to find a ground state on a quantum computer could be as follows. The first step is to train a VQE to output states with low energy.[12] These might not be the exact ground states, but they will hopefully be similar (in jargon, they have a large overlap with the ground state). As a second step, we append a QPE circuit, which will not only report the energy of the VQE states, but also has a fair probability of changing these states into perfect ground states (in jargon: it projects onto the ground state). Running the VQE + QPE combination a few times will almost certainly give the lowest energy states, assuming the VQE produces proper approximations of it.

Further reading on simulation algorithms

Various more technical and sophisticated methods exist, for which we refer to other more technical sources. These require expert knowledge of quantum chemistry.

'Introduction to Quantum Algorithms for Physics and Chemistry' (2012),[13] a pedagogical book chapter.

'*Quantum Algorithms for Quantum Chemistry and Quantum Materials Science*' (2020),[14] a scientific overview article.

6.3 A hype around quantum computing for climate change

Some businesses make spectacular claims about how quantum computing could be a cornerstone in solving climate change, thanks to the boost to R&D on batteries, carbon capture, and more efficient chemical factories. However, rarely do we see any evidence – most seem to assume that quantum computers simply spit out blueprints for revolutionary sustainable technologies.

McKinsey takes the biscuit with their report titled 'Quantum computing just might save the planet'.[15] The article rightfully selects some of the most impactful technologies to reduce CO_2 emissions, like electrification of transport, improved solar panels, and even vaccines that reduce methane emissions by cattle (indeed, due to cow farts). The article concludes that the selected innovations could reduce global warming from 1.7–1.8 °C by 2050 down to just 1.5 °C. It is a mystery to us why they throw in quantum computing because there is no mention whatsoever about why specifically quantum algorithms would be the key enabling factor. This exemplifies what we see more frequently in popular articles: quantum computers are depicted simply as insanely fast computers that will magically solve the barriers to other new technologies on our wishlist.

What are the true prospects for quantum computing in the context of climate change? Sceptics may point out that technological innovations alone will not be sufficient to avert a climate disaster – we will remain agnostic

in this debate. A much more concrete issue is the mismatch in timelines. Climate experts agree that, to limit global warming to no more than 1.5° C, we need to act relatively soon. Imperial College London concludes on their website,[16] referencing the 2014 IPCC report:

> Limiting warming to 1.5°C will only be possible if global emissions peak within the next few years, and then start to decline rapidly, halving by 2030.

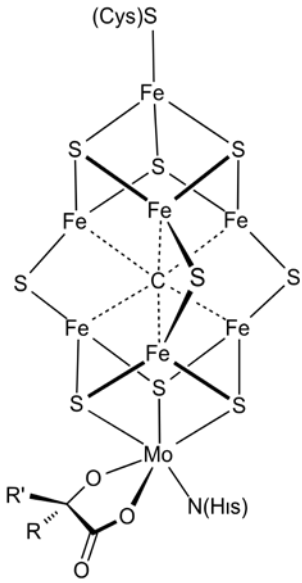
Our chapter on timelines shows that it is exceedingly unlikely that significant quantum utility is possible anywhere before the 2030s. Additionally, it will take several years before a computational discovery is sufficiently mature for large-scale deployment. For this reason, we don't see quantum computers as a good investment against climate change, but rather as a long-term development that can help us tackle other problems that humanity will face in the future.

Do we really have no concrete applications in climate science? Well, we do have some concrete leads. In the search for a killer application in chemistry, perhaps the most-studied topic is the enzyme Nitrogenase. Its active site is precisely a multi-metal system that classical methods struggle with, and as we'll soon see, it appears in reputable plans for decarbonisation. To understand the relevance of this molecule, we need to dive into the world of food production.

6.4 A case study of a potential killer application: FeMoco

Today's agriculture relies heavily on the use of artificial fertilisers. Without large-scale use of supplementary nutrients, we would not be able to sustain intensive farming practices and feeding our world's huge population would be problematic. In fact, about half of the nitrogen atoms in our body have previously passed a fertiliser factory!

Unfortunately, the production of fertiliser involves enormous energy consumption and carbon emissions. The main culprit is the ingredient ammonia (NH_3), of which we use as much as 230 Mton per year. Although our air consists mainly of molecular nitrogen (N_2), plants cannot directly absorb this. Instead, they rely on bacteria (or, in the case of artificial fertiliser, humans) to perform so-called nitrogen fixation, breaking the strong triple bond of molecular nitrogen and converting this into ammonia. Microorganisms can convert this into further nitrogen-containing compounds that the root system can absorb.



The chemical structure of the FeMo cofactor of the Nitrogenase enzyme. Figure credits: Smokefoot for www.wikimedia.org.

Pretty much all of the world's ammonia production facilities follow the so-called Haber-Bosch process, where hydrogen gas (H_2) and nitrogen gas (N_2) react together to form ammonia. This method has the benefit that it can be implemented in large, high-yield production lines but comes with the disadvantage of its staggering energy consumption. The inefficiency stems from two essential steps: first, producing sufficiently pure hydrogen and nitrogen gasses, and later, separating the H_2 and N_2 molecules into individual atoms. Breaking N_2 is especially challenging due to its strong triple bond. As an effect, factories operate at extreme conditions, with high temperatures (~400 degrees Celsius) and high pressure (over 200 atmospheres), driven mainly by natural gas. As much as 1.8% of the world's CO2 emission is caused by factories performing such reactions, consuming around 3–5% of the world's natural gas production!

Can't this be done more efficiently? We strongly suspect so. Certain bacteria are also capable of making ammonia, but in a seemingly more efficient way, without high temperatures or high pressure. It would be extremely valuable to copy this trick.

To imitate bacteria, we need to better understand a particular substance, the FeMo cofactor (in short: FeMoco), which acts as a catalytic active site during ammonia production. A perfect simulation of FeMoco is not possible on classical computers, as the structure of roughly 120 strongly reacting electrons rapidly becomes intractable. In 2016, researchers from ETH Zurich

and Microsoft were the first to report that a moderately large quantum computer could come to the rescue. A few years later, Google researchers refined the prospects even further, describing how simulations could be accomplished with about 4 million qubits and four days of computing time.

With FeMoco, we seem to finally have an example that confidently ticks all the boxes for quantum utility: classical methods are limited, we have well-understood quantum methods, and computational outputs have a significant commercial and societal impact. Unfortunately, there is yet another catch – innovation never comes so easily. A recent article[17] quotes that industrial production of a ton of ammonia costs around 26 GJ of energy, compared to at least 24 GJ (estimated) in bacteria. This is not the massive reduction we were hoping for. The article concludes that perhaps the true value lies in a better understanding of this process:

The chemical motivation to study nitrogenase is thus less to produce an energy-efficient replacement of the Haber-Bosch process but rather because it is an interesting system in its own right, and perhaps it may motivate how to understand and design other catalysts that can activate and break the nitrogen-nitrogen triple-triple bond under ambient conditions.

As a final note, we want to stress that quantum computers do not magically spit out recipes for fertilisers, nor for medicines, batteries, or catalysts. For real breakthroughs, we need collaborations between chemists, engineers, and many other experts who spend several years running experiments, having discussions, employing computer simulations, making mistakes, going back to the drawing board a few times, and slowly converging to practical solutions. We should not forget that quantum computers merely provide a new set of tools. The best we can hope for is that smart people will use them in the right way!

6.5 Further reading

(Scientific overview article) '*Prospects of quantum computing for molecular sciences*'

(Scientific overview article) '*Quantum Chemistry in the Age of Quantum Computing*'

(Scientific article) '*Toward the first quantum simulation with quantum speedup*'

6.6 Notes

1. Feynman, R.P. (1982) 'Simulating Physics with Computers,' *International Journal of Theoretical Physics*, 21(6), pp. 467–488. https://doi.org/10.1007/BF02650179.
2. Lee, S. *et al.* (2023) 'Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry', *Nature Communications*, 14(1), p. 1952. https://doi.org/10.1038/s41467-023-37587-6.
3. Santagati, R. *et al.* (2024). 'Drug Design on Quantum Computers', *Nature Physics*, 20(4), pp. 549–557. https://doi.org/10.1038/s41567-024-02411-5.
4. Hariharan, S., S. Kinge and L. Visscher (2024). 'Modelling Heterogeneous Catalysis using Quantum Computers: An Academic and Industry Perspective'. *ChemRxiv*. https://doi.org/10.26434/chemrxiv-2024-d2l1k-v2.
5. Daley, A.J. *et al.* (2022). 'Practical Quantum Advantage in Quantum Simulation', *Nature*, 607(7920), pp. 667–676. https://doi.org/10.1038/s41586-022-04940-6.
6. Chan, G.K.-L. (2024). 'Quantum Chemistry, Classical Heuristics, and Quantum Advantage'. *arXiv*. https://doi.org/10.48550/arXiv.2407.11235.
7. Leijnse, K. (2024). 'Photocatalysis for Water Splitting', *Quantum Application Lab*, 8 January. https://quantumapplicationlab.com/2024/01/08/photocatalysis-for-water-splitting/.
8. Von Burg, V. *et al.* (2021). 'Quantum Computing Enhanced Computational Catalysis', *Physical Review Research*, 3(3), p. 033055. https://doi.org/10.1103/PhysRevResearch.3.033055.
9. Hutchins, Mark. 'Quantum Physics, Supercomputers, and Solar Cell Efficiency'. *pv magazine International*, 4 August 2023. https://www.pv-magazine.com/2023/08/04/quantum-physics-supercomputers-and-solar-cell-efficiency/.
10. Choi, Charles Q. 'How Quantum Computers Can Make Batteries Better'. *IEEE Spectrum*. https://spectrum.ieee.org/lithium-air-battery-quantum-computing.
11. Santagati, R. *et al.* (2024) 'Drug design on quantum computers', *Nature Physics*, 20(4), pp. 549–557. https://doi.org/10.1038/s41567-024-02411-5. Quote from this article: 'Current classical quantum-chemistry algorithms fail to describe quantum systems accurately and efficiently enough to be of practical use for drug design'.
12. An interesting subtlety is how we measure the energy that the VQE is supposed to optimise. Fortunately, very short circuits exist that we can append to measure the output states in different *bases*. By running the VQE a relatively small number of times, we can make good estimates of the energy of its output states. This avoids

performing the more complex QPE during the optimisation phase. We only need the QPE to produce an accurate representation of the state we're searching for.

13. Yung, M.-H. *et al.* (2014) 'Introduction to Quantum Algorithms for Physics and Chemistry', in *Quantum Information and Computation for Chemistry*. John Wiley & Sons, Ltd, pp. 67–106. https://doi.org/10.1002/9781118742631.ch03.
14. Bauer, B. *et al.* (2020) 'Quantum Algorithms for Quantum Chemistry and Quantum Materials Science', *Chemical Reviews*, 120(22), pp. 12685–12717. https://doi.org/10.1021/acs.chemrev.9b00829.
15. Cooper, P. *et al.* (2022) *Quantum computing just might save the planet*, *McKinsey*. https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/quantum-computing-just-might-save-the-planet.
16. *How and When Do We Need to Act on Climate Change?* (no date) Imperial College London. https://www.imperial.ac.uk/grantham/publications/climate-change-faqs/how-and-when-do-we-need-to-act-on-climate-change-/.
17. Chan, G.K.-L. (2024). 'Quantum Chemistry, Classical Heuristics, and Quantum Advantage'. *arXiv*. https://doi.org/10.48550/arXiv.2407.11235.

7 The impact on cybersecurity

In the world of quantum computers, the most convincing exponential speedup lies in codebreaking. Anyone who wants to understand the impact of quantum computers must know the basics of cryptography. Let's start at the beginning.

7.1 Cryptography is much more than just secrecy

Why do we actually use cryptography? Pretty much everyone will immediately think of:

- **Privacy/confidentiality:** ensuring others cannot read your data (especially when messages are sent over a network).

However, there are many more threats that cryptography protects us from. Most people wouldn't normally worry about them, but when any of the following is missing, cybercriminals can cause a lot of harm:

- **Authentication/identification:** You want to verify that a message really came from the entity that claims to send the message. For example, during online banking, you want to be 100% sure that you are communicating with your bank and nobody else.
  Another example is when installing a new piece of software. When executing the latest Windows update, your computer makes sure to check that there is a **'digital signature'** that belongs to Microsoft. Imagine how unsafe your laptop would be if anyone could send fake updates!
- **Integrity:** You want to verify that nobody changed the message during transit. Imagine the damage when anyone can alter emails or file transfers, or when the commands coming from an air traffic control tower are modified. Similarly, any software installer confirms that the software wasn't changed by anyone but the original publisher, by verifying a digital signature.
- **Exchanging secret keys:** How do you negotiate a new secret key with a brand new web shop that you have never visited before? This is a seemingly impossible task if anyone can read bare internet traffic, but modern cryptography has a solution.

There are many other vital functionalities, like non-repudiation and availability, that we don't discuss here. Remember the bold-faced terms above, as we will come across these frequently.

A@#15GZ$de*_TVq47%B&86€HJfhjpw3KL4a^Z("927

We hope that this introduction makes you aware of the enormous importance of proper cryptography and the sheer number of cryptographic checks required for the proper functioning of our IT. You would be surprised how often you use cryptography on a daily basis through your laptop, phone, car keys, or smart cards.

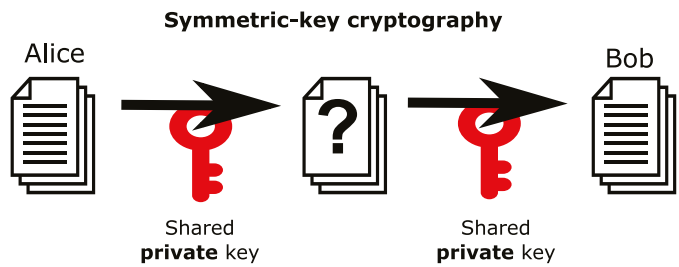
7.2 The quantum threat is mainly to public key cryptography

A common misconception, which we see a lot in popular literature, is that the quantum threat can be summarised as follows. (Both of the statements below are **incorrect**!)

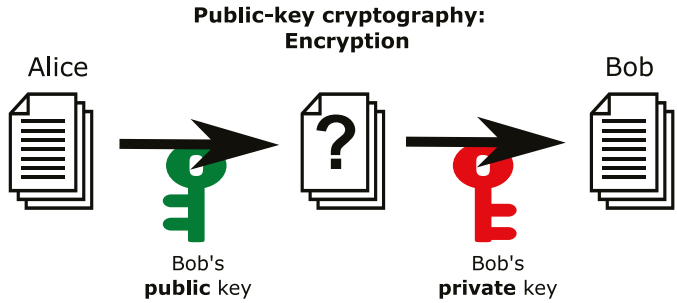
– 'A quantum computer will break all of today's cryptography'.
– 'A quantum internet is needed to keep our cryptography safe again'.

To better understand this, let's first look at what cryptography a quantum computer will break, and which it won't. Later, we will look at the necessity of a quantum internet.

In line with common cryptography jargon, we will typically have two parties, Alice and Bob, who want to communicate with each other. We distinguish two different types of cryptography: the symmetric and the asymmetric (public key) variants.

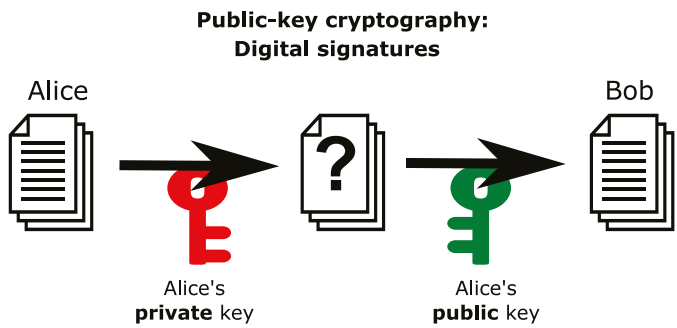


In **symmetric (or private key) cryptography,** we assume that both Alice and Bob already know a secret key. This could be a password that they both know or, more commonly, a very long number represented by, say, 128 bits in their computer memory. Alice can use the key to encrypt any message using a cipher like AES. Bob can then use the same key to decrypt this message. The details of how encryption and decryption work are unimportant for our purposes. The only relevant thing is that our computers can do this very efficiently and that it's considered sufficiently safe: without the key, nobody could reasonably break this encryption.

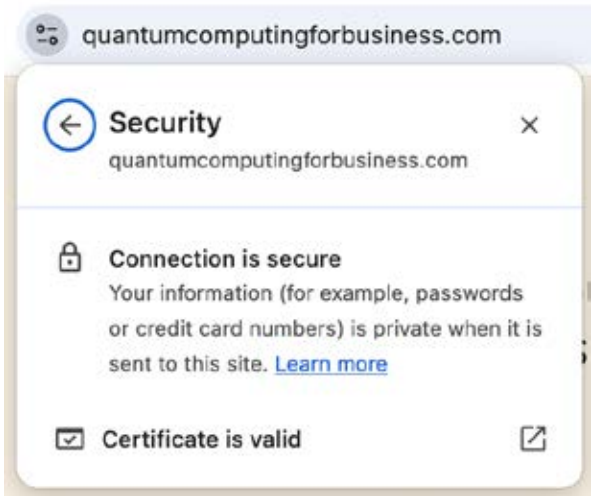


In asymmetric cryptography, more often called **public key cryptography (PKC)**, each participant has two keys: a public key and a private key. The public key can be shared with anyone, while the private key must be kept secret. That's why we use the suggestive colours green (save to share) and red (keep private!). If Alice wants to send an encrypted message to Bob, she uses Bob's public key to encrypt the message. The message can only be decrypted using Bob's private key, ensuring that only Bob can read the message.

The setting with two keys offers more functionality. For example, using public key cryptography, Alice could securely send a secret key to Bob that they can subsequently use for symmetric cryptography, which is faster in practice. When public key cryptography is built for this purpose, we call it a **key encapsulation mechanism** (KEM).



Furthermore, the protocol works in 'reverse'. Alice can use her private key to encrypt a message, which then anyone in the world (including Bob) can decrypt using the corresponding public key. Bob should then be confident that Alice is the only person who could have encrypted this message. Indeed, something encrypted with the private key can *only* be decrypted with the public key, and vice versa. The encrypted message is much like a signature that only Alice can produce. This forms the basis of digital signatures and certificates.



You can see public key cryptography in action whenever you visit a web page. Your browser (like Chrome or Firefox) will display that the connection is secure, which means that it verified that the digital signature is valid, amongst other things. This guarantees authenticity (the page came from a registered server) and integrity (the site arrived unchanged).

It should come somewhat as a surprise that public key cryptography is even possible at all! It's a small miracle that encryption and decryption with two totally different keys can be made to work, thanks to some powerful mathematics. However, it turns out that the delicate relationship between the two keys is also a weak spot…

How good are quantum computers at cracking cryptography?

Symmetric-key cryptography is quite safe against quantum hackers. The biggest problems are brute-force attacks, where an attacker effectively tries every possible secret key. Using a key size of 128 bits, the total number of possible keys is 2^{128} – that's an incomprehensibly large number, much more than the number of atoms in a human body.

We know that Grover's algorithm speeds up brute-force search by reducing the number of attempts from 2^{128} to its square root, which is 2^{64}. This is something that cryptographers are not happy about, but considering the slowness and extra overhead that comes with quantum computers, this doesn't seem to be a problem in the foreseeable future. Still, to be on the safe side, it is recommended to double key lengths, hence, to use the same algorithm with 256-bit keys. Changing this in existing IT infrastructure is relatively straightforward, although one

shouldn't underestimate the time and costs for such changes within large organisations.

The situation is entirely different with **public key cryptography.** The most-used algorithms today, RSA and ECC, can be straightforwardly broken by a large quantum computer. We discussed the details of Shor's algorithm earlier and saw that around 20 million qubits and eight hours are needed to retrieve a secret RSA key. Fortunately, there exist PKC systems that are believed to be safe against quantum computers, and an obvious way forward is to start using these. We call such systems **post-quantum cryptography**, and despite the confusing name, they're built to work on conventional computers. We discuss the rabbit hole of migrating to new cryptography in a different chapter.

Unfortunately, even today's communication could be at risk due to a practice called **harvest now, decrypt later.** Encrypted messages that are sent over a network can be intercepted and stored for many years, until a quantum computer can efficiently decrypt the messages. Even though we use public key encryption mainly to establish temporary keys for symmetric cryptography, a smart attacker could still retrace all the intermediate steps and retroactively spy on our communication. It is unclear at what scale storage of sufficiently detailed internet data is genuinely happening, but it seems plausible that security agencies of larger nations are already doing this.

The following table summarises how our cryptosystems are threatened:

	Symmetric	**Public-key**		**Quantum networks**
	Today (AES, …)	Today (RSA, ECC)	PQC	QKD
Safe against classical computers	✔	✔	✔	✔
Safe against quantum computers	✔* *with double key lengths	**Unsafe**	✔	✔

Why don't we switch to symmetric cryptography?

Public key cryptography solves a very fundamental problem: how can Alice and Bob agree on a secret key before they have a means of encryption in the first place? They cannot just send a new key over the internet without any form of encryption, because anyone would be able to read this. This is the fundamental **problem of key distribution**. Let us look at the functionality offered by the two types of cryptography:

	Symmetric	Public-key	Quantum key distribution
Confidentiality (privacy)	Only with pre-shared keys	✔	✗
Authentication / Integrity	Only with pre-shared keys	✔	✗
Establishing secret keys	✗	✔	✔* *Only when another mechanism takes care of authentication.

If only we could somehow give Alice and Bob pre-shared keys in a secure way, we would resolve most of these problems. Without public key cryptography, there are other options:

- **Trusted courier.** Alice and Bob could meet every other week to exchange USB drives with secret codes.
- **Trusted third party.** Alice and Bob could both trust a large 'key server'. If both share a secret key with the key server, they can securely ask the server to generate a new secret key that they can use together.
- **Quantum key distribution.** We discuss this solution further below.

Unfortunately, trusted couriers or trusted third parties are rarely an attractive alternative to public key cryptography, especially when scaling up to networks with thousands or millions of connected users. Couriers are simply too slow for today's standards, and single trusted parties would pose a particularly interesting target for attackers.

7.3 What solutions exist?

There is a clear need for post-quantum cryptography to replace commonly used cryptosystems like RSA and ECC. Fortunately, back in 2016, the American National Institute of Standards and Technology (NIST) started a competition to select a new cryptosystem, which should balance safety and practical usability (for example, it should not be too slow or memory-inefficient). They invited experts from around the globe to propose cryptographic algorithms, which peers assessed. Four rounds and several broken algorithms later, NIST selected a first set of winners that are suitable for large-scale use. As of August 2024, the first three PQC algorithms are now official NIST standards.

Even though this effort was coordinated by an American institute, the process was backed and carried out by cryptographers from around the world. A broad majority of cybersecurity experts have confidence in NIST's competition and recommend the final standards. National security organisations from other countries like BSI (Germany) and ANSSI (France) may prefer different algorithms but have also explicitly stated that this does not mean that they consider NIST's standards unsafe.

The results of the competition are as follows. Firstly, NIST selected one Key Encapsulation Mechanism that can be used to establish secret keys over an unencrypted connection – remember the problem of communicating with a web shop that you had never encountered before.

Functionality	NIST Name	Problem family	Documentation	Original name
Key Encapsulation Mechanism	ML-KEM	Module-Lattice based	FIPS 203	CRYSTALS-Kyber

Secondly, NIST selected three different Digital Signature Algorithms. These are used for authentication and integrity – remember how we don't want our messages to be altered in transit or how we want to prevent malware injected in software updates.

Functionality	NIST Name	Algorithm family	Documentation	Original name
Digital Signatures Algorithm	ML-DSA	Module-Lattice based	FIPS 204	CRYSTALS-Dilithium
Digital Signatures Algorithm	SLH-DSA	Stateless Hash-Based	FIPS 205	SPHINCS$^+$
Digital Signatures Algorithm	FN-DSA	Fast-Fourier Transform over NTRU-Lattice based	FIPS 206	FALCON

You might wonder why three algorithms were selected. Unfortunately, all three standards come with downsides, for example, because the keys can take up more memory or because the performance (time to sign or verify) is worse. The real-world impact will differ per use case. ML-DSA is the main cryptosystem recommended for general use, whereas SLH-DSA and FN-DSA may be beneficial in specific circumstances.

Are the new standards considered safe?

The short answer is yes: the new PQC standards are considered ready for use, and choosing algorithms such as ML-KEM or ML-DSA is widely regarded as a sound decision. There may be exceptions in specific high-security scenarios, but if you are operating in such a context, you are likely already aware of these nuances.

However, there seems to be some uncertainty within the cryptographic community regarding whether the new PQC standards will be as reliable as our trusted RSA or ECC. The new standards have not yet stood the test of time, and it is possible that unexpected weaknesses – whether minor implementation flaws or fundamental vulnerabilities – may still be present. To illustrate, a PQC method called SIKE[1] was in the race to become a new NIST standard and made it all the way to the fourth round until it was proven unsafe.

To mitigate any unexpected vulnerabilities in the new standards, most authorities recommend a **hybrid** implementation that combines the strengths of both conventional and post-quantum PKC. Moreover, organisations are generally advised to invest in **cryptographic agility**, a broad term used to describe the ability to easily update cybersecurity defences.

The above may sound somewhat negative, but we don't expect the slightly lower trust to stand in the way of adoption. Cryptographic algorithms themselves are rarely the weakest point, so it seems wise to focus on other potential vulnerabilities instead.

What about Quantum Key Distribution (QKD)?

Quantum key distribution is also presented as a solution for key exchange, making it a potential alternative to RSA, ECC and ML-KEM.

Still, many security authorities warn against adopting QKD today. Although the idea is promising, today's hardware is still immature. Moreover, QKD doesn't provide any functionality for digital signatures, thus we will need the migration to PQC anyway.

It is somewhat of a pity that QKD is not so mature yet, because it would be a viable weapon against Harvest Now, Decrypt Later. Nevertheless, since a quantum threat could be here as soon as the early 2030s, experts warn that companies and governments should fix their PQC first. At a later stage, QKD can be considered as an add-on for further security.

What about Quantum Random Number Generators (QRNG)?

Good random number generators are exceptionally important in cryptography, and QRNGs could provide a good alternative to the hardware random number generators that are widely used today.

However, all they do is generate random numbers – that doesn't make any protocol in itself quantum-safe. As a general warning: **products with 'quantum' in the name do not automatically protect against Shor's algorithm!**

7.4 Conclusion

Cryptography is strongly intertwined with quantum computing through Grover's algorithm, Shor's algorithm, and Quantum Key Distribution. Security experts recommend that there is an obvious way forward:

- Replace current public key cryptography with new, quantum-safe protocols (PQC);
- Double key lengths in symmetric cryptography.

Especially the first bullet is a major challenge. There are many legacy systems on the internet that can not be updated so easily. Billions of devices are all interconnected, so updating one device may cause incompatibilities somewhere else. Moreover, PQC protocols will likely require more CPU power, memory, and bandwidth than today's trusted methods. Companies may need to update the core code of hundreds or even thousands of applications. Lastly, the new protocols haven't been tested as extensively as our conventional methods, so it is not unlikely that new security issues will be found. Before they are even built, quantum computers are already causing headaches to cryptographers and cybersecurity managers.

7.5 Further reading

Cloudflare's resource page '*The State of the Post-Quantum Internet*' explains many aspects of the migration to post-quantum cryptography.

The NSA publishes recommendations on which cryptographic algorithms should be used and sketches a concrete timeline about when governmental security systems should be updated.

The PQC Migration Handbook is a free guide for corporate managers on how to tackle the upcoming cryptography migration, written by Dutch research organisations TNO, CWI, and the secret service AIVD.

In the context of Harvest Now, Decrypt Later, the urgency to migrate depends on how long your data should remain confidential, according to Mosca's Theorem.

7.6 Note

1. Goodin, Dan. 'Post-Quantum Encryption Contender Is Taken out by Single-Core PC and 1 Hour'. *Ars Technica*, 2 August 2022. https://arstechnica.com/information-technology/2022/08/sike-once-a-post-quantum-encryption-contender-is-koed-in-nist-smackdown/.

8 Applications of quantum networks

If we're building computers that deal with qubits, superposition, and entanglement, wouldn't these computers also need some way to send qubits to each other? This is the dream of the quantum internet: a network parallel to our well-known classical internet that allows the transmission of qubits.

There is a bit of a paradox here. On the one hand, a full-blown quantum internet that stretches across the globe is very, very far away – it will require quantum repeaters to bridge longer distances, purification mechanisms to repair imperfections, and many more technologies that we're only just figuring out. On the other hand, it is often said that quantum networks have a higher Technology Readiness Level than computing. That sounds like a contradiction, right?

The main explanation is that there are some applications for small-scale 'imperfect' quantum networks, particularly in the context of cryptography.

In a sense, quantum networking applications have always been ahead of quantum computing. Already in 1984, long before quantum computers were seriously considered, quantum pioneers Charles Bennett and Gilles Brassard discovered a method to securely negotiate a secret key (think of a password) between two distant parties based on sending individual photons. Their result is now famously known as the BB'84 protocol. Similarly, the commercialisation of network technologies has long been ahead of computing. Early quantum startups like MagiQ Technologies and ID Quantique were founded around the start of this century, and their first commercial networking products were brought to the market in 2003 and 2004. This technology, where a quantum network is used to generate a secret key at two endpoints, is called Quantum Key Distribution (QKD) – an application that we will address in much more detail below.

8.1 The promises of the quantum internet

There is a long list of arguments why we should be excited about the quantum internet. Here are some of the applications that we hear most frequently:

- **Clustering quantum computers:** By connecting multiple smaller computers, one might build a much larger computer with more combined memory, allowing it to tackle more complex problems.
- **Securing classical communication.** The main contender here is Quantum Key Distribution (QKD), sometimes dubbed the 'unhackable' network. This

allows two distant users to create a secret key (think of a password) that can be used in further cryptographic applications.
- **'Blind computing': Encrypting data while still allowing someone else to process it.** What if you hire an Amazon cloud computer to do calculations on your data, but you don't want Amazon to actually see the data itself? It turns out that you can make quantum computers do their computations even while the data remains encrypted, with some caveats. Similarly, one could use 'encrypted' software to solve someone else's problem without them discovering this algorithm. Such applications often go by the name of blind computing or private computing.
- **Position verification:** Can you prove that you are currently at a given location in a way that cannot be spoofed?
- **Protocols with multiple parties, where not every participant can be trusted**, such as leader election or Byzantine agreement. You can find many more in the Quantum Protocol Zoo.
- **Make quantum sensors more effective.** There exist proposals to combine different telescopes or gravitational wave detectors, and plans to synchronise quantum clocks.

8.2 How useful is the quantum internet in practice?

The impact of many quantum network applications will depend on how much we will use quantum computers. If quantum computers become widespread in the future, then communication between them also seems to be extremely worthwhile. On the other hand, our current outlook of quantum computers focuses on special-purpose devices used to solve isolated problems. In the latter scenario, the value of exchanging quantum data is not immediately clear.

There is an intriguing road map to build a reliable quantum internet in the future (involving fascinating tricks like entanglement distillation and teleportation), but this would require multiple error-corrected quantum computers by itself! Therefore, in this book, we're not yet ready to look ahead at applications like clustering computers, multi-party computations, private computing, or making sensors more effective. Regarding clustered quantum computers, we frequently hear arguments that one can make a bigger quantum computer by connecting individual ones, giving us access to larger numbers of qubits in a single calculation. It seems that building these computers right next to each other (and calling it a single computer) is much more effective than transporting fragile quantum data over large distances – clustering seems useful in extremely small networks.

In the foreseeable future, the first interesting applications are those that work over a 'noisy' connection and transport just one qubit at a time (or perhaps a handful of them). For practical interest, **Quantum Key Distribution (QKD)** is by far the most interesting application.

8.3 The case for QKD

To fully understand QKD, we require a bit more background about cryptography, especially the key distribution. For a full account, we recommend first reading the chapter on cryptography. In short, we're wondering how Alice can agree on a secret key with her distant friend Bob in a world where everyone can read plain data sent over the internet. Surely, they can't just send their secrets or passwords over to each other without having any encryption in the first place! This problem is commonly solved using *public key cryptography* (which we know will be revamped in the following years). If you really don't trust public key cryptography, the main alternative is to physically transport a USB stick by a trusted courier.

Compared to conventional cryptography, the unique selling point of QKD is that it is fundamentally impossible for cybercriminals to obtain the secret key as it is being distributed. As long as our understanding of quantum mechanics is correct (and we're convinced it is, as it's arguably the most well-tested theory in science), no amount of computational power or mathematical breakthroughs will let an attacker gain information about the key. Of course, this assumes that the protocol is executed precisely as prescribed and that there are no other vulnerabilities in the actual hardware or software.

This fundamentally differs from today's approach to public key cryptography, which must rely on certain mathematical assumptions. We know for sure that, with sufficient computational power, these codes can be broken, but we argue that this takes such a painfully long time that nobody will bother. Still, such statements about computation times are based on assumptions, and our trust derives from the empirical evidence that our smartest cryptographers have not found any weaknesses yet. In fact, well-regarded cryptosystems do get broken from time to time.

That said, although QKD is 'unhackable' in theory, the actual hardware *and* software are likely to contain vulnerabilities. Contrary to well-trusted public key cryptography, no QKD system has received proper certification and accreditation, and a significant fraction of historical products have been hacked.

QKD has the downside that it requires specialised hardware, although it is much less demanding than other quantum internet applications we mentioned. It can already be practical with a basic point-to-point network with just two connected parties, with one party limited to sending photons and the other limited to just measuring them. Moreover, the qubits need only be sent and measured one at a time, so no quantum memory or extensive quantum computations are needed. There have already been several demonstrations that use standard telecom fibre (the stuff that's already in the ground) or satellite-based systems that communicate through air. QKD hardware is fancy and expensive but not completely out of reach.

The fundamental downside of QKD is that it features no intrinsic way to confirm who the person on the other end of the line is. Some form of authentication is still needed – which is done with secret keys that should already be present in the first place! This makes QKD just a partial solution to the key distribution problem: it's mostly a key *extension* protocol, creating arbitrary amounts of key material based on a small initial key.

What do experts say?

Cybersecurity experts (indeed, the people who have been diligently keeping our classical computers safe for decades) are typically sceptical about QKD. In fact, all major security authorities that we are aware of currently advise against the use of QKD. They find the use of additional, uncertified hardware too large of a security risk and stress that there is a better solution that works on conventional computers: post-quantum cryptography (PQC). From their perspective, PQC offers all the required functionalities, and is currently more practical to test, certify and implement.

Be careful not to confuse the abbreviations PQC and QKD. QKD is about communication with a fancy quantum network. PQC runs on conventional hardware. You may call both of them 'quantum-safe' cryptography, as they should both resist attacks from a large-scale quantum computer.

A fair argument in favour of QKD stems from the harvest now, decrypt later attacks that could be done today. These imply that even the privacy of today's messages is at risk, which could be an argument for organisations to rapidly switch to QKD to protect their most sensitive data. Still, for those willing to go the extra mile for their privacy, looking at more mature and readily available solutions might be more worthwhile. For example, there exist certified solutions that rely on symmetric encryption with trusted couriers.

What's left is a niche use case for the most forward-thinking organisations that deal with fierce security requirements. It is a pity that QKD is not so mature today, as many organisations will start a migration to quantum-safe

cryptography soon. Widespread adoption of QKD would likely lower the costs of quantum networks and make it easier to expand to a large-scale quantum internet in the future. Nevertheless, since a quantum threat could be here as soon as the early 2030s, we stick with the recommendation to migrate to post-quantum cryptography first and to consider QKD as an add-on for additional security later, if needed.

8.4 Conclusion

In conclusion, most applications of a quantum internet will not be immediately relevant in the foreseeable future, with an exception for QKD. And even QKD might not be the killer applications that many investors are hoping for – it most definitely shouldn't be called 'unhackable'.

Still, it seems unfair to dismiss a quantum internet because it would be 'too technologically challenging' or 'too expensive'. These arguments are correct today but could be naive on a scale of several decades. Would anyone from the 1970s have believed that today, almost everyone on the globe is streaming videos on a mobile phone for just a few dollars per month? Who knows what the quantum internet will look like thirty years from now?

8.5 Further reading

Much more about the various quantum network applications can be found in an online Quantum Internet magazine by TU Delft or on the website of the Quantum Internet Alliance.

A video explanation of QKD for laymen or experts.

A nature commentary on why practical long-range QKD is still out of reach.

9 Optimisation and AI: What are companies doing today?

The earlier chapter on quantum applications discussed whether quantum computers can offer practical speedups for optimisation and AI. We concluded that this is a subtle case and that it is still unclear how much utility quantum computers can provide in these fields. Still, a large body of literature claims some form of near-term speedup in specific quantum applications. What's really going on here?

This chapter aims to build a more detailed intuition of how different organisations are exploring quantum applications. We will assess some example research papers, examine the problems they tackle, and analyse how convincingly they point to quantum utility. Moreover, we will look at the most fruitful directions for finding new and useful quantum algorithms. But before we dive into these details, let's take a step back and ask ourselves: what must a quantum algorithm do to be a genuine improvement over its classical counterpart?

9.1 Comparing Algorithms and Oranges

It is not straightforward to compare two algorithms. Perhaps one is faster on a particular computer, while another works better on a phone. Maybe one is best written in programming language A, and the other in language B. Computer scientists don't like dealing with such tedious details and resort to simply counting the number of fundamental computational steps an algorithm takes. In other words, they abstract away the actual computer and see the algorithm as a purely mathematical sequence of well-defined steps. Admittedly, the precise definition of step is still vague and machine-dependent. Therefore, algorithms are compared by their '**asymptotic complexity**' (or: 'asymptotic scaling'), which describes how the required number of steps grows as the problem becomes increasingly complex. What do we mean by a more complex problem? Intuitively, this is the case when an algorithm receives more data to parse, like larger numbers to factor, more locations on a map to route through, larger molecules to simulate, and so forth. The relative increase in the number of steps turns out to be completely machine-independent, allowing a fair comparison. Scientists use a systematic language to describe asymptotic scaling called


THE FUTURE LOOKS SO BRIGHT THAT YOU'LL HAVE TO ADJUST THE BRIGHTNESS
-
+
PAPARA

Big O notation (see the Box 'What does asymptotic runtime mean?' in the chapter on applications), making it straightforward to recognise and compare the efficiency of algorithms.

From the perspective of asymptotic scaling, a broad spectrum of quantum algorithms exists that could speed up optimisation tasks. Scientifically, it is downright fascinating that these algorithms can provide such advantages, using the laws of exotic physics to save trillions of computational steps. However, this book is about quantum computing for *business*, so while we appreciate the marvels of nature, at the end of the day, we want to know what the most *practical* way to solve our problems is. No matter what abstract mathematics says, all we care about is the actual wall clock time for our specific niche of problems.

At this point, the competition from classical computers becomes fierce. Today's processors from companies like AMD or Nvidia are so incomprehensibly fast that a quantum algorithm must be quite special before it can overcome the relative slowness of a quantum computer. Moreover, quantum computers will have a fair amount of overhead from error correction that conventional computers don't have to worry about. If we're looking at wall clock time, the race between quantum and classical is much tighter!

Even when we compare *classical* algorithms, asymptotic complexity isn't always the best indicator. For example, the Coppersmith-Winograd algorithm can multiply huge matrices relatively efficiently – asymptotically, it's much faster than the naïve brute-force methods used today. Large matrices are abundant in computationally hungry fields like engineering and AI, so one might expect Coppersmith-Winograd to be widely adopted. Nevertheless, it appears that hardly any professional software implementations actually use this algorithm, nor any of its relatives.[1] It turns out to be difficult to work with and enabling its speedup requires even larger matrices than we handle today. Asymptotic complexity is a useful tool, but no silver bullet.

Moreover, the theory of asymptotic complexity is unsuitable when comparing heuristic algorithms. For example, a class of problems that we call 'NP-complete' is hard to solve in theory, while we have software tools like Gurobi and CPLEX that solve such problems quite well on a daily basis.

The only truly fair comparison is **benchmarking.** It involves standardised tests to indicate the performance of an algorithm or a machine. The tests could be as simple as a set of reference problems that should be solved as quickly as possible. For example, supercomputers are commonly compared through the LINPACK benchmark, whereas algorithms for the Traveling Salesman Problem can be tested in TSPlib. The field of AI has been playing this game for a long time, focusing on fuzzy problems like producing natural

English texts or recognising what's on an image – stuff that's hard to formally define in mathematics. For example, neural network architectures for image recognition cannot be taken seriously until they have been tested on standardised datasets like MNIST and ImageNet.

To assess the advantage of quantum computers, we'll need to compare them to classical machines in similar benchmarks. Unfortunately, today's hardware is far from adequate, and, so far, the best comparisons are based on resource estimates and heuristic arguments. Today, it seems nearly impossible to prove the utility of a quantum optimisation algorithm.

Nevertheless, it is not hard to find articles that boldly claim a business-ready speedup with just a few thousand noisy qubits, and we strongly recommend being sceptical about such sources. There are many ways in which such results can be misleading. For example, many articles merely report that a quantum computer *can* solve a problem but fail to quantify how fast or accurate it is in comparison to the best-known classical method. These articles can still have very suggestive titles that make one believe that a quantum computer is faster. Sometimes, researchers compare their quantum algorithm only to 'weak' contenders, like classical brute force search or a simplified algorithm that's rarely used in practice. Such situations are likely to occur when analysing some obscure dataset or solving a problem that nobody has seriously looked at before. Occasionally, a quantum algorithm is benchmarked against a classical machine learning model trained by the same researchers. Optimising AI methods is finicky, and such reports make us sceptical about whether the classical method was treated just as carefully as the quantum approach. All of these examples indicate the importance of testing quantum algorithms against well-studied classical approaches.

This all sounds quite negative, but we still see it as a positive development when companies perform early explorations of quantum algorithms, often testing accessible algorithms like variational circuits on sector-specific toy problems. Quantum computing can be incredibly complex, and it will take time to gain experience, train a qualified workforce, and tackle all the barriers that stand in the way of taking a quantum algorithm to production. It would be best for the field if everyone is honest when the outcome of a proof-of-concept is primarily a set of learned lessons, without inflating the result as a revolutionary speedup.

To conclude, quantum algorithms will need to prove their worth in standardised benchmarks, similar to how leading AI methods are assessed today. While we are waiting for the hardware to mature, the most relevant information comes from rigorous resource estimates. One should be careful with claims purely based on an algorithm's performance on relatively small-scale problems.

Further reading

The scientific paper 'Better than classical? The subtle art of benchmarking quantum machine learning models' performs a systematic test on several quantum machine learning models.

Olivier Ezratty proposes a framework to assess quantum computer case studies.

Metriq is a platform that collects several early quantum benchmarks.

(Technical) The Quantum Economic Development Consortium (QED-C) proposes benchmarks based on several optimisation tasks.

Microsoft Azure features a resource estimator that helps gauge the number of qubits and the amount of time needed to run certain quantum algorithms.

9.2 Where should we look for a new killer application?

Well, we simply don't know! However, some useful technical hints may be:

- We'd most likely require an *exponential,* a large *polynomial*, or some *heuristic* speedup. This is much more likely achieved on problems where we don't already know very efficient classical algorithms.
- When reading data is a limiting factor (for example, in big data applications), quantum computers appear to be relatively slow. Getting the data into a quantum computer seems to take at least as long as processing the data on a much cheaper supercomputer. This holds, for example, when searching through a large database, but also for data-intensive simulations like weather forecasting.

– Similarly, if the desired output is a large amount of data (such as a very large list or table), then a quantum computer is likely not efficient. Most quantum algorithms look at a global property of a function or dataset that can be encoded in a very small output (like Deutsch-Jozsa or Shor's algorithm when interpreted as finding the period of a function).
– Some people would say that if quantum computers are not 'faster', perhaps they might solve a problem 'more accurately' (for example, they might produce a more reliable forecast). However, when we look at speedups, then accuracy is already taken into account: we compare the number of needed to achieve a given accuracy.
– Classical computers are already incredibly fast, and the bottleneck for many real-world computational problems is not in a computer's clock speed. If an application does require a supercomputer today, then it's unlikely that anyone will invest in a quantum computer soon.

9.3 Examples of results in different sectors

To gain further understanding of the commercial applications of quantum computers, we reach a point where we can no longer provide any generic wisdom. The best way to understand this field is by looking at various examples. In this section, we present three industries that are commonly mentioned in the context of quantum applications: pharmaceuticals, finance, and energy. For each of these, we briefly highlight typical use cases and discuss one or two technical reports.

The reports are picked for no particular reason except that they should provide a decent amount of technical information – much more than a typical press release or blog post would. Moreover, these reports cover a broad spectrum of results, tackling different problems, featuring different types of companies, and taking different perspectives on the degree of utility that quantum computers would offer. We limit ourselves to use cases in optimisation and AI, because quantum simulation and cybersecurity are already covered in more depth in different chapters.

Note

The application areas and use cases highlighted here are speculative: there is no hard guarantee that quantum computers will offer significant advantages

for these applications. We selected the examples below because they have notable *potential*, meaning that further investigation is justified (and will likely happen in the following years).

Moreover, this section is meant to give examples, and it's far from exhaustive.

Pharmaceutical industry & health

The pharmaceutical sector seems willing to make long-term investments, mainly because IP and patents can be very profitable. Indeed, the larger corporations file some 50–100 'quantum' patents each year.[2] Part of the enthusiasm is justified because computational chemistry R&D is part of their core business. The broader health industry, including parties like hospitals and manufacturers of medical equipment, may have less focus on quantum simulations but are still frequently mentioned.

Some of the most studied themes include:

- Computer-aided drug discovery, where a (quantum) computer simulates how a proposed drug reacts with compounds in the human body. In particular, quantum-mechanical interactions may be relevant when estimating the binding strength between a drug and biological compounds;
- Optimising strategies for drug synthesis;
- Simulation of the molecular spectra expected in NMR or spectroscopy experiments.

Even though the chemical nature of drug design lends itself well to exponential speedups, some restraint is warranted. The most important quantum speedups are expected for *strongly correlated systems* that exhibit large amounts of superposition and entanglement. A recent overview article states the following about drug design:[3]

> [Classical methods] offer good-enough accuracy for most systems. This is because most oral drugs are small closed-shell organic molecules (they need to pass through the gut wall to be absorbed) which generally lack strong correlation.

This leads them to conclude:

> [I]f the advantage of quantum computers is limited to strongly correlated systems, they might have limited practical significance in drug design.

Nevertheless, there are still plentiful computational challenges that classical computers haven't solved, both in the areas of quantum simulation and optimisation. Whether quantum computers will address just a small niche of strongly correlated systems or prove to have broader applicability is still an open question.

Example results

Exploring the Advantages of Quantum Generative Adversarial Networks in Generative Chemistry

The paper is based on Generative Adversarial Networks (GAN), where two neural networks are trained simultaneously. One network is a 'discriminator', which has to detect whether a structure (graph) of a molecule derives either from a fixed dataset or whether it is created by the other network, the 'generator'. By training both networks in parallel, they become increasingly adept at their task, such that eventually, the generator mimics natural molecule structures very accurately.

The paper constructs the GANs partially from **variational quantum circuits** (VQC) and sees improvements in some benchmarks. Note that this has only been tested for relatively small molecules.

My subjective view is that this looks like an overall interesting approach. The abstract does get us sceptical due to a claim that the authors 'demonstrate the quantum advantage of a VQC in the discriminator of GAN' because the VQC performs certain tasks better than a classical neural network while using fewer internal parameters. A comparison to just one self-written classical contender is never fair. Moreover, a quantum model with fewer parameters can still take more time and resources to train or optimise.

Press release: https://zapata.ai/news/zapata-foxconn-insilico-medicine-university-toronto-quantum-generative-ai-for-drug-discovery/.

Paper reference: Kao, Po-Yu, Ya-Chu Yang, Wei-Yin Chiang, Jen-Yueh Hsiao, Yudong Cao, Alex Aliper, Feng Ren, et al. 'Exploring the Advantages of Quantum Generative Adversarial Networks in Generative Chemistry'. *Journal of Chemical Information and Modeling* 63, no. 11 (12 June 2023): 3307–3318. https://doi.org/10.1021/acs.jcim.3c00562.

Organisations involved: Insilico Medicine, Foxconn, Zapata.

Hybrid Quantum Image Classification and Federated Learning for Hepatic Steatosis Diagnosis

In this work, the authors train a neural network to assess photos of livers with the aim of diagnosing non-alcoholic fatty liver disease (NAFLD). They compare a standard (classical) convolutional neural network with a 'hybrid' model that contains a variational quantum layer. The paper claims that the quantum version is more accurate by 1.8 percentage points.

My personal evaluation would be quite positive if it weren't for an important detail that the quantum layer uses just five qubits. It seems unlikely that such an architecture would outperform classical methods in a fair comparison, especially because simulating five qubits is trivial for a classical computer. A possible explanation is that the classical network wasn't properly optimised (and the paper doesn't share the necessary details to check this). This hypothesis seems supported by one of the paper's own plots, where the classical model's accuracies drop when it gains access to more training data. This shows why it's important to compare algorithms on well-studied benchmarks.

Press release: https://www.einpresswire.com/article/735111499/quantum-algorithm-outperforms-current-method-of-identifying-healthy-livers-for-transplant.

Paper reference: Lusnig, Luca, Asel Sagingalieva, Mikhail Surmach, Tatjana Protasevich, Ovidiu Michiu, Joseph McLoughlin, Christopher Mansell, et al. 'Hybrid Quantum Image Classification and Federated Learning for Hepatic Steatosis Diagnosis'. *Diagnostics* 14, no. 5 (6 March 2024): 558. https://doi.org/10.3390/diagnostics14050558.

Organisations involved: Terra Quantum, University of Trieste

See also:

(Scientific overview article) 'Drug design on quantum computers', https://www.nature.com/articles/s41567-024-02411-5 (open access: https://arxiv.org/abs/2301.04114).

(Scientific overview article) 'Quantum Computing for Molecular Biology', https://doi.org/10.1002/cbic.202300120.

Finance

There is an extensive body of literature on applications in the financial services sector. Our intuition tells us that this is mainly thanks to two top-down reasons: small algorithmic improvements can quickly lead to large monetary gains, and institutions like banks have relatively long investment horizons, making them more willing to invest in technologies that could be several years away. Unfortunately, at this point, there is little evidence for rigorous exponential speedups in this sector, so the focus is primarily on polynomial and heuristic improvements.

Some of the most commonly studied themes include:

- Optimising investment portfolios (for high profit and low risk);
- Analysing risk and studying future market scenarios;
- Estimating the price of complex assets, such as options;
- Fraud detection.

Example results

Quantum Deep Hedging

A hedge is an investment chosen specifically to offset the potential for loss in other investments. For example, a bank with many assets in a volatile market might also invest in a sector that typically moves in the opposite direction. The problem can be cast in a conventional reinforcement learning framework, where a computer program makes virtual investment decisions and receives rewards depending on its performance, allowing it to learn better strategies. Deep hedging is an existing classical method to train a good software agent using deep (multi-layer) neural networks.

This paper investigates the potential of quantum computers in this area. Amongst other things, the authors replace certain network layers with quantum variants. Compared to the classical approach, they achieve comparable scores while using fewer trainable parameters. They also produce qualitatively different investment strategies, hence offering something unique compared to the conventional approach. The new methods are tested on Quantinuum's H1–1 and H1–2 trapped ion computers using up to 16 qubits.

Our subjective interpretation is that this is an interesting and sound paper that focuses on rigorous analysis rather than extravagant claims. As a downside, we are not aware of any standardised benchmark in this field, nor is there evidence that the quantum approach could lead to faster computation times (as the reduction in parameters suggests).

Press release: https://www.jpmorgan.com/technology/news/jpmorgan-chase-qcware-evolve-hedging-for-a-quantum-future.

Paper reference: Cherrat, El Amine, Snehal Raj, Iordanis Kerenidis, Abhishek Shekhar, Ben Wood, Jon Dee, Shouvanik Chakrabarti et al. 'Quantum Deep Hedging'. *Quantum* 7 (29 November 2023): 1191. https://doi.org/10.22331/q-2023-11-29-1191.

Organisations involved: JPMorgan Chase, QCWare, Université de Paris

Quantum portfolio optimisation by Citi Innovation Labs and Classiq

The portfolio optimisation problem is as follows. You receive a list of possible stocks you may invest in and a ***probabilistic*** outlook of their expected gains and volatility (i.e. the riskiness of the stock). The gains can be correlated. Given that you're allowed only to take a certain amount of risk, what would be the optimal set of stocks to invest in?

In this work, the authors optimise assets using the Quantum Approximate Optimisation Algorithm, an example of a variational quantum circuit. There are no methodological innovations, but the authors do a good job of combining existing building blocks into a full end-to-end implementation: the algorithm is written in a high-level software package (by Classiq), using real-world data (by Yahoo finance) in a standard Python data processing pipeline (using Pandas), and running the resulting quantum program through the cloud (through AWS, albeit on a classical simulation in this case). There is no comparison with any classical methods.

In our subjective interpretation, this is more a marketing outing (showcasing the technical wit of the parties involved) than a newsworthy result. Nonetheless, several news outlets picked this up, most likely thanks to the large companies involved.

Press release: https://www.classiq.io/insights/citi-and-classiq-advance-quantum-solutions-for-portfolio-optimization-using-amazon-braket.

Blog reference: 'Citi and Classiq Advance Quantum Solutions for Portfolio Optimization Using Amazon Braket | AWS Quantum Technologies Blog', 7 February 2024. https://aws.amazon.com/blogs/quantum-computing/citi-and-classiq-advance-quantum-solutions-for-portfolio-optimization/.

See also:

(Scientific overview article featuring JP Morgan researchers) 'Quantum computing for finance', https://www.nature.com/articles/s42254-023-00603-1 (open access: https://arxiv.org/abs/2307.11230).

(Scientific overview article featuring QC Ware researchers) 'Prospects and challenges of quantum finance', https://arxiv.org/abs/2011.06492.

Energy

The energy sector is another branch where we see much enthusiasm for quantum technologies, possibly because the sector generally focuses heavily on new innovations to transition away from fossil fuels. It comprises various parties involved in the production and distribution of electricity, gas, and oil, making for a diverse community of utility organisations, petrochemical industries, shipping companies, and many others.

We make a distinction between two types of use cases: those based on quantum simulation (chemistry and material science) and those based on optimisation and AI.

Optimists will point out that there is much potential for large speedups in chemical R&D, which could be one of the earlier application areas. Commonly studied applications are, for example:

- The development of new battery types, which ideally have a high capacity and low weight, cause limited pollution, are recyclable, and rely primarily on materials that are not too difficult to acquire. Better batteries have obvious uses in consumer electronics, electric vehicles, load balancers and emergency power supplies.
- Efficient water splitting: refining the production of hydrogen gas from plain water. The hydrogen itself can be used as high-capacity fuel.
- Finding carriers for hydrogen fuel. These carriers can absorb hydrogen such that it becomes faster, easier or safer to transport.
- Simulations of nuclear fission or fusion, contributing to improved reactor designs.

In its current state, the optimisation side has the obvious issue of relying on polynomial or heuristic speedups. Nevertheless, there is a broad range

of high-performance computing challenges waiting to be solved. Some of the most-studied quantum use cases include:

- Management of electricity grids. This includes balancing the load over different cables/stations (such that currents remain in a safe range and losses are limited), simulating exotic situations, and computing the optimal placement of new electricity lines.
- Prediction of electricity supply and demand.
- Electricity price forecasting.
- Finding optimal sites for oil and gas extraction.

Example results

Practical Quantum K-Means Clustering: Performance Analysis and Applications in Energy Grid Classification

K-means clustering is a widely used unsupervised learning problem. Given a set of datapoints, can we group these into k different clusters, such that all the vectors in a cluster are 'similar'? Here, 'similar' means that the distance between two datapoints (vectors) is small.

The paper applies this to the context of the German low-voltage electricity grid. They selected 1037 regions that they call 'subgrids', and gathered 26 characteristics for each of these. Example characteristics are the average (non)-renewable energy load, operating voltage, power line thickness, and so forth. The goal is to identify subgrids that are similar, which has applications in predictive maintenance: it can be expected that similar subgrids will experience similar failures.

Unsurprisingly, the paper concludes that small datasets with few clusters gave the most reliable results, as noisy quantum computers struggle with larger problems. The paper is honest about the fact that, at this moment, quantum computers offer little advantage over classical methods. The goal is to look ahead and build experience before executing these algorithms on more powerful devices in the future.

Paper reference: DiAdamo, Stephen, Corey O'Meara, Giorgio Cortiana, and Juan Bernabé-Moreno. 'Practical Quantum K-Means Clustering: Performance Analysis and Applications in Energy Grid Classification'. *IEEE Transactions on Quantum Engineering* 3 (23 June 2022): 1–16. https://doi.org/10.1109/TQE.2022.3185505.

Organisations involved: E.ON Digital Technology, Technische Universität München

See also

(Scientific overview article) 'Quantum Computing and Simulations for Energy Applications: Review and Perspective'. https://doi.org/10.1021/acsengineeringau.1c00033.

9.4 Further reading

Some news websites report on new developments in quantum applications. For example, *Quantum Computing Report* covers mostly business-oriented news, while *Quanta Magazine* takes a more scientific perspective.

The Quantum Application Lab describes how it tackles several real-world problems with quantum computers in well-written and accessible blog posts.

XPRIZE runs a competition to design and employ quantum algorithms that address global challenges.

(YouTube, technical) Ronald de Wolf presents an in-depth overview of known speedups in quantum optimisation algorithms aimed at viewers with a strong mathematics background.

(Scientific overview article) In 'Quantum Optimization: Potential, Challenges, and the Path Forward', a large group of researchers discuss the field's open questions and elaborate on the importance of benchmarking.

9.5 Notes

1. For example, see this discussion on StackOverflow: https://mathoverflow.net/questions/101531/how-fast-can-we-really-multiply-matrices/.
2. Newton, W. (2023) 'Quantum medicine: how quantum computers could change drug development', *Clinical Trials Arena*, 24 February. https://www.clinicaltrialsarena.com/features/quantum-computers-drug-development/.
3. Santagati, R. *et al.* (2024) 'Drug design on quantum computers', *Nature Physics*, 20(4), pp. 549–557. https://doi.org/10.1038/s41567-024-02411-5.

Part 3

The hardware and strategic actions

10 Quantum hardware

Conventional computer hardware is extremely reliable. Professional servers are supposed to run non-stop for years without any hardware failures. If you take a new product out of a box, you can be reasonably sure that it will work precisely as advertised – and if does not, it should be straightforward to replace. Moreover, classical IT is extremely well-standardised. No matter what supplier you buy a computer from, you can be reasonably sure you can run your favourite applications on them. Thanks to such high reliability and clear compatibility, it is rather easy to compare different machines, for example, by looking at speed (e.g. floating-point operations per second, FLOPS) and memory size.

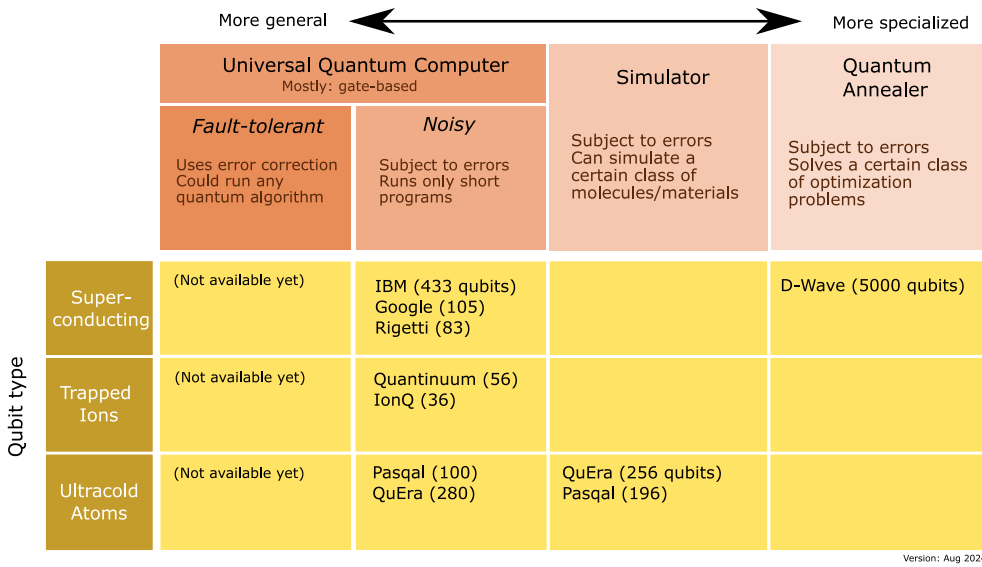
We will see that this is radically different for quantum computers. Devices make mistakes, have limited functionalities, and memory is scarce compared to classical computing standards. Several manufacturers focus on niche applications, making trade-offs in certain features to enhance performance in others. In this chapter, we take a high-level perspective at quantum computing hardware. We address the two most important aspects:

- What functionality does a device have?
- What type of qubits are used?

10.1 Different functionalities

The figure below shows three different functionalities that quantum computers can have (top, red), along with some examples of products on the market (yellow), built from different building blocks. This list is by no means complete! It should, at best, give an indication of the current state of the art. Let us start by taking a closer look at the functionalities.

Our biggest dream is to have a '**universal quantum computer**'. The word 'universal' indicates that it can execute any quantum algorithm (or, technically, it can approximate any algorithm's output to arbitrary precision). For comparison, your laptop, phone, and even a modern coffee machine are universal classical computers, making them capable of running any classical application you can think of: spreadsheets, 3D games, data encryption, and so on. Similarly, a proper universal quantum computer is suitable for any quantum application, regardless of whether it is already known today or invented in the future.



The definition of 'universal' is blind to some details, such as memory limitations (it assumes you will never run out of RAM), and omits tedious details about software compatibility (a PlayStation game won't run on an Xbox). In our high-level overview, such details are unimportant: the main point is that there also exist devices that can *not* run just any algorithm.

Does a universal computer need to be 'gate-based'?

No, there are various computational models that are universal.

There are different ways to make a 'universal quantum computer'. The most popular way is to use a **gate-based** approach, where elementary operations ('gates') change the data stored one or two qubits at a time. This perspective is most intuitive for those used to conventional logical circuits (with AND, OR and NOT gates), and most quantum algorithms are presented in this language. Other alternatives include **adiabatic** computation and **measurement-based** computation, which can theoretically run any algorithm written for a gate-based computer without issues and vice versa.

Currently, gate-based computers are by far the most widespread and appear to be the most popular approach in the race towards a million-qubit quantum computer: nearly all large tech companies rely on this architecture. There is one important exception. Some **photonics startups** are working towards measurement-based computing, as this overcomes the challenges in performing 'entangling' quantum gates with photons. In the following, we will focus mostly on gate-based computers.

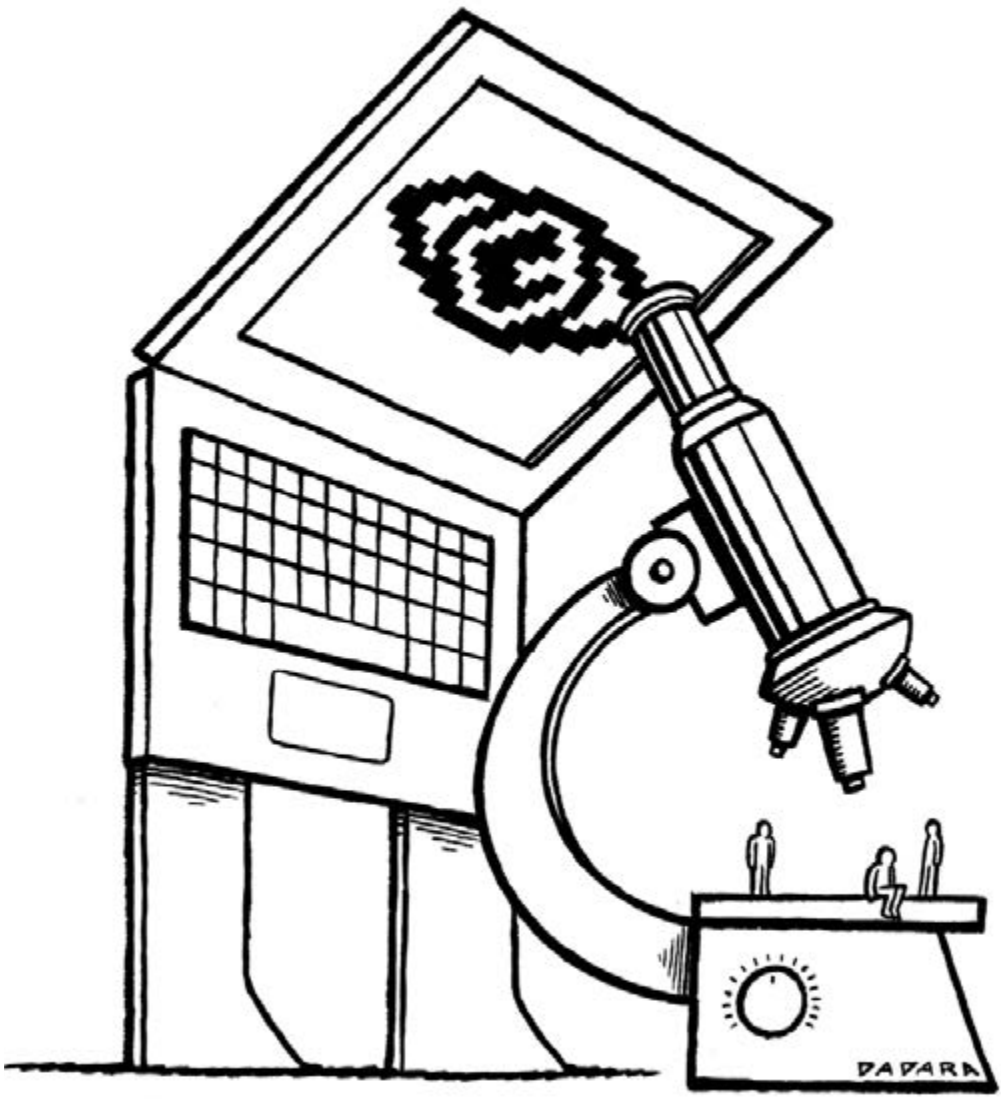
No matter what architecture or qubit type you pick, today's technology will only allow you to run relatively short computations. This is due to the inherent imperfections in qubit construction and control methods. The imperfections cause errors to accumulate, so after some number of steps, the result is almost surely corrupted and unusable. For longer computations, fixing errors on the fly is essential, using so-called **error correction**.

At the time of writing, we live in the so-called NISQ era, with **Noisy Intermediate-Scale Quantum** devices. Many are theoretically fully universal, except that they are limited both in the number of qubits and, most of all, in the number of steps they can execute. Companies like IBM, IonQ, Quantinuum, and Pasqal all have NISQ computers available to test over the cloud.

A universal computer is a jack-of-all-trades, but it excels at nothing. Engineers can make **special-purpose devices** that improve in certain areas (like the number of qubits or clock speed) by omitting certain functionalities. A **quantum simulator** specialises in mimicking the behaviour of a particular class of materials or molecules. The precise capabilities can be described in the mathematical language of a 'Hamiltonian' that specifies which materials qualify. For example, Harvard-spinoff QuEra offers a quantum simulator over the cloud that mimics a quantum Ising model.[1] Today's simulators (like QuEra's) are fairly similar to a universal NISQ computer, missing only a few essential ingredients, and similarly having restrictions due to noise. Although they look similar, they are not designed to run conventional (gate-based) algorithms.

The jargon around simulators can be a bit confusing. Firstly, the term 'quantum simulation' is also used when a classical computer tries to calculate the output of a quantum algorithm. To differentiate, some prefer the term 'emulation' for such classical approaches. Secondly, we often hear a distinction between 'analogue' and 'digital' simulation. Ironically, both approaches tend to discretise information over discrete qubits (which we call digital). In practice, the terms are rather used to distinguish between continuous and discrete time steps. An analogue simulation would use longer, continuous operations on the qubits, whereas a digital simulation uses quantum gates that act in short, discrete bursts on the qubits.

Another special-purpose device is the **quantum annealer**, popularised mainly by the Canadian scale-up D-Wave. These special-purpose devices can solve a specific class of optimisation problems that goes by the name of QUBO: quadratic unconstrained binary optimisation. There is a well-developed theory of mapping various industrial problems into the QUBO


DADARA

formalism, making annealers fairly versatile machines. However, quantum annealers will never be able to take advantage of the various other quantum algorithms out there: even with enough qubits, we won't see them cracking codes using Shor's algorithm.

Further reading

D-Wave's introduction to its quantum annealing platform

Scale-up Pasqal reports on a material science simulation with 196 qubits. In another article, they explain why an 'analogue' quantum simulation has its advantages.

QuEra makes a 256 qubit simulator available over the Cloud.

10.2 Different building blocks

Another important question concerns the materials used to create qubits. Scientists have cooked up several competing approaches, such as superconducting materials, photons, individual atoms, or ions, each with their own strengths and weaknesses. When comparing different qubits, we use the terminology of qubit implementation, the qubit type, or (what we prefer) qubit **platform**.

The conventional computer electronics industry has settled on a single choice of material and manufacturing process: essentially, all computer chips are made using lithography on silicon wafers. On the contrary, there is an ongoing race between wildly different qubit platforms, and it is still unclear which will eventually be the winner — or whether we will converge to a single winner at all.

There is fascinating physics behind the different hardware types, but we won't delve into that in this non-technical book (would you care otherwise what material your classical CPU is made of?). However, as soon as you want

to test a prototype quantum program on real-world NISQ hardware, you probably want to learn more details. Interested readers are invited to take a look at the references below.

It is interesting to note that all these different functionalities (universal computers, annealers, and simulators) can, in principle, be built using any type of qubit. Returning to the figure at the top, you can see that specific qubit platforms have been used for multiple purposes, and it's likely that the empty fields will also be populated in the future.

10.3 Further reading

Different types of qubits explained by Sifted.eu

Different types of qubits at IQC Waterloo

Different types of qubits on Wikipedia

A MOOC about different hardware types by TU Delft

10.4 Note

1. Gemelke, N. and Lukin, A. (2022) *Hamiltonian simulation on QuEra's 256-qubit Aquila machine*, *QuEra*. https://www.quera.com/events/hamiltonian-simulation-on-queras-256-qubit-aquila-machine.

11 Error correction

At a glance

To run long computations, we need to dramatically reduce the likelihood of error in each computational step – not just a little bit, but by a factor of millions.

Error correction is the most effective method to achieve extremely low error probabilities. It combines a small number of 'physical' qubits (think of several hundred) into a single 'logical' qubit that suppresses errors *exponentially*.

Logical qubits are still not perfect: the 'number of steps' that they can survive is an important specification that determines whether they can a particular application.

It's 2024 and we're seeing a major shift in the road maps of quantum computer manufacturers. Several companies no longer put their bare qubits in the spotlight, but instead focus on *logical qubits.* Error correction seems to be an essential component of large-scale quantum computing, adding yet another facet in which these devices differ from their classical counterparts. Although this is a relatively advanced topic, we find it so important that it deserves a dedicated chapter in this book.

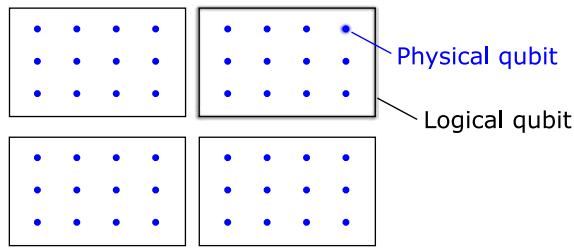
As with many aspects of quantum computing, error correction can be rather confusing. A statement (that is incorrect!), which we often hear is:

> Logical qubits (or error-corrected qubits) are resilient to errors that occur during a computation. Once we have logical qubits, we can increase the length of our computations indefinitely.

What's the problem here? Well, not every logical qubit is created equally. In the near future, we expect to see logical qubits that are perhaps 2x more accurate than today's bare hardware qubits, and later 10x, and in the future perhaps 1000x. Error correction is a trick to *reduce* the probability of errors, but it will not eliminate errors completely. In the following decade, we expect gradual improvements, hopefully down to error rates of 10^{-10} and below.

11.1 What is error correction?

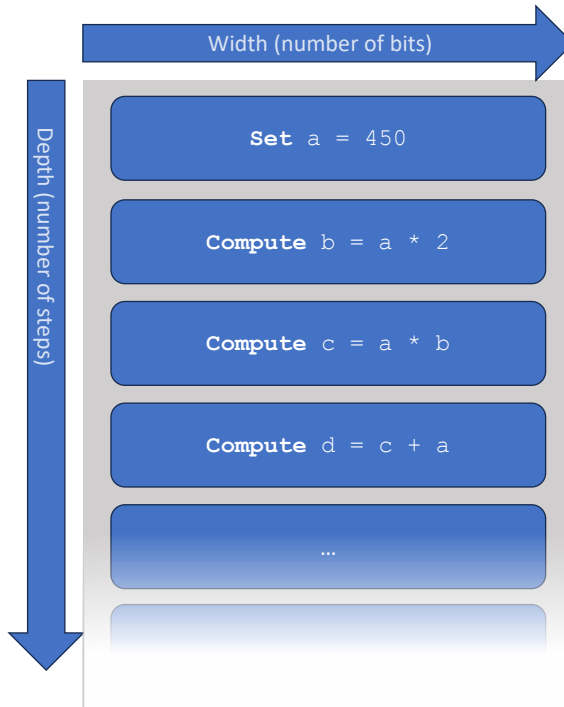
In quantum error correction, we combine some number (think of hundreds or thousands) of **'physical'** hardware qubits into a virtual **'logical'** qubit. The logical qubits are the information carriers used in an algorithm or application. Error correction methods can detect whenever tiny errors occur in the logical qubit, which can then be 'repaired' with straightforward operations. Under the assumption that the probability of hardware errors is sufficiently low (below a certain error threshold), the overall accuracy improves exponentially as we employ more physical qubits to make a logical qubit. Hence, we obtain a very favourable trade-off between the number of usable qubits and the accuracy of the qubits.



Doesn't measuring a quantum state destroy the information in the qubits?
Indeed, if we naively measure all the physical qubits, we destroy potentially valuable information encoded in the qubits. However, quantum error correction uses an ingenious way to measure only whether or not an error occurred. It learns nothing about the actual information content of the qubit. It turns out that this way, the data stored in the logical qubit is not affected.

Why are errors so much of a problem? How do errors screw up our computations?
In short, even tiny errors are a problem because we want to perform an astonishing number of quantum operations successively — think of billions or trillions of them.

Let's make this more concrete. A computer program is essentially a sequence of **'steps'**, each of which a computer knows how to perform. We say that a program or algorithm has a **width**, which is the number of qubits it requires. It also has a **depth**, which is the number of consecutive steps that need to be performed. You may interpret one step in early hardware as a single quantum gate (although, in practice, gates may be performed in parallel, making the impact of errors slightly more complicated).



The concept of 'width' is pretty straightforward: if the computer doesn't have enough memory, it cannot run the program. Dealing with 'depth' is harder. To run a program of 10^9 steps, we need to limit errors to roughly the inverse, say, a probability of 10^{-9} per step. If the error is larger, it becomes extremely unlikely that the quantum computer will produce the correct outcome. These are not hard numbers: a computer with 10^{-10} error would be a significant improvement (resulting in much fewer mistakes), and a computer with 10^{-8} error might be pushed to also find the correct answer after many tries. However, as the imbalance between depth and error grows, the probability of finding a correct outcome is reduced *exponentially*. We illustrate this in more detail in the box below.

To illustrate, why do we need such small error rates?

Let's look at a simple model of a computer, which is not unlike what happens inside a quantum computer or a modern (classical) CPU. As above, the computer is supposed to work through a list of instructions. We can consider various specifications of a computer:

– The available memory, measured in bits (or perhaps megabytes or gigabytes, if you like).

- The speed at which the computer operates, measured in steps per second.
- The 'probability of error', describing the likelihood that one gate introduces a mistake. This is given as a number between 0 and 1 (or a percentage between 0 and 100%). Many sources use the word 'fidelity' instead, which can be roughly interpreted as the opposite (fidelity ≈ 1 – probability of error). In this text, we sometimes just say 'error' while we mean its probability.

In this simple model, the time taken to complete the computation equals 'depth' x 'speed'. You can make the calculation faster by increasing the speed of the computer or by writing a 'better' program that takes fewer steps.

The influence of errors is harder to track. For contemporary computers, we typically don't worry about hardware mistakes at all, as every step has essentially 100% certainty to output the correct result. However, let's see what happens when this is not the case.

Assume that each step has a 1% (= 10^{-2}) probability of error. What will the impact be on the final computation? Below, we compute the probability to finish the computation without any errors, for various numbers of computational steps.

Error probability: 1%	
Number of steps	P(success)
1	$(0.99)^{1} = 99\%$
100	$(0.99)^{100} = 37\%$
1000	$(0.99)^{1000} = 0.004\ \%$
10,000	$(0.99)^{10,000} = 10^{-44}$

In this simple model, we assume that *any* error is catastrophic. This is quite accurate for most programs. You might argue that there is a miniscule probability that two errors cancel, or that the error has little effect on the final result, but it turns out that such effects are statistically irrelevant in large computations.

Now, if we improve the hardware to have an error rate of just 0.1% (=10^{-3}), we find the following.

Error probability: 0.1%	
Number of steps	P(success)
1	$(0.999)^{1} = 99.9\%$
100	$(0.999)^{10} = 90\%$
1000	$(0.999)^{1000} = 37\%$
10,000	$(0.999)^{10,000} = 0.004$

A 37% probability of succeeding may sound bad, but for truly high-end computations, we might actually be okay with that. If the program results in a recipe for a brand-new medicine or tells us what the perfect design is for an aeroplane wing, then surely we don't mind repeating the computation 10 or 100 times, after which we're very likely to learn this breakthrough result. On the other hand, if the probability of success is 10^{-44}, then we will *never* find the right result, even if the computer repeats the program billions of times.

In the table above, we see a pattern: to reasonably perform 10^{2} steps, we require errors of roughly 10^{-2} or better. To perform 10^{3} steps, we need roughly a 10^{-3} probability of error. These are rough order-of-magnitude estimates, but they lead to a very valuable conclusion when dealing with very large circuits (or very small errors): if you want to execute 10^{n} steps, you'd better make sure that your error probability is not much bigger than 10^{-n}.

This simplified model assumes that an operation either works correctly or fails completely, with nothing in between. In reality, quantum operations act on continuous parameters, and therefore, they have an inherent scalar-value accuracy. For example, a quantum gate with 99% accuracy might change a parameter from A to A+0.49, where it's supposed to do A+0.5. Fortunately, for our discussion, these details don't matter much. It suffices to see a '99% accurate' quantum gate as simply having a 99% probability of succeeding. We also overlook various other technical details, like operations carried out in parallel, different types of errors, native gate sets, connectivity, and so forth — these make the story much more complicated but will not change our qualitative conclusions.

Why don't we just make the hardware more stable?
To some degree, we can further reduce errors by creating more accurate hardware. However, quantum objects are so incredibly fragile that even getting down to 10^{-2} errors requires some of the world's most astonishing engineering. We definitely hope to see two-qubit gate errors reduced to 10^{-3} and perhaps even 10^{-4}, but achieving targets of 10^{-9} seems unlikely with incremental hardware engineering alone. On the other hand, quantum error correction is incredibly effective: the error drops dramatically at the cost of adding a modest number of qubits, which is assumed to be scalable anyway. That's why experts agree that error correction is the right way forward.

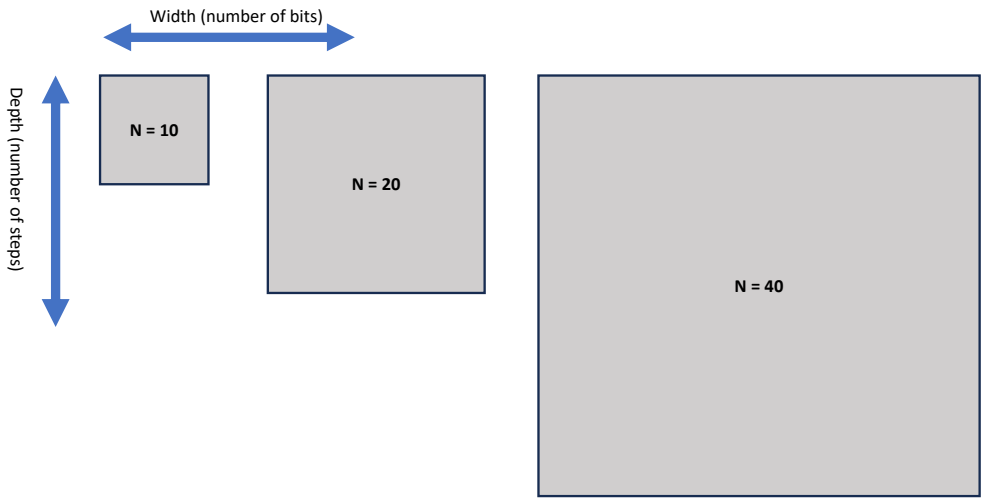
Do we use error correction in classical computers too?
This might be a good moment to appreciate the incredible perfection of classical computer chips. While doing billions of steps per second, running for months in a row, sometimes with hundreds of cores at a time, errors in CPUs practically never occur. We were hoping to find hard numbers on this, but companies like Intel and AMD apparently keep this data under stringent non-disclosure agreements. However, some research shows that errors well under 10^{-20} are easily attained as long as we don't push processors to their limits (in terms of voltages and clock speeds), sufficiently low that error correction is rarely needed.

Memory (RAM) for high-performance computers still frequently has built-in error correction, and some form of CPU error correction was sometimes used in older mainframes and (even today) in space probes.

11.2 Longer computations need more qubits

As problems become more complex, they typically require better computer hardware, both in terms of width (number of bits) and depth (number of steps). We illustrate this below. We define a number 'N' that indicates the difficulty or the size of the problem. For example, we might consider the task of 'factoring a number that can be written down using at most N bits').

Requirements to solve a problem, depending on it's 'size' N

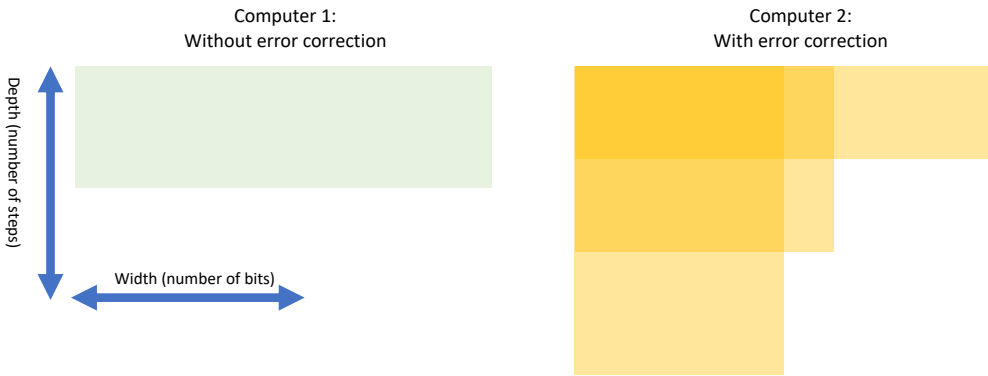


Remember that we're talking about the requirements to solve a problem; so, here, width indicates *logical* bits. If a computer does not have error correction, then one logical bit is simply the same as one physical bit – or its quantum equivalent.

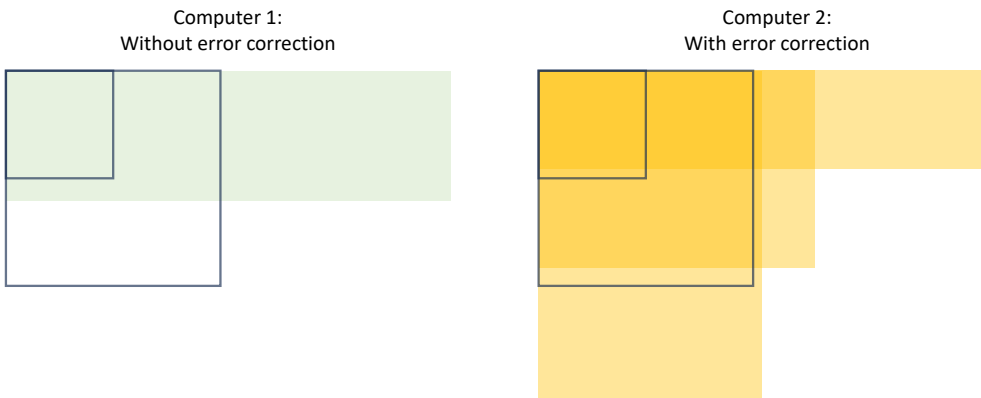
For 'perfect' classical computers, the situation is straightforward: if a problem gets bigger, we need more memory, and we need to wait longer before we obtain the result. For (quantum) computers that make errors, the situation is more complex. With increasing depth, not only do we need to wait longer, but we also need to lower the error probabilities and, hence, need more extensive error correction.

Let's consider two computers for which we show the width and depth that they can handle (where the available 'depth' is assumed to be 1 / 'probability of error'). On the left is a computer without error correction (hence, it has a small, fixed depth). The other is an error-corrected computer that can trade between depth and width (in certain discrete steps).

Example quantum computers



The computer without error correction might have enough memory to solve a problem but often lacks the depth. Even an error-corrected computer might not have a suitable trade-off to solve the hardest problems. Looking at the above example, it seems that both computers can solve the N=10 problem. Here, only the error-corrected computer can solve the N=20 problem, as depicted below. For the N=40 problem, which would be represented by an even larger box, the error-corrected computer might have sufficient depth OR sufficient width, but it doesn't have both at the same time. Hence, neither computer could solve the N=40 problem.



In terms of cracking the N=40 problem, our best bet is to upgrade the error-corrected computer to have *more physical qubits*. Using error correction, these can be traded to achieve sufficient depth (whilst also reserving just enough *logical qubits* to run the algorithm).

We have found a paradoxical conclusion here. Larger problems not only require more memory (to store the calculation) but also more depth, which requires more qubits again! To summarise:

'Harder' problems -> More depth -> Better error correction -> More physical qubits

Once we reach an era of error correction, scaling the number of physical qubits will still be at the top of our wishlist, as this will be the key enabler of longer computations.

11.3 What is the current state-of-the-art?

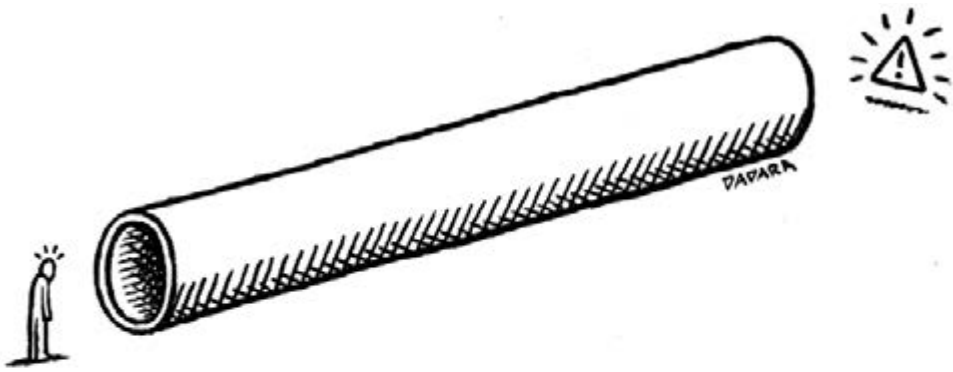
This section is more technical and can be safely skipped. As of 2024, there have been several demonstrations of error correction (and the slightly less demanding cousin: error *detection*), but these have all been with limited numbers of qubits and with very limited benefit to depth (if any at all). However, we seem to be at a stage where hardware is sufficiently mature that we can start exploring early error correction.

Below are the three most popular approaches to error correction. Each of them can be considered a 'family' of different methods based on similar ideas:

- Surface codes;
- Colour codes;
- Low-Density Parity Check (LDPC) codes.

The surface code (or toric code) has received a lot of scientific attention, as this seems to be on the roadmap of large tech companies like Google and IBM. Their superconducting qubits cannot interact with each other over long distances, and the surface code can deal with this limitation. Many estimates that we use in this book (such as the resources required to break RSA or to simulate FeMoco) are based on this code. It has already been tested experimentally on relatively small systems:

Colour codes are somewhat similar to surface code but typically lack the property that only neighbouring qubits have to interact. This makes them less interesting for superconducting or spin qubits, but they appear to work extremely well for trapped ions and ultracold atoms.



THE ERROR AT THE END OF THE TUNNEL

LDPC codes are now rapidly gaining attention. They build on a large body of classical knowledge and could have (theoretically) more favourable scaling properties over the surface code.

Which code will eventually become the standard (if any) is still completely open.

What are the main challenges?

Firstly, we would need just *slightly* more accurate hardware. We mentioned a certain accuracy threshold earlier: state-of-the-art hardware seems to be close to this threshold but not comfortably over it. Secondly, error correction also requires significant classical computing power, which needs to solve a fairly complex 'decoding' problem within extremely small time bounds (within just a few clock cycles of a modern CPU). Classical decoding needs to become more mature, both at the hardware and the software level. It is likely that purpose-built hardware will need to be developed, which for some platforms might be placed inside a cryogenic environment (placing stringent bounds on heat dissipation). Theoretical breakthroughs can still reduce the requirements of classical processing.

Lastly, it turns out that 'mid-circuit measurements' are technically challenging. Without intermediate measurements, one might retroactively detect errors, but one cannot repair them. We should also warn that many related terms exist, such as 'error mitigation' and 'error suppression'. They might be useful for incremental fidelity improvements, but they don't bring an exponential increase in depth like proper error correction does.

11.4 Conclusion

The bottom line is that one shouldn't naively take 'logical qubits' as perfect building blocks that will run indefinitely. A logical qubit is no guarantee that a computer has any capabilities; it merely indicates that some kind of error correction is applied (and it doesn't say anything about how well the correction works). A much more interesting metric is the probability of error in a single step (in jargon: the fidelity of an operation), which gives a reasonable indication of the number of steps that a device can handle!

11.5 Further reading

'*The Quantum Threat Timeline Report*' asked several experts what they find the most likely approach to fault-tolerance (section 4.5).

British startup Riverlane builds a hardware chip that decodes which error occurred on logical qubits. They provide an accessible press release and a more technical scientific article.

Craig Gidney (Google) has a more technical blog post on why adding physical qubits will remain relevant in the following decades.

(Technical) Some *scientific* work speaks of 'early fault-tolerant' quantum computing, such as:

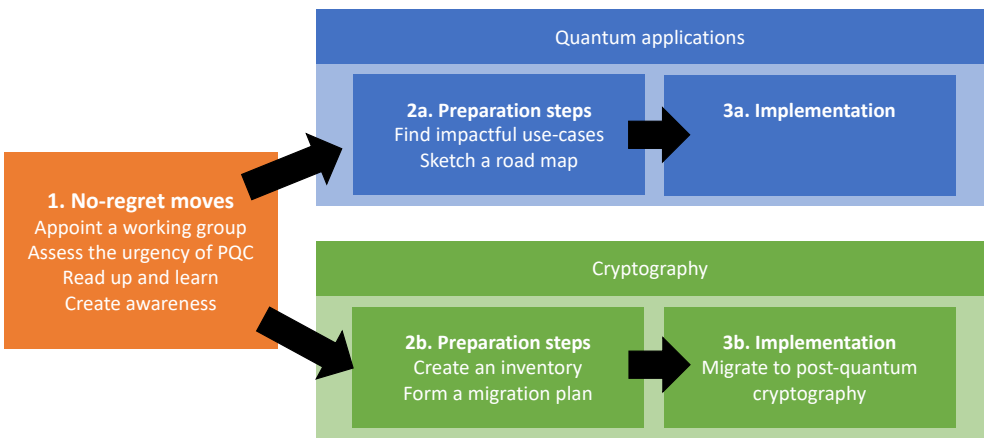
'Early Fault-Tolerant Quantum Computing', discussing how we can squeeze as much as possible out of limited devices.

'*Assessing the Benefits and Risks of Quantum Computers*' takes a similar width x depth approach as we do here, but uses it to assess what applications will be within reach first.

12 What steps should your organisation take?

In the previous chapters, we discussed the use cases, the threats, and the timelines of quantum technologies. We will now look at the strategic perspective of a typical non-quantum enterprise. We will assume a typical large-scale organisation that does not sell IT products per se, but relies heavily on computing infrastructure to optimise its operations, supervise processes, communicate with suppliers and clients, and potentially invest in computer-aided R&D. While these organisations may be excited about the potential of quantum computing, they may also feel vulnerable – whether due to competitors advancing ahead or due to hackers attacking legacy cryptography.

We outline the typical process an organization undertakes in three steps. The first steps, like growing expertise, finding adequate staff, and doing first proof-of-concept studies, will be largely sector-independent. Further steps can become more organisation-specific, and we will highlight several tools for tailored assessment



12.1 Common first steps

Step 1: Start with no-regret moves
Most companies start with early steps aimed at better understanding the situation. These can be done with very little financial risk.

Some must-do actions:
- Appoint a quantum lead or a quantum working group tasked with following the developments.
- Read up and learn. If you've come this far in this book, you're already doing a fantastic job. We have a separate chapter on further learning resources.
- Create internal awareness. Many employees will enjoy inspirational talks, tours or demonstrations that academics or quantum manufacturers can provide.

Optionally:
- Put quantum on the agenda with senior management.
- Involve collaborators, suppliers and vendors, and make your interest in quantum known. It is to your benefit if suppliers are well-prepared.
- Participate in a workshop, hackathon, or similar event.

In terms of more concrete follow-up actions, it makes sense to split your quantum journey into two different categories:

a. Preparing for **quantum applications,** where the goal is to leverage quantum technologies to gain some competitive advantage (for example, by strengthening your R&D, further optimising your logistics, improving a product, etc).
b. Migrating to **quantum-safe cryptography,** where the goal is to keep your IT secure against attackers with a quantum computer.

These endeavours serve very different purposes and are likely spearheaded by different departments. Hence, it seems logical to break these down into separate projects. We discuss further steps in both directions separately.

12.2 Prepare to use quantum applications

Step 2a: Explore use cases

At this stage, most organisations will want to make low-regret moves that get them prepared to leverage quantum technologies fairly soon after practical utility becomes available. Some of the bottlenecks could be the lack of in-house knowledge, a limited available workforce, or a long timeline to integrate quantum applications in production environments.

Must do:
- Identify the most impactful use cases in your sector.
- Sketch a road map for the coming years.

Optionally:

- Start concrete proof-of-concept projects. Right now, these are unlikely to offer practical utility and will likely tackle just a toy problem. However, these help build experience in setting up quantum projects and can uncover 'unknown unknowns'. For staff with a strong physics or mathematics background, it is relatively accessible (and fun!) to get acquainted with quantum programming packages and implement a first test algorithm.
- Find strategic partners. Organisations can save costs by collaborating on early, pre-competitive exploration.
- Create PR! We notice that many companies are actively promoting their early results on quantum applications, even if these do not offer significant advantages yet.
- Hire staff with a strong background in quantum technologies who understand the market, have the right skills to lead proof-of-concept studies, and can offer advice for strategic decisions.

Step 3a: Implementing actual applications, whenever ready

From here onwards, it gets increasingly difficult to give concrete advice, as priorities may depend on your business and on the way the field of quantum computing will progress. Several sources will simply tell you do 'develop a long-term strategy' or similar. Others highlight the need to 'remain agile' to quickly adapt to this rapidly evolving field.

For inspiration or a dot on the horizon, you may think towards a competence centre for quantum computing, similar to how many companies have special departments for data science and/or AI. A concrete task could be to elaborate on the list of impactful use cases from the previous step, benchmarking the performance of various quantum and classical software tools. Another task could be to professionalise an earlier proof-of-concept project, bringing it closer to implementation in a production environment.

Identifying fruitful use cases

From a top-down perspective, it is a good exercise to identify your current needs in high-performance computing. What do you currently spend your computing budget on? Are there any areas where new tools in computation or modelling could provide serious business value (for example, by being faster, tackling bigger problems, or delivering higher accuracy)? Which quantities would you ideally have calculated but are beyond the reach of current computers? This results in a longlist of use cases where new computational tools are worth further investigation. The next step would

be to research to what extent a quantum computer (or whichever other new computational tool) offers any advantage.

We recommend this top-down approach because it can lead to conclusions sooner, especially because it avoids studying use cases that are *not* worth your time (for example, because additional computational power provides little value).

It is also possible to take a bottom-up approach. Looking at the available quantum algorithms, which would speed up processes in your existing IT? Would any of them provide value for your business? This more technical perspective requires some in-depth quantum expertise but can definitely be worth the effort, especially if you have people with the right skills available.

> The Quantum Application Lab is a collaboration between various Dutch research organisations. They invite end-users to explore the benefits of quantum computers in projects that last anywhere between three and twelve months, ranging between a first exploration of use cases to advanced development of quantum prototype software. Several example projects can be found on their website: www.quantumapplicationlab.com.

Further reading

Scientists propose a framework to discover which real-world problems are potentially accelerated by quantum computers.

Consultant Olivier Ezratty proposes a framework to assess the maturity of quantum computing case studies.

(YouTube) A recording of Quantum.Amsterdam's online seminar 'What do companies get out of quantum projects today?'

What does an R&D collaboration with academia look like?

Several end-users have started collaborations with universities to better understand the use cases of quantum computing. This is often a win-win situation, as companies can learn from renowned experts at relatively low costs, whereas academics benefit from additional funding and showcasing that their research has practical interests. Moreover, many countries provide subsidies for so-called 'public-private partnerships'. Below, we sketch a personal experience with the process of starting such a partnership.

You will most likely be dealing with a university's tech transfer office (TTO), which specialises in making in-house knowledge available externally. As a first step, it is important to agree on the scope of the project: what are the research questions, what are the expected outcomes, how long will the project run, and so forth. Ideally, this would be a discussion between an expert from your organisation and a university's (assistant) professor. The professor will most likely take a supervising role, as the actual work is often executed by a junior researcher employed as a PhD candidate or a postdoctoral (PD) researcher. PhD programmes take relatively long, 3–5 years depending on your locale, and it may take some time before the first results come in. Postdoc projects often take 1–3 years and can lead to results sooner, but as of 2024, it can be much harder to hire a postdoc with the right competencies.

When the topic and duration of the project are clear, it is important to discuss details around intellectual property (IP), often done by legal experts. For universities, it is important that researchers can keep building upon the project's IP in an academic setting. Moreover, they will demand that the results can be published in scientific journals. At the same time, a paying company will want sufficient options to patent new discoveries and will require exclusive use of the IP within their sector. These demands do not necessarily conflict with each other, and in principle, it should be possible to find an arrangement that satisfies both parties.

A straightforward way to ensure that the company learns from the academic developments is by organising meetings or workshops throughout the collaboration project, in which the ongoing R&D is discussed with company staff. The occasional dialogue with company staff is arguably more important than a shiny final report or paper, which risks disappearing in someone's drawer.

12.3 Migrating to post-quantum cryptography

This section relies on technical knowledge from the previous chapter on cybersecurity.


SUPERPOSITIONMAN
00
10
01
11
PADARA

Step 2b: Prepare your migration

Cryptography is a completely different beast, with a more concrete goal, and more urgent timelines for most organisations. Contrary to the applications in the previous section, the cryptography migration is not optional. Fortunately, most organisations face the same problem, and there is ample research on effective steps. The core challenge is to upgrade *all* existing public key cryptography to Post-Quantum Cryptography (PQC) in the next decade, which could be spread over hundreds or thousands of different applications. Many businesses, especially those dealing with critical infrastructure, may additionally deal with regulators who may or may not have guidelines ready. Moreover, IT transitions can be incredibly slow – it is not uncommon to see plans that cover five or even ten years.[1]

Authorities seem to agree that the following initial steps should be taken urgently by all large organisations.

- Create awareness: make sure that the quantum threat is well-understood in your security departments and among IT managers and product owners throughout the organisation.
- Create an inventory of cryptographic assets used within the organisation. This should include both software and hardware and should clearly specify the used algorithms, whether developed in-house or purchased from a vendor. Some parties refer to a 'cryptographic bill of materials' (CBOM).
- Determine the risk and urgency of PQC migration. Most organisations already perform regular risk assessments of their IT infrastructure. Additionally, organisations should assess whether they classify as an urgent adopter of PQC (see below).
- Create a migration plan. This is a more complex step, which should at least prioritise which assets must be migrated first and indicate whether the migration of all urgent systems can be realistically achieved in time, before the arrival of cryptographically relevant quantum computers.

For more details, we recommend following the *PQC Migration Handbook*, a free guide written by the Dutch secret service AIVD and research organisations CWI and TNO. Security authorities in other countries have made similar guidance available.

Are you an urgent adopter?
Planning ahead to transition to new cryptography can be more critical depending on the organisation. We can distinguish between regular and urgent adopters. You are an urgent adopter when you:
- Handle sensitive or personal data with a long confidentiality span;
- Handle critical infrastructure on which large groups of people rely;
- Provide systems with a long lifespan; hence, your products will still be around when quantum computers are available.

Based on these criteria, a significant fraction of organisations would classify as urgent adopters, such as banks, governments, car manufacturers, grid operators, hospitals, and so forth. Examples of non-urgent adopters could be schools, webshops, travel agencies, some construction agencies, etc. Urgent adopters are encouraged to start their migration as soon as possible if they haven't already.

Step 3b: Migrate
This is a much more technical step for which you will need a well-prepared migration plan from the previous step.

Organisations are strongly discouraged from implementing their own cryptographic functions. The best practice is to rely on standard libraries written by cryptographic experts, which should be safe against a broad spectrum of attacks and have seen careful reviews. We expect NIST's standards to soon be available in popular open-source packages like OpenSSL or BouncyCastle. This makes the migration less technical, although organisations still deal with the operational challenge of updating a huge number of applications within a limited time.

Due to harvest now, decrypt later attacks, most organisations will focus on updating key exchange algorithms before updating digital signature methods.

On the technical side, cryptographic experts recommend the use of **hybrid** algorithms that combine the strengths of PQC (to defend against quantum attacks) with a proven conventional public key algorithm (which guarantees at least the original security in case the new PQC algorithm turns out to be less safe than expected). For example, early versions of quantum-safe connections with the Chrome web browser use a combination of X25519 (elliptic curves) and Kyber-768 (ML-KEM).

Moreover, the practice of **cryptographic agility** is strongly encouraged, meaning that security protocols can be easily updated and replaced. This is a vague term that isn't just a software feature – it requires alignment with business protocols and internal policies.

12.4 Further reading

To learn more about transitioning to quantum-safe cryptography, we strongly recommend the *PQC Migration Handbook* written by the Dutch secret service AIVD and research organisations TNO and CWI.

An extension to the handbook is the *PQChoiceAssistant*, a tool that recommends what cryptographic algorithms are best used in specific situations.

In 2022, the NSA published requirements for national security systems. They indicate a timeline with concrete deadlines between 2025 and 2033.

12.5 Note

1. To illustrate, the *PQC Migration Handbook* mentions that: 'Judging from previous migrations this process might take well over five years'. The NSA's requirements for national security systems, published in 2022, demand that quantum-safe algorithms be exclusively used from 2033 onwards. NIST has indicated that quantum-unsafe standards will be deprecated in 2030 and will be disallowed around 2035.

Part 4

The final bits

13 Further reading

Below, we give a selection of recommended sources to learn more about this fascinating topic.

13.1 I want to learn the technical details

For (older) high school students (or those who followed high-school level mathematics):

Quantum Quest [Book/website] is an intensive five-week online course about the theory (mathematics) of quantum computing. Materials are freely available for self-study.

Quantum in Pictures (Cooke) [Book] teaches the theory (mathematics) of quantum computing using diagrams.

Undergraduate (Bachelor's) university level:

Quantum.Country [Website] – the 'Duolingo of Quantum Computing', a very well-written introduction for those with a late high-school or early university-level math background.

Quantum Computation and Quantum Information (Nielsen, Chuang) [Book] – the 'bible of quantum computing'. Perhaps not the most up-to-date, but definitely the most well-known resource in our field. Sets the standards for jargon and notation.

Quantum Computer Science: An Introduction (Mermin) [Book] – a well-written introduction, with quite some focus on manipulating quantum circuits.

Quantum Computing Since Democritus (Aaronson) [Book] – Aaronson is an authority in the field. His book touches upon many topics, such as the foundations of computer science, black holes and consciousness, making it a good read for those looking for something much broader than just quantum computing.

Graduate (Master's) level:

These assume no prior knowledge about quantum physics but require a strong background in mathematics (i.e. linear algebra, calculus, advanced inequality bounds and approximations, etc.). In exchange, they go into much more detail.

Lecture Notes for UvA course 'Quantum Computing' by Ronald de Wolf, which is frequently updated and features some cutting-edge algorithms. Via the course website, you can find the link and password to view all the recorded lectures.

Lecture Notes for Caltech course 'Quantum Computing' by John Preskil

Scientific overview papers

The papers below are aimed at scientists from fields other than quantum computing itself. All papers we mention are open-access and peer-reviewed, making them very suitable for citation.

'Quantum Algorithms: An Overview' (Ashley Montanaro)

'The Potential Impact of Quantum Computers on Society' (Ronald de Wolf), also available as recorded lecture.

Scientific opinions and discussions

Scott Aaronson's blog. Although written from a theoretical computer science perspective, this blog addresses a very broad range of quantum computing topics. Prof. Aaronson has a strong authority in the field, and his posts attract readership and comments from a broad range of prominent scientists.

13.2 I want to learn to program a quantum computer

Several programming packages for quantum computers exist, mostly maintained by major hardware providers. All of them offer great introductory tutorials. The ones we recommend below are all in Python:


Qiskit, the language by IBM, probably features the largest catalogue of learning materials. To start from scratch, we recommend following the 'Basics of Quantum Information', which teaches both the mathematics behind qubits and the usage of the package itself.


Cirq is a very similar package developed by Google. As of 2024, they have a more focused tutorial to explain the programming package itself without extensive theory of quantum mechanics.

QWorld Bronze offers tutorials in the form of Jupyter notebooks and hosts various training days around the world, mostly focused on Qiskit and sometimes ProjectQ.

PennyLane is a package by startup Xanadu with a strong focus on machine learning applications.

Classiq is one of the largest players that focuses on a higher-level programming language. This makes it easier to re-use code and to synthesise circuits for different types of hardware, but it also requires more background knowledge to get started.

13.3 I want to stay up to date with the latest developments

Major business conferences

Q2B (organised by QCWare)

IQT (Inside Quantum Technology)

Quantum.Tech (organised by Alpha Events)

Commercialising Quantum (organised by The Economist)

Major scientific conferences

The following are very technical and only recommended for those acquainted with the field. They take place at a different location each year.

Quantum Information Processing (QIP)

Theory of Quantum Computation, Communication and Cryptography (TQC)

Quantum Computing Theory in Practice (QCTIP) (mostly based in the UK)

Business News

Quantum Computing Report - don't be fooled by the basic look on the website. The content is written with a very critical eye and with very relevant contextual information, making it our favourite source for quantum-related news.

The Quantum Insider

Scientific news

The sources below do not focus exclusively on Quantum Technology, but offer high-quality scientific news (and surely none would miss any important quantum breakthroughs).

Quanta Magazine

Phys.org

13.4 I want to learn more about business implications

Several sources cover similar topics as this book. Most of these come from consultants of hardware providers who have a financial interest in making

others get started with quantum. In our opinion, the articles are sometimes too optimistic and predict that quantum applications will come much sooner than the typical expert would anticipate. On the other hand, they collect insightful details about financial matters.

McKinsey publishes yearly 'Quantum Technology Monitor' reports, focusing on the economic impact that quantum computers will have.

Cloudflare's support pages contain an incredibly complete bible of Post-Quantum Cryptography

Are you looking for a much more extensive source that covers pretty much everything there is to know about quantum computers? French consultant Olivier Ezratty regularly updates a 1500+ page book, Understanding Quantum Technologies.

Workshops and trainings

Short workshops will likely cover content similar to this book. A one-afternoon training can be particularly useful to inspire your colleagues and friends.

The Workshop General Awareness Quantum Computing follows the same philosophy as this book: an introduction to business opportunities that should be understandable for everyone.

Qureca is a British startup that offers several trainings, such as 'Quantum for Everyone' and 'Quantum Training for Business'.

14 Overview of quantum computers available today

This list shows a selection of the larger quantum computers as of August 2024, based on publicly available sources. The list is not exhaustive, there are many other systems that are not mentioned here.

Company	**#Qubits** **+ chip name**	**Platform** + notes
IBM	**1121** **"Condor"** **127** **"Eagle"**	**Superconducting** + Fast + Precise gates – Limited connectivity
Rigetti	**79** **"Aspen-M-3"** **83** **"Ankaa-2"**	
Google	**105** **"Sycamore"**	
University of Science and Technology of China, Hefei	**66**	
IQM	**20** **"Garnet"**	
PsiQuantum	**0**	**Photonic** + Fast – Imprecise – Different formalism
Quix	**20 modes**	
University of Science and Technology of China, Hefei	**100 modes,** **50 photons,** (equivalent to roughly 90 limited qubits).	
IonQ	**36** **"Forte"**	**Trapped ions** + Connectivity + Precise – Slow operations
Quantinuum	**32** **"H1-2"**	
	56 **"H2-1"**	
Alpine Quantum Technologies	**24**	
Pasqal	**100 as computer** **196 as simulator**	**Cold atoms** + Connectivity – Slow operations
QuEra	**280**	
D-Wave	**5000**	D-Wave's Quantum Annealers use **superconducting** qubits which specialize in a single algorithm: **annealing**.

15 Quantum Hype Bingo

'Unprecedented capabilities'	'Our algorithm solves …' (without comparison to classical computers)	'Future-proof your business'	Straightforwardly solving generic (NP-)hard optimisation problems
'Harness the commercial potential'	'Game-changing'	Trying all solutions at once	'Transformative'
'Unhackable'	Solving climate change	'The next frontier'	'X times faster' (without fair benchmark)
Quantum parallelism	Quantum computers will replace classical computers	Get quantum-ready	Enable artificial general intelligence (AGI)

16 Acknowledgements

Writing this book was a process of multiple years, during which I relied on the expertise of many others to bring this book to life. Through this section, I'd like to thank everyone who helped and supported me.

The initial content was born through a blog series I started with Joran van Apeldoorn. I want to thank him for fruitful brainstorming sessions and for helping to establish a clear set of messages for a well-defined audience – a crucial first step in crafting a meaningful story.

I am also indebted to many other friends, colleagues, and peers in the field of quantum computing who have helped me with advice, feedback, and insights. I want to thank Dimitri van Esch and Christian Schaffner for their broad tips and feedback, especially for helping me grasp many subtle details about cybersecurity. I'm grateful to Leonie Cazemier, Jonas Helsen, Victor Land, and Mischa Vos for their proofreading and helpful feedback. Also, many thanks go to Seenivasan Hariharan for extensive feedback and invaluable guidance on the topic of chemistry and material science. Moreover, I want to express my appreciation to Craig Gidney and Kareljan Schoutens for helping me with many broader insights and discussions, and to many people at QuSoft for insightful exchanges over the past years. Last but not least, a special thanks to Ronald de Wolf for proofreading the entire book and for sharply pointing out several mistakes and nuances.

I am grateful for the support of various organisations I work with, including the University of Amsterdam, Centrum Wiskunde & Informatica, QuSoft, Quantum.Amsterdam, and Quantum Delta NL, which made the publication of this book possible. Also, thanks to QuTech for allowing me to use the beautiful pictures of their labs.

17 Bibliography

Aaronson, S. (2015) 'Read the fine print', *Nature Physics*, 11(4), pp. 291–293. Available at: https://doi.org/10.1038/nphys3272.

Abbas, A. *et al.* (2024) 'Quantum Optimization: Potential, Challenges, and the Path Forward'. arXiv. Available at: https://doi.org/10.48550/arXiv.2312.02279.

Arute, F. *et al.* (2019) 'Quantum supremacy using a programmable superconducting processor', *Nature*, 574(7779), pp. 505–510. Available at: https://doi.org/10.1038/s41586-019-1666-5.

Babbush, R. *et al.* (2021) 'Focus beyond Quadratic Speedups for Error-Corrected Quantum Advantage', *PRX Quantum*, 2(1), p. 010103. Available at: https://doi.org/10.1103/PRXQuantum.2.010103.

Baker, B. (2023) *IBM Details Road to 100,000 Qubits by 2033*, *IoT World Today*. Available at: https://www.iotworldtoday.com/industry/ibm-details-road-to-100-000-qubits-by-2033 (Accessed: 26 September 2024).

Bauer, B. *et al.* (2020) 'Quantum Algorithms for Quantum Chemistry and Quantum Materials Science', *Chemical Reviews*, 120(22), pp. 12685–12717. Available at: https://doi.org/10.1021/acs.chemrev.9b00829.

Begušić, T. and Chan, G.K.-L. (2023) 'Fast classical simulation of evidence for the utility of quantum computing before fault tolerance'. arXiv. Available at: https://doi.org/10.48550/arXiv.2306.16372.

Bermejo, P. *et al.* (2024) *Quantum Convolutional Neural Networks are (Effectively) Classically Simulable*, *arXiv.org*. Available at: https://arxiv.org/abs/2408.12739v1 (Accessed: 7 September 2024).

Beverland, M.E. *et al.* (2022) 'Assessing requirements to scale to practical quantum advantage'. arXiv. Available at: https://doi.org/10.48550/arXiv.2211.07629.

Bobier, J.-F. *et al.* (2024) *The Long-Term Forecast for Quantum Computing Still Looks Bright*, *BCG Global*. Available at: https://www.bcg.com/publications/2024/long-term-forecast-for-quantum-computing-still-looks-bright (Accessed: 26 September 2024).

von Burg, V. *et al.* (2021) 'Quantum computing enhanced computational catalysis', *Physical Review Research*, 3(3), p. 033055. Available at: https://doi.org/10.1103/PhysRevResearch.3.033055.

Cao, Y. *et al.* (2019) 'Quantum Chemistry in the Age of Quantum Computing', *Chemical Reviews*, 119(19), pp. 10856–10915. Available at: https://doi.org/10.1021/acs.chemrev.8b00803.

Chan, G. (2022) 'Is There Evidence of Exponential Quantum Advantage in Quantum Chemistry?' Berkeley Quantum Colloquium, 12 April. Available at: https://www.youtube.com/watch?v=DZPH7ENcRLU (Accessed: 19 September 2024).

Chan, G.K.-L. (2024) 'Quantum chemistry, classical heuristics, and quantum advantage'. arXiv. Available at: https://doi.org/10.48550/arXiv.2407.11235.

Chapman, P. (2020) 'Scaling IonQ's Quantum Computers: The Roadmap', *IonQ*, 9 December. Available at: https://ionq.com/posts/december-09-2020-scaling-quantum-computer-roadmap (Accessed: 26 September 2024).

Cherrat, E.A. *et al.* (2023) 'Quantum Deep Hedging', *Quantum*, 7, p. 1191. Available at: https://doi.org/10.22331/q-2023-11-29-1191.

Choi, C.Q. (2022) *How Quantum Computers Can Make Batteries Better*, *IEEE Spectrum*. Available at: https://spectrum.ieee.org/lithium-air-battery-quantum-computing (Accessed: 28 August 2024).

Cookson, C. (2021) 'PsiQuantum expects commercial quantum computer by 2025', 13 March. Available at: https://www.ft.com/content/a5af3039-abbf-4b25-92e2-c40e5957c8cd (Accessed: 26 September 2024).

Cooper, P. *et al.* (2022) *Quantum computing just might save the planet*, *McKinsey*. Available at: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/quantum-computing-just-might-save-the-planet (Accessed: 23 September 2024).

Daley, A.J. *et al.* (2022) 'Practical quantum advantage in quantum simulation', *Nature*, 607(7920), pp. 667–676. Available at: https://doi.org/10.1038/s41586-022-04940-6.

Das Sarma, S. (2022) 'Quantum computing has a hype problem'. Available at: https://www.technologyreview.com/2022/03/28/1048355/quantum-computing-has-a-hype-problem/ (Accessed: 26 September 2024).

DiAdamo, S. *et al.* (2022) 'Practical Quantum K-Means Clustering: Performance Analysis and Applications in Energy Grid Classification', *IEEE Transactions on Quantum Engineering*, 3, pp. 1–16. Available at: https://doi.org/10.1109/TQE.2022.3185505.

Dunhill, J. (2021) *Chinese Scientists Create Quantum Processor 60,000 Times Faster Than Current Supercomputers*, *IFLScience*. Available at: https://www.iflscience.com/chinese-scientists-create-quantum-processor-60000-times-faster-than-current-supercomputers-61475 (Accessed: 29 September 2024).

Feynman, R.P. (1982) 'Simulating physics with computers', *International Journal of Theoretical Physics*, 21(6), pp. 467–488. Available at: https://doi.org/10.1007/BF02650179.

Finke, D. (2020) 'Google Goal: Build an Error Corrected Computer with 1 Million Physical Qubits by the End of the Decade', *Quantum Computing Report*, 5 September. Available at: https://quantumcomputingreport.com/google-goal-error-corrected-computer-with-1-million-physical-qubits-by-the-end-of-the-decade/ (Accessed: 26 September 2024).

Finke, D. (2024) 'PsiQuantum Receives $940 Million AUD ($620M USD) to Install a 1 Million Qubit Machine in Australia by 2027', *Quantum Computing Report*, 30 April. Available at: https://quantumcomputingreport.com/psiquantum-receives-940-million-aud-620m-usd-to-install-a-1-million-qubit-machine-in-australia-by-2027/ (Accessed: 26 September 2024).

Gemelke, N. and Lukin, A. (2022) *Hamiltonian simulation on QuEra's 256-qubit Aquila machine*, *QuEra*. Available at: https://www.quera.com/events/hamiltonian-simulation-on-queras-256-qubit-aquila-machine (Accessed: 10 September 2024).

Gidney, C. and Ekerå, M. (2021) 'How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits', *Quantum*, 5, p. 433. Available at: https://doi.org/10.22331/q-2021-04-15-433.

Goings, J.J. *et al.* (2022) 'Reliably assessing the electronic structure of cytochrome P450 on today's classical computers and tomorrow's quantum computers', *Proceedings of the National Academy of Sciences*, 119(38), p. e2203533119. Available at: https://doi.org/10.1073/pnas.2203533119.

Goodin, D. (2022) *Post-quantum encryption contender is taken out by single-core PC and 1 hour*, *Ars Technica*. Available at: https://arstechnica.com/information-technology/2022/08/sike-once-a-post-quantum-encryption-contender-is-koed-in-nist-smackdown/ (Accessed: 26 September 2024).

Hackett, R. (2020) *IBM plans a huge leap in superfast quantum computing by 2023*, *Fortune*. Available at: https://fortune.com/2020/09/15/ibm-quantum-computer-1-million-qubits-by-2030/ (Accessed: 26 September 2024).

Hariharan, S., Kinge, S. and Visscher, L. (2024) 'Modelling Heterogeneous Catalysis using Quantum Computers: An academic and industry perspective'. ChemRxiv. Available at: https://doi.org/10.26434/chemrxiv-2024-d2l1k-v2.

Herman, D. *et al.* (2023) 'Quantum computing for finance', *Nature Reviews Physics*, 5(8), pp. 450–465. Available at: https://doi.org/10.1038/s42254-023-00603-1.

How and when do we need to act on climate change? (no date) *Imperial College London*. Available at: https://www.imperial.ac.uk/grantham/publications/climate-change-faqs/how-and-when-do-we-need-to-act-on-climate-change-/ (Accessed: 23 September 2024).

How Quantum Computers Break The Internet... Starting Now (2023). Available at: https://www.youtube.com/watch?v=-UrdExQWocs (Accessed: 20 September 2024).

Hutchins, M. (2023) *Quantum physics, supercomputers, and solar cell efficiency*, *pv magazine International*. Available at: https://www.pv-magazine.com/2023/08/04/quantum-physics-supercomputers-and-solar-cell-efficiency/ (Accessed: 28 August 2024).

Jordan, S. (2024) *Quantum Algorithm Zoo*. Available at: https://quantumalgorithmzoo.org/ (Accessed: 27 September 2024).

Kao, P.-Y. *et al.* (2023) 'Exploring the Advantages of Quantum Generative Adversarial Networks in Generative Chemistry', *Journal of Chemical Information and Modeling*, 63(11), pp. 3307–3318. Available at: https://doi.org/10.1021/acs.jcim.3c00562.

Kim, Y. *et al.* (2023) 'Evidence for the utility of quantum computing before fault tolerance', *Nature*, 618(7965), pp. 500–505. Available at: https://doi.org/10.1038/s41586-023-06096-3.

Lee, J. *et al.* (2021) 'Even More Efficient Quantum Computations of Chemistry Through Tensor Hypercontraction', *PRX Quantum*, 2(3), p. 030305. Available at: https://doi.org/10.1103/PRXQuantum.2.030305.

Lee, S. *et al.* (2023) 'Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry', *Nature Communications*, 14(1), p. 1952. Available at: https://doi.org/10.1038/s41467-023-37587-6.

Leijnse, K. (2024) 'Photocatalysis for Water Splitting', *Quantum Application Lab*, 8 January. Available at: https://quantumapplicationlab.com/2024/01/08/photocatalysis-for-water-splitting/ (Accessed: 28 August 2024).

Liu, Y., Arunachalam, S. and Temme, K. (2021) 'A rigorous and robust quantum speed-up in supervised machine learning', *Nature Physics*, 17(9), pp. 1013–1017. Available at: https://doi.org/10.1038/s41567-021-01287-z.

Lusnig, L. *et al.* (2024) 'Hybrid Quantum Image Classification and Federated Learning for Hepatic Steatosis Diagnosis', *Diagnostics*, 14(5), p. 558. Available at: https://doi.org/10.3390/diagnostics14050558.

Matt Langione *et al.* (2023) *Quantum Computing Is Becoming Business Ready*, *BCG Global*. Available at: https://www.bcg.com/publications/2023/enterprise-grade-quantum-computing-almost-ready (Accessed: 26 September 2024).

Matuschak, A. and Nielsen, M. (2019) 'Quantum Country'. Available at: https://quantum.country (Accessed: 23 September 2024).

McKinsey Digital (2024) 'Quantum Technology Monitor'. Available at: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/steady-progress-in-approaching-the-quantum-advantage (Accessed: 26 September 2024).

Montanaro, A. (2016) 'Quantum algorithms: an overview', *npj Quantum Information*, 2(1), pp. 1–8. Available at: https://doi.org/10.1038/npjqi.2015.23.

Mosca, M. and Piani, M. (2023) *Quantum Threat Timeline Report 2023*. Available at: https://globalriskinstitute.org/publication/2023-quantum-threat-timeline-report/.

Mounier, E. (2024) *Slowly but Surely, Quantum Computing Will Transform Industry*, *EE Times Europe*. Available at: https://www.eetimes.eu/slowly-but-surely-quantum-computing-will-transform-industry-and-society/ (Accessed: 19 September 2024).

Newton, W. (2023) 'Quantum medicine: how quantum computers could change drug development', *Clinical Trials Arena*, 24 February. Available at: https://www.clinicaltrialsarena.com/features/quantum-computers-drug-development/ (Accessed: 24 September 2024).

Quantinuum accelerates the path to Universal Fully Fault-Tolerant Quantum Computing (2024) *Quantinuum*. Available at: https://www.quantinuum.com/blog/quantinuum-accelerates-the-path-to-universal-fault-tolerant-quantum-computing-supports-microsofts-ai-and-quantum-powered-compute-platform-and-the-path-to-a-quantum-supercomputer (Accessed: 26 September 2024).

Robert Davis (2023) 'How Ewin Tang's Dequantized Algorithms are Helping Quantum Algorithm Researchers', *Qiskit*, 15 March. Available at: https://medium.com/qiskit/how-ewin-tangs-dequantized-algorithms-are-helping-quantum-algorithm-researchers-3821d3e29c65 (Accessed: 26 September 2024).

Santagati, R. *et al.* (2024) 'Drug design on quantum computers', *Nature Physics*, 20(4), pp. 549–557. Available at: https://doi.org/10.1038/s41567-024-02411-5.

Saran, C. (2018) *Microsoft predicts five-year wait for quantum computing in Azure*, *ComputerWeekly.com*. Available at: https://www.computerweekly.com/news/252440763/Microsoft-predicts-five-year-wait-for-quantum-computing-in-Azure (Accessed: 26 September 2024).

Scott Aaronson (2024) 'Quantum Computing: Between Hope and Hype', *Shtetl-Optimized*, 22 September. Available at: https://scottaaronson.blog/?p=8329 (Accessed: 30 September 2024).

Shieber, J. (2018) *The reality of quantum computing could be just three years away, TechCrunch*. Available at: https://techcrunch.com/2018/09/07/the-reality-of-quantum-computing-could-be-just-three-years-away/ (Accessed: 26 September 2024).

Tindall, J. *et al.* (2024) 'Efficient Tensor Network Simulation of IBM's Eagle Kicked Ising Experiment', *PRX Quantum*, 5(1), p. 010308. Available at: https://doi.org/10.1103/PRXQuantum.5.010308.

Wang, B. (2020) 'PsiQuantum Targets Million Silicon Photonic Qubits by 2025', 23 April. Available at: https://www.nextbigfuture.com/2020/04/psiquantum-targets-million-silicon-photonic-qubits-by-2025.html (Accessed: 26 September 2024).

What will million-qubit computers look like in a few years? (2022) *ICV TAnK-icv*. Available at: https://www.icvtank.com/newsinfo/629365.html (Accessed: 26 September 2024).

de Wolf, R. (2017) 'The potential impact of quantum computers on society', *Ethics and Information Technology*, 19(4), pp. 271–276. Available at: https://doi.org/10.1007/s10676-017-9439-z.

Wyciślik-Wilson, S.E. (2019) *Google creates quantum chip millions of times faster than the fastest supercomputer, TechRadar*. Available at: https://www.techradar.com/news/google-creates-quantum-chip-millions-of-times-faster-than-the-fastest-supercomputer (Accessed: 29 September 2024).

Yoram Avidan *et al.* (2024) *Citi and Classiq advance quantum solutions for portfolio optimization using Amazon Braket | AWS Quantum Technologies Blog*. Available at: https://aws.amazon.com/blogs/quantum-computing/citi-and-classiq-advance-quantum-solutions-for-portfolio-optimization/ (Accessed: 24 September 2024).

Yung, M.-H. *et al.* (2014) 'Introduction to Quantum Algorithms for Physics and Chemistry', in *Quantum Information and Computation for Chemistry*. John Wiley & Sons, Ltd, pp. 67–106. Available at: https://doi.org/10.1002/9781118742631.ch03.

Zhong, H.-S. *et al.* (2020) 'Quantum computational advantage using photons', *Science*, 370(6523), pp. 1460–1463. Available at: https://doi.org/10.1126/science.abe8770.

18 Index

How will businesses use quantum technology in the future? What problems will a quantum computer solve? How long will it take before these devices become commercially relevant?

With the first generation of quantum computers on the horizon, understanding their impact is more relevant than ever. Luckily, you don't need a physics degree to understand the functionality of these computers – just like you don't need to know how a transistor works to excel in conventional IT.

This book is the perfect introduction to the opportunities and threats of quantum technologies. It equips you with the necessary knowledge to join cutting-edge discussions and make strategic decisions.

KOEN GROENLAND is a theoretical physicist with a PhD in the near-term applications of quantum computers. He works as an innovation officer at the University of Amsterdam, where he is responsible for setting up research collaborations and developing lifelong learning education for professionals. He is one of the driving forces behind Quantum. Amsterdam, the innovation hub that drives the commercialisation of quantum technologies around the Dutch capital.

"Easy to read and full of insights, a must-read for anyone looking to understand the real-world impact of quantum computing."
– Diederick Croese, Director of Center for Quantum and Society

"This book offers a well-rounded, scientifically accurate overview of quantum technology, highlighting its significant potential for innovation." – Christian Schaffner, Professor in Theoretical Computer Science, Director of QuSoft

AUP.nl

9 789048 568987