Introduction

Introduction

Welcome to *Quantum Computing in Practice* — a course that focuses on today's quantum computers and how to use them to their full potential. It covers realistic potential use cases for quantum computing and best practices for running and experimenting with quantum processors having 100 or more qubits.

Quantum utility

It's an exciting time for quantum computing. After many years of theoretical and experimental research and development, we're approaching a point at which quantum computers can begin to compete with classical computers and demonstrate *utility*.

Utility is not the same thing as *quantum advantage*, which refers to quantum computers *outperforming* classical computers for meaningful tasks. Classical computers have incredible power and adaptability, and quantum computers can't beat them yet. We've seen decades of advancements in classical computation — not only in computing hardware but also in algorithms for classical computers — and we can observe with clarity that electronic digital computing has radically changed our world.

Quantum computing, on the other hand, is at a different stage in its development. Quantum computing places extreme demands on our control of quantum mechanical systems and pushes the boundaries of today's technology — and we cannot realistically expect to master this new technology and beat classical computing immediately. But we are seeing suggestive signs that quantum computers are starting to compete with classical computing methods for selected tasks, which is a natural step in the technological evolution of quantum computing known as *quantum utility*.

As the technology advances and new quantum computing methods are developed, we can reasonably expect that its advantages will become increasingly pronounced — but this will take time. As this happens we'll likely see a back-and-forth interaction with classical computing: quantum computing demonstrations will be performed and classical computing will respond, quantum computing will take another turn, and the pattern will repeat. And one day, when a quantum computer's performance can't be matched classically, we'll hypothesize that we've seen a quantum advantage — but even then we won't know for sure! Proving impossibility results for classical computers is itself an impossibly difficult problem as far as we know.

Simulating Nature

Classical simulators — computer programs running on classical computers that simulate physical systems — can make predictions about quantum mechanical systems. But classical simulators are not quantum and cannot directly emulate quantum systems. Instead, they use mathematical calculations to approximate quantum behavior. As the sizes of the simulated systems grow the overhead required to do this increases dramatically, placing limits on which quantum systems can be simulated classically, how long the simulations take, and the accuracy of the results.

Quantum computers, on the other hand, can emulate quantum systems more directly — and as a result the overhead they require scales significantly better as the system size grows. This, in fact, was Richard Feynman's idea in the 1980s that first motivated an investigation into the potential of quantum computers. We'll have more to say about this later!

IBM researchers published a paper in 2023 that showed, for the first time, that a quantum computer can compete with state-of-the-art classical techniques for simulating a particular physical model. Its results can still be matched by advanced techniques running on classical computers — but it bested brute-force algorithms, and it also offers a new data point to which different simulation methods can be compared.

A focus on larger quantum processors

Prior users of IBM quantum hardware may have noticed that the smaller processors we previously made available to the public have been taken offline, making way for larger processors (having 100+ qubits). Those smaller processors could easily be simulated classically. So, although they were stepping stones in an advancing technology, they could not possibly demonstrate quantum utility: anything that could be done with them could just as easily be done with a classical simulation.

At around 100 qubits, however, this is no longer the case; quantum processors of this size can no longer be simulated classically. This represents a phase-transition of sorts, into a new era of quantum computing technology where the *potential* for outperforming classical computation exists. This is where IBM has chosen to focus — to look for quantum computational power and reach toward an eventual quantum advantage.

We encourage our users to use these new devices to their full potential, to experiment with them and push their limits, and to carry forward lessons learned to the next generation of quantum processors currently in development. The purpose of this course is to enable you to do this!

Audience and course goals

This course is for anyone that aims to develop new applications for quantum computers, wants to scale up their current work in quantum computing, or learn how to use quantum processors within their workflow. This includes not just physicists and computer scientists, but also engineers, chemists, materials scientists, and anyone else interested in mastering quantum computing hardware.

The course will be hands-on and focused on the practical use of quantum computers. The following topics and skills are among those it covers:

- Running utility-scale jobs on quantum processors through Qiskit Runtime
- Using error mitigation techniques to improve hardware results
- Potential application areas for near-term quantum computers

This course does not cover the introductory theory of quantum computing and assumes a basic familiarity with qubits and quantum circuits. The *Basics of quantum information* course covers this material and is recommended first for those new to quantum computing.

Pre-course Survey

Before we begin, please take a moment to complete our pre-course survey, which is important to help improve our content offerings and user experience.

Quantum Computing in Practice: Pre-Course Survey

Taking the "Quantum Computing in Practice" course from IBM? Please fill out this short form about your experience and expectations.

Your insights will help us improve our IBM Quantum Learning offerings. Your responses will be kept confidential and will only be accessed by the IBM Quantum design research and learning teams. If you have questions about this survey or how the data will be used, please email quantum.design@us.ibm.com.

Powered by Qualtrics

The story of computation

Quantum computing is an exciting new technology in an early stage of development — but it's just one chapter in a story that goes back thousands of years. It's the story of *computation* and its multifaceted connections to the *physical world*.

Computing devices since ancient times

Since ancient times, we as humans have needed to perform *computations* — or, in other words, to process information according to certain rules and constraints — to enable communication, construction, commerce, science, and other aspects of our lives. We've looked to the physical world for assistance, and through ingenious discoveries we've constructed devices to help us compute.

Long ago, devices made from wood, bone, and knotted ropes stored information and facilitated calculations. Mechanical devices built from levers, gears, and other machinery advanced from early astronomical clocks, to calculators, to sophisticated computing devices such as differential analyzers that solved equations using wheels and rotating disks. Even the technology of writing has played an important part in this story by allowing *people* to perform computations they wouldn't be able to otherwise.

When we think about computers today, we tend to think about electronic digital computers. But this is in fact a fairly recent technology: electronic digital computers were first built in the 1940s. (In contrast, the Sumerian abacus is believed to have been invented somewhere between 2700 and 2300 BC.) The technology has advanced dramatically since then and computers are now ubiquitous. They're found in homes, places of work, and the vehicles that transport us between them, and many of us carry them with us wherever we go.

We also have *supercomputers*, which are large collections of powerful classical processors hooked up in parallel. They're among the best tools humankind has ever built for solving difficult problems, and their power and reliability continues to advance. But still, there are important computational problems that even these behemoths will never be able to solve, due to the inherent computational difficulty of these problems.

Connections to the physical world

Computers have many uses. One important use for computers is to *learn* about the physical world and better understand its patterns. Historical uses in this category have included the prediction of eclipses and tides, understanding the movement of astronomical bodies, and (in somewhat more recent times) modeling explosions. Today there's scarcely a physics lab in the world without a computer.

More generally, physics and computation have always been intertwined. Computation can't exist in a vacuum: information requires a medium, and to compute we need to harness the physical world in some regard. Rolf Landauer, a physicist (and IBMer), recognized decades ago that information is physical, existing only through a physical representation. *Landauer's principle* establishes a connection between information and the laws of thermodynamics, but in fact there are many connections.

The aim of physics is to understand the physical world, but it's a two-way street. Through this understanding, we're able to harness new technologies to help us compute, and through them we continue to learn about the physical world — essentially pulling physics and computational technology up by the bootstraps.

Moore's law

Moore’s law is an observation that the maximum number of transistors in an integrated circuit doubles approximately every two years. Over the past five decades or so, we've observed this trend and reaped its rewards. With more transistors on a chip we can perform more complex computations and we can do them faster. This is why computers have become more and more powerful over time.

However, Moore’s law is, by necessity, coming to an end. Experts disagree on when this will happen, and some argue that it already has. But we know for certain that it must inevitably end because there's a theoretical limit to the miniaturization of computing components. We can't make a transistor smaller than an atom! While that may sound exaggerated, this is the wall we're approaching.

The solution is not to give up and say, "Well, that’s as good as it gets." This goes against human nature. Instead, we must look to the physical world for new computational tools, which is where quantum computing comes into play.

Quantum computing

Quantum mechanics and computation

Quantum mechanics was discovered in the early 20th century, and it's already played an important role in computation. Indeed, our understanding of quantum mechanics has, in part, made modern day computers possible. Without quantum mechanics, for instance, it is difficult to imagine the solid-state hard drive having been invented.

Quantum computing in theory

When Richard Feynman first proposed the notion of a quantum computer in 1982, his focus was on simulating quantum mechanical systems. The calculations required to do this seemed too hard for ordinary computers — but perhaps, with a computer that operates according to a quantum mechanical description of the world, the systems could be emulated directly.

Today this is one of the most promising avenues for quantum computing. To the best of our understanding, Nature is not classical — it's quantum. And so, quantum computers may be valuable tools for understanding it. Classical computers, on the other hand, can only approximate what actually occurs in Nature, and in some cases those approximations are very limited.

One way to think about this is through an analogy to wind tunnels. Fluid dynamics are notoriously hard to simulate and predict mathematically. For example, it's too costly and impractical to simulate a car driving through wind, so instead car manufactures actually build tunnels with blowing wind and drive cars through them to test their performance. That is, they create wind rather than simulating it. Building a quantum computer to study the physical world is kind of like building a wind tunnel to study how wind affects cars. Quantum computers can directly emulate the laws of Nature on a molecular level because they act in accordance with those laws: they emulate Nature rather than simulating it through formulas and calculations.

Others followed up on Feynman's ideas — and they linked these ideas with a theory of *quantum information* that was already being developed. The field of *quantum information and computation* was born. It has since developed into a rich, multidisciplinary field of study, and numerous advantages of quantum over classical information and computation have been identified in a wide variety of *theoretical* settings involving communication, computation, and cryptography.

Quantum computing in practice

In practical terms, two things are needed to transfer these sorts of theoretical advantages to real world advantages: the devices themselves and the methodologies to unlock their potential.

Unlike classical computers, quantum computers aren't stashed away in anyone's back pocket. Until very recently, if you wanted to experiment with a quantum computer, you had to build and maintain one yourself (usually in a sad, basement lab in a university or research facility), and you would have just a few, very noisy qubits at most. No longer is this the case. In 2016, IBM put the first quantum processor on the cloud. It had only five qubits and fairly high rates of errors, but we've come a long way since then. We'll summarize the current state of the technology in a section below.

In addition to building quantum computers, we also need to develop methodologies for using them effectively. While theoretical advances in quantum algorithms and protocols suggest a strong potential, we still need to find practical uses for quantum computing. Today's quantum computers cannot yet perform the fault-tolerant computations required to transfer known theoretical advantages into practical advantages. But they are beyond the reach of classical computer simulations, and we can attempt to leverage this fact for computational power.

With these advances we find ourselves with a new tool for computation, and it's up to us to figure out what we can do with it.

Potential applications

We don't expect to use quantum computers to study how cars perform in wind. But there are other physical processes — such as ones involved in the design of batteries or in certain chemical reactions — where a quantum computer's ability to emulate Nature could lead to a quantum advantage. More generally, there are many problems that are too difficult or costly even for cutting-edge supercomputers, including problems that are highly relevant to our society. Quantum computing may not offer solutions to all of them, but it could offer solutions to some.

The following three application areas represent targets in the noisy quantum computing era, prior to the implementation of quantum error correction and fault-tolerance.

- Optimization
- Simulating Nature
- Finding structure in data (including machine learning)

We'll discuss these topics in greater detail later in the course.

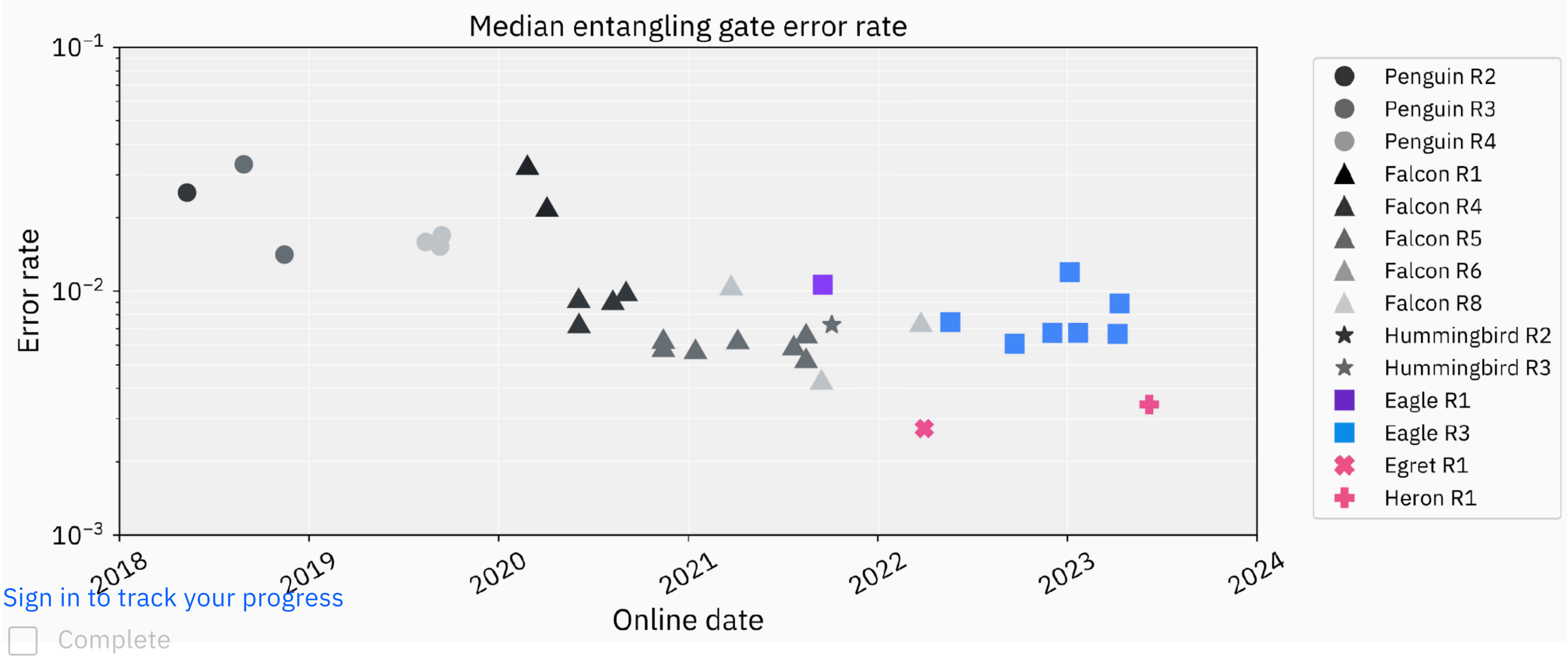
State of the technology

Building quantum computers is a difficult technological challenge, and it's only been eight years since small quantum computers have been publicly available. In those eight years, we have made progress on many fronts.

Numerous IBM quantum processors are now accessible through the cloud, all of which have over 100 qubits. But it isn’t just the size of the processors that is important — that's just one metric we care about. Gate quality has dramatically improved, and we've also introduced methods of reducing and mitigating errors intrinsic to quantum systems, even as we push forward to the creation of fault-tolerant systems. Three basic metrics — *scale*, *quality*, and *speed* — are vital to tracking performance improvement.

- **Size.** More qubits are obviously better, but only if increasing the number doesn't degrade performance (which can happen). We want more *good quality* qubits that don’t interfere with one another through crosstalk when we don’t want them to. The way the qubits are connected to one another is also important, and figuring out how best to do this represents a challenge for superconducting qubit circuits.

- **Quality.** Another important performance metric we track is two-qubit gate fidelity. Gates that run on single qubits are not as prone to errors as two-qubit gates, which are therefore the larger concern. (Two-qubit gates are also crucial because they're responsible for creating entanglement between the qubits, which helps give quantum computing its power.)
- **Speed.** Last is speed and efficiency. In short, the time spent running a program (including both quantum and classical parts) should be as short as possible.



Sign in to track your progress

□ Complete

Running Quantum Circuits