



ILLUSTRATION BY THE PROJECT TWINS

# HOW TO GET STARTED IN QUANTUM COMPUTING

Tempted to try your hand at a new technique? These tools will help you on your way. **By David Matthews**

**T**o the untrained eye, a circuit built with IBM's online Quantum Experience tool looks like something out of an introductory computer-science course. Logic gates, the building blocks of computation, are arrayed on a digital canvas, transforming inputs into outputs.

But this is a quantum circuit, and the gates modify not the usual binary 1 or 0 bits, but qubits, the fundamental unit of quantum computing. Unlike binary bits, qubits can exist as a 'superposition' of both 1 and 0, resolving one way or the other only when measured. Quantum computing also exploits properties such as entanglement, in which changing the state of one qubit also changes the state of another, even at a distance.

Those properties empower quantum computers to solve certain classes of problem more quickly than classical computers. Chemists could, for instance, use quantum computers to speed up the identification of new catalysts through modelling.

Yet that prospect remains a distant one. Even the fastest quantum computers today have no more than 100 qubits, and are plagued by random errors. In 2019, Google demonstrated that its 54-qubit quantum computer could solve in minutes a problem that would take a classical machine 10,000 years. But this 'quantum advantage' applied only to an extremely narrow situation. Peter Selinger, a mathematician and quantum-computing specialist at Dalhousie University in Halifax,

Canada, estimates that computers will need several thousand qubits before they can usefully model chemical systems.

"The stage of quantum computers now is something like classical computing in the late 1980s," says Sara Metwalli, a quantum-computing researcher at Keio University in Tokyo. "Most of the work done now is to prove that quantum, in the future, may have the ability to solve interesting problems."

## Fast-moving field

Still, progress is happening fast. IBM hopes to have a 1,000-qubit machine by 2023, and quantum-computing advocates enthuse that the field is ripe for development. For those who want to see what the fuss is about, a growing

collection of online tutorials, programming languages and simulators are making it easier than ever to dip their toes into quantum computing.

The digital logic underlying classical computers is well known:  $1 \text{ AND } 0 = 0$ , for instance. But quantum computers are much more fluid, and researchers must come to grips with how qubit states are expressed mathematically to understand how they behave. “Quantum computing is essentially matrix vector multiplication – it’s linear algebra underneath the hood,” says Krysta Svore, principal manager of the quantum-computing group at Microsoft Research in Redmond, Washington.

Several online guides build up from the basics. Physicist Michael Nielsen and software engineer Andy Matuschak, both based in San Francisco, California, have produced a walk-through resource called Quantum Computing for the Very Curious (see [go.nature.com/3qazj2p](http://go.nature.com/3qazj2p)). And IBM has created an interactive toolkit to accompany its Qiskit quantum language, with exercises that can be run in a Jupyter computational notebook.

Scientists also need to wrap their heads around quantum circuits, says Jeannette Garcia, senior manager for the quantum applications, algorithms and theory team at IBM Research in San Jose, California. Running from left to right and looking a bit like a musical stave, these circuits visually represent how qubits are transformed by logic gates – similar to the AND, OR and NOT gates from which electronic circuits are built – before being measured to reveal their state. IBM’s Quantum Experience allows users to drag and drop logic gates to create their own circuits in a web browser, and to run them remotely on a real quantum computer.

## Lingua quantum

From there, dedicated software frameworks and programming languages allow researchers to simulate, execute and explore the quantum circuits they design. Several of these languages were described in a 2020 review (B. Heim *et al. Nature Rev. Phys.* **2**, 709–722; 2020).

Microsoft, IBM and Google have all created tools – Q#, Qiskit and Cirq, respectively – that draw heavily on the Python programming language, and have built user-friendly development environments with ample documentation to help coders get started. Microsoft, for example, has created a full quantum development kit (QDK), containing code libraries, a debugger and a resource estimator, which checks in advance how many qubits an algorithm will require.

And it’s not just the technology giants that are involved. Rigetti Computing in Berkeley, California, which has its own 31-qubit machine, has released a quantum-software development kit called Forest, which includes a Python library called pyQuil. And

UK-based Cambridge Quantum Computing has launched tket, with the associated pytket library.

Another option is Silq, a language released last year by a team at the Swiss Federal Institute of Technology (ETH) in Zurich. One of its key advantages, says co-creator Benjamin Bichsel, involves ‘uncomputation’. The language automatically resets the temporary values used by a quantum program, rather than forcing programmers to do this tedious work manually.

Somewhat less user-friendly is Quipper. Unlike Python, Quipper is not an ‘imperative’ language – one in which the program details a series of steps that change the state of the software, says Selinger, who is one of Quipper’s creators. Rather, it is ‘functional’, more akin to a series of mathematical functions. “You never update anything, there are no variables,” Selinger says.

Although not immediately useful for current small-scale devices, Quipper’s functional nature could ultimately make it easier to mathematically verify that a quantum program is bug-free and actually solving the problem

## “The stage of quantum computers now is something like classical computing in the late 1980s.”

you want it to, Selinger says. But it also makes the language less accessible. “If you want a non-specialist, such as a chemist, to try quantum computing, then it is best to lower the threshold of entry and start with a programming language that most people are already familiar with,” says Selinger. He suggests Qiskit or one of the other imperative, Python-based languages.

Actual quantum computers are largely in the hands of private technology firms, who offer access to the hardware on a variety of terms.

IBM makes a five-qubit machine freely available, but to use the company’s more-powerful machines, research organizations need to be part of its Quantum Network, comprising universities, laboratories and companies. Although IBM doesn’t make its pricing public, it does give out ‘access awards’ to scientists who have a “cool research idea and want access to a device to try it out”, says Garcia. For instance, a team at the University of Chicago in Illinois, announced last November that it had used IBM’s machine to explore an ‘exciton condensate’, a highly electrically conductive quantum system (L. M. Sager *et al. Phys. Rev. Res.* **2**, 043205; 2020).

Microsoft offers access to other firms’ quantum computers through its new Azure Quantum platform. This is at a free ‘limited preview’ stage, says Svore, and research

institutions can apply to become early adopters.

Google doesn’t sell access to its quantum machines. But Markus Hoffmann, who heads its quantum-computing partnerships and programs team, says that any scientist with a strong proposal for an experiment that could be deployed on Google’s hardware should get in touch. “Based on the research impact in the field, we will find a way to make that experiment happen,” says Hoffmann, who is based in Munich, Germany.

Ashley Montanaro, a quantum-computing researcher at the University of Bristol, UK, runs his quantum programs through Amazon Web Services, a cloud-computing platform that plugs into other firms’ quantum devices. It costs him around US\$1 to test one quantum circuit, but because researchers might want to test thousands of such circuits, “the cost can rack up”, he cautions.

## Start with simulations

Curious scientists can also experiment with an emulator that simulates a quantum computer on a classical machine. Microsoft’s QDK, for example, has a built-in emulator that can simulate a 30-qubit device on a laptop.

“I would suggest to anyone: start on an emulator,” says Thomas O’Brien, European quantum algorithms and applications lead at Google’s Quantum AI research team, who is based in Munich. “[An] emulator is much more predictable. It allows you to actually see the quantum states,” he says. Inspecting the state of a real quantum computer just causes it to collapse, making troubleshooting difficult, he says. And stray background heat or magnetic fields can easily knock qubits out of their existing state.

But scientists should still run their programs on a real quantum computer if they can, Montanaro advises, to get used to their noisy, error-prone behaviour. “It just tells you things that you just don’t get from emulation,” he says.

As research advances and quantum devices improve, such headaches will diminish. But even then, quantum computers are unlikely to replace their classical counterparts. Instead, they will sit embedded within a larger classical architecture, crunching those problems for which they provide an exponential speed-up.

Researchers still need to home in on which problems those are, but the search is on. “This is really the big question, and I think the only way to answer it is through exploration,” says Eric Johnston, co-author of *Programming Quantum Computers* (2019), who is based in Boston, Massachusetts. “If you’re a scientist who knows some classical computing, there is so much unexplored terrain in quantum computation that you’ll never be bored.”

**David Matthews** is a freelance writer based in Berlin.