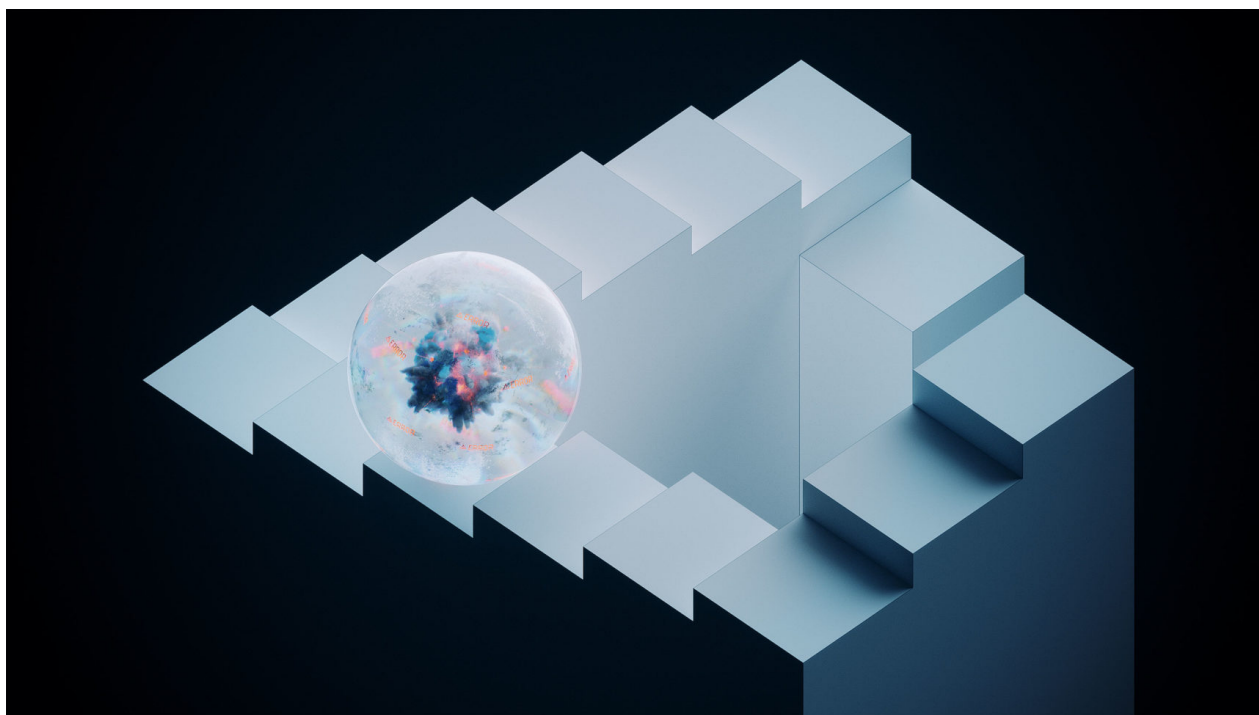




Quantum Algorithms Struggle Against Old Foe: Clever Computers

The quest for "quantum supremacy" – unambiguous proof that a quantum computer does something faster than an ordinary computer – has paradoxically led to a boom in quasi-quantum classical algorithms.

By Ariel Bleicher



[Josef Bsharah](#) for Quanta Magazine

A popular misconception is that the potential — and the limits — of quantum computing must come from hardware. In the digital age, we've gotten used to marking advances in clock speed and memory. Likewise, the 50-qubit quantum machines now coming online from the likes of Intel and IBM have inspired predictions that [we are nearing "quantum supremacy"](#) — a nebulous frontier where quantum computers begin to do things beyond the ability of classical machines.

But quantum supremacy is not a single, sweeping victory to be sought — a broad Rubicon to be crossed — but rather a drawn-out series of small duels. It will be established problem by problem, quantum algorithm versus classical algorithm. “With quantum computers, progress is not just about speed,” said [Michael Bremner](#), a quantum theorist at the University of Technology Sydney. “It’s much more about the intricacy of the algorithms at play.”

Paradoxically, reports of powerful quantum computations are motivating improvements to classical ones, making it harder for quantum machines to gain an advantage. “Most of the time when people talk about quantum computing, classical computing is dismissed, like something that is past its prime,” said [Cristian Calude](#), a mathematician and computer scientist at the University of Auckland in New Zealand. “But that is not the case. This is an ongoing competition.”

And the goalposts are shifting. “When it comes to saying where the supremacy threshold is, it depends on how good the best classical algorithms are,” said [John Preskill](#), a theoretical physicist at the California Institute of Technology. “As they get better, we have to move that boundary.”

‘It Doesn’t Look So Easy’

Before the dream of a quantum computer took shape in the 1980s, most computer scientists took for granted that classical computing was all there was. The field’s pioneers had convincingly argued that classical computers — epitomized by the mathematical abstraction known as a Turing machine — should be able to compute everything that is computable in the physical universe, from basic arithmetic to stock trades to black hole collisions.

Classical machines couldn’t necessarily do all these computations efficiently, though. Let’s say you wanted to understand something like the chemical behavior of a molecule. This behavior depends on the behavior of the electrons in the molecule, which exist in a superposition of many classical states. Making things messier, the quantum state of each electron depends on the states of all the others — due to the quantum-mechanical phenomenon known as entanglement. Classically calculating these entangled states in even very simple molecules can become a nightmare of exponentially increasing complexity.

A quantum computer, by contrast, can deal with the intertwined fates of the electrons under study by superposing and entangling its own quantum bits. This enables the computer to process extraordinary amounts of information. Each single qubit you add doubles the states the system can simultaneously store: Two qubits can store four states, three qubits can store eight states, and so on. Thus, you might need just 50 entangled qubits to model quantum states that would require exponentially many classical bits — 1.125 quadrillion to be exact — to encode.

A quantum machine could therefore make the classically intractable problem of simulating large quantum-mechanical systems tractable, or so it appeared. “Nature isn’t classical, dammit, and if you want to make a simulation of nature, you’d better make it quantum mechanical,” the physicist Richard Feynman famously quipped in 1981. “And by golly it’s a wonderful problem, because it doesn’t look so easy.”

It wasn’t, of course.

Even before anyone began tinkering with quantum hardware, theorists struggled to come up with suitable software. Early on, Feynman and [David Deutsch](#), a physicist at the University of Oxford, learned that they could control quantum information with mathematical operations borrowed from linear algebra, which they called gates. As analogues to classical logic gates, quantum gates manipulate qubits in all sorts of ways — guiding them into a succession of superpositions and

entanglements and then measuring their output. By mixing and matching gates to form circuits, the theorists could easily assemble quantum algorithms.



[Tamiko Thiel](#)

Richard Feynman, the physicist who came up with the idea for a quantum computer in the 1980s, quipped that “by golly, it’s a wonderful problem, because it doesn’t look so easy.”

Conceiving algorithms that promised clear computational benefits proved more difficult. By the early 2000s, mathematicians had come up with only a few good candidates. Most famously, in 1994, a young staffer at Bell Laboratories named [Peter Shor](#) proposed [a quantum algorithm](#) that factors integers exponentially faster than any known classical algorithm — an efficiency that could allow it to crack many popular encryption schemes. Two years later, Shor’s Bell Labs colleague Lov Grover devised [an algorithm](#) that speeds up the classically tedious process of searching through unsorted databases. “There were a variety of examples that indicated quantum computing power should be greater than classical,” said [Richard Jozsa](#), a quantum information scientist at the University of Cambridge.

But Jozsa, along with other researchers, would also discover a variety of examples that indicated just

the opposite. “It turns out that many beautiful quantum processes look like they should be complicated” and therefore hard to simulate on a classical computer, Jozsa said. “But with clever, subtle mathematical techniques, you can figure out what they will do.” He and his colleagues found that they could use these techniques to efficiently simulate — or “de-quantize,” as Calude would say — a surprising number of quantum circuits. For instance, circuits that omit entanglement fall into this trap, as do those that entangle only a limited number of qubits or use only certain kinds of entangling gates.

What, then, guarantees that an algorithm like Shor’s is uniquely powerful? “That’s very much an open question,” Jozsa said. “We never really succeeded in understanding why some [algorithms] are easy to simulate classically and others are not. Clearly entanglement is important, but it’s not the end of the story.” Experts began to wonder whether many of the quantum algorithms that they believed were superior might turn out to be only ordinary.

Sampling Struggle

Until recently, the pursuit of quantum power was largely an abstract one. “We weren’t really concerned with implementing our algorithms because nobody believed that in the reasonable future we’d have a quantum computer to do it,” Jozsa said. Running Shor’s algorithm for integers large enough to unlock a standard 128-bit encryption key, for instance, would require thousands of qubits — plus probably many thousands more to correct for errors. Experimentalists, meanwhile, were fumbling while trying to control more than a handful.

But by 2011, things were starting to look up. That fall, at a conference in Brussels, [Preskill speculated](#) that “the day when well-controlled quantum systems can perform tasks surpassing what can be done in the classical world” might not be far off. Recent laboratory results, he said, could soon lead to quantum machines on the order of 100 qubits. Getting them to pull off some “super-classical” feat maybe wasn’t out of the question. (Although D-Wave Systems’ commercial quantum processors could by then wrangle 128 qubits and now boast more than 2,000, they tackle only specific optimization problems; many experts doubt they can outperform classical computers.)

“I was just trying to emphasize we were getting close — that we might finally reach a real milestone in human civilization where quantum technology becomes the most powerful information technology that we have,” Preskill said. He called this milestone “quantum supremacy.” The name — and the optimism — stuck. “It took off to an extent I didn’t suspect.”

The buzz about quantum supremacy reflected a growing excitement in the field — over experimental progress, yes, but perhaps more so over a series of theoretical breakthroughs that began with [a 2004 paper](#) by the IBM physicists [Barbara Terhal](#) and [David DiVincenzo](#). In their effort to understand quantum assets, the pair had turned their attention to rudimentary quantum puzzles known as sampling problems. In time, this class of problems would become experimentalists’ greatest hope for demonstrating an unambiguous speedup on early quantum machines.



[Lulie Tanett](#)

David Deutsch, a physicist at the University of Oxford, came up with the first problem that could be solved exclusively by a quantum computer.

Sampling problems exploit the elusive nature of quantum information. Say you apply a sequence of gates to 100 qubits. This circuit may whip the qubits into a mathematical monstrosity equivalent to

something on the order of 2^{100} classical bits. But once you measure the system, its complexity collapses to a string of only 100 bits. The system will spit out a particular string — or sample — with some probability determined by your circuit.

In a sampling problem, the goal is to produce a series of samples that look as though they came from this circuit. It's like repeatedly tossing a coin to show that it will (on average) come up 50 percent heads and 50 percent tails. Except here, the outcome of each "toss" isn't a single value — heads or tails — it's a string of many values, each of which may be influenced by some (or even all) of the other values.

For a well-oiled quantum computer, this exercise is a no-brainer. It's what it does naturally. Classical computers, on the other hand, seem to have a tougher time. In the worst circumstances, they must do the unwieldy work of computing probabilities for all possible output strings — all 2^{100} of them — and then randomly select samples from that distribution. "People always conjectured this was the case," particularly for very complex quantum circuits, said [Ashley Montanaro](#), an expert in quantum algorithms at the University of Bristol.

Terhal and DiVincenzo showed that even some simple quantum circuits should still be hard to sample by classical means. Hence, a bar was set. If experimentalists could get a quantum system to spit out these samples, they would have good reason to believe that they'd done something classically unmatchable.

Theorists soon expanded this line of thought to include other sorts of sampling problems. One of the most promising proposals came from [Scott Aaronson](#), a computer scientist then at the Massachusetts Institute of Technology, and his doctoral student Alex Arkhipov. In [work posted on the scientific preprint site arxiv.org in 2010](#), they described a quantum machine that sends photons through an optical circuit, which shifts and splits the light in quantum-mechanical ways, thereby generating output patterns with specific probabilities. Reproducing these patterns became known as boson sampling. Aaronson and Arkhipov reasoned that boson sampling would start to strain classical resources at around 30 photons — a plausible experimental target.

Similarly enticing were computations called instantaneous quantum polynomial, or IQP, circuits. An IQP circuit has gates that all commute, meaning they can act in any order without changing the outcome — in the same way $2 + 5 = 5 + 2$. This quality makes IQP circuits mathematically pleasing. "We started studying them because they were easier to analyze," Bremner said. But he discovered that they have other merits. In work that [began in 2010](#) and culminated in [a 2016 paper](#) with Montanaro and Dan Shepherd, now at the National Cyber Security Center in the U.K., Bremner explained why IQP circuits can be extremely powerful: Even for physically realistic systems of hundreds — or perhaps even dozens — of qubits, sampling would quickly become a classically thorny problem.

By 2016, boson samplers had yet to extend beyond [6 photons](#). Teams at Google and IBM, however, were verging on chips nearing 50 qubits; that August, Google quietly [posted a draft paper](#) laying out a road map for demonstrating quantum supremacy on these "near-term" devices.

Google's team had considered sampling from an IQP circuit. But [a closer look](#) by Bremner and his collaborators suggested that the circuit would likely need some error correction — which would require extra gates and at least a couple hundred extra qubits — in order to unequivocally hamstring the best classical algorithms. So instead, the team used arguments akin to Aaronson's and Bremner's to show that circuits made of non-commuting gates, although likely harder to build and analyze than IQP circuits, would also be harder for a classical device to simulate. To make the classical computation even more challenging, the team proposed sampling from a circuit chosen at

random. That way, classical competitors would be unable to exploit any familiar features of the circuit's structure to better guess its behavior.

But there was nothing to stop the classical algorithms from getting more resourceful. In fact, in October 2017, a team at IBM [showed how](#), with a bit of classical ingenuity, a supercomputer can simulate sampling from random circuits on as many as 56 qubits — provided the circuits don't involve too much depth (layers of gates). Similarly, [a more able algorithm](#) has recently nudged the classical limits of boson sampling, to around 50 photons.

These upgrades, however, are still dreadfully inefficient. IBM's simulation, for instance, took two days to do what a quantum computer is expected to do in less than one-tenth of a millisecond. Add a couple more qubits — or a little more depth — and quantum contenders could slip freely into supremacy territory. "Generally speaking, when it comes to emulating highly entangled systems, there has not been a [classical] breakthrough that has really changed the game," Preskill said. "We're just nibbling at the boundary rather than exploding it."

That's not to say there will be a clear victory. "Where the frontier is is a thing people will continue to debate," Bremner said. Imagine this scenario: Researchers sample from a 50-qubit circuit of some depth — or maybe a slightly larger one of less depth — and claim supremacy. But the circuit is pretty noisy — the qubits are misbehaving, or the gates don't work that well. So then some crackerjack classical theorists swoop in and simulate the quantum circuit, no sweat, because "with noise, things you think are hard become not so hard from a classical point of view," Bremner explained. "Probably that will happen."

What's more certain is that the first "supreme" quantum machines, if and when they arrive, aren't going to be cracking encryption codes or simulating novel pharmaceutical molecules. "That's the funny thing about supremacy," Montanaro said. "The first wave of problems we solve are ones for which we don't really care about the answers."

Yet these early wins, however small, will assure scientists that they are on the right track — that a new regime of computation really is possible. Then it's anyone's guess what the next wave of problems will be.

Correction on February 7, 2018: The original version of this article included an [example](#) of a classical version of a quantum algorithm developed by Christian Calude. Additional reporting has revealed that there is a strong debate in the quantum computing community as to whether the quasi-quantum algorithm solves the same problem that the original algorithm does. As a consequence, we have removed the mention of the classical algorithm.

This article was reprinted on [Wired.com](#).