word count / 2666

QUANTUM COMPUTATION: HOPES AND STRUGGLES

Quantum computation: hopes and struggles

You're in the middle of solving a physics problem. You reach a point where two small numbers have to be multiplied. Maybe you do the calculation in your head, but then another hurdle appears out of nowhere: you find yourself having to multiply two 8 digit numbers. This immediately makes you reach for your calculator, a Casio fx-991EX – a capable device, no doubt. Type in your input, get the output, move on. This goes on for a few hours, when you reach something truly unexpected. The next step requires you to use integration by parts – twice over. ***„Gee! This is all too much maths, let me just get to the physics already!"*** – you cry out. You make a painstaking effort to carry out the integration, but you just don't seem to be able to perform basic arithmetic at this point. How much easier would it be if you could just type the integral into a computer and get back the results – ah yes, WolframAlpha will do just that. So you fire up the website, insert your query and realise your integral was wrong in the first place. Your head hurting from confusion, you convince yourself that you are in fact enjoying this, and somehow, miraculously correct the mistake, now inputting without doubt the right integral to the search box and hitting enter. After about 15 seconds, you are welcomed by the disappointing message *standard computation time exceeded*. ***„Bugger!"*** you exclaim, subconsciously wishing the supercomputer from Hitchhiker's Guide was real and accessible to you and you never had to go through all this computational pain.

After closing WolframAlpha in a frustrated manner, just as you start giving up and commence browsing the other open tabs on your laptop, a news article appears in your recommended, claiming quantum supremacy has been achieved at Google[1,2], rendering all conventional computers officially obsolete. It also tells you the program they ran would have taken 10,000 years on a high-end supercomputer[3] – you relate to the struggle. You read on, of course, being a student of physics, and shortly stumble upon the sentence ***„IBM makes a five-qubit machine freely available"***[3] – a glimmer of hope appears in your eyes – shortly, the phrase ***„Jupyter Notebook"*** is mentioned, boosting your confidence and confirming to you that this is in fact, the solution to your struggle.

After days of learning about logic gates and completing an online course in quantum algorithms, you are finally ready. You have extracted the necessary piece of code from WolframAlpha's servers for analytic integration and you are about to run it on the quantum computer. Pouring yourself a cup of the finest Earl Grey, you sip and press ***run code***. Twenty long seconds pass, then a warning appears: ***standard computation time exceeded***. You stare at the message in disbelief – what good is quantum then?

The quantum advantage

There are things quantum computers are better at, it just turns out it's not this. They are not even marginally faster when running the same algorithm that was designed for a classical computer (that's what the type you are using right now is called, the one with 1s and 0s in the background). That would be like a washer-dryer being better at the washing part than a simple washing machine – it doesn't do anything special if you just ask it to wash, but it ***is*** much better at the task of producing clean ***and dry*** clothes out of your laundry. Have you ever tried asking a washing machine to dry your clothes? Its efforts will be contemptible at best (it just waits until they dry by themselves, of course). This is analogous to how quantum compares to classical computing. There are some problems, like multiplication (a basic washing-sort-of-task) that, while you ***can*** do on a quantum machine, it's quite boring – takes just as long as a conventional processor. Going the other way however, factorisation is one of those calculations that are so significantly faster with qubits that

identity	$\begin{pmatrix}1&0\\0&1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}1&0\\0&1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$
negation	$\begin{pmatrix}0&1\\1&0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}0&1\\1&0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}$
set to zero	$\begin{pmatrix}1&1\\0&0\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}1&1\\0&0\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}1\\0\end{pmatrix}$
set to one	$\begin{pmatrix}0&0\\1&1\end{pmatrix}\begin{pmatrix}1\\0\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}0&0\\1&1\end{pmatrix}\begin{pmatrix}0\\1\end{pmatrix}=\begin{pmatrix}0\\1\end{pmatrix}$

Figure 1: The four single-bit operations with their matrix representations, acting on a 0 and a 1 bit.

there is reason to fear the collapse of modern encryption when a powerful enough quantum chip is built – just as the drying part is far superior with a washer-dryer.

If you ask a computer scientist, they'll say it all boils down to the smallest unit of information the two systems are built on: ***cbits*** and ***qbits***[4,5] (or qubits[6]). It will be helpful for us to represent the

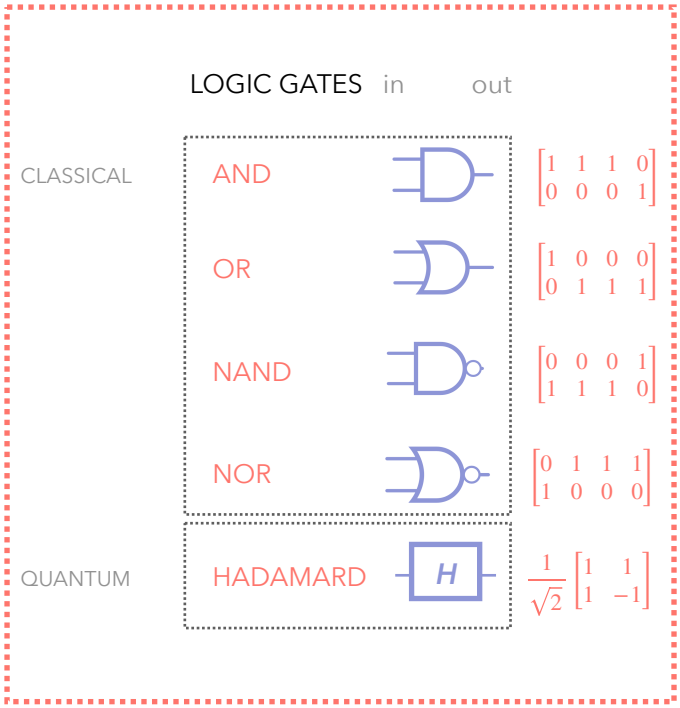

Figure 2: Basic logic gates for classical and quantum computing. There are several more complicated ones, these are just a few of the most common ones selected to illustrate the concept. *(icons courtesy of Studio Refine from Noun Project)*

former, classical bits by two possible 2-vectors: $\begin{pmatrix}1\\0\end{pmatrix}$ for the value zero and $\begin{pmatrix}0\\1\end{pmatrix}$ for one. There are four fundamental operations that can be performed on these, each one corresponding to a 2×2 matrix (see Figure 1). Then there are the more complex logic gates, such as **AND, OR** etc. that process more than one bit (see Figure 2). All of classical computation is built from these elements, from video game boss fights to the blue screen of death.

The quantum counterpart of classical bits, qbits (often called qubits\cite{schumacher}) have an advantage; they can take on any complex value for the 2-vector elements, except the magnitudes have to sum to 1 for it to be correctly normalised, meaning for a qbit $\binom{a}{b}$,

$$|a|^2+|b|^2=1, \tag{1}$$

where a, b are both complex.

Quantum mechanics tells us that a measurement collapses – transforms – the observable (in this case the qbit) to one of its possible (eigen-) states, which are 0 or 1 in our case. Until we measure it though, it's in a superposition of these states – like Schrödinger's cat, if you like that sort of analogy. Here, the two elements a and b can be thought of as the probability amplitudes of the two possible states, where $|a|^2$ gives the probability of the qbit collapsing to a 0-bit, and $|b|^2$ to a 1-bit. You might notice this doesn't rule out the two cbits we introduced above, they just have a 100% probability of collapsing to either 0 or 1 (depending on whether it's $\begin{pmatrix}1\\0\end{pmatrix}$ or $\begin{pmatrix}0\\1\end{pmatrix}$ of course) when measured. So cbits are, in fact, just two special qbits. Throughout the history of modern computation, we have been limited to 2 of the infinite values a bit can take on, imagine the possibilities it would unlock if we could choose from an infinity of complex numbers to use. Except there doesn't seem to be quite as many useful applications as new qbits, for now. There are a few interesting things about these little things that are immediately obvious however. For example, consider a qbit $\begin{pmatrix}1/\sqrt{2}\\1/\sqrt{2}\end{pmatrix}$. If you look carefully, you'll find this is the qbit that will collapse to 0 or 1 with equal probability (it's not $\begin{pmatrix}\frac{1}{2}\\\frac{1}{2}\end{pmatrix}$, because that would

not be correctly normalised – see Equation (1). Well okay, you might say, but how do we build a reliable algorithm if we don't now for sure which qbit will spontaneously turn into what? It will produce a different outcome every time, not like the robust, deterministic calculations we are used to. And you're right, this can be a problem, there is whole field of quantum computer science devoted to error correction. But it's not just a bug, it can also be a feature if handled right. We don't ***need*** other gates and operations for a qbit system – the old classical methods will still work – but to exploit all of the capabilities of such a computer, we can introduce special gates that can make use of the nature of superposition – such as the ***Hadamard gate*** (in Figure 2). What's more, we can use more than one qbit, entangle them with each other and amplify certain outcomes we would like to appear with higher probability. For the advantages to truly take off, we need upwards of a 1000 qbits. This is not as simple as just adding them one by one, and is in fact ***the*** bottleneck in current research projects.[7]

The Algorithms

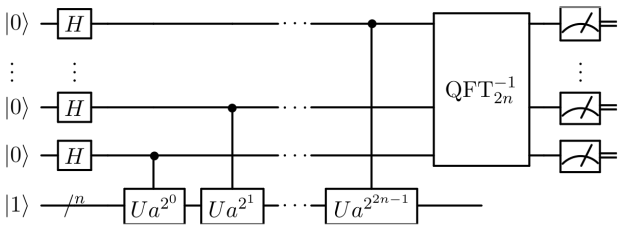

Figure 3: This is a high-level circuit diagram of Shor's algorithm. The input qbits (left) go through a series of quantum logic gates, reaching a Quantum Fourier Transfrom (QFT) block, after which they are each measured for output.[8]

As mentioned above, factorisation is a task where these machines excel, provided we use a specially tailored program. Behold **Shor's algorithm[9]** (in Figure 3), able to find the prime factors of the input integer N in ***polynomial time***, meaning the time it takes to run is less than the order of $O(N^k)$ for some positive constant k. This is opposed to the best classical methods (see ***general number field sieve***) achieving sub-exponential ($2^{o(N)}$) time, which means it's currently infeasible to factor larger numbers, such as the ones used in RSA encryption. Its advantage lies in the QFT, or Quantum Fourier Transform part, which is able to find the periodicity of a function incredibly fast, which turns out is mathematically the same problem as integer factorisation.[10]

A note on encryption

Very simply put, much of modern cryptography is built upon the concept of one-way functions. Specifically, the ubiquitous method of RSA encryption, used throughout the web, relies on integer factorisation being hard – taking impossibly long for large enough numbers. The ***keys*** required for decrypting a secret message are the prime factors of a given 1024 bit number (for 1024-bit RSA), therefore anyone cracking the code would have to somehow guess the factors – good luck with that.

The ever-growing world of cryptocurrencies is also at risk, as the one-way, so-called ***hash*** functions used there, combined with some coin-specific vulnerabilities mean these systems are not quantum-safe, and are also prone to attacks from a computer running modified versions of Shor's and Grover's algorithms (see latter below).[11]

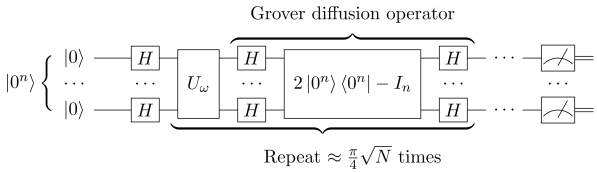

Figure 4: Grover's algorithm, again, on a high-level circuit diagram. Here we start with *n* input qbits, with the key operations being performed on their *phases*. The algorithm finds the only entry satisfying the search condition with high probabilty when measured at the outputs.[12]

In '96, Lov K. Grover published his algorithm that could theoretically perform a search for a single entry in an ***unsorted*** (sorted ones are

much easier) database with N records, in $O(\sqrt{N})$ steps – classical methods are fundamentally ineffective here, allowing only $O(N)$ steps (comparison illustrated in Figure 4). Somewhat surprisingly, it can also be used to find the inverse $f^{-1}(y)$ of a function $y = f(x)$, opening the door to potentially cracking other widespread encryption methods.[13]

Simulations

However trivial it may seem at first, quantum computers will most certainly be superior in a specific, very complex area of science: simulating quantum systems. The machine being a quantum system itself, this may not come as a shock, but other than raising deeply unsettling questions – such as whether we live in a world that is itself a simulation on a quantum machine[14] – this area could prove to be game-changing in a number of research objectives. Chemical reactions could be simulated efficiently, speeding up drug development, better catalysts for fertiliser could be produced, prototyping new materials for aerospace would be possible along with many other exciting potential applications. Classical circuits are not suited to handle the complexity that arises from interfering wavefunctions and many-particle quantum systems. It is easy to see how the limitations of a deterministic, binary system (with its 0s and 1s) has to have more bits to represent the inherently probabilistic nature of reality, where everything is built from superpositions of the conventional discrete states.

Where we're at and the challenges ahead

Given the high stakes and potential reward involved in a functional quantum computer, it should come as no surprise that the world's leaders in technological research are heavily invested in its development. NASA and Google's AI department have achieved quantum supremacy a few years back, meaning they successfully ran a program that would have taken ***thousands of years*** on even the most powerful classical supercomputer in a few seconds[1]. IBM has a freely accessible version online and, along with Microsoft, their own implementation of a quantum programming environment[3]. All of these projects however use somewhere around a hundred qbits, which is an alarmingly low number considering the thousands needed for remotely useful computation. There are of course a multitude of challenges to overcome, not to mention the looming possibility that the final concept can never actually come to fruition because of the exponentially growing noise in the system.[15]

The first and most difficult problem to overcome is decoherence[16], which is essentially information loss due to interactions with the environment. The system can never truly be isolated from the outside world in practice, and entanglement and interference will have an effect that grows in strength as more qubits are added to the circuit.

There are also supply chain challenges that are not expected to be resolved unless there is stable demand for components and diversification, as well as innovation in the relevant manufacturing sectors. The superconducting cables needed for building the circuits are only made by a single mainstream company in Japan, and it can easily take over a year to acquire the dilution refrigerators needed for cooling the equipment to the necessary temperatures for stable qbit creation (they need to be cooler than outer space).[17] We'll just have to wait and see how the field develops… or join in the struggle for a great quantum computer. Either way, don't expect a consumer product next week.

References

[1] Tavares, F. (2019) *Google and NASA Achieve Quantum Supremacy.* http://www.nasa.gov/feature/ames/quantum-supremacy [Accessed Jan 10, 2022].

[2] Gibney, E. (2019) Hello quantum world! Google publishes landmark quantum supremacy claim. *Nature.* 574 (7779), 461-462. 10.1038/d41586-019-03213-z.

[3] Matthews, D. (2021) How to get started in quantum computing. *Nature.* 591 (7848), 166-167. 10.1038/d41586-021-00533-x.

[4] Andrew Helwer. (2018) *Quantum Computing for Computer Scientists.* https://www.microsoft.com/en-us/research/video/quantum-computing-computer-scientists/ [Accessed Jan 10, 2022].

[5] Mermin, N. D. (2002) From Cbits to Qbits: Teaching computer scientists quantum mechanics. 10.1119/1.1522741.

[6] Schumacher, B. (1995) Quantum coding. *Physical Review A.* 51 (4), 2738-2747. 10.1103/PhysRevA.51.2738.

[7] Nielsen, M, & Chuang, I 2010, Quantum Computation and Quantum Information : 10th Anniversary Edition, Cambridge University Press, Cambridge. Available from: ProQuest Ebook Central. [10 January 2022].

[8] Bender2k14. *Quantum subroutine in Shor's algorithm.* https://commons.wikimedia.org/wiki/File:Shor%27s_algorithm.svg#/media/File:Shor's_algorithm.svg. [10 January 2022]

[9] Shor, P. W. (1995) Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. 10.1137/S0097539795293172.

[10] N. David Mermin. (2007) *Quantum Computer Science.* Cambridge University Press. https://www.lassp.cornell.edu/mermin/factoring-princeton1.pdf [Accessed Jan 10, 2022]

[11] Buterin, V. *Bitcoin Is Not Quantum-Safe, And How We Can Fix It When Needed.* https://bitcoinmagazine.com/technical/bitcoin-is-not-quantum-safe-and-how-we-can-fix-1375242150 [Accessed Jan 10, 2022].

[12] Fawly. CC BY-SA 4.0. *Quantum circuit representation of Grover's algorithm.* https://commons.wikimedia.org/w/index.php?curid=106362482.

[13] Bernstein, D. J., Buchmann, J., & Dahmén, E. (2009). Post-quantum cryptography. Berlin, Springer. https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=276427.

[14] Lloyd, S. (2011) The Universe as Quantum Computer. In: LLoyd, S. (ed.).*A Computable Universe.* , WORLD SCIENTIFIC. pp. 567-581. https://www.worldscientific.com/doi/abs/10.1142/9789814374309_0029.

[15] Kalai, G. (2019) The Argument against Quantum Computers. https://arxiv.org/abs/1908.02499v1.

[16] Schlosshauer, M. (2003) Decoherence, the measurement problem, and interpretations of quantum mechanics. 10.1103/RevModPhys.76.1267.

[17] Giles, M. (2019) *We'd have more quantum computers if it weren't so hard to find the damn cables.* https://www.technologyreview.com/2019/01/17/137811/quantum-computers-component-shortage/ [Accessed Jan 10, 2022].