Ryan Coyne

David Edwards

ENGL 125C

12/07/2022

The Reality of Quantum Computing

Quantum computers are devices that make use of some of the mathematical quirks of quantum mechanics to perform calculations that would take a classical (referring to the use of classical mechanics, not the technology’s age) computer an unreasonable or impossible amount of time to complete. They are computers in the original sense that they perform computations, whereas modern classical computers have much more complex functions layered on their base layer of computation. Classical computers perform operations on bits, binary states represented by a 1 or a 0. However, a quantum computer performs operations on qubits, which are quantum mechanical systems with two potential states. As an example, a qubit may be a Yb+ atom (Dumitrescu et al. 2). This is the critical difference between the two types of machines and is the source of the quantum computer’s power.

Many people whom I have spoken with hold the misconception that quantum computers will one day replace our commonplace classical computers. While it is possible that the technology will advance so far that this will one day be true, our current understanding of the math and physics utilized by quantum computers suggests that it would be far too impractical. Current quantum computers are very expensive to build and operate, and while this may change over time, it is not the only limitation. Quantum computers are able to perform the same computations that classical computers can perform but cannot match the speeds of classical computers when performing most tasks that classical computers are already used for. Quantum algorithms are also almost always slower than classical algorithms for simple versions of the problems, which would further lower the performance of quantum computers at the things we currently use computers for. This is because at those low levels, exponential time, the speed at which the slowest classical algorithms run, is often faster than polynomial time, the speed at which most quantum analogs run.

Algorithms designed for quantum computers can theoretically complete certain mathematical problems in a much shorter time than the most powerful classical computers in existence. An important example of this is their theoretical ability to find the factors of very large numbers in a competitively short amount of time, which is essential for modern encryption protocols. One such protocol, RSA, is used to encrypt messages when the sender cannot be given the means to decrypt the cipher. RSA uses the product of very large prime numbers, and , as a part of the public key, and the prime numbers themselves are used to determine the private key. The public key is used to encrypt a message and can be public knowledge while maintaining the security of the encryption, while the private key is used to decrypt the message and is kept a secret. Since the public key is available to any that can send a message with this encryption and the private key is uniquely determined by the prime factors of the public key, determining those prime factors would allow a person to decrypt any message that was encrypted with that public key. Encryption protocols other than RSA, such as DSA and ECDSA, can also be solved in polynomial time by a quantum computer running Shor’s algorithm (Bernstein 1).

Today, the best factoring algorithms for classical computers can factor all integers in sub-exponential time, meaning that the run time of an algorithm increases more quickly than any polynomial but more slowly than any exponential function. Consequently, a modern classical computer would require more time than is possible to wait (often suggested to be in the billions of years) to decrypt a message by factoring the relevant portion of the public key. In contrast, a quantum computer could compute the factors of any integer in polynomial time, and a problem that can be solved in polynomial time is considered to be tractable (Cobham).

The algorithm that a quantum computer could use to factor large numbers is known as Shor’s algorithm and was discovered in 1994 by the American mathematician Peter Shor. This algorithm uses the fact that quantum states can be represented by a wave function, , and that the probability that the qubit will be in a particular state is to compute factors of very large integers. In RSA encryption, the private key is determined by the modular multiplicative inverse of a component of the public key, , with respect to the modulus which is the least common multiple of and . That is, the private key is (mod ) where . Here we are performing modular arithmetic, which is a system of integers where there are no integers greater than or equal to some number , called the modulus, and if an operation results in a number higher than , the wraps back to the beginning.

This type of arithmetic is similar to a clock which, for the purposes of this example, will have twelve hours. If it is currently seven o’clock and three hours pass, it will be ten o’clock, but if three more hours pass, the time will wrap back around to one o’clock. Modular arithmetic would start at zero and end at eleven instead of one and twelve and can include more operations than addition. However, the clock analogy gives a very good intuition for the differences to ordinary arithmetic. Two integers, *b* and *c* are congruent modulo if is an integer. This congruence is written . The multiplicative inverse of an integer, , under a modulus, , is an integer such that For example, if , then and the remainder of is 1. So , and thus 4 and 10 are multiplicative inverses of one another under the modulus 13.

Because RSA encryption depends so heavily on modular arithmetic, it is only fitting that Shor’s algorithm would take advantage of modular arithmetic to render it obsolete. The first step in Shor’s algorithm is to choose a random integer, , that is between 1 and the relevant piece of the public key, , and compute the greatest common denominator of the two. If the greatest common denominator is 1, then is not a factor of , and we continue with the algorithm. Otherwise, we have found a factor of and there is no need to continue. For the next step, we analyze the function , where x can be any integer from 1 to . It can be shown that repeats with a period, , which is less than but may be close to . For small numbers, it is possible to find by hand, and for some larger numbers, can be found on a classical computer. For example, if and , the period of is 630 (Coyne). However, the time complexity of this operation is greater than polynomial time for a classical computer, and so as approaches the orders of magnitude required for secure encryption, they can no longer complete the algorithm in a reasonable amount of time. At this point, the property that an uncertain quantum state can be described as a wave function, and the property that two quantum systems can interact so that their wave functions interfere, become important. The quantum Fourier transform will make use of these properties to determine the period of (Aaronson; Shor 16-17).

The Fourier transform is a linear transformation that decomposes a function into frequencies, which are the inverse of the period. To understand this, let us suppose there is a function which can be written as a linear combination of an unknown number of unknown wave functions. Each wave function has a frequency that may be different from the others. Given , the Fourier transform determines the frequencies of the wave functions which can be summed to produce . On a classical computer, a variant of the Fourier transform called the fast Fourier transform runs in exponential time. The quantum Fourier transform is a variant of the Fourier transform that acts on quantum states which runs in polynomial time (Gyongyosi and Imre, 23).

Quantum computers perform computations by manipulating their qubits so that they interfere with one another, producing superposition of all of the qubits involved. A superposition is a linear combination of two or more quantum states where each state has a probability of occurring when a measurement is taken of the system. In a quantum computer, this superposition is a linear combination of the wave function of each qubit. The goal of quantum algorithms such as the quantum Fourier transform is to produce a superposition with an almost guaranteed probability of resulting in the desired state when measured. If properly utilized, these properties can dramatically increase the speed of quantum algorithms because, in a quantum computer, the math being done is a physical thing, as opposed to the more abstract manipulations that classical computers perform. Period is a fundamental property of waves, and with these wave-like properties of qubits and the knowledge that the Fourier transform operates on wave functions to produce frequencies, it seems to make sense that they could quickly perform to determine a period.

Once the period is determined, it is trivial to find the factors of . If we let denote the period determined by the quantum Fourier transform, then two factors of , and , are , where gcd() represents the greatest common denominator of and . A multiple of two primes, such as those used in RSA encryption, has only those two prime factors, so and are the same primes that were used to determine the private key and can now be used to determine the private key by another party through the same method.

This theoretical discussion of quantum computers is fascinating but is any progress being made? Yes, on October 23rd, 2019, researchers at Google announced that they had achieved quantum supremacy. This phrase, “quantum supremacy”, makes it sound like they had created a Skynet-like artificial intelligence to rule over humanity, but the reality is quite underwhelming. Quantum supremacy refers to the milestone where a quantum computer is first able to perform a computation that no classical computer could perform in a reasonable amount of time. The quantum computer that achieved this milestone did so by checking the outputs of a quantum random number generator, which is a computation that would take a classical computer more than 10,000 years to complete. It is evident that quantum computer technology has progressed a great deal since the first quantum computer was created in 1998 (Chuang et al.). Unfortunately, quantum computers have a long road ahead before becoming practical in any real-world use case.

The computation that Google achieved quantum supremacy with has very little application in the real world, and the highest number that has been reliably factored using Shor’s algorithm on a quantum computer is a very underwhelming 21, which is a far cry from the enormous numbers on the order of that are currently used for RSA encryption. Quantum computers are still this limited even though the first was created more than 24 years ago because qubits are extremely unstable. This instability arises because any interaction can significantly change the state of a quantum system. There are two major classes of errors that a quantum computer can experience: coherent and incoherent errors. Coherent errors result from undesirable interactions between qubits within the computer and are the limiting factor on the possible number of qubits in the quantum computer. Incoherent errors result from interactions with the environment around the quantum computer, which are always undesirable and occur because it is impossible to completely isolate the qubits from the environment.

Steady progress has been made since the first quantum computer, which had just two qubits. On November 9th, 2022, IBM announced a quantum computer with 433 qubits, the most in the world at the time of writing and more than three times the number in the previous IBM model. Also in 2022, a paper was published in which the researchers were able to maintain greater stability by structuring the quantum circuits in terms of two types, which were arranged in accordance with Fibonacci words where a 0 represents one of the types and a 1 represents the other (Dumitrescu et al. 2). A Fibonacci word is formed by concatenating the previous two entries to form a new entry, where the first entries are defined as 0 and 01, so the third entry would 010. According to Dumitrescu et al., this structure increases stability because it is quasi-periodic, which increases the order of the system without repeating (3).

Quantum computation is a fascinating field with incredible potential, but it is still in its infancy and will most likely remain there for some time. Quantum computers will still not be commonplace for many years to come and will never be the magical devices that some would expect from the popular culture portrayal of quantum mechanics. Although, it is not yet possible to buy a home quantum computer, there is a way to start using them from home. An organization by the name of Qiskit has created a freely available, open source, quantum computing SDK which anyone interested in quantum computing should take advantage of. The SDK can simulate quantum computing on a classical computer, and for anything that needs to be run on a real quantum computer it can access one of IBM’s.

Works Cited

Aaronson, Scott. “Shor, I’ll Do It.” *Shtetl-Optimized*, 24 Feb. 2007, <https://scottaaronson.blog/?p=208>.

Bernstein, Daniel J. “Introduction to Post-Quantum Cryptography.” *Post-Quantum Cryptography*, edited by Daniel J. Bernstein et al., Springer, 2009, pp. 1–14. *Springer Link*, <https://doi.org/10.1007/978-3-540-88702-7_1>.

Chuang, Isaac L., et al. “Experimental Implementation of Fast Quantum Searching.” *Physical Review Letters*, vol. 80, no. 15, Apr. 1998, pp. 3408–11. *APS*, <https://doi.org/10.1103/PhysRevLett.80.3408>.

Cobham, Alan. “The Intrinsic Computational Difficulty of Functions.” *Logic, Methodology and Philosophy of Science: Proceedings of the 1964 International Congress (Studies in Logic and the Foundations of Mathematics)*, edited by Yehoshua Bar-Hillel, North-Holland Publishing, 1965, pp. 24–30.

Coyne, Ryan. *ModPeriod/Mod.Py at Main · Rainvm/ModPeriod*. <https://github.com/rainvm/modPeriod>. Accessed 7 Dec. 2022.

Dumitrescu, Philipp T., et al. “Realizing a Dynamical Topological Phase in a Trapped-Ion Quantum Simulator.” *Nature*, vol. 607, no. 7919, July 2022, pp. 463–67. *arXiv.org*, <https://doi.org/10.1038/s41586-022-04853-4>.

Lee, Jane Lanhee. “IBM Launches Its Most Powerful Quantum Computer with 433 Qubits.” *Reuters*, 9 Nov. 2022. *www.reuters.com*, <https://www.reuters.com/technology/ibm-launches-its-most-powerful-quantum-computer-with-433-qubits-2022-11-09/>.

Shor, Peter W. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.” *SIAM Journal on Computing*, vol. 26, no. 5, Oct. 1997, pp. 1484–509. *arXiv.org*, <https://doi.org/10.1137/S0097539795293172>.

Gyongyosi, Laszlo, and Sandor Imre. *An Improvement in Quantum Fourier Transform*. arXiv:1207.4464, arXiv, 18 July 2012. *arXiv.org*, <https://doi.org/10.48550/arXiv.1207.4464>.