Quantum Algorithms

by Ted G. Lewis

Editor’s Introduction
This essay was written for computer scientists seeking to understand quantum computing from an algorithmic point of view. It sacrifices rigorous physics so anyone with a computer science background can understand it. Note that it is about algorithms, not machines. This is intentional.
Quantum Algorithms

by Ted G. Lewis

A startling discovery was made in 1900 by Max Planck (1858–1947) that turned the deterministic physics of Newton upside down. Planck’s experiments on blackbody radiation convinced him that energy is quantized rather than continuously flowing like water. It comes and goes in packets called quanta.

His concept of energy quanta conflicted with everything known about matter and energy up to that time. It was so radical that Planck himself was skeptical. But as evidence piled up through the work of Einstein and others, Planck gradually came around. At very small scales, the Universe dissolved into quanta. The classical physics of Newton was banished in favor of quantum physics—at least at small scales.

The Newtonian idea of a solid particle gave way to the notion of matter waves acting sometimes like a vibration and other times like a solid. Even the exact location of a particle became probabilistic instead of definite as dimensions scaled downward. At the so-called atomic level (approximately one nanometer), nothing can be certain except that matter and energy are wave-like. This required a complete rethinking of reality. A reality described by quantum mechanics.

Quantum theory and the modern model of quantum objects such as atoms, electrons, protons, neutrons, etc. emerged from a rather small group of physicists around the turn of the 20th century through the mid 20th century. Today we would say they formed a social network as illustrated in Figure 1. The most influential members of this network included Ernest Rutherford (1871–1937), discoverer of the proton, alpha, and beta rays, and a number of other fundamental properties of matter. Rutherford is known as the “father of nuclear physics” because of his enormous contributions and influence on others.
Figure 1. The most influential atomic scientists of the twentieth century formed a social network with J. J. Thompson, Erst Rutherford, and A. J. W. Sommerfeld at the center because of their large number of students and associates. Note: The number of connections is shown as the inscribed $g$ number.

The New Reality

A burst of activity by an unusually erudite network of pioneers gave us our current understanding of quantum mechanics. They were a close-knit group. Today we might say they were prominent members of a social network, exchanging posts the old-fashioned way—by postman and face-to-face communications. These communications often took place on a train crossing Europe, by writing each other letters, or through their graduate students at universities.

Ernest Rutherford, Sir J. J. Thomson (1856–1940), discoverer of the electron, and A. J. W. Sommerfeld (1868–1951) influenced an entire generation of students and colleagues. Rutherford had 36 students, including the chief designer of the largest-ever detonated nuclear bomb—Tsar Bomba—Yulii Borisovich Khariton (1904–1996). Ten of Rutherford’s 36 students were subsequently awarded Nobel prizes in physics.

Rutherford was the son of a New Zealand farmer who had immigrated from Scotland. Thomson admitted Rutherford to the Cavendish Laboratory, University of Cambridge where he “invented” a radio receiver, only to discover later that Marconi had beaten him to it. He rose through the scientific ranks by sheer hard work and genius.

If you worked with Thomson, the chances were very good that you would win an achievement award of some kind. Thomson led the English component of this social network and A. J. W. Sommerfeld led the German component. Thomson taught 17 graduate students. Eight of them received Nobel prizes, including his son. Sommerfeld taught 31 students, including Werner
Heisenberg (1901–1976). Seven received Nobel prizes. Thomson and Sommerfeld hold the unofficial title for advising the most Nobel prize-winning students in history.

Thomson also came from humble circumstances. His father ran a bookstore, and his mother descended from textile workers. Like many prodigies, he was admitted into college at a young age and excelled at mathematics. When he died in 1940 his ashes were appropriately buried in Westminster Abbey, next to the graves of Sir Isaac Newton, and his student, Ernest Rutherford.

On the other side of the English Channel was Arnold Johannes Wilhelm Sommerfeld (1868–1951), a pioneer of atomic and quantum physics. Like Thomson, Sommerfeld excelled at mathematics and received his Ph.D. at the age of 22. One of his teachers was David Hilbert, the man who almost beat Albert Einstein to the theory of relativity. Today, Hilbert space mathematics remains the formal basis of quantum mechanics. Quantum computing takes place in Hilbert space.

What is so remarkable about this small and close-knit group of men was their radical ideas about the nature of reality. Matter does not exist as a solid, but rather as waves of probability clouds. An electron is actually a fog of matter that is denser in some places than others. Atomic particles are matter waves that behave according to a wave function described by Edwin Schrödinger’s wave equation. It is the matter-wave duality that makes quantum computing possible.

Some properties of this fog are found to be in multiple states at the same time. Like musical chords on a piano, the state of a quantum particle is composed of multiple notes playing at the same time—a chord. The note becomes a state, and a chord becomes a superposition of multiple states in the language of quantum mechanics. A quantum computer operates on these states in parallel to solve a problem “in parallel.”

In 1926 Edwin Schrödinger published his famous equation that describes the time-dependent motion of quantum matter waves—the evolution of the state of matter waves over time. Assuming a binary state for each particle, a quantum system of \( n \) particles can be in \( 2^n \) states at once. An operation on one particle is an operation on all particles at once if the \( n \) particles are treated as a single state space. That is, the \( 2^n \) states exist simultaneously so that an operation in state space modifies all \( 2^n \) states at once. In each step of a quantum algorithm, it is possible to change \( 2^n \) values, simultaneously without additional circuits or storage.

Coherence is the term Schrödinger used to describe a system in multiple states of superposition. As long as it is not disturbed, a quantum system remains coherent. But coherence collapses when observed (or disturbed), ending any computations taking place while in superposition. Coherence vanishes and the \( 2^n \) states collapse into one state.
Also, in 1926, one of J. J. Thomson's students, Max Born (1882–1970), made the connection between the wave function and the probability of a quantum system being in a certain state. The Born Rule says that the probability of finding a particle at a given point when the system collapses is proportional to the square of the amplitude of the particle's wave function at that point. Quantum algorithms use the Born Rule to equate wave amplitudes with numerical values typical of classical algorithms.

A Fantastic Train Ride

In 1925, Albert Einstein, Wolfgang Pauli, and Paul Ehrenfest chased Niels Bohr’s train across Europe, meeting him at a stopover in Hamburg, Germany. Ehrenfest—a student of Ludwig Boltzmann (1844–1906) who committed suicide two years after Ehrenfest earned his Ph.D.—had urgent news that couldn’t wait for Bohr to arrive at his destination in Leiden, Netherlands. He had to tell Bohr about his thought experiment in which he imagined fleas randomly jumping back and forth between two dogs—a model of how atomic particles are distributed throughout space!

Later, Werner Heisenberg and Pascual Jordan met Bohr at the Gottingen, Germany stop. Again, they were eager to report their results to the prevailing expert on the atomic structure of matter. At the atomic level, matter becomes fuzzy and according to Heisenberg, uncertain. The deterministic physics of Newton crumbled at the atomic level. Probability reigned in its place. The realization that Newton was wrong at the atomic level was spreading across Europe like the Black Plague.

Even later, Pauli intercepted Bohr again in Berlin. Ditto—Pauli anxiously explained his exclusion principle—if atoms really had electrons circling a nucleus as Bohr claimed, they could only take on certain energy levels and were otherwise excluded from all other levels. Pauli’s exclusion principle elaborated on Planck’s quantum theory of 1900 and provided further proof that actions at a very small scale are quantized and probabilistic.

The interruptions continued as Bohr’s train approached Leiden. By the time the train reached its destination, most of quantum theory was decided—half of the meeting had already taken place! 1925 marks the beginning of the quantum era. By 1927 the Copenhagen interpretation of how the world works at the sub-atomic level was settled. (Note that Bohr was Danish).

Why were these famous men so eager to meet Niels Bohr, the scientist who, along with Rutherford, developed the Rutherford-Bohr model of the atom? Bohr had worked out the math that explained how atoms worked. According to the Rutherford-Bohr model, atoms consist of a nucleus with orbiting electrons that can take on only certain states—quantum states dictated by
Planck’s constant. Electrons jump from one state to another level of energy in discrete jumps—quanta—according to the new theory. Quantum theory contradicted almost everything in the classical Newtonian description of the world. It was an exciting time to be a scientist.

Sadly, many of the quantum pioneers met a tragic end. Sommerfeld was killed in a traffic accident while walking his grandchildren. Ehrenfest shot his disabled son and then himself in 1933. Planck’s son Erwin was executed by the Nazis in 1945 for participating in a failed attempt to assassinate Adolf Hitler. While Pascual Jordan supported his Jewish colleagues, he later joined the Nazi party in 1933 and spent the rest of his life in politics. Paul Dirac died in 1984 after he turned down a knighthood because he did not want to be addressed by his first name.

Heisenberg was awarded the 1932 Nobel Prize in Physics for “the creation of quantum mechanics.” The United States Office of Strategic Services sent agent Moe Berg to “attend Heisenberg’s lecture carrying a pistol, with orders to shoot Heisenberg if his lecture indicated that Germany was close to completing an atomic bomb” [1] Wolfgang Pauli died of pancreatic cancer on December 15, 1958, at the relatively young age of 58.

A Modern Shell Game

The Cosmic Joker brushes aside her flaming red hair and points a flaming red fingernailed digit right at me in the first row of the audience. I look around, but she is pointing at me! “Come up here, old man.” She says in a scolding voice. I begrudgingly obey and slowly walk to the stage like a schoolboy going to the principal’s office.

Before her is a row of three half-coconut shells, numbered left-to-right from 0 to 2, which she shifts around so as to distract the audience. “I will place this fig under shell number 2, and then watch,” she shouts to the audience. She proceeds to exchange the first and third shells, followed by the second and first shells. Then looking me straight in the eye says, “Where is the fig?”

I am old but not dead, so I tracked the motion of her hands as she rearranged the shells. “It is under the first shell—number 2—I said with confidence. The audience gasped as she lifted all three shells to reveal their contents. A fig appears under both the first and second shells but not the third.

The Cosmic Joker failed to tell the audience and me that there might be other figs hiding in the shells. Her power lies in her ability to manufacture figs or not, at will. I soon learned that regardless of what I expected, figs could magically appear or disappear under a shell. The joke was always on me! Was this some sort of modern shell game?
The Cosmic Joker repeated her deception several more times, each time embarrassing me more and more in front of the audience. I guessed the correct number and placement of figs only once out of a dozen attempts. It turned into a whack-a-mole game of chance with the house winning most of the time. The Cosmic Joker giggled and continued tormenting me.

I decide to record the results to see if a pattern emerges. When a fig is revealed under a shell, I record a 1, otherwise a 0. That is, the state of the shell is either a 1 or a 0, and the state of the three shells is a string of three 1/0s. For example, when a fig is found under the first shell its state is 1 and the state of the shell game is 1,0,0. If a fig appears under shell 0 and 2, the state of play is 1,0,1. Finally, I count the number of times each state occurs and convert the counts into frequencies. This is an estimate of the probability that each state is produced by the Cosmic Joker’s magic.

I wonder what the Cosmic Joker is up to? She repeats shuffling followed by exposing the contents of the shells again, so rapidly that I can barely keep track. Meanwhile, I furiously record the results—writing down the three-digit state each time. This reveals something strange. A fig appears under a shell—any one of the three shells—without any apparent pattern. Regardless of which shell I chose, a fig either appears or not. I watch for a while and tally the results. Each of the eight possible states occurs roughly equally, one-eighth of the time. It is as if the shells contain both a fig and no fig at once. The state of each shell is indeterminate until the shell is turned over.

After many more shuffles, it becomes apparent that each of the eight possible states occurs equally often—one-eighth of the time. But what the Cosmic Joker is keeping a secret is this: Each shell contains nothing and a fig, both at once! Both results are present at once. Furthermore, the result is indeterminant—the shell might contain a fig or not, equally likely. Shells don’t have to decide whether they contain a fig or not until exposed. They can be in both states simultaneously.

The random appearance (or not) of figs under coconut shells is independent of one another. The probability of a fig in shell 0 is independent of the states of the other two shells. Thus, all eight possible configurations of figs occur with equal probability. The equal probability of all states of coconuts illustrates a Hadamard state in quantum mechanics. A Hadamard gate is a quantum computing operation that puts qubits into superposition with an equal probability of occurring when it collapses.

But what if they are not independent? What if the presence of a fig under shell 0 forces the other two shells to be empty? The system of shells is somehow correlated or what is called entangled in quantum mechanics. Entanglement in quantum state space allows one to construct the
equivalent of a circuit in classical engineering. While superposition is the power of quantum computing, entanglement is the basis of quantum computing.

If the three shells were entangled, the number of configurations reduces to three—001, 010, 100—each appearing one-third of the time upon revealing their contents. In a sense, the state space transitions from eight independent states to three because the shells are no longer independent. They are correlated. Somehow the Cosmic Joker is able to correlate the contents of the coconut shells such that the appearance of a fig under one shell forces the disappearance of figs under all other shells.

**Spooky Correlation**

In 1935 Albert Einstein (1879–1955), Boris Podolsky (1896–1966), and Nathan Rosen (1909–1995) published a paper in Physical Review titled, “Can Quantum Mechanical Description of Physical Reality Be Considered Complete?” The paper, subsequently abbreviated to EPR, describes a Gedankenexperiment—a thought experiment of the kind that Einstein was famous for. This EPR paper contained a startling theoretical idea; that a quantum system of particles can be entangled regardless of how far apart they are from one another. That is, two or more particles are correlated—one or more of their properties are connected so that changing one particle affects the other particle. Furthermore, the change is instantaneous.

*Whenever two or more particles become entangled, they share a single quantum state.*

Suppose a quantum particle on the moon is entangled with a “twin” particle on Earth. Further, suppose that we read the state of the lunar particle and discover it is UP, then the state of the terrestrial particle is automatically and instantaneously DOWN. Or, if the lunar particle is DOWN, the terrestrial particle is automatically and instantaneously UP. This quantum entanglement is what Einstein called “spooky action at a distance.”

The EPR paper remains in the top 10 most impactful papers ever published in a Physical Review journal. The sophistication of quantum theory evolved quickly partially due to this paper. But Einstein was not convinced of the validity of his own paper. Later he said, “It did not come out as well as I had originally wanted.” It was Podolsky’s idea, really, but it stimulated a half-century of debate.

Superposition and quantum entanglement have since become building blocks of quantum algorithms. Superposition allows us to encode 1s and 0s as information qubits (quantum bits),
and entanglement allows us to connect qubits together to form “circuits” and perform operations analogous to electrical circuits in classical computers.

The EPR paradox defined *Verschränkung* (German for entanglement) as follows: Two particles can become entangled such that when the state of a particular property is measured in one particle, the opposite state is observed on the entangled particle instantaneously. This is true regardless of the distance separating the entangled particles, so measuring the state of one particle instantaneously informs the other particle.

Of course, if UP always correlates with DOWN when entangled, we may just as well say UP is entangled with UP because we can flip the state of the entangled particle. The important thing is that once the first quantum state is decided, the second one is automatically decided. The first quantum particle may collapse to 0 or 1 with equal probability, but the second particle always agrees with 100 percent probability, if fully correlated.

The perfect entanglement of two qubits with equal probabilities of collapsed state is known as the Bell state. The degree to which particles are entangled is called concurrence. Figure 2 shows how concurrence ranges from 0 to 1 [2]. Generally, quantum algorithms use two states: 0 and 1, and concurrence of either 0 or 1.

![Figure 2. Entanglement concurrence varies from 0 to 1. Entanglement is used to construct quantum logic gates.](image)

**The C-NOT Gate**

To illustrate how entanglement is used in quantum computers, consider the simple C-NOT gate (Control-NOT). It has two inputs, control A, and target B (see Figure 3). The C-NOT gate operates on a quantum register consisting of two qubits and flips the second qubit (the target qubit) if and only if the first qubit (the control qubit) is $|1\rangle$. In simple English, “if A equals 1 flip B, otherwise
do nothing to B.” The truth table in Figure 3 shows the results obtained for each of the four possibilities of A and B.

![Truth Table](image)

**Figure 3.** The quantum C-NOT gate uses quantum entanglement to compute exclusive-or: $B = A \text{ XOR } B$. Entanglement is shown schematically as a line connecting the two qubits A and B. We use Dirac notation: $|0\rangle$ means the qubit is simultaneously 0 and $|1\rangle$ means it is also 1.

The first experimental realization of a C-NOT gate was accomplished in 1995 using a single Beryllium ion. The two qubits were encoded as an optical state and a vibrational state of the ion.

**Programming a Quantum Computer**

The basic unit of information in a quantum computer is the qubit (quantum bit), named by Benjamin Schumacher and William Wootters in a paper published in 1995. Schumacher writes, “This is accomplished by replacing the classical idea of a binary digit with the quantum two-state system, such as the spin of an electron. These quantum bits, or ‘qubits’, are the fundamental units of quantum information” [3].

A qubit can simultaneously be in two states at once. Two qubits can be in four states at once, and three qubits in eight states at once. In general, $n$ qubits can be in $2^n$ states at once. This is the power of quantum computing and the source of “quantum supremacy.” Whereas a classical computer can process binary bits sequentially, a quantum computer with $n$ qubits can process $2^n$ bits at once, as long as it is coherent.

High-level programming languages have been modified to include quantum operations like the Hadamard gate and the C-NOT gate. Primitives are often built in to supply basic operations like setting values and measuring results. To illustrate consider the Q# modifications to Python provided by Microsoft [4]. The simple quantum program in Q# shown in the Example Quantum Program sidebar below illustrates how to use the Hadamard function and C-NOT function with superposition and entanglement.
**************** Sidebar 1: Example Quantum Program ****************

Two qubits, q1, and q2, are entangled with superposition values of 0 and 1. We want to count the number of times the two qubits agree with one another and the number of times the state is equal to 0 and 1.

```
operation TestState(count : 1000, initial : Result) : (Int, Int, Int) {
    mutable nOnes = 0; //Mutable means changeable
    mutable agree = 0; //Count the number of ones and number that agree
    using ((q0, q1) = (Qubit(), Qubit())) { //Declare two qubits, q1, and q2
        for (t in 1..1000) { //Loop 1000 times
            Set(initial, q0);
            Set(Zero, q1);
            Hadamard(q0); //Create a superposition
            CNOT(q0, q1); //Return 1 if they disagree
            let res = Measure(q0); //res = 0 (agree) or 1 (disagree)
            if (Measure(q1) == res) { //Measure the result: either 0 or 1
                set agree += 1; //Count the number of times they agree
            }
            if (res == One) { //Count the number of 1s
                set nOnes += 1;
            }
        } //End loop
        Set(Zero, q0); //Clean up: return qubits to zero
        Set(Zero, q1);
    }
    // Return number of times we saw a 0 and number of times we saw a 1
    return (count - nOnes, nOnes, agree);
}
```

This code uses five subroutines, `Set`, `Measure`, `Reverse`, `Hadamard`, and `CNOT`. The `Set` routine sets the argument to a classical state—either 1 or 0. `Set` measures the qubit, and if it is in the state we want, it leaves it alone. Otherwise, it changes the qubit to the desired state. It does this by executing the primitive `Reverse` operation, which changes the qubit state to a new state in which the probabilities of a measurement returning zero and one are reversed.

```
operation Set(desired : Result, q1 : Qubit) : Unit {
    if (desired != Measure(q1)) {
        Reverse(q1);
    }
}
```

The results of `TestState` are non-deterministic, so for 1,000 trials, approximately half should be 0 and half should be 1. When this program was executed on a quantum computer owned by Microsoft, it produced approximately 500 counts, each:
Init: Zero 0s=499 1s = 501 agree = 1000
Init: One 0s=490 1s =510 agree = 1000

The first qubit has a 50/50 chance of being a 0 or a 1 (499 versus 501), but now when we measure the second qubit, it is always the same as what we measured for the first qubit (490 versus 510), because they are entangled.

*End Sidebar 1*

The Strange Language of Quantum Mechanics

Paul Dirac received the Nobel Prize in Physics along with Schrödinger in 1933. Schrödinger said only Einstein exceeded Dirac with his ability to understand the universe. Another physicist declared Dirac more than a genius—a magician—because his work astounded people that understood it. It was hard to see how any human could have imagined Dirac’s results [5].

Dirac invented a notation commonly used today—the bra-ket notation—a compact way of representing wave functions as vectors in Hilbert space. Mathematically, superposition is simply a linear combination of two vectors in Hilbert space. Dirac’s bra-ket notation for superposition of two states in qubit A is a linear combination of vectors $|0\rangle$ and $|1\rangle$ representing the two states:

$$\langle A | = a_0 |0\rangle + a_1 |1\rangle$$

Superposition of two waves = two states

This represents a superposition of 0 with probability $|a_0|^2$ and 1 with probability $|a_1|^2$, such that $|a_0|^2 + |a_1|^2 = 1$. But we don’t know which state the qubit is in when it collapses until the qubit collapses. As long as coherence exists, qubit A is assumed to be in both states at once and processing can take place “in parallel.”

The bra-ket notation for two maximally entangled qubits—Bell states—is:

$$|AB > = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

Entangled qubits A and B

The bra-ket notation for a maximally independent pair of qubits, A and B, called separable, is:

$$|AB > = \frac{1}{\sqrt{2}}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

Separable qubits A and B

---

1 Think of Hilbert space as a Euclidean space containing functions in place of dimensions and points, and as many dimensions as you want.
If we cannot predict with certainty what a collapsed system will produce, how can we depend on it for calculations? Instead of deterministic results, we must settle for highly probable results. Quantum algorithms are algorithms that iterate until the most probable state approaches 1.0, thus squeezing out uncertainty. We pay a price in terms of uncertainty, but we get parallel computation in return.

Superposition transforms coherent cubits in parallel states all at once. State space is where computation is done and where solutions are found. Quantum computation in state space is a form of data-parallel computing that uses superposition of values of a quantum particle in place of duplication of (parallel) transistors. Quantum computers replace multiple (parallel hardware) units in classical computation with multiple (parallel states) of quantum particles.

When combined, superposition produces data-parallelism and iteration increases the probability of the “right answer” close to 1.0. Think of quantum computing as moving a point in state space closer and closer to a solution. Exactly how this is done is the rest of the story.

**Compute It Before It Collapses**

The rules of quantum algorithms are still emerging, but what we know now, and what we can exploit for the purposes of computation are:

1. Qubits can be made from a variety of quantum particles and properties: spin of an electron or polarized photon (horizontal and vertical polarization).

2. Superposition is real and can be exploited to yield an exponential increase in information content. A qubit contains multiple states (values) of information all at once. A quantum computer must remain in superposition (coherent) long enough for the algorithm to finish.

3. A coherent superposition state collapses to a basis state when observed. When the result of the quantum algorithm is observed, the quantum computer collapses into its de-coherent base state.

4. When a qubit is observed and collapses to its basis state, it takes on the value of 1 or 0 with equal probability, unless operated on. Many algorithms find a solution by adjusting the probabilities of either a 1 or 0 to fractional values as the computation evolves. For example, if the state of a qubit is initially 1 with probability of 0.5 and 0 with probability of 0.5, a quantum operation may alter the probabilities to .05 and .95 after a number of quantum operations.
5. Entanglement is used to build gates such as the C-NOT gate, to carry out logic, analogous to a classical computer. Degrees of entanglement, called concurrence, are manipulated by quantum gates to give the desired result. Concurrence is simply a correlation, where -1 says if A is UP, then B is DOWN; and +1 says that if A is UP, so is B.

6. A Bell state is created by entanglement with a 50-50 chance of collapsing to 1 or 0. The Hadamard state (not entangled) and the Bell state (entangled) or often initial states of a quantum algorithm.

Rise of Quantum Machines

Quantum computing began in the early 1980s, when physicist Paul Benioff (1930–2022) proposed a quantum mechanical model of the Turing machine. Richard Feynman (1918–1988) and Yuri Manin (1937–2023) later suggested a quantum computer had the potential to simulate things that a classical computer could not, but no one offered an example.

In 1994, Peter Shor developed a quantum algorithm for factoring integers that had the potential to decrypt public key encrypted communications based on prime numbers, see Sidebar 2 below. A number \( N \) that cannot be factored is considered a prime number. That is, it is divisible by one and itself, only. Numbers that can be factored are called a composite number. For example, 15 is composite because it can be factored into three times five, but 13 is prime. Public key encryption is not the only method of securing information, but it is widely used because it is asymmetric—users are given a public and private key—rendering it ideal for key exchanges.

**************************Sidebar 2: Shor's Algorithm**************************

Shor proposed a quantum algorithm for parallel trials that produces 1 and \( N \), when \( N \) is prime and the factors when it is not a prime number. We only need to test odd numbers because even numbers are composite numbers easily factored by dividing by 2.

Shor’s algorithm is classical except in step 2, whereby quantum computation finds the period \( r \) using superposition.\(^1\) For example, factor odd \( N = 15 \), e.g., find \( X \) and \( Y \) such that \( XY = N \), where \( \text{GCD}(X, N) \) is the greatest common factor of \( X \) and \( N \):

1. Randomly select \( X \) between 1 and \( N \); say \( X = 7 \);
2. Use quantum superposition to find period \( p \) of \( X^i \text{mod} N \):
   e.g., do these operations all at once:
   \[ p = 1: 7^1 \text{mod} 15 = 1 \]
   \[ p = 2: 7^2 \text{mod} 15 = 4 \]
   \[ p = 3: 7^3 \text{mod} 15 = 13 \]
\[ p = 4: 7^4 \mod 15 = 1, \text{ therefore } p = 4. \]

3. If \( p = N-1, N \) is a prime number. Stop.

4. If \( p \) is odd, go back to step 1.

5. If \( X^{p/2} + 1 = 0 \mod N \) go back to step 1.
   e.g.,
   \[ 7^2 + 1 = 50 \mod 15 = 5 \]

6. Compute a factor: \( f = \text{GCD}(X^{p/2} - 1, N) \)
   e.g.,
   \[ f = \text{GCD}(48, 15) = 3; \]

7. The factors are \( f \) and \( N/f \).
   e.g., 3, and 15/3 = 5;

(*) [https://www.geeksforgeeks.org/shors-factorization-algorithm/](https://www.geeksforgeeks.org/shors-factorization-algorithm/)

This discovery kicked the field into high gear because it was the first practical application of quantum computing. But it has flaws. First and foremost, it requires large numbers of qubits for ordinary numbers used by encryption and related applications. Typically, millions of qubits are needed to factor large numbers commonly used by public key security. Only three-digit numbers have been factored to date.

Although Shor demonstrated the realization of a full-scale quantum computer has the potential to provide a truly significant increase in computing speed, the main contribution was to spark a great deal of interest in the design of quantum algorithms that endures today. Shor showed it could be done even if not practical.

According to Shor, quantum algorithms were potentially much more powerful than classical algorithms because their time complexity scaled as the logarithm of problem size while classical computers scaled exponentially, see Figure 4. In addition, quantum computers may be able to surpass classical data parallel computing for extremely large problems. This is known as “quantum supremacy.”
Figure 4. General algorithm performance of classical versus quantum computation. Classical algorithms take exponential time while quantum algorithms typically take logarithmic time.

A Physicist Walks into A Bar

It is the late 1990s and the bar is located somewhere near the famous Bell Labs in Murray Hill, New Jersey. A man dressed in a tan casual sports jacket and white shirt without a tie steps up to the bar and orders a Diet Coke. He is an Indian American with advanced degrees in physics from the best schools in India (Indian Institute of Technology, 1981) and America (Stanford University, 1985). Nobody notices and nobody cares because nobody ever heard of Lov Grover—until much later.

Within a decade, Grover would become a superstar of quantum computing. John Phillip Preskill, an American theoretical physicist and professor of theoretical physics at the California Institute of Technology, said, “If quantum computers are being used 100 years from now, I would guess that they will be used to run Grover’s algorithm or something like it.” Grover’s algorithm is the starting point of many quantum computing algorithms, today. It illustrates the analog nature of quantum computing whereby a physical analog of an abstract mathematical problem is simulated in a quantum state space.

Back at the bar, I begin by asking what Grover’s algorithm is and what it does. He turns to me and says, “Well, that is a mouthful. It may take some time and more than one diet cola. My algorithm searches a list of numbers for a certain value and returns its location. But, instead of taking $O(N)$
time, it exploits superposition and entanglement to find the value in $O(\sqrt{N})$ time. It holds the speed record for search.”

“Think of the numbers in a list to be searched as waves, each with an amplitude proportional to the value of the number. Likewise, the search key is a wave with amplitude. We want to transform the list of waves in a way that enhances the amplitude of the desired wave so it “sticks out” compared to the others. This is done by iteratively using the search wave to enhance the waves in the list, increment-by-increment. After $O(\sqrt{N})$ iterations, the amplitudes of all but the desired item go to zero or near zero, leaving the amplitude of the search item greater than all others.”

“More precisely, the algorithm iterates $\lceil \sqrt{N} \rceil$ times over two quantum operations called oracle and diffuser, as follows:

1. Initialize the state space $|s\rangle$ with Hadamard gates (equal probabilities).
2. Repeat $O(\lceil \sqrt{N} \rceil)$ times:
   2.1. Transform $|s\rangle$ using oracle $U_f$.
       Applying the oracle to the qubits negates the amplitudes.
   2.2. Transform $U_f|s\rangle$ using diffuser $U_s$.
       The diffuser flips the probabilities about their mean value.”

“Simple, eh?”

Grover takes a cautious sip of cola directly from the plastic bottle and waits for my reaction. “I sort of understand what you are saying,” I lie. “Your algorithm made huge waves in the media. How does it work, really?”

“Well, first you need some hardware, but assuming one can build a quantum computer from an appropriate collection of qubits, you begin by initializing the spins of the protons with a strong magnetic field, aligning both of them in the UP direction. Then you give each particle a fainter dose of magnetism, just enough energy to change the spin state to a superposition that is 50 percent UP and 50 percent DOWN (a "50 percent flip").”

“Wait, you are going too fast,” I protest. “Suppose I have a list of eight numbers, and I want to use your algorithm to search for, say, number 3, which is 011 in binary. What happens to this list?” Grover breaks it down in simple steps, see Sidebar 3.
Sidebar 3: Grover’s Algorithm

List: 0, 1, 2, 3, 4, 5, 6, 7: Search for 3.

I am trying his patience with such a trivial problem. I find it difficult to imagine the list being in all states of solution space at one, but that is what it amounts to. That is, while coherent, all solutions exist at once—000, 001, 010, 011, 100, 101, 110, and 111—each with a probability determined by the wave’s amplitude when the quantum state collapses. Grover’s algorithm finds the correct location of 011 by enhancing the amplitude of the correct item and diminishing the amplitudes of all other items.

Grover exhales a deep sigh, takes out a pencil and pad of paper, and speaks as he writes, “You can encode this in three superposition qubits since each of the eight states has a probability of 1/8 and amplitude of \( \frac{1}{\sqrt{8}} = 0.353 \). In Dirac’s bra-ket notation this is:

\[
.353 |000\> + .353 |001\> + .353 |010\> + .353 |011\> + .353 |100\> + .353 |101\> + .353 |110\> + .353 |111\>
\]

Now comes the heart of the algorithm. The oracle changes the sign of the amplitude of the target state \( |011> \) to minus 0.353:

\[
.353 |000\> + .353 |001\> + .353 |010\> - .353 |011\> + .353 |100\> + .353 |101\> + .353 |110\> + .353 |111\>
\]

That is possible because, in a sense, when the quantum computer is in the target state (and all others at the same time), it can verify that it is indeed in the right state and can then invert the phase in that state. Note that the state space contains all states (answers) at once, because the probabilities—that is, the squares of the amplitudes—remain coherent and in superposition.”

This seems a bit of hand waving to me, but I am not a physicist, so I let it pass for now. My rudimentary understanding is that \( |011> \) simultaneously compares each of the superposition values at once, but doesn’t tell anyone, yet, because the probabilities—that is, the squares of the amplitudes—remain coherent and in superposition.

“Next the diffuser transformation replaces amplitudes with an inversion about the average. If you imagine the average value as a crossbar whose height is equal to the average value of the amplitudes, with the various individual amplitudes jutting up or dangling down from it, you invert each amplitude by flipping it over to the opposite side of the bar,” Grover mumbles and writes some numbers barely legible on his pad.

Looking over his shoulder, I watch as he computes the transformation on my list of eight numbers. “The average amplitude is 6(.353)/8 = .264 because the negative .353 cancels one of the positive .353s. The diffuser adjusts the positive amplitudes .353 - .264 = .089 and for the negative .353 the adjustment is .353 + .264 = .617, so we add .617 + .264 = .881 and get the revised amplitude, which is .881^2 = .776 in terms of probability,” he plows on.
“Now we have to adjust the probabilities of the other qubits by dividing up the remainder after subtracting .776, This is $\frac{1 - .776}{7} = .032$ or $\sqrt{.032} = .178$ in terms of amplitude. This yields the next iteration, in terms of bra-ket notation.”

$.178|000> + .178|001> + .178|010> + .881|011> + .178|100> + .178|101> + .178|110> + .178|111>

The cloud of magic lifts as Grover recites, “The third and final iteration of the transformation yields the most probable result of the search. The average amplitude is $\frac{7(.178) - .881}{8} = .046$. The adjustment to amplitudes are $.178 - .046 = .132$ and $.881 + .046 = .927$, respectively. Thus, the revised value of the negative amplitude is $.927 + .046 = .973$, or $.973^2 = .947$ in terms of probability. Adjusting the positive amplitudes accordingly, $\frac{1 - .947}{7} = .0076$, or .087 in terms of amplitude.”

$-.087(|000> + |001> + |010> + .973|011> - .087(|100> + |101> + |110> + |111>)$

“When coherence collapses, the answer is 011 with probability $(.973^2) = .947$.”

Sure enough, the probability of the matching item in the list increases while all others decrease. After three iterations we get the answer. Convergence to the search item 011 is shown pictorially in Figure 5. Amplitude of the search item 011 is transformed as follows: $.353 -> .353 -> .881 -> .973$, while all other amplitudes diminish: $.353 -> .178 -> -.087$. The probability of the search item being found at location three converges in three steps: $.125 -> .881 -> .955.$

Figure 5. The diffuser transformation increases the amplitude of the matched item in a list while decreasing the amplitude of all other items.

************ End Sidebar 3 ******************
I protest, “Of course one can use a classical data parallel computer to search in parallel. The search key (3) and one item of the list is distributed to $N = 8$ processors where they simultaneously compare one data point with the key. If it matches, they return 1, otherwise 0. This is also very fast and does the search in one clock cycle.”

Grover replies, “Yes, but you have to consider the time it takes to move each of $N$ data points to separate processors and include the cost of $N$ processors. With quantum computing, the duplication is in the quantum space, instead of additional hardware and there is no need to move data. For very large lists, the time required to move lots of data points and the cost of hardware can be prohibitive. Suppose $N$ is 1,000,000! Who can afford a million processors?”

Grover finishes his second diet cola and relaxes while his explanation sinks in. It has been a good day for learning about quantum algorithms, but I am not convinced they are good for any practical problems. But then, what do I know?

A Successful Failure

If you were a promising student in the Roaring 20’s (1920–29) looking to make a name for yourself in science, the obvious choice was physics. Einstein and the traveling cohort that promoted the Copenhagen interpretation were the rock stars of science. If you were lucky enough to study under someone as great as Arnold Sommerfeld or J. J. Thomson, you were destined for fame. That is how Wilhelm Lenz (1888–1957) felt about it, anyway.

Lenz was one of the lucky ones who studied under Sommerfeld from 1906 to 1911 and then became his assistant until the young Lenz was recruited into the German army as a radio operator in 1914. He returned to academic life in 1920, where he trained Wolfgang Pauli, Pascual Jordan, and Ernst Ising. Together, Lenz and Ising rose to fame for the Lenz-Ising model of ferromagnetism in 1925.

Years later I interviewed Ernst Ising (1900–1998) and asked him about Lenz and others in the Copenhagen conspiracy. “What was it like working with the great atomic physicists during the 1920s,” I inquired? By now, Ising was showing signs of aging, with gray hair and thick spectacles. He was dressed meticulously in a sports suit and tie.

“Oh, it was quite exciting,” his enthusiasm showing as he spoke. “Germany was the place to be during the quantum revolution, at least until the Nazis took over.” He and his wife survived WWI but got out of Germany in 1938 and emigrated to the USA in 1947. “I never got involved in politics, but the war was trying for us.” I sensed he did not want to talk about it. After all, he is a gentle
human uninterested in politics. People who knew him said, “He was a sensitive, artistic man who loved travel and the arts. He had a keen mind and sharp sense of humor, but was a gentle, quiet individual who always seemed a little shy when questioned about his famous model” [6].

I had heard of the Ising model but really didn’t understand it until it became an important model for solving optimization problems using a quantum computer. Of course, Ising died before his model became a key ingredient in quantum computing. He studied ferromagnetism instead of computing. So, I avoided the reference to computing and asked him about ferromagnetism. “You and Lenz invented the model to explain magnetism, correct?” He was embarrassed to talk about himself.

“Yes, I was his student in 1922–1924, and earned my doctorate from the University of Hamburg in 1924 under Wilhelm. He was rather emphatic that I do my dissertation in the new field of atomic physics, so I complied. Atomic physics was all the rage in 1922.” The Copenhagen gang was in the midst of formulating quantum theory, but most people were unaware of the radical nature of the Copenhagen interpretation.

Prior to the discovery of atoms and electrons, it was believed “currents in matter” gave rise to magnetism. It was never very clear how these currents worked or what they were made of, but it was the only explanation without a theory of atoms. But confirmation of atoms as the building blocks of matter opened up an entirely new avenue as wide as a 16-lane turnpike.

“How does it work,” I asked. Maybe he was unaware that the Lenz-Ising model has been referenced by others more than 16,000 times and still more than 800 times each year. After all, he became somewhat obscure after 1948 when he accepted a physics professorship at Bradley University, in Peoria, Illinois. He spent the rest of his life teaching physics.

“Well, it is actually very simple, really. I don’t know what the fuss is all about, especially in today’s age of computing. Computers did not exist in 1922, so everything was done with math,” he became animated with his hands waving in the air. As it turned out, the math was too difficult even for Ising.

“Imagine you have a bunch of atoms, strung together like pearls in a necklace. The atoms behave like magnets, with north (+) and south (−) spins or poles. Initially, they point north or south randomly, as shown graphically,” he takes out a sheet of paper and draws boxes containing plus or minus signs representing polarization. Roughly 50 percent point north, and 50 percent point south, initially.” (See Figure 6a).
Ising continues, “With roughly an equal number of atoms pointing north and south, the necklace is in a state of maximum disorder—what physicists call entropy. If we want to create a magnetic field from this disorder, we must arrange the atoms, so they all point north, or all point south. This takes energy. My dissertation attempted to mathematically explain how order can emerge from disorder, thus giving the necklace a kind of magnetism,” he paused to let that sink in.

“One wants to understand the relationship between atoms and ferromagnetism. The theory in 1922 was unproven, [to] be fairly obvious—magnetism appears when the atoms all orient themselves in the same direction—either north or south. I derived a mathematical equation showing under which conditions the atoms align themselves thus giving rise to magnetism. When iron comes under the influence of a strong magnetic force, the atoms align themselves in the direction of the external field.” He set aside the pencil.

I tried to summarize, “Your mathematical model used classical physics to explain how energy from an outside applied field could transform the disordered atoms into alignment, thus showing how ferromagnetism arises from random atoms. Is that it?” He appeared distressed by my clumsy explanation.

“Well, I used something called the Hamiltonian which is math for potential and kinetic energy in a system. As disorder declines, so does the Hamiltonian.” Figure 6b illustrates the change in the Hamiltonian at time t-1 versus time t—a classical state diagram. In the 1D case, disorder never dissipates. The state diagram cycles through different values of the Hamiltonian but never settles down. This was not the result Ising wanted.

Unfortunately, Ising was unable to show that his model worked for the 1D case shown in Figure 6. His dissertation, published in 1925 was disappointing, because it solved the 1D case but gave a negative result. Another student assistant of Lenz, Lars Onsager (1903–1976) presented a modified model based on a different Hamiltonian formulation of the Ising model and showed that it worked for 2D (see Figure 7). That is, in two dimensions convergence to all + or all – polarization is possible under certain conditions. This is called a phase transition and occurs if the Hamiltonian can be minimized.

Regardless of the failure, the Lenz-Ising model retains its name after Lenz and Ising, when in fact, the model was perfected by Onsager and others. The trick was in correctly defining the Hamiltonian in a ferromagnet. Potential is defined as the difference in polarization of adjacent atoms, rather than strictly the direction of polarization. In quantum physics, the Hamilton is defined as the total energy of a system—potential and kinetic—without regard for a coordinate system.
The Hamilton of an Ising system is the sum over all pairs of polarized atoms with value $s_i = \pm 1$:

$$s_i s_{i \pm 1}$$ for 1D,

Or sum over adjacent atoms above, below, left, and right of each atom:

$$s_i (s_{i \pm 1} + s_{j \pm k})$$ for 2D, where $j$ and $k$ are above and below $s_i$.

Clearly, the Hamilton increases with the number of adjacent atoms that differ in polarity versus when they are the same. Potential is embedded in the difference instead of the similarity of polarization.

Thus, for certain conditions, the 2D model converges to a minimum energy level when random atoms are repeatedly flipped such that the Hamilton decreases as shown in Figure 7b. The Hamiltonian is simple for a computer to calculate but the result is surprising. Start with random polarization (equal north and south atoms). Select an atom at random. Flip its polarity if the Hamiltonian decreases. If it does not decrease the Hamiltonian, flip it anyway with probability $P$.

$$P = e^{-\beta(H - H_0)}$$; $H, H_0$ are before and after Hamiltonians.

**************************** Sidebar 4: Ising Algorithm ****************************

A computer simulation algorithm for the modern version of Ising is given here:

Ising: Randomly flip polarities of all atoms (50% - 50%).
Repeat until $H$ reaches a minimum:

$$H_0 = \text{calcHamilton}();$$
Randomly select atom $i$
$k = i; \quad // \text{Save it for later}$
$i = -i; \quad \quad \quad // \text{Flip state}$
$H = \text{calcHamilton}()$
if($H > H_0$) {
$P = \exp(-\beta(H - H_0))$
    Restore state of $i$ to $k$ with probability $(1-P)$
}
End.
Simulated Ising models converge to a minimum Hamiltonian state for certain values of beta, and cycle forever, otherwise. Convergence is considered a phase change in physics, but it is merely convergence to a ground state where change no longer occurs. Thus, the Lenz-Ising model is a type of optimization model whereby the objective function is the Hamiltonian.

End of Sidebar 4

The Lenz-Ising model is important because it is a path to solving optimization problems using quantum computers. If a problem can be modeled as an analog of Lenz-Ising, then it can be solved by the simple Ising algorithm above. That is, the Hamiltonian in an analog quantum system is minimized, thus finding a ground state that corresponds with the optimization of the stated problem.

This is the idea behind many optimization algorithms that apply quantum superposition parallelism to run fast. Using superposition, all steps in the algorithm above are done simultaneously, driving the system to its ground state all at once. Once again, superposition is responsible for performance of the computation.

(a). Initially, atoms randomly point north (+1) or south (-1) with equal probability.

(b). Atoms flip due to the influence of neighbors but repeat Hamiltonian cycles as shown by the state space diagram.

Figure 6. The idealized Lenz-Ising model of ferromagnetism. (a). The 1-D model of Ising initially contains random north and south poles. (b). The 1-D model repeats cycles of north-south polarization as shown in the state space diagram.
Figure 7. Modified Lenz-Ising model using Hamiltonian as a measure of ferromagnetism. (a) 2-D model of initial state of random polarization. (b) Eventually, all atoms point in the same direction when $\beta$ is large enough.
The Quantum Traveling Salesman Problem

In quantum computing, finding a minimum value uses the Lenz-Ising model of minimum Hamiltonian as an analog. The key is to formulate the problem as a quantum energy minimization experiment that runs on a quantum computer. This is the approach used in many NP-complete problems such as the Traveling Salesman Problem (TSP). Minimum Hamilton energy is the analog of minimum distance traveled by the salesman.

Instead of polarized atoms and a Hamilton function defined by differences in polarized pairs of atoms, quantum fluctuations are introduced into a quantum system and coherent superposition is maintained long enough for the fluctuations to “settle down” and find the minimum. This was done for the TSP in 2004 by Martonak, Santoro, and Tosatti at ETH in Switzerland and the International School for Advanced Studies, in Italy [7]. The quantum Ising model is also used by D-Wave Quantum Systems Inc., a Canadian quantum computing company based in Burnaby, British Columbia. The D-Wave QPU uses quantum annealing to find the minimum of an energy landscape defined by couplings of qubits in the form of a Hamiltonian.

Martonak et al. divided the TSP into two Hamiltonian parts: one part computes the length of a tour, $H_{pot}$, and the other part, $H_{kin}$, is a suitable representation of the kinetic energy involved in quantum fluctuations. Quantum fluctuations are gradually dampened by shifting $s$ from one to zero in:

$$\text{Hamiltonian} = (1-s)H_{pot} + s H_{kin}$$

Initially, $s = 1$ so Hamiltonian = $H_{kin}$ is 100 percent quantum fluctuations. Eventually, $s = 0$, so Hamiltonian = $H_{pot}$, and the optimal tour is found.

The challenge is to devise an expression for quantum fluctuation, $H_{kin}$. $H_{pot}$ is simply the length of a tour. What is $H_{kin}$?

The selection of $H_{kin}$ is somewhat arbitrary. It could be simply transposition of neighboring cities, or transposing paths between cities. For example, tour,

$$1\rightarrow 2\rightarrow 5\rightarrow 8\rightarrow 7\rightarrow 6\rightarrow 3\rightarrow 4\rightarrow 1$$

becomes,

$$1\rightarrow 2\rightarrow 3\rightarrow 6\rightarrow 7\rightarrow 8\rightarrow 5\rightarrow 4\rightarrow 1$$
after transposing 3 and 5, and 8 and 6, noting that tours are bi-directional. $H_{\text{kin}}$ embodies transpositions that simulate quantum fluctuations.

When compared with simulated annealing on a classical computer, Martonak et al. found that quantum annealing converged in fewer steps. However, the algorithm presented by Maronak et al. was slower than highly optimized algorithms on classical computers. According to the authors, “We believe that gaining further experience with the effects of artificially introduced quantum fluctuations in classical complex problems represents a very promising and challenging route, particularly in view of future developments in quantum computation.”

**The Promise of Quantum Computing**

There remain major challenges to quantum computing. Error-correction and scaling to millions of qubits are the biggest obstacles. Decoherence is a constant threat. Quantum computers currently must be isolated from the world. This means they have to be cooled down to nearly zero degrees Kelvin, insulated against vibration, heat, and earth’s electromagnetic field, etc. In other words, they are impractical for common everyday use. But some important applications require them.

At this point we cannot tell if quantum computers are the beginning of a sea change in computing or whether it is a dead end. Alternatives are already being proposed as a challenge to quantum computers described here. For example, quantum dot computing uses properties of quantum dots rather than quantum states to represent information. Quantum dots represent bits more like classical computers except they employ entanglement to simulate a finite state machine. Instead of coherence and superposition, finite state machines operate like a classical computer jumping from state to state.

A quantum dot is a tiny semiconductor particle with quantum properties. When illuminated by ultraviolet light an electron is excited to a state of higher energy. This corresponds to the transition of an electron from the valence band (0) to the conductance band (1). The excited electron can drop back into the valence band (0), thus switching states. This is how solar energy cells work thus quantum dots are already a reality.

Quantum computing is a take on the 1940s style of computing using analog computers. The idea is simple—construct a physical analog of the mathematical problem you want to solve. For example, classical analog computers used analog circuits to solve differential equations representing a real-life problem. This is rather straightforward because electrical circuits are easily simulated by differential equations and vice versa.
Currently quantum computers are like analog computers in that the solution is found by simulating actions at the quantum level. Each problem must be modeled with a different and unique quantum analog. Thus, programming requires knowledge of quantum physics. This is a high price—in terms of expertise—to pay.

REFERENCES

[1] Wikipedia contributors. "Werner Heisenberg," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Werner_Heisenberg&oldid=1212247950 (accessed March 15, 2024).

[2] Freytes, H., Giuntini, R., Leporini, R., and Sergioli, G. Entanglement and quantum logical gates. Part I. International Journal of Theoretical Physics 54 (2015), 4518–4529.

[3] Schumacher, B. Quantum coding, Phys. Rev. A51 (1995), 2738–2747.

[4] Tutorial: Explore quantum entanglement with Q#. Jan. 12, 2024. Microsoft Learn.

[5] Paul Dirac: The purest soul in physics. Feb. 1, 1998. Physics World.

[6] Kuzemsky, A. L. Biography Of Ernst Ising. 2006.

[7] Martonak, R., Santoro, G. E., and Tosatti, E. Quantum annealing of the traveling salesman problem. arXiv:cond-mat/0402330v1. Feb. 12, 2004.

ABOUT THE AUTHOR

Ted Lewis is a retired professor of computer science interested in network science, social media, and emerging technologies, and has published more than 30 books on topics ranging from personal computing to complexity theory.

DOI: 10.1145/3650178