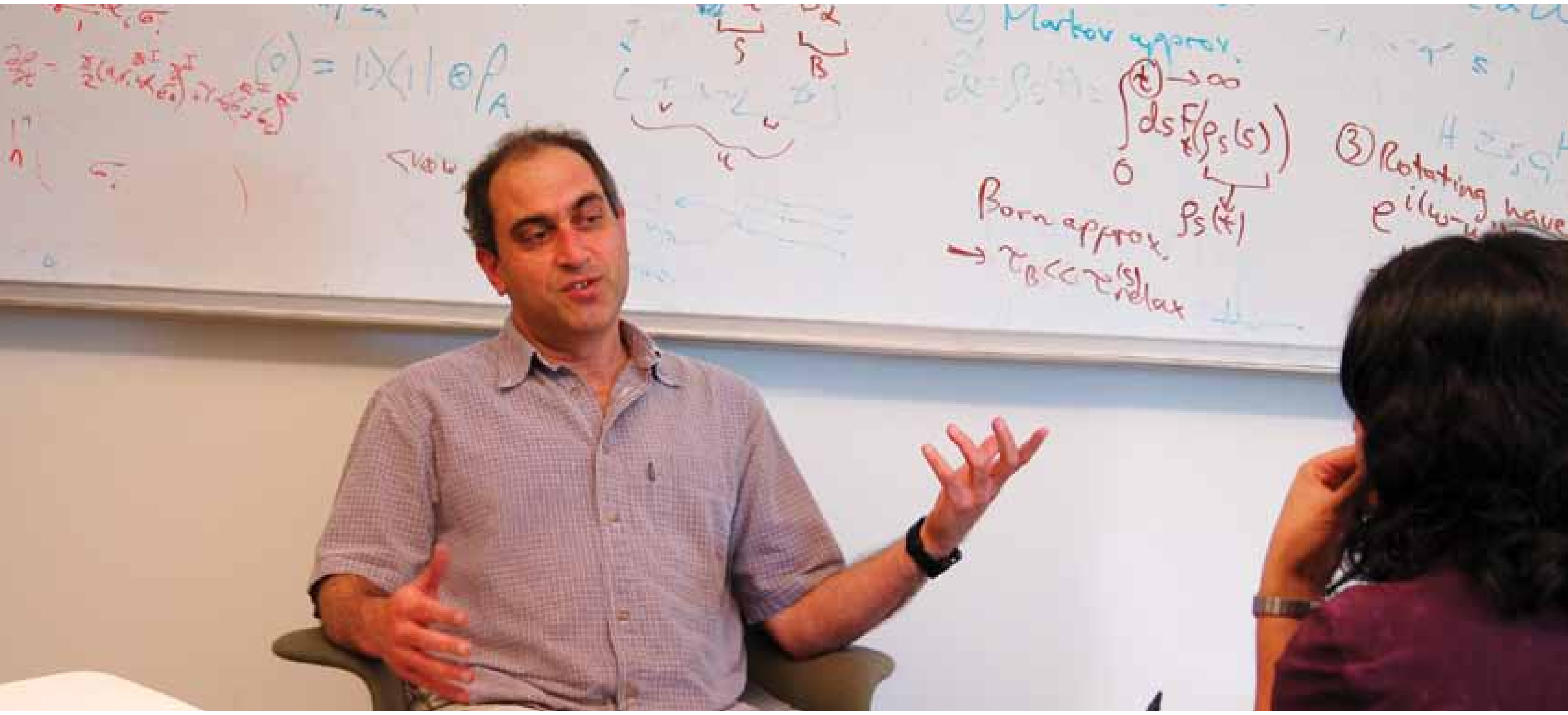

Quantum Algorithms

A Test for the Laws of Physics

by Leonard J. Schulman

The relationship between science and engineering is unequal. Science has custody of the noble truths; engineering is in charge of getting things done. One engineering proposal, however, defies this asymmetry. A successful quantum computer would verify as-yet untested predictions of quantum mechanics. Such verification is not a foregone conclusion.

Forty-five years ago, when computers began to enter the academic and commercial world, researchers asked what types of problems could foreseeably be solved on them. As computers improved from year to year, it was clear that this question must be couched in a suitably abstract framework. Soon, the mathematician Jack Edmonds and others had focused on what we still consider a fundamental distinction: between the problems that can be solved in time bounded by a *polynomial* in the input size, and those that cannot. For computer scientists, the divide between problems solvable in polynomial time and all others is a useful, if rough, demarcation between the problems that are tractable (can be solved with reasonable effort) and those that are intractable.

Early computer scientists studied a wide variety of physical implementations of computing devices, but when they modeled these mathematically, they discovered that the class of problems solvable in polynomial time never changed, and they called this class P. (Some prefer a variant, BPP, but we digress.) The class P could be perceived, then, as a property of physical reality—a limit on the computational power of physical devices. Although the concept of polynomial time was originally formulated to answer an *engineering* question, the class of problems P was quickly absorbed into a *scientific* assertion about what is physically possible in our universe.

One problem that seemed to be intractable, or outside of polynomial time, is this: given a whole number, find its factorization into primes. This task has intrigued mathematicians at least since the early nineteenth century. Confidence in its intractability was so strong that in the 1970s, in work for which, in 2002, they were given the Turing award (the highest award in computer science), Ron Rivest, Adi Shamir, and Leonard Adleman invented a cryptosystem (RSA) whose security depended on this assertion [A method for obtaining digital signatures and public-key cryptosystems. *C. ACM*, 21:120-126, 1978]. Today their cryptosystem is widely used for commercial and other transactions.

Early in the 1980s the physicist Richard Feynman observed that computers were having difficulty with another kind of computation: simulating quantum mechanical dynamics [Simulating physics with computers. *International Journal of Theoretical Physics*, 21 (6/7):467-488,1982]. This may at first seem unremarkable: all manner of physical processes, such as the weather, are hard to simulate. But Feynman's difficulty was altogether greater. In the mathematical theory of quantum mechanics, the number of parameters needed to describe a many-particle system grows *exponentially* in the number of particles. This is because each particle of the system is, to varying degrees, in each of its possible states at once—what is called a "superposition"—and because to write down the state of the whole system we need to keep track of each way of *combining* the states of all the particles. As far as we know, the system cannot be simulated without doing so. The problem of simulating quantum dynamics seems to lie far outside of P.

What quantum computation has done is to take Schrödinger's improbable feline spectre out of the realm of theory and into the arena of testable experimental predictions.

Now, something about this picture is suspicious. Many problems are hard to compute, but the problem of simulating quantum dynamics should not be on that list. After all, the universe performs these computations all the time—in real time. What gives? Feynman suggested two possible resolutions. The first is:

(1) There is some clever, mysterious way of computing quantum mechanical simulations that doesn't require writing down and updating all those exponentially-many parameters.

Feynman couldn't think of one, nor has any other physicist, computer scientist, or mathematician. Indeed, the possibility seems to run counter to how quantum mechanics works. For the remainder of this essay, we'll dismiss the possibility.

Feynman's second suggestion was:

(2) Devices operating on the principles of quantum mechanics have inherently greater computational power than those operating on the principles of classical mechanics.

Feynman did not have the mathematical framework (known as complexity theory) to take possibility (2) further, but a decade later, the computer scientists Ethan Bernstein and Umesh Vazirani did [Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411-1473, 1997. (STOC 1993)]. They were able to show (under certain abstract assumptions) that the class of tractable (polynomial-time solvable) problems is indeed greater in a quantum-mechanical world than it would be in a classical world. This is a deep *scientific* statement about what is or is not physically possible in our universe. Within a year, the computer scientist Peter Shor had derived from it a great *engineering* accomplishment: a polynomial time algorithm for factoring numbers on a quantum computer [Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Computing*, 26:1484-1509, 1997. (FOCS 1994)]. So it turns out that polynomial time, in a quantum mechanical universe, is adequate to solve problems that seem to require far *more* than polynomial time if you rely only on *classical* physics processes.

To date, however, the prototype quantum computers that have been built are very limited. Shor's algorithm, to be useful, must be run on a quantum computer large enough to produce, and maintain and manipulate over an extended time, particular kinds of quantum superpositions involving (at least) hundreds of particles. Superpositions like this have never been observed. (Which is why we still get away with using RSA.) Indeed, the prediction that they exist has troubled physicists since the inception of quantum theory. Erwin Schrödinger, a founder of the theory, memorably told of a (hypothetical) cat in a simultaneous superposition of two states: alive and dead. The whole point of this image is that it is ridiculous—nothing as complex as a cat has ever straddled reality so delicately. Yet subatomic particles are always in superpositions, and quantum theory knows no size limit: what it prescribes for particles, it predicts for cats...and for computers.

What quantum computation has done is to take Schrödinger's improbable feline spectre out of the realm of theory and into the arena of testable experimental predictions. Since the computational implications of these predictions are remarkable, it behooves us to consider an alternative remarkable possibility—that a quantum computer of a useful size is a physical impossibility, that large numbers cannot be quickly factored, that Schrödinger's cat was never in danger—in short, a third possible way of resolving Feynman's conundrum:

(3) Quantum theory is incorrect for large, complex systems.

Large quantum systems are so hard to control in the laboratory that our theory for them is only an extrapolation of what we know for small systems. Like earlier extrapolations—Newtonian mechanics, which Albert Einstein revised at high velocities, or the flatness of the earth, which the ancient Greeks revised at large distances—it might be wrong. Quantum computers, as computers, will probably not be useful until they contain hundreds of "quantum bits" (basically, particles involved in the computation). As experimental tests of quantum mechanics, however, they are already charting new terrain: recent experiments have reached a dozen quantum bits.

Where do we stand? Quantum algorithms—nothing but engineering designs—are so powerful that they pose a test to the laws of physics. In a manner of speaking, these algorithms have given teeth to Schrödinger's troublesome cat, who is forcing us to discover something startling about the reality we live in. What will it be? Scenario (2), in which our previous understanding of computation is revealed to have been fundamentally flawed? Or scenario (3), in which our understanding of physics turns out to have been no better? The only test is in the laboratory. ENG

Leonard Schulman is Professor of Computer Science
Visit: *http://www.cs.caltech.edu/~schulman*