Aspects of Computability in Physics

Joseph Shipman

Abstract

This paper reviews connections between physics and computation, and explores their implications. The main topics are computational "hardness" of physical systems, computational status of fundamental theories, quantum computation, and the Universe as a computer.

Introduction

In the last decade many connections have been discovered between physics and the theory of computation, illuminating both subjects. This paper reviews these diverse connections and explores some of their implications.

The first section explores the computational "hardness" of physical systems -- the difficulty of predicting the evolution of a dynamical system, or more generally the result of an experiment, to various degrees of accuracy. We will deal with results from classical dynamical systems and the classical theory of computational complexity (time and space complexity), and then explore some more speculative connections involving quantum-mechanical systems and the algorithmic information theory of Kolmogorov and Chaitin, with implications for the foundations of mathematics. We will discuss Church's thesis and define and assess several complexity-theoretic analogues.

Section II deals with the computational status of fundamental quantum field theories, especially Quantum Electrodynamics (QED). The difficulties in formulating these theories on a sound algorithmic basis are examined, and some related mathematical issues are treated.

The third section of the paper reviews recent work in Quantum Computation by Feynman and others and the implications for complexity theory.

In the final section we discuss the question "Is the Universe a Computer?", an affirmative answer to which has been argued by Fredkin and Toffoli, and the more general question "Why is computation (as we know it) possible?".

I. Computational "Hardness" of Physical Systems

A. What does it mean to "solve" a dynamical system?

In the last century much effort was spent trying to find "closed form" solutions for dynamical systems. The classical example was the "n-body problem", in which a system of n point masses obeying an inverse square law of attraction is to be "solved". The ultimate objective was to prove that the solar system was stable for some suitable notion of stability.

But what is the meaning of "closed form solution"? Originally, it meant that the dynamical variables could be expressed as a function of the initial conditions and time, where the function involved was a "known" function. The repertoire of "known" functions included trigonometric and exponential functions and was augmented over the years by "special functions" such as the gamma function, Airy and Bessel functions, elliptic functions, and so on, as well as any function obtainable from these by standard operators such as differentiation, integration, and functional composition and inversion. Ideally, the value of a variable at a given time could be obtained by "plugging in" the initial conditions and t to the function.

The next step was the definition of an "integrable (dynamical) system". This is a system in which sufficiently many "constants of the motion" (that is, independent sufficiently smooth functions of the initial conditions which are invariant on all trajectories) can be defined that the change of variables given by the implicit function theorem transforms the system into a

0-8186-3420-0/93 $03.00 © 1993 IEEE

trivial one.

In the n-body problem, there are 10 classical "constants of the motion", also known as "first integrals", corresponding to the laws of conservation of energy (1 integral), momentum (6 integrals), and angular momentum (3 integrals). When n=2, this reduces the system to one which can be solved explicitly.

The change of variables will not, in general, be given by a well-known function, but will presumably be fairly smooth and computable. Again, the point is that after the initial change of variables the solution is a simple function easily computed for all t.

The use of the word "integrable" here is interesting. The correct analogy is not with a (Riemann) integrable function from R to R, but with a function "integrable" in the freshman calculus sense: f can be "integrated" if you can find an antiderivative which can be evaluated at the endpoints of the interval, avoiding tedious numerical integration.

From the perspective of modern computation theory, what was sought for, say, the n-body problem, was a practical way of computing the state of the system at time t for given initial conditions. They certainly knew how to compute this by numerical integration, but numerical integration is slow. It requires computational resources (running time for a single processor, or running time * number of processors for a parallel computation) proportional to t. Actually, the resources required for a numerical integration might be even greater, say (t * log t), because of the computational overhead involved in keeping track of the computation and bounding round-off error.

In principle, the "running time" (in a broader sense) is never going to exceed t asymptotically for many dynamical systems because you can just set up the system and watch it evolve. The types of systems for which such a simulation is possible have been investigated by Pour-El and Richards ["Abstract Computability and its relation to the General-Purpose Analog Computer", Transactions of the A.M.S. 199 (1974), 1-28].

In addition to computing where the system is at time t, one might want to know some long-term qualitative feature of its behavior, such as whether a certain region of the phase space will ever be visited for a given initial condition. In an n-body problem, the question might be whether there is a "gravitational escape", where one of the "planets" moves arbitrarily far away from the others as t goes to infinity. This type of question is straightforward for an integrable system, when the "first integrals" are computable easily, but difficult in general because a numerical integration will only take you out to time t but tell you nothing about what happens after that.

For the n-body problem with n > 2, Poincare showed [Acta Math., 11 (1887)] that the classical integrals corresponding to energy, momentum, and angular momentum are the only ones. Thus the solar system (in the Newtonian approximation) is not an "integrable system" and there is no "solution" to the n-body problem in the classical sense.

However, just because a system isn't integrable does not mean that there is no practical way of "solving" it computationally. Let us make some definitions:

Definition. A dynamical system is "weakly computationally solvable" if there is a Turing machine algorithm to compute the values of the dynamical variables as a function of the initial conditions and time, such that the running time of the algorithm is o(t). It is "strongly computationally solvable" if there is such an algorithm whose running time is polynomial in (log t). ("Compute the values of the dynamical variables" means provide arbitrarily many decimal places of these real numbers, asking for additional digits of precision of the initial conditions as necessary.)

The first notion of solvability says that the system can be "speeded up": there is a way to predict the behavior of the system faster than simply watching it go. The second notion says that the prediction algorithm is a feasible computation in the sense that the running time is polynomial in the size of the input data (to specify the time t requires an amount of information proportional to log t). This implies, of course, that only a polynomial amount of information is needed to specify the initial conditions precisely enough: see the remarks on chaotic systems, below.

(Remark: The identification of "polynomial-time computation" with "feasible computation" is slightly controversial. An O(n^100) computation may not be very feasible in practice, and the recognition of prime numbers is considered a feasible problem even though it has only been shown to be in "random polynomial time". Since all known non-contrived polynomial-time problems actually have fairly low-degree polynomial algorithms, and since "random polynomial time" is almost certainly equal to "polynomial time" [J. Shipman, "Why P=R is extremely likely to be true", in preparation], we will treat "feasible" and "polynomial-time" as synonymous. End of remark.)

Definition. A dynamical system is "computationally predictable" if there is an algorithm which, given an initial condition and a region of phase space, decides whether or not the resulting trajectory ever passes through that region (and always halts except when the trajectory passes through the region's boundary but not its interior; in this exceptional case the algorithm would have to ask for infinitely many bits if it were to perform correctly for nearby initial conditions).

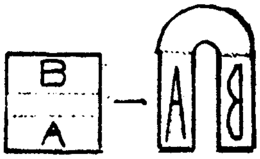
This is a strong condition. It means that you can say something about the behavior of the system for all time, not just compute where it is at any given time. The 2-body problem is computationally predictable because the orbit structure is completely understood. This condition is not meaningful for ergodic systems, where almost all trajectories are dense in the phase space.

In principle, dynamical systems which are not integrable may still be computationally solvable or predictable. The famous Kolmogorov-Arnold-Moser theorem states that for sufficiently small and smooth perturbations of an integrable Hamiltonian system, most trajectories remain confined to "invariant tori", behaving in a "quasi-periodic" fashion. [Moser, "On the theory of quasi-periodic motion", SIAM Review (1966)]. For this type of system it is often possible to say something about the long-term behavior (computational predictability) and achieve a speedup in computing the behavior over a finite time interval (computational solvability). The important question is whether the perturbation is small enough that the KAM theorem applies: this is often difficult to prove, so that in practice it may be very hard to achieve these computational goals. For example, Arnold has shown that for sufficiently small planetary masses and orbital eccentricities and inclinations the motion of an n-body gravitating system is quasiperiodic for most initial conditions, but his theoretical bounds are exceeded in the case of our own solar system. [Arnold, "Small denominators and problems of stability of motion in classical and celestial mechanics", Uspehi Mat. Nauk, 18 (113) (1963), 13-40].

B. Harder systems: Chaos (sensitive dependence on initial conditions)

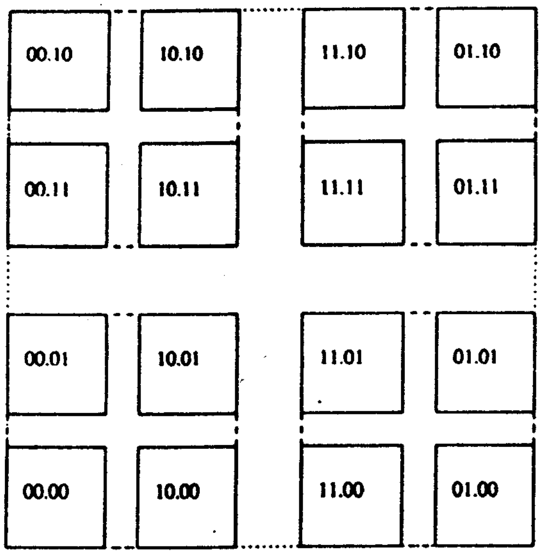
What happens when the system is not integrable or close enough to integrable that the KAM theorem applies? A great deal of work, beginning with Poincare and continuing today, has been done on the structure of these dynamical systems. M. Berry's article ["Regular and Irregular Motion", AIP Conference Proceedings Vol. 46 (1978), American Institute of Physics, New York] is a good review of this work.

The most important recent work has investigated "chaotic systems". It has been shown that many natural systems exhibit "chaotic" behavior, which can be defined for our purposes as "sensitive dependence on initial conditions". This means that in certain regions of the phase space, trajectories which are initially close diverge exponentially. Thus, to say where the system is at time t, you need to know the initial conditions very precisely: specifically, you need to know f(t) digits of the initial conditions where f(t) is proportional to t. The classic example is Smale's "horseshoe map" [Smale, in "Differential and Combinatorial Topology", edited by S.S. Cairns (Princeton Univ. Press, Princeton, NJ 1963) pp. 63-80].



{Diagram 1 : horseshoe map}

Any system which contains somewhere in its phase space a region which after some time interval is stretched, folded, and mapped back into itself as in the "horseshoe" diagram contains an invariant Cantor set on which the dynamics is chaotic. In particular, each point on the Cantor set can be given an "address" consisting of a two-sided infinite binary sequence (see diagram 2),



{Diagram 2 : invariant Cantor set}

so that the horseshoe map (equivalent to letting the system evolve for a fixed time interval) acts on the address as the "shift map":

$$...a_{-3}\ a_{-2}\ a_{-1}\ .a_0\ a_1\ a_2... \text{-->} ...a_{-2}\ a_{-1}\ a_0\ .a_1\ a_2\ a_3....$$

Clearly such a system is not "computationally solvable", since you need an amount of inital data proportional to t to determine where the system is at time t. On the other hand, the system (restricted to the invariant Cantor set) is trivially easy to simulate: to tell where a point will be n time units from now (the unit is the length of the time interval involved in defining the horseshoe map), simply shift the "address" of the point left by n bits. It is not much more difficult to compute the behavior at non-integral times, because you can numerically integrate the flow over the fractional part of the time interval after doing the shift; this numerical integration takes a constant amount of time for all t (for a given amount of precision in the answer).

The system is not "computationally predictable" because to tell whether a certain region of the phase space is ever visited corresponds to knowing whether a certain bit sequence occurs ANYWHERE in the "address" of a point. For an arbitrary initial point which you are given one bit at a time, there is no way to tell except by continuing to look until you see the sequence, but if it isn't there you'll never know for sure. For particular initial conditions which are known EXACTLY, such as a point with rational coordinates (both bit sequences a_{-1} a_{-2}... and a_0 a_1 a_2... are eventually periodic), on the other hand, the question of whether a particular region of phase space will be visited can be easily answered.

Although the type of system described above is chaotic, it is not complex: the dynamics are given by a very simple map. The computation of a trajectory cannot be speeded up, but the structure of the dynamics (identifying periodic points and basins of attraction, finding the rate of divergence of trajectories) may well be easy to compute. Although chaotic systems may contain Cantor-set-like (or "fractal") "strange attractors", these features are often easy to define computationally. The long-term behavior of such a system may be easy to predict (in particular, the attractors and basins of attraction can be defined and computed).

C. Even harder systems: computational incompressibility.

A standard result in the theory of computation is that arbitrary computations cannot be "speeded up". In particular, the function f(i,t) "whether machine i (started on a blank tape) halts after running for t time steps" cannot be computed by any Turing machine whose running time is o(t). (Actually, a technical condition about the length of the description of machine i should be added here, or the result is trivial. Requiring (length[i])^2 < t should suffice.)

But Turing machines, as instantiated in real computers, are physical systems! Hence the behavior of a physical system which is a computer is impossible to predict in general (up to a linear factor) any faster than by "letting it run".
(Remark: Implicit here is a "Computational Complexity Church's thesis". I do not know of any official fomulation of such a thing, so I propose the following : "There is a polynomial p such that any function which we can 'compute in time f(input size)' can be computed by a Turing machine in time p(f)." The empirical evidence for this assertion is that all known "physically sensible" models of computation are at most polynomially faster than Turing machines. In fact, multi-tape Turing machines come within a logarithmic factor of real (single-processor) computers such as the IBM PC; the fastest physically sensible model known seems to be 3-dimensional cellular automata. We will discuss this more below. End of Remark.)

Furthermore, the long-term (t -> infinity) behavior of a computer is impossible to predict because of the unsolvability of the halting problem. This is very strong type of unpredictability: even when the "initial conditions" for the computer are known exactly, questions about whether a certain state will ever be reached are essentially unanswerable.

Wolfram [Commun. Math. Phys. 96, 15 (1984)] has suggested that this type of "computational irreducibility" is really a common property of physical systems. To formulate this in terms of classical dynamical systems, though, is difficult. It has long been known that Cellular Automata, a kind of discrete dynamical system, can do computation. However, the natural attempt to model Cellular Automata in a continuous dynamics gives systems with infinitely many degrees of freedom. The "billiard ball computer" of Fredkin and Toffoli has a natural continuous dynamics, but also has infinitely many degrees of freedom.

In an important advance ["Unpredictability and Undecidability in Dynamical Systems", Phys. Rev. Lett. 64, 2354 (1990)], Cristopher Moore has shown how a very simple dynamical system (a particle moving in a three-dimensional potential) can simulate a Turing machine. His construction involves "generalized shift" maps which resemble Smale's horseshoe map but can alter a finite number of bits arbitrarily as well as shift all of the remaining bits. There is an invariant Cantor set (under the transformation given by letting the system evolve for one unit of time), and the bits in the "address" of the state of the system act like the bits on the tape of a Turing machine under this transformation.

Moore's system has only three degrees of freedom but is computationally unpredictable in a very strong sense, even though the divergence of trajectories is subexponential and the system is not chaotic. It raises the possibility that relatively simple "real" dynamical systems may be computationally intractable, even in nonchaotic regions of the phase space.

One major problem with Moore's construction is that it depends on dynamical variables being meaningful to arbitrary precision. A simple Turing machine computation which uses, say, 100 squares of the machine's tape is going to correspond to a precision of 50 bits in the dynamical variables, but any real physical system is going to be subject to perturbations (e.g. from the gravitational influence of nearby objects) of this order which would ruin Moore's computation. Of course, this makes the system even harder to predict, but it remains desirable to come up with an example of a dynamical system which is robust under small perturbations but can still "compute" and is therefore computationally unpredictable.

One way of constructing such a system is by modeling, not Turing machine computations, but "counter machine" computations. A Turing machine can be thought of as a finite-state machine with a memory consisting of words in some alphabet (for a 1-tape Turing machine which uses the binary alphabet, the two "words" are the tape to the left and to the right of the currently scanned square, up to the outermost "1"). The permitted operations are reading, adding, deleting, or changing the letter at the beginning of a word (and branching depending on what the letter is).

A "counter machine" is simply a Turing machine with a unary alphabet. In other words, it is a finite-state machine with a memory consisting of a number of registers, each of which contains not a word in some alphabet, but a non-negative integer. The permitted operations are adding or subtracting one to a register, and testing whether it contains "zero". ("Subtracting one" from a register which is already 0 leaves it unchanged.) Counter machines can do everything Turing machines can do, but more slowly.

It is simple to simulate a Turing machine by a 3-counter machine: the first two counters represent the tape, "decoded" from binary to unary. Inserting, deleting, or reading the first letter of half the tape correspond to doubling, halving, and parity-testing the integer in the corresponding counter. These operations are easily performed by using the third counter to keep track. In fact, a 2-counter machine will suffice [Minsky, "Computation: Finite and Infinite Machines", Prentice-Hall, Englewood Cliffs, NJ 1967], but the construction is much more elaborate. A characteristic of counter machine simulations of Turing machines is that there is an exponential slowdown due to the continual decoding from binary to unary and back.

In another paper [J. Shipman, "Robust undecidability in systems with finitely many degrees of freedom", in preparation] I will show how to define a dynamical system with finitely many degrees of freedom which simulates a 3-counter machine. The "memory" is represented not by the bits of the dynamical variables, but by the positions of certain particles. A register containing the integer n is represented by a particle "n units away". Checking or changing the value of a register involves sending out a "test particle" and takes time proportional to the value, rather than unit time, so there is a "quadratic" slowdown on top of the exponential slowdown that comes from simulating a Turing machine with a counter machine.

Although no general way to exponentially speed up the behavior of counter machines is known, the system described above is certainly "strongly computationally solvable" because its behavior can be so greatly speeded up by running the corresponding Turing machine instead. However, for an appropriate Turing machine the "counter machine system" is computationally unpredictable because the counter machine halts if and only if the corresponding Turing machine does, so the "t -> infinity" questions are just as unanswerable as for the systems described in Moore's paper.

D. Hardest of all: Noncomputability

Are there systems which cannot be computationally simulated at all? The strong version of Church's thesis says that any mathematical function that we can "compute" by some physically realizable setup can be computed by a Turing machine. It is hard to see how a dynamical system specified by differential equations could NOT be tracked by a numerical integration, unless the functions in the formulation were already noncomputable (i.e. if one of the coefficients involved was a noncomputable real number). This doesn't seem physically plausible.

If we don't restrict ourselves to classical dynamical systems the possibilities are more interesting. For example, Pour-El and Richards ["Noncomputability in analysis and physics". Adv. Math 48 (1983) 44-74] have shown that the wave equation with "computable" initial data can have a "noncomputable" solution where the term "computable" here refers to real functions rather than integer functions. However, this doesn't lead to a violation of Church's thesis because their result does not lead to any physically realizable experiments (one would have to somehow specify initial conditions to infinite precision).

A big problem in attempting to show a physical system is "noncomputable" in some appropriate sense is the inherent uncertainty in our knowledge of physical constants. When the mathematical formulation of the system depends on physical constants whose value is empirically determined rather than defined from first principles, the precision of the predicted result depends on the precision to which the constants have been measured. Theoretically, this issue can be addressed by defining the "experiment" to include a procedure for measuring these constants to the precision necessary. Unfortunately, those procedures may ultimately depend on other physical constants, so that you end up having to include more and more complication in the "experiment". It may be possible to stop this process at some finite point, where each of the constants involved depends on the others in some measurable way and a "bootstrapping" process allows simultaneous increases in the precision to which they all are known [J. Shipman, "A theory of 'bootstrapping' greater precision in measuring physical constants", in preparation].

Another issue is whether "arbitrary precision" is physically realistic. If we want an experiment to give us an outcome that involves a "noncomputable number" as the value of some physical quantity, it has to be a quantity that is meaningful to arbitrarily high precision, because any finite amount of information can be coded into a Turing machine's program. However, it seems unlikely that "classical" physical quantities like length and electric charge can be meaningfully interpreted as having arbitrary precision, in view of the atomic nature of matter and the uncertainty principle. This suggests several points:

(1) We should deal with quantum systems rather than classical systems, because classical theories are approximations already and we should be dealing with the fundamental scale of matter and the uncertainty principle directly.

(2) We should look at systems containing only a few particles, so that we can specify the situation as exactly as possible and avoid approximations.

(3) The numbers we are measuring should be "dimensionless", since the numerical values of quantities which have a "unit" attached depend on physical constants and are difficult to interpret to arbitrary precision .

Dimensionless numbers ultimately come down to ratios or probabilities. In the last analysis, something must be counted, for example the number of times a particle which has two possible decay modes decays in each mode. The half-life of a radioactive atom is not dimensionless (it is expressed in seconds), but the ratio of a half-life to another half-life or the period of an atomic clock is, and we can obtain more bits of precision by repeating the experiment. There is no a priori limit on the number of bits we can obtain, although we would have to perform a trillion "experiments" rather than a million to get 20 bits of precision because of the statistical fluctuations. Counting the number of decays in a sample of a trillion atoms may be difficult, but in principle there is no problem.

Going back to the issue of noncomputability, we can say that a physical system is "noncomputable" if there is some well-defined procedure for extracting arbitrarily many bits of "information" from the system, such that the sequence of bits cannot be produced by any Turing machine. (It may be necessary to add a condition that the sequence is only obtained with some non-zero probability. In the case of a pure probability p, the rule "perform 2^(2.1n+10) trials before estimating the nth bit of p" guarantees a very good chance that fluctuations will never lead to the wrong answer.) The existence of a sequence not obtainable by a Turing machine is a standard result of computation theory; a famous example is Chaitin's number _O_ (Omega), representing the probability that a particular universal Turing machine halts (treating the contents of the semi-infinite input tape as the binary expansion of a real number between 0 and 1, _O_ is the measure of the set of input tapes which cause halting). [See G. Chaitin, "Algorithmic Information Theory", Cambridge University Press 1987.]

According to Church's thesis, no noncomputable physical systems exist: any sequence of bits we can generate in a well-defined manner can be obtained from a suitable Turing machine program. Geroch and Hartle ["Computability and Physical Theories", Found. Phys. 16, 533 (1986)] call a number "measurable" if there is a precisely specified experimental procedure by which a technician with an abundance of raw materials and time could generate estimates of the number to within any predetermined margin of error. In their terminology, Church's thesis says that all measurable numbers are computable. If a mathematically definable number like _O_ were part of a physical theory and accessible experimentally, then Church's thesis would be false. How could such a thing occur?

It is interesting to imagine how a non-computable real number (bit sequence) could arise in a physical theory. Non-computable bit sequences are usually defined as "recursively enumerable but not recursive" ("renotrec" is the ugly but standard abbreviation) sets, such as the set of TM's which halt (on blank input), the set of Diophantine equations with a solution, or the set of finite presentations of the trivial group (all with some standard coding into the set of natural numbers). These sets all have the property that anything in them can be reliably identified by a standard procedure, but non-elements may be impossible to definitively reject because any procedure which rejects only non-elements fails to halt on some inputs. Thus, you can run a TM until it halts, try out solutions to a Diophantine equation until you find one, and systematically generate all words equal to the identity element until you find that the group's generators are, but these enterprises are doomed to go on forever in cases where the desired property doesn't hold.

All known non-contrived instances of renotrec sets are recursively equivalent to the halting problem (and hence to each other), so it doesn't matter too much which we use if all we are interested in is the fact of their noncomputability. (It is also possible to define even harder sets, such as the set of TM's which halt on ALL inputs, which are not even recursively enumerable, but renotrec sets will suffice for our purposes). In terms of practical usefulness there is a big difference: from the first n bits of _O_ we can derive the halting behavior of the first 2^n TM's, but very, very slowly, while the first 2^n bits of an oracle for the halting problem solve this immediately. Thus, for renotrec sets there is a tradeoff between information density and "decoding time". (Also of practical importance is whether the number is available directly, as an "oracle" where getting another bit isn't much extra work, or indirectly, as a probability which must be estimated, with each successive bit costing two or even four times as much as the last one).

So how might a renotrec set enter into a physical theory? It's hard to see how Turing machine halting might be relevant. Diophantine equations look more promising, especially since it is known that we can restrict ourselves to Diophantine equations of bounded degree and dimension and still get renotrec sets: in particular, there is a "universal Diophantine equation" [M. Davis, "Computability and Unsolvability", Dover Publications 1982]. However, I don't know of any argument for the direct physical relevance of Diophantine equations.

A better candidate is the group isomorphism problem, of which the group triviality problem mentioned above is a special case (they are both equivalent to the halting problem). It is well-known that any finitely presented group can be obtained as the fundamental group of a compact 4-manifold; hence, telling whether two compact 4-manifolds are homeomorphic is at least as hard as telling whether two finitely presented groups are isomorphic, and in fact recognizing whether 4-manifolds are homeomorphic is equivalent to the halting problem. [A. Markov, "Insolubility of the problem of homeomorphy", Proc. Internat. Congr. Math. (1958), pp. 300-306.] Thus 4-manifolds cannot be "classified", and there is no clear way to perform a calculation which depends, say, on a sum over homeomorphism classes of 4-dimensional simplicial topologies. But this is exactly the kind of thing called for in some speculative theories of quantum gravity (see the Geroch and Hartle paper, above). In Quantum Electrodynamics, by way of comparison, we perform a sum over topologically distinct "Feynman diagrams"; we can do the calculation because in this case we can tell for sure whether or not two diagrams are equivalent. (Even granting this, there are still mathematical problems with the QED calculation; see the next section.)

Just because 4-manifolds can't be classified doesn't mean we won't find some clever way to add things up "all at once". The fact that the obvious try doesn't work doesn't imply that a (mathematically well-defined) sum is indeed non-computable. On the other hand, it might be possible to prove such a thing if there is a suitable computable "modulus of convergence" for the sum: use of this and an oracle for the value of the sum might allow one to distinguish between two otherwise indistinguishable 4-manifolds by deducing that the sum required contributions from both of them and therefore they could not be homeomorphic. This is very speculative, and much work needs to be done, but it certainly seems logically possible that a number from a physical theory may have a mathematical definition that allows us to prove it's a noncomputable but experimentally accessible number, and hence falsifies Church's thesis.

It is important to note that the philosophical significance of such a result does NOT depend on our being able to get arbitrarily many bits of information out of a system. As Chaitin has pointed out ["Godel's Theorem and Information", Int. J. Theor. Phys., December 1982], for any mathematical axiom system, any definable but noncomputable bit sequence embodies truths not obtainable within the system (for if the value of each bit were derivable within the system, the algorithm "generate all proofs until the value of the bit is settled" would compute the sequence, contradicting its noncomputability). Thus, for each noncomputable sequence there is some bit whose value is not settled by the ZFC axiom system, and we only have to go out this far with our physics experiments to derive a "new mathematical truth" unobtainable by purely mathematical investigation. Chaitin has also derived explicit bounds on the information implicit in axiom systems, in particular giving an upper bound on the number of bits of _O_ ZFC can determine.

Geroch and Hartle propose that a criterion for evaluating a physical theory ought to be that all its measurable numbers are computable: such theories are to be preferred, in the same way that theories which are simple, general, and "elegant" are preferred. They feel that a theory whose measurable numbers are not computable would represent an "inconvenience" but that it would still be possible to test the theory against experiment: "predictions are always in principle available. It is just that ever increasing degrees of sophistication would be necessary to extract those predictions. The prediction process would never become routine." In saying this, they are making the assumption that mathematics, with sufficiently many "new ideas", can provide ever-increasing knowledge about a non-computable set.

However, this implies that the reasoning processes involved go beyond any given axiom system. In particular, once the generally accepted mathematical axioms (ZFC or what have you) have exhausted their power to say anymore about the noncomputable number, mathematicians will start disagreeing about the appropriate extension of the axiom system and thus will disagree about the experimental predictions too. At this point it is no longer useful to talk about "making predictions to test the theory". If theory has passed all the tests so far, it makes more sense to assume the theory as a new axiom and start deriving new mathematics by resorting to experiment! This shows that mathematics is not logically prior to physics; for another result in that direction see the last section of my thesis [J. Shipman, "Cardinal Conditions for Strong Fubini Theorems", Trans. Am. Math. Soc. 321 (1990), pp. 465-481].

Hartle's version of quantum gravity theory is the only example I know of where noncomputability in a physical theory has been plausibly proposed. Kreisel [Synthese 29 (1974)] suggests that for sufficiently complicated molecules the ratio of two eigenvalues of the Schrodinger equation might be noncomputable, but Shankar ["Principles of Quantum Mechanics", Plenum Press 1980, chapter 13] argues that (in the nonrelativistic approximation) in theory we can numerically integrate the Schrodinger equation to get the eigenvalues to arbitrary precision. It is conceivable that in the full relativistic theory of a large, complex molecule noncomputable numbers may be involved, but this depends first on the computational status of quantum electrodynamics, which is treated in section II of this paper. Other attempts to show that something in a physical theory is noncomputable (see a paper of Komar [Phys. Rev. B 133 (1964), p. 542], or the recent book of Pour-El and Richards ["Computability in Analysis and Physics", Springer-Verlag 1990]) do not seem to lead to experimental ways to extract noncomputable numbers or functions.

In the absence of a mathematically tractable "Theory of Everything", which would allow us to do things like define the fine-structure constant from first principles, the prospect of extracting a non-computable number (bit sequence) from physical experiments seems remote. Nonetheless, the logical possibility of doing so tells us some things about our universe, as we will see in the last section of this paper.

II. The Computational Status of Fundamental Theories

A. Quantum Electrodynamics

Quantum Electrodynamics, or "QED", is the archetype of a successful physical theory. Richard Feynman, in his book "QED: The Strange Theory of Light and Matter" [Princeton University Press, 1985], calls it "the jewel of physics-- our proudest possession". It has been tested over an incredible range of conditions with no significant difference between theory and experiment: at the present time theory and experiment agree to the limits of their accuracy, which is about nine or ten decimal digits for certain experiments.

The key phrase here is "to the limits of their accuracy". It is easy to understand how the accuracy of our experimental setups may be limited, but in fact there is currently more uncertainty in our theoretical predictions than in our experimental results. According to Feynman, for example, the measured and predicted magnetic moment of the electron agree (as of 1983) to nine decimal places; the experimental value has an uncertainty of 4 x 10^(-11), while the theoretical value has an uncertainty of 2 x 10^(-10), about five times as much (uncertainty in the value of the fine-structure constant accounts for some but not all of this). All uncertainty in the values of physical constants (except the fine-structure constant), experimental control variables, phenomena ouside the theory [such as gravity and nuclear processes], and so on are subsumed in the former number: but how can there be comparable uncertainty in the theoretical value, which is just the result of a computation?

This last question has a surprising answer. The theory of Quantum Electrodynamics, amazingly successful as it has been in practice, does not have a fully satisfactory mathematical and computational foundation. We don't know why it works so well!

Oversimplifying slightly, we can say that theoretical predictions in QED are expressed as power series in the fine-structure constant, a dimensionless number which has an experimentally measured value of about 1/137.03597. As Feynman says, "All good theoretical physicists put this number up on their wall and worry about it....It's one of the greatest damn mysteries of physics: a magic number that comes to us with no understanding by man....We know what kind of a dance to do experimentally to measure this number very accurately, but we don't know what kind of a dance to do on a computer to make this number come out--without putting it in secretly!". As we saw above, there is not necessarily a way to compute this number, although if we could not then Church's thesis would be false.

However, the problem with QED is deeper than this. Even if we assume a particular value for this constant, uncertainty enters in another way. Technically, each coefficient in the power series is a sum of terms obtained from "Feynman diagrams", which are topologically distinct ways in which the event in question can occur, involving the emission and reabsorption of "virtual particles", and so on. The coefficient for the nth term in the power series is a sum over all the Feynman diagrams involving 2n couplings. The calculated value of the magnetic moment of the electron referred to above came from a calculation which looked at all diagrams with up to six couplings; this gave three terms of the power series. The calculation of the third term involved about 70 diagrams; Feynman says that a calculation of the fourth term deals with over 900 eight-coupling diagrams.

Furthermore, the calculation of the contribution of a particular diagram is extremely involved. Theoretically, it is supposed to represent an integral over the function space of all "paths" with the topology of the diagram, but there are great mathematical difficulties in defining this integral without having it come out equal to infinity. The problem was that as the "coupling points" approached zero separation, terms blew up. In 1948-49 Feynman, Schwinger, and Tomonaga proved that if the integral is only added up over "paths" whose coupling points get no closer together than a certain cutoff distance, consistent predictions of experimental results can be obtained (the intermediate numbers depend on the cutoff distance, but in the limit the predictions of experimental results do not). This process is called "renormalization", and it gave a well-defined way to assign a number to each Feynman diagram; the calculation of this number is still very complicated, involving about a hundred thousand terms for an eight-coupling diagram.

This explains the uncertainty in the theoretical predictions of QED (over and above the uncertainty in the value of the fine-structure constant): the computer calculations are so massive that round-off error limits the precision we can practicably achieve.

However, the real problem with QED is even deeper than this. What we actually have is a sequence of algorithms: the nth algorithm computes the first n terms of the power series, adding up the contributions of Feynman diagrams up to order 2n, using the complicated process of renormalization. But we do not know if this sequence of algorithms gives a converging sequence of predictions! Physicists call each additional term a "correction" to the previous sum, and these terms have been getting smaller because the extra power of the fine-structure constant (about 1/137) more than compensates for the increase in the number of diagrams considered. But there is no reason to suppose that the terms will not eventually grow without bound: the number of Feynman diagrams of order 2n grows faster than the nth power of any constant. (In particular, the number of diagrams of order 1002 certainly is a lot more than 137 times the number of diagrams of order 1000; the actual growth function must be at least factorial in n.) Thus we have a power series which probably does not converge. It is reasonable to suppose that at some point, as we calculate to higher and higher orders, the power series terms will start increasing in size and our predictions will begin to diverge from experiment. We will have reached the limits of the theory. The mystery is why it works so well at these low orders!

The mathematical difficulty here should not be confused with the problems of "renormalization". Although renormalization, which Feynman calls a "dippy process", is hard to justify in terms of the mathematics of function spaces, there is no question that it gives a well-defined combinatorial method to obtain numerical answers. Attempts have been made to reformulate the functional analysis to reflect what is going on numerically (of particular interest is R. Fittler's "Some Nonstandard Quantum Electrodynamics" [Helv. Phys. Acta 1985], which uses nonstandard analysis to construct a "toy" (that is, in 2 dimensions instead of 4) model of quantum field theory where Feynman integrals can be consistently defined). However, for the purposes of this paper we are regarding a physical theory as a prescription for computing experimental results, ignoring the mathematics of its "model of reality".

Summarizing the situation with QED, we have a sequence of progressively more complicated algorithms, the first few of which have shown extraordinarily good agreement with experiment. Although each algorithm we have been able to carry out so far has given more accurate predictions than its predecessor, we have reason to believe that after some point the algorithms will start diverging. We certainly have been unable to prove that the sequence of algorithms converges.

B. Other Fundamental Theories

The situation for other "fundamental" physical theories is even worse. The "next best" fundamental theory we have is the electroweak theory, a unification of the electromagnetic and weak interactions developed by Glashow, Salam, and Weinberg in the 1970's. Without getting into too many details, we can say that the theory is of the same general type as QED, although the mathematics is more complicated because it is a "non-Abelian gauge field theory" (in this terminology, QED is an Abelian gauge field theory). There are many more "free parameters" which must be measured experimentally (the only free parameter in QED is the fine-structure constant, although you must also build in the muon/electron mass ratio if you wish to apply the theory to muons and the tau/electron ratio as well to apply the theory to the tau lepton). Because of these free parameters and the mathematical complications, the electroweak theory is not nearly as precise as QED. Its greatest success has been the prediction of the masses of the W and Z intermediate vector bosons, which experiments showed to be correct to about one part in 40 or 50.

The electroweak theory is hampered by its abundance of free parameters, for some of which (for example the mass of the "Higgs boson") there is no theoretical and little experimental information on their size. (In the case of the Higgs boson all we know is that we haven't seen it yet, so if it exists its energy is probably higher than we've been able to reach with our particle accelerators.) On the other hand, predictions and successive refinements of the parameters are possible: the parameters our experiments are sensitive to are the ones we can gradually measure more precisely. The process of bootstrapping greater sensitivity in our measurements and predictions will lead to either continual refinement, falsification of the theory, or an experimental impasse [this trichotomy is explored in more detail in the "bootstrapping" paper mentioned in section I-D]. Even if everything goes well, the same mathematical problems which were seen to threaten QED will ultimately limit the electroweak theory, because the calculations still come down to Feynman diagrams and there is no reason to suppose that the answers will stabilize as we move to more and more detailed calculations.

The theory of the strong nuclear interaction, "Quantum chromodynamics" (QCD), explains nuclear phenomena in terms of particles called quarks and gluons. Here the computational situation is much worse: the theory does not lend itself to computation by perturbative expansion in powers of some coupling constant, because the coupling constants would be greater than 1 so that the terms wouldn't even decrease for a while before starting to blow up. By qualitative and analogical reasoning and ad hoc approximations, some predictions can be made, but there is no systematic procedure. The fact that the ad hoc stuff works fairly well much of the time is encouraging, but until theoretical advances are made QCD must be regarded as a descriptive rather than a predictive theory.

The best numbers have come from "lattice gauge theories": approximations to QCD in which spacetime is regarded as a discrete lattice and all interactions are discretized. Ian Lawrie, in "A Unified Grand Tour of Theoretical Physics" [Adam Hilger, Bristol, England 1990] remarks "Such calculations [i.e. masses of hadrons formed from quarks and gluons] have yielded encouraging results, but they do not, at the present time, appear capable of giving precise, reliable information." This is not only because our computers cannot simulate very fine lattices yet, but because there is no clearly defined theoretical connection between the lattice approximations and the "real" theory. From the computational perspective, it may make more sense to regard a lattice gauge "approximation" as the real theory and the standard continuous version as an ill-defined abstraction, until someone can give it clear numerical content (as we have seen, this need not imply computability, just that the numbers can be unambiguously defined).

Several "Grand Unified Theories" (GUTs) have been proposed which bring together the electroweak and strong interactions in a single non-Abelian gauge theory. This reduces the number of free parameters involved, but the energies at which we would need to conduct experiments in order to measure these parameters and test the theories are a trillion times higher than can be reached in contemporary accelerators. The one prediction GUTs make that the "standard model" (electroweak plus QCD) doesn't is the decay of the proton, but this has not been observed. This experimental impasse could be broken by a mathematically more encompassing theory which explained/predicted numbers like the ratios of the quark masses, which are currently data put in to the theory rather than predictions pulled out of it. Until some such theoretical breakthrough occurs, the GUTs must be regarded as having no computational content.

(Some physicists would argue that the predicted half-life of the proton is a number computed from the theory, but in fact what occurs is that the GUTs themselves change as the proton decays fail to occur. Feynman describes this process: "So they fiddle around with the numbers, putting a higher mass into the new particle, and after much effort they predict that the proton will decay at a rate slightly less than the last measured rate the proton has been shown not to decay at. When a new experiment comes along and measures the proton more carefully, the theories adjust themselves to squeeze out from the pressure." Lawrie puts it more diplomatically: "The value of grand unified theories lies much more in their aesthetic theoretical appeal in providing a completely unified description of the three interactions ... than in their utility for interpreting hard experimental data.")

The most ambitious fundamental theories of all are the ones which try to unify gravity with the other fundamental interactions. These "theories of everything" (TOEs) share some common mathematical features, especially a "supersymmetry" between bosons and fermions, which leads to a cancellation of infinities and (theoretically) finite answers for calculations of physical quantities. None of these theories can compute anything yet, but the mathematical requirements a consistent quantum theory of gravity must satisfy are so restrictive that there is hope the ultimate theory will have very few free parameters (at most one or two) and actually be able to explain numbers like the fine-structure constant and particle mass ratios. As we saw above, some of the numbers a TOE predicts may be noncomputable; before this can be investigated the theorists must find a TOE that is mathematically consistent.

Many physicists are skeptical of the enterprise, which is occupying dozens of physicists and mathematicians. If it succeeds, and a truly all-encompassing, mathematically consistent theory is discovered, we will be in a better position to assess Church's thesis and the role of computation in the Universe.

C. What Can Be Proved?

We have seen that the fundamental theories of physics are not very computationally satisfactory. There are several ways in which the situation could be clarified. We will discuss them in the context of QED, but the same possibilities apply to other theories, including potential Theories of Everything.

(1) Someone might prove that the sequence of algorithms described in section 2-A converges. This would be a great advance in mathematics as well as physics, but it seems unlikely. Such a proof would give us great confidence in QED's "rightness" ("See? Against all odds, the terms cancel. This must be the real thing."). Of course, it can be proved that for each n, the algorithm to compute the first n terms of the perturbation series converges (this is what Feynman, Schwinger, and Tomonaga essentially did) -- see, for example, Bjorken and Drell's text "Relativistic Quantum Fields". The proof depends on some standard results from mathematical analysis (e.g. the Heine-Borel theorem). But which n shall we take for our final algorithm, and why?

(2) Someone might prove that the sequence diverges, identifying the point beyond which the terms get larger. Then we would be forced to cut ourselves off at some finite n and say "this is THE algorithm"; we would have reached the limit of accuracy of QED. Experiments aimed to be more precise than QED will tell us something new about the world, and other experiments will have to take QED's proven margin of error into account.

(3) The theory might provide mathematically definable numbers with no way to compute them. This is the kind of thing we saw in section I-D might lead to a violation of Church's thesis. It seems unlikely for QED but may be the case in a theory of everything.

(4) There is some algorithm which computes definitive answers, but no proof the algorithm converges. This unlikely case might occur in QED if the sequence of perturbation algorithms does converge but this fact can't be proved. Many physicists have been acting as if this is the actual situation, but if it is not they will find out when their brand-new supercomputer with which they are doing the next generation of Feynman-diagram calculations gives worse agreement with experiment than their old one did!

(5) The entire theory might be reformulated in some finitary way (as a lattice gauge theory or a cellular automaton, for example), so that all predictions are straightforwardly defined and computed. This is probably the most philosophically pleasant solution. A major drawback is that it is hard to imagine a finitary reformulation of physics which is not a "local hidden variables theory"; these theories were shown to be untenable by Bell (theoretically) and Aspect (experimentally). See Feynman's paper "Simulating physics with computers" [Int. J. Theor. Phys., June 1982]. The inherent nonlocality of quantum mechanics seems to place limits on the effectiveness of naive local reformulations. A non-local, finitary reformulation may be possible but I've never heard of any good candidates for such a thing.

These possibilities are not exhaustive. The challenge to physicists is to reformulate physical theories in an explicitly algorithmic way, not only to clarify their empirical content but also to shed light on the relationship between physics, mathematics, and computation.

D. How Real Are Real Numbers?

Many of the mathematical problems with physical theories result from the use of real numbers and the concomitant assumption that infinitely many decimal places are meaningful. Feynman, in the paper cited above, says it seems strange that an infinite amount of information should be required to specify what is going on in a finite region of space-time. Given the quantum nature of matter and the uncertainty principle, it seems unlikely that our theories can give any meaning to distances below the Planck length of about 10^(-35) m, and the same is true for other physical quantities. Even "dimensionless" numbers like the fine-structure constant may not "really" (by which I mean Platonically, in the mind of God) exist to infinite precision if we live in a finite universe, unless there is a mathematical definition for them and the ultimate Theory of Everything has no free parameters.

Criticism of the "completed infinite" goes back to Kronecker, and before him to Gauss. The key question, from our point of view, is whether physical theories can be formulated from a computational viewpoint without requiring the logical and set-theoretical assumptions of classical mathematics.

One sense in which this can be done is by denying infinities of any sort, completed or potential, and treating the Universe as if it were a lattice of space-time points or a cellular automaton. As remarked above, this is philsophically pleasant but there are reasons for believing that the Universe really is not like that. Another viewpoint, which we have adopted implicitly so far, is to restrict our focus to algorithms and avoid speculation about what is "really" going on. Unfortunately, we are still in the position of having to prove, with the mathematical tools at hand, that our algorithms converge and give consistent results. Also, the algorithms themselves were formulated from physical theories based on real numbers, and we must either abandon their connection to these theories and all the "physics" which informed their creation, leaving us rootless, or accept the mathematical and philosophical problems that come with the theories.

A more ambitious program is to recast physical theories in the spirit of constructive mathematics [see E. Bishop's great text, "Foundations of Constructive Analysis", McGraw-Hill 1967]; algorithms (or "constructive procedures") will be built in from the start, and every statement asserted will have "constructive content". In analogy with the term "Continuum", which denotes the locus where classical analysis takes place, H. Resnikoff has proposed the term "Arithmeticum" (originally coined by R. O. Wells, Jr.) for the place where we do a kind of analysis in which arithmetic plays a central role. In his paper "Foundations of Arithmeticum Analysis: Compactly Supported Wavelets and the Wavelet Group" [Aware, Inc. 1990], he says "The Arithmeticum is, we think, the natural domain for analysis mediated by digital computation and, as such, it is also the source of natural models for representing physical phenomena which can be directly observed and measured".

Resnikoff has initiated a program to "arithmetize analysis from the perspective of digital computation". A major tool has been the "compactly supported wavelets" -- compactly supported complete orthonormal systems of functions in L^2(R). Some wavelet bases have the interesting property that although the functions are continuous and easily calculable on the dyadic rational numbers (by a recursion which lends itself very naturally to digital computation), they are nowhere differentiable. Continuous, nowhere differentiable functions have been known since Weierstrass but have been considered "pathological"; their appearance in the theory of wavelets suggests that the "pathology" really stems from taking something which is naturally defined on a dyadic grid and extending it to arbitrary real numbers. The computations only pay attention to the dyadic points, so why bring in all the rest?

All this goes to show is that the real number continuum we are familiar with is not necessarily the best place to do mathematical analysis if we want to compute things. The reformulation of physical theories in computational terms "from the ground up" (that is, using constructive mathematics) is hereby proposed; it would be premature to speculate on where this could lead, but as we have seen, the problems with the current formulations are so significant that any attempt to resolve or avoid them is worthwhile.

III. Quantum Computation

In the last section we approached quantum physics from a computational perspective. Now we will look at computation from the viewpoint of quantum mechanics. In particular, what happens if our "computers" are not Turing machines, but quantum mechanical devices?

A "quantum computer" is a device whose physical behavior is modeled by the laws of quantum mechanics, which can be prepared in a discrete set of "input" states (specifiable by a finite set of integers), and which has associated with it a set of discrete observables called "outputs", which may or may not commute with each other. There have been two serious attempts to define and analyze quantum computers, by Deutsch ["Quantum theory, the Church-Turing principle and the universal quantum computer", Proc. Roy. Soc. Lond. A400 (1985), pp. 97-117], and by Feynman and Margolus [see Feynman, "Quantum Mechanical Computers", Found. Phys. 16 (1985) 507-531; Margolus, "Parallel Quantum Computation", MIT Lab. for Comp. Sci. 1990, and other papers].

Both Feynman and Deutsch specify their quantum computers abstractly, and their constructions resemble Turing machines. Their computers are systems with a finite "central processor" and an unspecified number of "memory elements", which are two-state quantum systems (Feynman calls them "atoms" but they could be spin-1/2 particles like electrons as well). The memory elements are arranged in a sequence, akin to a Turing machine's tape. There are internal variables corresponding to the location of the currently scanned tape location (Feynman calls it the "cursor") and a special "halt bit" which is 0 at the beginning of the computation and is set to 1 if and when the computation halts.

The dynamics is generated by a unitary operator, which depends on which Turing machine is being simulated; because "universal Turing machines" which can simulate all other TM's exist, it is really only necessary to specify one such operator. Deutsch gives an existence proof, but Feynman actually constructs the Hamiltonian operator by showing how various "logic gates" can be built up from very simple primitive operators (and quoting results on how to build a universal computer from logic gates). There seems to be no reason, in principle, why a quantum computer along the lines of Feynman's model could not be built.

These quantum computers operate like Turing machines (e.g. the "cursor position" only changes by one unit at a time and the corresponding memory element is the only one which might get flipped), but there is one interesting constraint on them. Classical computers (and Turing machines) are irreversible: information can be erased and the "previous configuration" cannot necessarily be deduced from the current configuration. A unitary evolution is reversible: it must be possible for the computation to go backwards! Fortunately, Fredkin and Toffoli (and before them C. Bennett) have shown that universal computation can be performed in a logically reversible fashion: it is possible to carry along "garbage bits" without erasing them or having them grow too numerous, so that at any point the entire history of the computation can be recovered. Fredkin and Toffoli ["Conservative Logic", Int. J. Theor. Phys 21 (1982), 219-253] have shown how to build a universal computer from reversible logic gates, keeping the garbage bits under control and getting rid of them at the end.

Since the computation can go backwards as well as forwards there must be a way to ensure that the computation goes in the right direction. Feynman has shown how this can be done by applying an external force that pulls the cursor along, or by preparing the system with the cursor having different amplitudes to be on different sites representing a "spin wave packet" with a nonzero momentum. The "halt bit" is measured and if it is 1 then the answer to the computation can be read off from the memory observables. It is necessary to test the halt bit first in order to avoid interfering with an incomplete computation.

Margolus, in the paper cited above, has shown how Feynman's model can be extended so that there is more than one "active site" at a time: he has shown how to construct a quantum computer which simulates a one-dimensional array of Turing machines (which can be thought of as a 2-D grid, each row of which is the tape of a TM, where adjacent TM's can communicate when their tapes are at the same coordinate). This provides a form of parallel quantum computation.

These constructions show that quantum computers can simulate Turing machines, and hence do everything classical computers can. However, what is most interesting about quantum computation are the things that quantum computers can do that classical computers cannot, or cannot do as efficiently. These fall under several headings: miniaturization, parallelism, randomness, nonlocality, speedups and complexity, fault-tolerance, and doing physics.

Miniaturization: The most obvious advantage of quantum computers is that if the components are the size of atoms, computers can be much faster and more powerful. Even in the models above, which had only one-dimensional memories, it is clear that a megabyte doesn't take up very much space! With atomic-scale separations between the components, clock cycles on the order of femtoseconds (the time it takes light to go 3000 angstroms) are conceivable. As classical computers are further and further miniaturized, the behavior of their components will have to be analyzed in quantum mechanical terms, so that in this sense quantum computers are already approaching.

Parallelism: Quantum computers will be so small that very massive "classical" parallelism (duplication of hardware) will be possible. However, as Deutsch has pointed out, there is a special kind of "quantum parallelism" in which a SINGLE processor can actually carry out parallel computations. Deutsch starts his quantum computer in a superposition of n states corresponding to the computation of f(i) for i = 1 to n. After executing this "program", the computer is in a superposition of "answer states". At most one of these is accessible directly, because the measurement of the answer will "collapse the wave-function"; Deutsch, using the "many-universes" interpretation of quantum mechanics, regards the computations as taking place in different universes.

However, although we can't directly read all n of the answers, we may be able to compute some function of them by letting them interfere with each other in a further computation! For example, suppose n=2 and we set our quantum computer to simultaneously compute f(1) and f(2) (we shall suppose f is a Boolean function so that f(i) = 0 or 1), ending up in a superposition of two states corresponding to the two "basic" computations. Suppose further that we are not interested in f(1) and f(2) individually, but in the combination f(1) xor f(2). Deutsch shows how to define an observable which, when measured at the end of the computation, will give the outcome "fail" with probability 1/2 and the outcome f(1) xor f(2) with probability arbitrarily close to 1/2. Thus, if we only have one quantum processor we don't have to compute f(1) and f(2) serially to get the combination f(1) xor f(2): with probability 1/2 we can get that answer in half the time by doing both computations in parallel.

Deutsch has shown that any N-fold parallelizable quantum computation (a computation that can be done N times as fast by using N processors instead of 1) can be done N times as fast by a single processor which can obtain the correct answer with probability 1/(N^2 - 2N + 2). For example, a computation that can be done in time x using 3 processors can be done in time 3x by a single processor working serially, while a single processor doing all 3 computations at once will take time x but only return the correct answer 1/5 of the time. (If this parallel computation is redone until it returns an answer, the expected value of the running time is 5x). He conjectures that it is impossible to get the right answer more than 1/(N^2 - 2N +2) of the time when running an N-fold parallel computation. This may be true for arbitrary parallelizable computations, but for certain ones a greater speedup is possible (e.g. a quantum computer can compute f(1) xor f(2) xor f(3) xor f(4) serially in running time 4x and in parallel in time x, obtaining the correct answer 1/8 of the time; this improves on Deutsch's bound of 1/10 for arbitrary 4-fold computations [J. Shipman, "Quantum Speedups", in preparation]).

Deutsch proves that this sort of parallelism can never decrease the EXPECTED running time of a computation, but the decrease in "best-case" running time can be useful. He gives the following fanciful example: if your computer can predict tomorrow's stock market prices (by a 2-fold parallelizable algorithm) but the computation takes a day and a half, it is better to run the algorithm in parallel--half the time you will get your answer and can make money, and the other half the time you don't get an answer and sit on the sidelines for a day. (This is a silly example because it would be simpler to borrow the money to buy a second computer, so as to have the right answer every day and make money twice as fast!)

Randomness: There are at least two ways in which the randomness inherent in quantum mechanics makes quantum computers useful. First of all, quantum computers can have "built-in" random number generators. Classical computers must rely on deterministic "pseudo-random number generators", which are necessarily slower (in a quantum computer, generating a random bit is a primitive operation taking one time step, whereas a pseudo-random bit generator must do some computation) and less random. In fact, any given algorithm to generate bits looks non-random for certain applications--the output of such algorithms is not "algorithmically random" (also known as "Chaitin random"), whereas the output of a quantum random number generator almost certainly will be. It is possible to devise problems for which a true "randomized algorithm" will work better than any deterministic algorithm (though these problems are infeasible: not solvable by polynomial-time computations).

The second advantage of quantum randomness is that a quantum computer may generate bits with certain probabilities directly, without any coding. If it is necessary to have a bit which is 1 with probability cos^2 (pi/4) = sqrt (1/2), a classical computer could only do this by a complex process, probably involving the computation of sqrt(2) and thus gradually requiring more work per bit generated, while a suitable quantum computer might have this as a primitive function.

Nonlocality: One big difference between quantum physics and classical physics is that the former is "nonlocal". Two particles which are separated in space can be correlated in a way which cannot be modeled classically. Thus, two widely separated parts of a quantum computer can be "connected" in a way not possible for classical computers. It is not clear whether this allows "normal" computations to be done any more efficiently, but it certainly means that physical experiments exhibiting nonlocal phenomena (the EPR/Bell experiment, for example) can be simulated by quantum computers but not by classical computers. The term "simulation" is somewhat vague: a classical computer can certainly reproduce the results of the EPR/Bell experiment, but only by sending some sort of signals back and forth between separate locations, which the quantum computer need not do; the quantum computer can get the right results in "real time" which is what we mean by "simulation". See Deutsch's paper; see also Feynman's "Simulating physics with computers" [Int. J. Theor. Phys. 21 (1982), 467].

Another use for the nonlocal correlations in a quantum computer may be in "quantum cryptography" [see C. Bennett et al, "Quantum Cryptography", in "Advances in Cryptology", Plenum Press, New York 1983].

Speedups and complexity: We have seen how quantum computation allows for certain parallelizable computations to be speeded up in the "best case". It is also true that serial computations on a quantum computer may occasionally go faster than in a classical computer: the computation can "tunnel through". However, this occurs very seldom, and the expected running time remains proportional to that of a classical computer.

It is interesting to ask how the framework of complexity theory can be applied to quantum computers. In particular, what is the right definition of the complexity classes P and NP?

We can define "quantum polynomial time" (QP) as those algorithmic problems which can be solved by a quantum computer with expected running time bounded by a polynomial in the input size. "Quantum nondeterministic polynomial time" (QNP) consists of those algorithmic problems for which there is a quantum computer program which, given an instance of the problem as input, halts in polynomial time, and returns a solution with nonzero probability whenever a solution exists. (This definition is not quite right; we should add the condition that the log of the probability be bounded by a polynomial in the input size; otherwise the possibility of "tunneling" would put all quantum computations in the class QNP).

Clearly QP contains P. It is not hard to see that QNP contains NP: given a problem in NP, we can program our quantum computer to guess a solution at random and test it; if there is a solution it will be selected a sufficient fraction of the time, verified, and returned as output.

Wolfram, Pitowsky, and others have speculated that QP might contain NP: that is, there is some quantum computer which solves the satisfiability problem (or some other NP-complete problem) in polynomial expected running time. If this is proven, it would be a tremendous breakthrough which would completely redefine our notions of feasible computational problems (and, incidentally, falsify the "complexity version of Church's thesis" defined in section I). The hope is that some way will be found to take advantage of quantum parallelism to "try out all solutions at once". However, Deutsch's results imply that the obvious way to do this cannot improve the expected running time.

Although the P-NP question seems hard for quantum computers too, the NP-coNP question has a surprising answer. For classical computers, the conjecture "NP=coNP" says that all tautologies have short proofs: there is some nondeterministic computation which, given a tautology (or a member of some other coNP-complete set), will verify that it is a tautology (if it makes the right guesses). Whether a classical computer can do this is questionable, but a quantum computer does it easily. Using Deutsch's method, compute in parallel the truth value of the expression in question for all 2^n sets of values of the variables and take the logical "and" of these 2^n bits. The answer will show up at the end of the computation about one time in (2^n)^2. The expected running time is proportional to 2^2n, which compares unfavorably to the serial running time of 2^n, but we have still shown that coNP is contained in QNP. It is not much harder to show that in fact QcoNP=QNP (with QcoNP defined as sets whose complements are in QNP), because a quantum computer can run quantum computations in parallel too.

There are other ways to define quantum complexity classes. For example, we could define QNP differently by saying a set is in QNP if each instance has a short "certificate", which can be verified by a polynomial-time quantum computation that never verifies a certificate for a non-instance. This is more restrictive than the previous definition because the certificate is some finite piece of information which can be transferred, rather than a lucky sequence of guesses inside the quantum computer.

Quantum complexity theory is still in its infancy, but already shows some important differences from the classical theory.

Fault-tolerance: Deutsch has addressed the problem of computing in the presence of "noise". Suppose that an N-fold parallelizable function is to be computed, with NR processors each of which fails with probability p. Classically, the best solution is to set R processors working on each of the N tasks; the computation fails if and only if all R processors assigned to any one subtask fail. This occurs with probability 1-(1-p^R)^N. For suitable values of N, P, and R it is better to give each of the NR processors all N tasks in parallel.

Then each processor can fail in two ways (a hardware failure with probability p, or not being lucky enough to have the answer show up), but only one of the NR processors must succeed. For example, take N=R=2, p=1/2. The classical computation succeeds with probability 9/16 (each subtask succeeds with probability 3/4 because only one of its two processors need work), while the quantum computation succeeds with probability 175/256 (each processor fails with probability 3/4 but for all four to fail only happens 81 times in 256), an improvement of 31/256.

Doing physics: We have seen that quantum computers are better than classical computers at simulating quantum systems, and may provide a more natural way to solve certain physical problems than classical computers do. Deutsch's universal quantum computer can not only simulate classical computers, but also any quantum system with a finite-dimensional state space. Deutsch states a generalization of Church's thesis which he calls the Church-Turing 'principle': "Every finitely realizable physical system can be perfectly simulated by a universal model computing machine operating by finite means". He claims this is a physical statement with empirical content, but for it to be so some universal model computing machine needs to be specified. Deutsch's own model cannot compute any (classically) noncomputable functions, so if his Church-Turing 'principle' refers to his own model then it is false if the classical Church-Turing thesis is. It is conceivable that some "universal quantum computer" can be defined which is also capable of generating the "measurable" but not necessarily computable numbers discussed in sections I and II.

The existence of a quantum computer with the capabilities of Deutsch's model has implications for the interpretation of quantum mechanics. It is natural to regard a quantum parallel computation as being split among several "universes"; on those occasions when the computation is successful, the results of the several subcomputations have been correctly combined, so in some sense all of those universes have contributed to the answer. Deutsch asks "WHERE was all that computation done?" and argues that only the many-universes interpretation provides a satisfactory answer to the question. In another paper [Int. J. Theor. Phys. 24 (1985), 1], Deutsch points out that if true "artificial intelligence" programs can be implemented on quantum computers, an experimental test of the "many-universes" interpretation becomes possible.

IV. Is the Universe a Computer?

Why is computation, as we know it, possible? That is, why can we compute certain functions but not others? One can imagine a universe in which far fewer functions are "computable": for example, a cellular automaton with a simple rule like Fredkin's "mod 2" rule. Such a system is not "computation universal" and can be completely understood by us. We can define model universes in which addition is computable but not multiplication, or in which "primitive recursion" is allowed but not "general recursion" (primitive recursion requires that all loops be bounded in advance; addition, multiplication, exponentiation, etc. are primitive recursive functions but if the sequence f1(x) = x+x, f2(x) = x*x, f3(x) = x^x, f4(x) = x^x^...^x [x times], etc., is defined, the "diagonal" function g(i) = f_i(i) is not primitive recursive).

However, it is hard to imagine intelligent beings in such a universe, because it seems that an "intelligent being" ought to be able to simulate a Turing machine (we certainly can). (Actually, the inhabitants of a universe in which primitive recursion but not general recursion was possible might be interesting, though limited in their capacity for reflection and self-awareness).

A universe in which universal computation is possible can allow the construction of both universal computers and "self-reproducing automata" (also known as Von Neumann machines). Von Neumann's original construction of these automata bears a striking formal resemblance to the processes of DNA replication and protein synthesis that were discovered a few years later. It seems clear that such universes can have "life" in them, although whether they suffice to support "intelligent" life is a hotly disputed question [see R. Penrose, "The Emperor's New Mind", Oxford University Press 1989].

We can also imagine a universe in which the halting problem was solvable (earlier in this paper we have speculated that our own universe might be like that). Beings in such a universe could, from our point of view, do infinite computations and know things beyond the reach of our investigations. They might still have their own "unsolvable problems", and might even speculate that a universe like ours in which Diophantine equations can't be solved is too simple to contain intelligent life as they know it!

Fredkin and Toffoli have argued that our Universe may really be a giant cellular automaton, with some fixed connective topology and transition rule. They have shown [see, for example, T. Toffoli, "How cheap can mechanics' first principles be?", MIT Lab. for Comp. Sci. 1990] that continuity, entropy, relativistic covariance, and variational principles can emerge naturally as epiphenomena of an underlying discrete structure such as a cellular automaton. The major defect of this thesis is that any "local" model cannot reproduce the long-range correlations that Bell showed quantum mechanics requires and Aspect experimentally demonstrated. [See J. Bell, "Speakable and Unspeakable in Quantum Mechanics", Cambridge University Press 1987]. Until someone shows how the EPR/Bell correlations can emerge from a local model, these models must be regarded as viable universes which, unfortunately, are not our "the" universe.

This doesn't mean, though, that the universe is not a computer, just that it's not a classical computer. Feynman's dictum that it shouldn't take an infinite amount of information to describe what is going on in a finite region of spacetime still seems reasonable, and the task before us is to come up with better theories and models, more constructive and finitary in nature but compatible with the quantum world we live in.