# Quantum-Mechanical Computers and Uncomputability

Seth Lloyd
*Complex Systems Group (T-13) and Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, New Mexico 87545*
(Received 3 December 1992)

The time evolution operator for any quantum-mechanical computer is diagonalizable, but to obtain the diagonal decomposition of a program state of the computer is as hard as actually performing the computation corresponding to the program. In particular, if a quantum-mechanical system is capable of universal computation, then the diagonal decomposition of program states is uncomputable. As a result, in a universe in which local variables support universal computation, a quantum-mechanical theory for that universe that supplies its spectrum cannot supply the spectral decomposition of the computational variables. A "theory of everything" can be simultaneously correct and fundamentally incomplete.

PACS numbers: 89.70.+c, 03.65.−w

When can a system compute? Three conditions must be met: First, one must be able to prepare the system in a state that corresponds to a program state for a computer, without necessarily knowing beforehand the result of the computation. Second, the system's dynamics on that state must realize the dynamics of the computer. Third, one must be able to make measurements on the system that allow one to determine if the computation has been completed, and if so, extract its result. These would seem to be not only sufficient but necessary conditions for computation.

Some quantum-mechanical models of computation that satisfy these conditions are the ballistic quantum computers of Benioff [1], Deutsch [2], and Feynman [3], Bennett's dissipative enzymatic and Brownian motion computers [4], and the quantum cellular automaton computers of Margolus [5], as well as the error-correcting quantum computers of Peres [6] and Zurek [7]. Other fundamentally quantum-mechanical systems to which these conditions apply are semiconductor-based digital computers, analog computers, and human beings calculating on their own, or with a slide rule or abacus.

A wide variety of quantum systems obey the requirements for computation outlined above. In this paper, it is shown that any quantum system that meets these three conditions possesses a block diagonal form for its time evolution operator, but that the ability to decompose the input, "program" states of the quantum computer in terms of this diagonal form allows one to determine immediately which programs result in complete computations, and the results of those computations. Accordingly, the problem of determining the diagonal or spectral decomposition of the program states of a quantum computer is at least as hard as any problem that the computer can solve. In particular, if the quantum computer is a universal, digital computer, the spectral decomposition of its program states is uncomputable.

The basic idea behind this result is simple: Knowledge of a quantum system's spectrum, together with the ability to decompose a given state in terms of eigenstates, allows the straightforward evaluation of the time evolution of the state. Spectral decomposition is a valuable tool because it makes time evolution completely transparent. Now suppose that a system possesses observables that support computation, where computation may be interpreted either narrowly to mean digital computation alone or broadly to mean a process in which those observables pass through a sequence of configurations, in the course of which information is transferred and transformed. In either case, the spectral decompositions of two configurations of the observables allow one to determine whether or not there is a nonzero amplitude for one configuration to evolve into the other. Finding the spectral decomposition of two such configurations must then be at least as hard as determining whether one has a chance of evolving into the other.

In particular, spectral decomposition makes it possible to determine immediately whether or not a given program state for a computer evolves into a given output state. But as noted above, for a universal digital computer there is no algorithm that can determine in a finite time whether or not an arbitrary program state evolves into a given output state. Essentially, spectral decomposition makes all future time evolution transparent, while the future time evolution of a universal computer is by necessity to some degree opaque. Complete spectral decomposition for observables that support universal computation is therefore impossible.

This result has consequences for quantum-mechanical theories that purport to describe the universe as a whole ("theories of everything"). Suppose that it is possible, starting from some statement of such a theory, to derive the theory's eigenstates and spectrum. In a universe that supports complicated information processing in terms of some observables, obtaining the spectral decomposition of those observables is as hard as evaluating the results of the process. If the universe supports universal computation, then spectral decomposition of even the simplest state may prove impossible, if that state results in an arbitrarily long computation. [Two primary physical requirements for the universe to support universal computation are (a) that it expand forever and (b) that excess

0031-9007/93/71(6)/943(5)$06.00

free energy always be available for information processing. Neither requirement is implausible.] As a result, a theory of everything can be correct, can supply spectrum and eigenstates, and yet fail to give a full account of what goes on in the universe.

To prove the results presented above, a quantum-mechanical account of computation is required. Suppose that a quantum system $Q$ is to compute the values of a function $f$. The system must be capable of being prepared in states corresponding to the arguments of the function, its dynamics must perform the computation, and measurements on the system must reveal the values of the function. The conventional quantum-mechanical description of this process (in the Schrödinger picture) is the following:

(0) The system and environment are in some initial state, described by a density matrix, $\rho_0 \in \mathcal{H} \otimes \mathcal{H}^*$, where $\mathcal{H}$ is the Hilbert space for system and environment.

(1) The system is prepared in a program state corresponding to the input $b$. If the quantum system is to be a useful computer, the process of preparing the system in an input or program state should not require knowledge of the result of the computation. The system and environment are now in the state

$$\rho_0(b) = P_{\text{in}}(b)\rho_0 P_{\text{in}}(b)/\text{tr}[P_{\text{in}}(b)\rho_0 P_{\text{in}}(b)] , \quad (1)$$

where $\{P_{\text{in}}(b)\}$ are a set of projection operators corresponding to the different inputs $b$: $P_{\text{in}}(b) = P_{\text{in}}^\dagger(b)$, $P_{\text{in}}(b)P_{\text{in}}(b') = \delta_{bb'}P_{\text{in}}(b)$.

(2) The system and environment evolve over time,

$$\rho_0(b) \rightarrow \rho_t(b) = U(t)\rho_0(b)U^\dagger(t) , \quad (2)$$

where $U(t)$ is a unitary operator.

(3) Measurements are made to determine whether the computation is completed, and if it is, to extract the result: Once again, the process of making these measurements should not require *a priori* knowledge of the result of the computation. Let $P_h$ be a projection operator corresponding to the computation being complete (or "halting"), and $P_{\bar{h}} = 1 - P_h$ be the projection operator corresponding to the computation being incomplete ("not halting"). $p_t(h|b) = \text{tr}P_h\rho_t(b)$ is the probability at time $t$ that a measurement will reveal the computation to be complete. The different results, $r$, of the computation correspond to projection operators, $P_{\text{out}}(r)$, $P_{\text{out}}(r) = P_{\text{out}}^\dagger(r)$, $P_{\text{out}}(r)P_{\text{out}}(r') = \delta_{rr'}P_{\text{out}}(r)$. The probability that the computation is complete at time $t$ and gives the result $r$ is

$$p_t(r,h|b) = \text{tr}P_{\text{out}}(r)P_h\rho_t(b)P_h . \quad (3)$$

For a conventional computer, for example, (0) corresponds to plugging the computer in, (1) to inputting a program, (2) to letting the machine run, and (3) to checking if the computation is complete, and if it is, to getting the answer. $Q$ can compute the values of a function $f$ if, when prepared in the initial state $\rho_0(b)$ corresponding to an input $b$ for which $f$ is defined, measurement at some later time has a nonzero chance of revealing the computation to be complete, and to give the output $f(b)$. That is, $p_t(h|b) > 0$ for some $t$, and for all such $t$,

$$p_t(f(b)|h,b) \equiv p_t(f(b),h|b)/p_t(h|b) = 1 . \quad (4)$$

The conditions of Eq. (4) describe a computer that, if it gives a result, always gives the right one. Less stringently, and more realistically, one can ask that the computer give the right result with probability $1-\epsilon$, where $\epsilon < \frac{1}{2}$. Sufficient repetition of a computation then allows the identification of the correct result to any desired degree of confidence. In the results that follow, the appropriate extension to such probabilistic computers will be given parenthetically. That is, the quantum system can compute $f(b)$ if, when programmed with the input $b$, at some later time it reliably gives the result $f(b)$ (where reliability may involve repeating the computation several times).

The account of quantum-mechanical computation given above, though very general, has nontrivial consequences. In particular, although $U(t)$ is in general an operator on an infinite dimensional Hilbert space, and need not be diagonalizable, the restrictions that $Q$ have a nonzero chance of completing the computation, and that a computation that halts gives the proper output, imply that the operator $U(t)$ possesses a block diagonal form. First, look at the set of all states $|\phi\rangle \in \mathcal{H}$ of $Q$ that never result in a complete computation:

$$\mathcal{H}_{\bar{h}} \equiv \{|\phi\rangle \in \mathcal{H} : P_{\bar{h}}U(t)|\phi\rangle = U(t)|\phi\rangle, \ \forall t \geq 0\} . \quad (5)$$

$\mathcal{H}_{\bar{h}}$ is a subspace of $\mathcal{H}$, and is invariant under the action of $U(t)$. The orthogonal subspace $\mathcal{H}_h = \mathcal{H}_{\bar{h}}^\perp$ is composed of states that can give a completed computation for some $t$.

Each possible result for a completed computation also corresponds to an invariant subspace:

$$\mathcal{H}_h(r) \equiv \{|\psi\rangle \in \mathcal{H}_h : P_{\text{out}}(r)P_hU(t)|\psi\rangle = P_hU(t)|\psi\rangle, \ \forall t \geq 0\} . \quad (6)$$

$\mathcal{H}_h(r)$ is the invariant subspace composed of states that eventually give a complete computation with the result $r$. The corresponding invariant subspace for computers that give the correct result with probability $1-\epsilon$ is spanned by states $|\chi\rangle \in \mathcal{H}_h$ such that

$$||P_{\text{out}}(r)P_hU(t)|\chi\rangle||^2/||P_hU(t)|\chi\rangle||^2 \geq 1-\epsilon, \ \forall t \geq 0 . \quad (7)$$

The requirements that the quantum computer, when programmed with the input $b$, have a nonzero chance of halting

at some time, and halt only on (usually on) the output $f(b)$, imply that the density matrix $\rho_0(b)$ corresponding to the program state $b$ must be a mixture of states that lie at least partly in the space $\mathcal{H}_h(f(b))$, with the remainder lying in the space $\mathcal{H}_{\bar{h}}$. The subspace $\mathcal{H}_{\text{comp}} \subseteq \mathcal{H}$ explored by $Q$ in the course of all possible computations is therefore equal to $\mathcal{H}_{\bar{h}} \oplus \sum_{r \in \text{range}(f)} \times \mathcal{H}_h(r)$. The unitary time evolution $U(t)$ possesses a block diagonal form $U_{\text{diag}}(t)$ on $\mathcal{H}_{\text{comp}}$, where each block gives the action of $U(t)$ on an invariant subspace in the direct sum decomposition of $\mathcal{H}_{\text{comp}}$.

In short, if a quantum system is to compute, each computation must have some chance of reaching completion, and when complete, must reliably give the correct result. Consequently, the Hilbert space of the system possesses invariant subspaces that correspond to each complete computation, and to computations that never halt. The system's unitary time evolution operator them possesses a block diagonal form in which each block corresponds to the time evolution within an invariant subspace.

The unitary time evolution for a quantum computer possesses a simple diagonal form. Can one decompose the program states of the computer in terms of this diagonal form? From the results presented above, the diagonal decomposition of an input state $\rho_0(b)$ determines immediately whether the computer halts on the input $b$, and if it halts, what its output is. Determining the diagonal decomposition of an input state is therefore at least as hard as performing the computation for which the input state is the program.

If the system in question is capable of universal digital computation, the situation is worse. It is well known from the theory of computation that there is no method that in a finite amount of time tells one whether a universal digital computer halts on a given input [8]. For such a system, the diagonal decomposition of program states is uncomputable.

More formally, let $Q$ be able to compute the function $T(b)$, where $T(b)$ is the output of a universal computer or Turing machine given the input $b$. $T(b)$ is undefined if the computer fails to halt on input $b$. The unitary evolution of $Q$, $U(t)$, has a block diagonal form $U_{\text{diag}}(t)$, in which each block corresponds to the action of $U(t)$ on the invariant subspaces, $\mathcal{H}_{\bar{h}}$, the set of states that never halts, and $\mathcal{H}_h(r)$, the set of states that halts only on (reliably on) the output $r$, for all $r$ such that $r = T(b)$ for some input $b$. Possession of the diagonal decomposition of $\rho_0(b)$ tells one whether $p_t(h|b) > 0$ for some $t$, and the result $r$ for which $p_t(r|h,b) = 1\,[p_t(r|h,b) \geq 1 - \epsilon]$. So possession of the diagonal decomposition of $\rho_0(b)$ for all $b$ gives the solution of the halting problem, and allows one immediately to evaluate $T(b)$, for all $b$. This is impossible [10].

A straightforward example of the uncomputability of the diagonal decomposition of program states for a quantum computer can be seen in Benioff's quantum computer [1]. The program states of the computer correspond to pure states $|b\rangle$, which over a period of time $\Delta t$ evolve to the state $U(\Delta t)|b\rangle = |\Upsilon(b)\rangle$, where $\Upsilon(b)$ is the logical state into which $b$ evolves over a single machine cycle of a digital computer. $\Upsilon$ must be a one-to-one function, i.e., such a computer must be logically reversible [4]. Logically reversible computers have a special "halt" flag, which when raised indicates the completion of the computation. $P_h$ ($P_{\bar{h}}$) projects out the space of states in which this flag is (is not) raised. $\mathcal{H}_{\bar{h}}$ can be defined as above, and is spanned by states in the course of whose evolution the halt flag is never raised. The question of whether a program state $|b\rangle$ lies in $\mathcal{H}_{\bar{h}}$ or not is unsolvable in general.

One can go further. It is not difficult to show that $U(\Delta t)$ is wholly diagonalizable, with both a discrete and a continuous spectrum, and that the state $|b\rangle$ can be decomposed in terms of eigenvectors whose eigenvalues fall in the discrete part of the spectrum if under repeated application of $\Upsilon$, $b$ returns to itself: $\Upsilon^m(b) = b$ for some $m$. If $b$ never returns to itself, then $|b\rangle$ can be decomposed in terms of eigenvectors whose eigenvalues fall in the continuous part of the spectrum.

But the answer to the question of whether a computational state of a universal digital computer ever returns to itself or not under the computational dynamics is uncomputable; the ability to answer this question translates into the ability to solve the halting problem. In fact, a prefix can be adjoined to every program for a logically reversible computer that causes the resulting program to return to itself if and only if the original program halts. As a result, in Benioff's computer, there is no effective procedure that allows one to discover whether an arbitrary program state $|b\rangle$ is to be decomposed in terms of the discrete or the continuous part of the spectrum. (Of course, one can usually discover for some program states whether they repeat or not. It is simply that there is no effective procedure for discovering for *all* program states whether they repeat.) The diagonal decomposition of the program state is uncomputable.

In conclusion, if a quantum system is to compute, one must be able to program it, verify that the computation has been completed, and extract the results of the computation. These requirements imply that the time evolution operator for the system possesses a block diagonal form, where each block corresponds to an invariant subspace of Hilbert space made up of states either that never halt or that halt on a particular output. The knowledge of how a given input state overlaps with these invariant subspaces translates directly into the ability to tell whether or not the quantum computer halts on that input, and if it halts, what its output is. Obtaining this knowledge is therefore at least as difficult as evaluating the computation for which the input is the program. Since there is no algorithm that tells whether or not a universal computer halts on an arbitrary input, one cannot in general obtain the knowledge of the decomposition of input states of such a computer in terms of invariant subspaces. (The decom-

position of some inputs can always be obtained, of course, just as the outputs of some programs can always be evaluated. But there is no algorithm that tells which inputs these are.)

These results have the following consequences for the attempt to reduce knowledge of the world around us to a knowledge of fundamental physics and its consequences. Suppose that one has a "theory of everything," such as superstring theory, that purports to explain the underlying quantum-mechanical dynamics of our Universe. Suppose that one has managed to obtain the theory's spectrum: That is, one has obtained the eigenvalues and eigenvectors of the time evolution operator—its diagonal form. Suppose further that the Universe, or some part of it, is capable of computation, defined either narrowly as digital computation or broadly as information processing in general. Then to decompose computational states of the Universe in terms of the eigenstates of the time evolution operator is as difficult as determining the end result of such computation. If the Universe is capable of universal digital computation, then it may be impossible to give the spectral decomposition of even the simplest states. A theory of everything, even if correct, is not necessarily an effective theory of the part of the Universe that can compute.

[1] P. Benioff, J. Stat. Phys. **22**, 563–591 (1980); Phys. Rev. Lett. **48**, 1581–1585 (1982); J. Stat. Phys. **29**, 515–546 (1982); Ann. N.Y. Acad. Sci. **480**, 475–486 (1986).

[2] D. Deutsch, Proc. R. Soc. London A **400**, 97–117 (1985); **425**, 73–90 (1989).

[3] R. P. Feynman, Opt. News **11**, 11–20 (1985); Found. Phys. **16**, 507–531 (1986); Int. J. Theor. Phys. **21**, 467–488 (1982).

[4] C. H. Bennett, IBM J. Res. Dev. **17**, 525–532 (1973); Int. J. Theor. Phys. **21**, 905–940 (1982).

[5] N. Margolus, Ann. N.Y. Acad. Sci. **480**, 487–497 (1986).

[6] A. Peres, Phys. Rev. A **32**, 3266–3276 (1985).

[7] W. H. Urek, Phys. Rev. Lett. **53**, 391–394 (1984).

[8] A. M. Turing, Proc. London Math. Soc., series 2, **42**, 230–265 (1936,1937).

[9] M. B. Pour-El and J. I. Richards, *Computability in Analysis and Physics* (Springer-Verlag, Berlin, 1989).

[10] In fact, $\mathrm{tr}P_h(r)\rho_0(b)$ is an uncomputable function of $b$, $r$, in the sense that any procedure that gives successively better approximations of this number converges more slowly than any computable function [9]. That is, even though one can approximate the spectral decomposition of an input state, there are always states for which the approximation converges to the desired accuracy arbitrarily slowly.