International Journal of Theoretical Physics, Vol. 42, No. 2, February 2003 (© 2003)

Looking at Nature as a Computer

Norman Margolus[1,2]

Received May 13, 2002

Although not always identified as such, information has been a fundamental quantity in Physics since the advent of Statistical Mechanics, which recognized "counting states" as the fundamental operation needed to analyze thermodynamic systems. Quantum Mechanics (QM) was invented to fix the infinities that arose classically in trying to count the states of Black Body radiation. In QM, both amount and rate of change of information in a finite physical system are finite. As Quantum Statistical Mechanics developed, classical finite-state models naturally played a fundamental role, since only the finite-state character of the microscopic substratum normally enters into the macroscopic counting. Given more than a century of finite-state underpinnings, one might have expected that by now *all* of physics would be based on informational and computational concepts. That this isn't so may simply reflect the stubborn legacy of the continuum, and the recency and macroscopic character of computer science. In this paper, I discuss the origins of informational concepts in physics, and reexamine computationally some fundamental dynamical quantities.

KEY WORDS: information; entropy; energy; action; cellular automaton; quantum mechanics.

1. INTRODUCTION

Viewed from a distance a digital image looks continuous, but if you look closely enough you start to see the pixels—it becomes apparent that there is only a finite amount of resolution. Similarly, it became apparent about a century ago that finite physical systems also have only a finite amount of resolution. For example, a gas of particles in a box has only a *finite* number of possible distinct states. This is described precisely by Quantum Mechanics (QM), which also says that the rate at which a finite physical system can move from one distinct state to another distinct state is finite. Thus Nature is a lot like a computer: every finite system has finite state and a finite rate of change of this finite state. Of course, Nature is more like a

[1] MIT Artificial Intelligence Laboratory.
[2] To whom correspondence should be addressed; e-mail: nhm@ai.mit.edu.

0020-7748/03/0200-0309/0

spatially distributed Cellular Automaton (Margolus, 1998; Toffoli and Margolus, 1987) than like a conventional von Neumann machine.[3]

Since the nonclassical properties of quantum physics (other than finite state) are rarely evident in the macroscopic world, for many purposes it will be indistinguishable whether the microscopic finite-state dynamics is classical or quantum. We might therefore expect basic quantities of macroscopic physics to have a classical informational/computational interpretation. That this is true has been understood about entropy for a long time. More recently, energy has been understood to be a computational quantity as well (Margolus and Levitin, 1998). For any isolated physical system, the energy determines the maximum rate at which it can go through a sequence of distinct states—its maximum number of operations per second.

In this paper, we begin with a review of how information first became an essential part of physics. We then develop a statistical description of ordinary computation using the language of QM. This provides a simple introduction to QM from a computer science point of view, clearly separating the formalism from its physical content. We use this approach to define energy and action in ordinary computation. We then discuss the difference between a finite-state classical world and a quantum world. Finally, we argue that since most of macroscopic physics is indifferent to this difference, classical finite-state models can play a distinguished role in understanding physical dynamics.

2. INFORMATION

Since this is all going to be about information, we should start off by saying what we mean here by *information*. We will be concerned here only with the information-carrying capacity of physical systems: How many bits can a given physical system hold, used as a computer memory? An n-bit computer memory has 2^n possible distinct physical states and, conversely, any physical system that has 2^n possible distinct states can, in principal, be used as an n-bit computer memory.

This is a classical notion of information. As Bennett (this issue) discusses, there is an expanded notion of information that deals with qubits and quantum entanglement and such, but here we will be concerned only with the classical information content of physical systems. One reason that this is an interesting focus is that classical information survives in the macroscopic limit. The whole basis of Quantum Statistical Mechanics is that QM is needed primarily to determine a set of distinct states—to give us something to count. The macroscopic combinatorics that give relative probabilities are purely classical. From this, we expect that all of the ordinary quantities in macroscopic physics should have a classical-information

[3] Interestingly enough, Cellular Automata were also invented by von Neumann (Burks, 1970): they are *unconventional* von Neumann machines.

interpretation. Also, classical information is easier to understand, so it's a good place to start.

3. ENTROPY

In discussing information in physics, the natural starting point is Entropy. Entropy was, for a long time, one of the most mysterious quantities in physics. You put in an integrating factor in some differential equations of Thermodynamics and you got a perfect differential. It was a very formal thing. Boltzmann and Gibbs proposed that the entropy S equals $\log(\omega)$, where ω is the number of possible microscopic physical states consistent with the macroscopic constraints on the system. Since computers and information hadn't been invented yet, no one thought of S as being like a number of bits in a computer memory. They just reasoned that to do statistics you need to count something, and this definition of S grows proportionately with the size of the system, as thermodynamic entropy does, and gets larger for less constrained systems. They guessed correctly that this S corresponds to the more abstract concept of entropy from thermodynamics.

Now the connection between entropy and the direction of macroscopic thermodynamic evolution was explained. When macroscopic constraints are relaxed—say, a wall is removed—then a physical system expands to make use of a larger set of possible states. Even though the microscopic evolution is reversible, it is very unlikely that the system will just happen to land on one of the relatively few states consistent with the original constraint. This relative likelyhood is reflected in the fact that a particle in a small box can be used to represent fewer bits of information than the same particle in a large box (Bennett, 1982; Bennett and Landauer, 1985).

Classically, in order to calculate the number of possible states, what they did was introduce some sort of coarseness both in positions and momenta.[4] Then you could just do a counting of the relative number of ways of getting different kinds of states, with and without a constraint. It didn't much matter what the coarseness was because when you compared two things, the size of the grain canceled out in the ratio of relative probabilities. It's like the relative probability of hitting a bullseye on a dart board, compared to hitting the next ring—it doesn't matter what units you use to measure the two areas.

This kind of coarse-graining worked well until they came to the problem of black body radiation, which was the first place where they really had to get the counting of states right. This necessitated the invention of QM. The situation considered is illustrated in Fig. 1. We have a cavity, and we have the radiation

[4] They also had to treat particles of the same kind as classically indistinguishable, like the two ones in the number 110: switching the two ones doesn't give a new number, and swapping two classically identical particles doesn't give a new state. They didn't understand why they had to adjust their counts like this to get the right answer. If computer scientists had been around, they might have suggested that particles are like patterns of bits in a computer memory, and swapping identical patterns between two memory locations doesn't produce a new state of the memory.

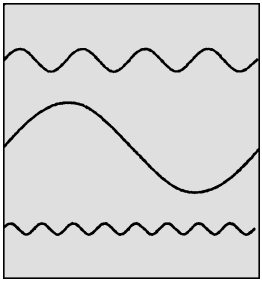

Fig. 1. Radiation field inside a cavity. The boundaries are periodic, and so each Fourier mode must be periodic.

field inside the cavity, and the boundaries in the illustration are periodic. We can analyze the field as a sum of some number of Fourier modes, each with an integer number of cycles that fit between the boundaries. There are an infinite number of these, with higher and higher frequencies. Classically, the energy of each mode only depends on its amplitude, and so if you have a unit of energy that you're adding to the system, you can add it to any one of these modes: there are an infinite number of places to put the unit of energy. Even if you coarse grain the amplitudes, there are *still* an infinite number of places to put the energy! You don't get a finite number by coarse graining.

Max Planck solved this problem around the beginning of the last century. What Planck proposed was that for each mode, there is a minimum energy that is proportional to the frequency, with a universal constant of proportionality that he introduced. Only integer multiples of the minimum energy can be added to a given mode. This means that given a unit of energy to add to the system, there are only a finite number of places it can be put—modes with energies too high cannot be "excited." Given the integer-multiple constraint, there are in fact only a finite number of way to divide up the given unit of energy to add it to the system.

Planck's strange new rule gave a finite count for the entropy, for the heat capacity, and even gave agreement with experiment! He didn't have to figure out the dynamics, he just initially figured out the counting that was needed: Quantum Statistical Mechanics is easier than QM, and was invented first. Soon, it was understood how to apply Planck's quantization to the rest of physics, to make the counting of physical entropy of every finite system finite. Planck's new constant was in fact the grain size that needed to be used to do coarse graining of classical *position* $\times$ *momentum* in general, to get the correct quantum counting of distinct states![5] This grain size was no longer an arbitrary unit that

[5] Since an integer number of wavelengths must fit across a cavity of width W, all mode frequencies must be integer multiples of $\nu_1 = c/W$. If just one mode is minimally excited (i.e., just a single *photon*), then with energy E there are $E/h\nu_1$ possible modes for the photon. Since a photon has two possible polarizations, there are actually twice this number of possible states. Since $E = cp$, the number of possible states for a single photon can be written as $2cp/h\nu_1 = W \times 2p/h$, which is just the number of states for a single particle in a box if the *position* $\times$ *momentum* grain size is h.

cancels out in Statistical Mechanics calculations: it was a new fundamental unit of Nature.

As an aside, we should note that QM defines what it means physically to have a set of distict states: this is a set of states that can, in principle, be distinguished from each other with perfect fidelity. Such a set of *mutually orthogonal* states must be used for doing the counting. A physical system with finite extent and finite energy has only a finite set of distinct possible states.

4. COMPUTATIONAL MECHANICS

The dynamics of finite-state systems is quite familiar to computer scientists, and if computers had existed when it was realized that the world is a finite state machine, it would have been natural to try to describe Nature in computational terms. Here we will develop a statistical description of classical finite-state systems, and lead into QM from that viewpoint—giving a classical computational interpretation of energy and action along the way.

Figure 2 shows a logic gate. This is just ordinary computer logic. A and B go in; A comes out and $A + B$ (modulo 2) comes out. This is a reversible logic gate: A is one of the outputs, and if you add the two outputs, modulo 2, you get B back again. In fact, this gate is its own inverse: if you apply it again to the two outputs, you get back the original inputs.

We can describe this gate's dynamics using vector notation. Here you should try to forget for a moment that you may have seen the bra-ket notation used in QM, and we'll reintroduce it in this classical context. The symbol $|00\rangle$ represents a vector in a vector space. We associate a distinct basis vector with each possible input state of our logic gate: $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ are the four possible input cases. Now we describe what the gate does to each possible input state using a linear operator U_{xor}:

$$\begin{aligned} U_{\text{xor}}|00\rangle &= |00\rangle, \\ U_{\text{xor}}|01\rangle &= |01\rangle, \\ U_{\text{xor}}|10\rangle &= |11\rangle, \\ U_{\text{xor}}|11\rangle &= |10\rangle. \end{aligned} \tag{1}$$

$|00\rangle$ and $|01\rangle$ are unchanged by this gate, and $|10\rangle$ and $|11\rangle$ turn into each other. If we apply U_{xor} twice, we always get back the state we started with. Note that we can represent *any* logic operation on a fixed set of bits using this kind of linear-algebra

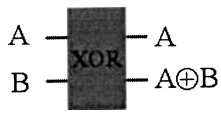

Fig. 2. A reversible logic gate with two inputs and two outputs.

notation, since we've associated a distinct basis vector with each possible state, and so we can decide separately what happens to each. By using 2^n basis states for an n-bit system, we are able to represent even nonlinear logic using linear operators.

5. CLASSICAL STATISTICAL ENSEMBLES

Now suppose you wanted to describe an ordinary statistical ensemble of computations that employ a gate like that of Fig. 2. You start off with state $|00\rangle$ with probability a, $|01\rangle$ with probability b, and so on. After we apply the gate once to the initial state, we get

$$a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle \rightarrow a|00\rangle + b|01\rangle + d|10\rangle + c|11\rangle \qquad (2)$$

For example, if we start off in $|10\rangle$ 1/3 of the time, then we end up in the state $|11\rangle$ 1/3 of the time, since $|10\rangle$ turns into $|11\rangle$. In other words, the probabilities just move around to follow the states. Statistical ensembles of invertible logic circuits are naturally described using linear algegra, since whatever probability p_s you had of starting the computer in a particular state $|s\rangle$, after applying n steps of computation U you end up in the nth successor state $U^n|s\rangle$ with the same probability p_s. If the computation is invertible, distinct states remain distinct, and so probabilites just move around between basis vectors.

Now suppose we simply take Eq. (2) and replace all probabilities with their square roots:

$$\sqrt{a}|00\rangle + \sqrt{b}|01\rangle + \sqrt{c}|10\rangle + \sqrt{d}|11\rangle \rightarrow \\ \sqrt{a}|00\rangle + \sqrt{b}|01\rangle + \sqrt{d}|10\rangle + \sqrt{c}|11\rangle. \qquad (3)$$

Since there is no mixing of probabilities under an invertible dynamics, it makes no difference informationally if we use probabilities directly as coefficients, or some encoded form of the probabilities, such as $\sqrt{p}$. The reason that $\sqrt{p}$ is convenient to use with a vector description of ensemble dynamics is that the sum of the squares of the coefficients is a meaningful geometrical quantity: the *vector length* is just the total probability, which is always one. This probabilistic character of vector length will be important when we want to analyze our systems in different bases. This notation, with square roots of probabilities (called *probability amplitudes*), is the one that is used in QM. The amplitude-weighted sum of basis states is called a *superposition*.

More generally, if we take any invertible logic circuit and describe its action on a set of bits using linear operators U as above, then the behavior of that circuit with a Gibbs ensemble of possible initial states is naturally described using basis vectors, probability amplitudes, superpositions, and length-preserving invertible operators (so-called *unitary operators*). There is no mixing of the amplitudes, they just follow the states. We could do Monte Carlo sampling, starting the circuit off in

different initial states with specified probabilities, running n steps of time evolution, and getting final states with the same probabilities predicted by applying U^n to the probability vector describing the ensemble of initial states. There is nothing mysterious about this quantum mechanical language—it is just a natural way to describe the classical evolution of ensembles of inputs.

6. THE ENERGY BASIS

We have described time evolution in terms of what it does to some complete basis of states. For ordinary invertible circuits, the basis states are just all possible configurations of the bits, where each bit is localized in space and is either zero or one. The dynamics of a *superposition* of configuration states describes the time evolution of a statistical ensemble.

Now suppose that we have a circuit that is governed by the time evolution U_τ, applied repeatedly. For example, U_τ might describe one clock-cycle of the operation of a reversible computer. The operator U_τ turns one configuration of the bits of the computer into the next. Since the computer is finite, some configuration of bits will eventually repeat. Since the evolution is deterministic, all subsequent configurations will then also be repeats. Since the evolution is invertible, the first repeated configuration will be the one we started with: all dynamical histories form closed cycles. For example, a cycle of configuration states of length N might look like

$$|X_0\rangle \to |X_1\rangle \to \cdots \to |X_{N-1}\rangle \to |X_0\rangle. \tag{4}$$

If we add together, with equal weight, all of the configuration states in a cycle, we produce a time-invariant state: the dynamics turns this superposition into itself!

$$|E_0\rangle = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} |X_m\rangle,$$

$$U_\tau |E_0\rangle = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} |X_m\rangle. \tag{5}$$

Such a state describes a circuit that is at some unknown point in time in a known repeating cycle of configurations. By adding the same set of configurations together with different sets of amplitudes, all of equal magnitude, we can form a complete basis out of time-invariant states. This is called the *energy basis*—the connection to the classical notion of energy will be discussed in the next section.

As a very simple example, consider a one-bit system with U_τ the NOT operation that changes the configuration $|0\rangle$ into $|1\rangle$ and vice versa. This system has only two configuration basis states, and they are turned into each other by U_τ to

form a cycle of length 2. We can construct a new two-element *energy basis* by adding the two configuration states in two different ways:

$$|E_0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}},$$
$$|E_1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}. \tag{6}$$

The original configuration states can be recovered by adding and subtracting the energy states, and so this is a complete basis. When U_τ acts on the energy states $|E_0\rangle$ and $|E_1\rangle$, we see that

$$U_\tau|E_0\rangle = \frac{U_{\text{not}}|0\rangle + U_{\text{not}}|1\rangle}{\sqrt{2}} = |E_0\rangle,$$
$$U_\tau|E_1\rangle = \frac{U_{\text{not}}|0\rangle - U_{\text{not}}|1\rangle}{\sqrt{2}} = -|E_1\rangle. \tag{7}$$

Each energy basis state is invariant under the dynamics, with only its overall sign changing with time. More generally, for a cycle of configuration states of length N, where U_τ takes us from one configuration $|X_m\rangle$ to the next configuration $|X_{m+1}\rangle$, we can construct

$$|E_n\rangle = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} e^{2\pi inm/N}|X_m\rangle,$$
$$U_\tau|E_n\rangle = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} e^{2\pi inm/N}|X_{m+1}\rangle,$$
$$= e^{-2\pi in/N}|E_n\rangle. \tag{8}$$

Since we only require the square of the magnitudes of our amplitudes to be probabilities, we are free here to use complex amplitudes in constructing the new basis, while retaining an equal-probability-for-all-configurations-in-a-cycle interpretation of the energy basis states. Using complex amplitudes provides enough different coefficients of equal magnitude to construct N orthogonal time-invariant basis states.

We should make it clear what we mean here by orthogonality. We define a vector $|A\rangle$ to be equivalent to a column vector consisting of its components in some basis, and the dual vector $\langle A|$ is equivalent to a row vector consisting of the complex conjugates of the same components. Then the inner product $\langle A|A\rangle$ is just the matrix product of these two quantities, and equals the sum of the squares of the magnitudes of the components. Thus if $|A\rangle$ is a superposition of configurations with unit total probability, $\langle A|A\rangle = 1$. It is then easy to verify that $\langle E_j|E_k\rangle = \delta_{j,k}$, as you would expect for a basis.

The energy and configuration bases are discrete Fourier transforms of each other: we can express $|X_m\rangle$ in terms of the $|E_n\rangle$ as

$$|X_m\rangle = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} e^{-2\pi i n m/N} |E_n\rangle. \tag{9}$$

This can be verified by substituting in the definition of $|E_n\rangle$ in terms of configuration states. If we interpret the magnitudes of the amplitudes in a superposition of energy basis states as square roots of probabilities, then the probability of different energy basis states doesn't change with time: this is a kind of normal-mode decomposition of a dynamics.

If U_τ represents the change in the circuit that happens in a time interval of length τ, then in time t this operator will be applied t/τ times, and the phase of $|E_n\rangle$ (see Eq. (8)) will decrease by $(t/\tau \times (2\pi n/N)$. The phase change is proportional to time. Since this is the phase of a complex amplitude, the amplitude itself will repeat in value cyclically as t gets larger, repeating with a frequency of

$$\nu_n = \frac{n}{N}\nu, \tag{10}$$

where $\nu = 1/\tau$. If we were computer scientists living in the year 1901 and had just heard about Planck's work, we might think that it would be natural to call $E_n = h\nu_n$ the energy of a Fourier mode that cycles with a period of ν_n. This is in fact what we will do.

The operation of U_τ on configuration states can be described in the energy basis. Both the length and the motion of the state vector are independent of the basis—changing bases is just changing coordinate axes. If we interpret the square-magnitudes of the components in the energy basis as probabilities, then since these states are invariant in time, so are all probabilities in the energy basis. We can compute a time invariant energy for any state by expressing the state in the energy basis. For example, if

$$|X\rangle = \alpha|E_0\rangle + \beta|E_1\rangle, \tag{11}$$

then the "average" energy of $|X\rangle$ is

$$E = |\alpha|^2 E_0 + |\beta|^2 E_1, \tag{12}$$

and this is constant in time.

7. WHAT IS ENERGY?

Classically, kinetic energy is a measure of how much motion a system has. For a classical system with energy E and lowest possible energy $E_{\min}$, the most energy that could possibly be changed into kinetic energy is given by $E - E_{\min}$. This is the energy that is not somehow inextricably tied up—that you can actually turn into

motion of particles. We will see that in our statistical treatment of computational dynamics, the "average cycle rate" that we've called energy is also a measure of motion: it is the maximum rate of state change.

On the face of it, the classical and statistical notions of energy seem very different. One has to do with particle motion and "potential" motion, whereas the other has to do with state change. But in considering computational systems, all "motion" is state change. As we'll see in Section 7.2, these concepts are essentially the same.

For a computation that passes through a cycle of N configurations $|X_m\rangle$ at a rate $\nu = 1/\tau$, the average value of the energy of a configuration is simply $h\nu/2$. We can see this directly from Eq. (9): all energy states E_n appear in this expression with equal and constant probabilities, and the corresponding energy values are

$$0, h\nu/N, 2h\nu/N, 3h\nu/N, \ldots, (N-1)h\nu/N \tag{13}$$

and so, for N large, the average is $E = h\nu/2$. We can also turn this expression around, to say that the rate at which this system passes through a set of configurations is governed by its energy:

$$\nu = 2E/h. \tag{14}$$

Perhaps the best way to express this is in terms of action. If we multiply both sides of this equation by t, the total time that we let the system run, then we see that the number of configurations $\Omega(t)$ that the system passes through in time t is

$$\Omega(t) = 2Et/h. \tag{15}$$

Suppose we have an array of bits that all change simultaneously in one particular relativistic reference frame. In almost any other reference frame, the bits will change one at a time. Thus in general the transitions we should be thinking about in our discussion above are configuration changes in which a single bit may change. This means that, in general, energy should be interpreted as the maximum rate at which bits can change (times $h/2$). A computer scientist might call this the maximum number of operations per second. Action is then the maximum number of bit changes in a given time.

7.1. Serial Versus Parallel

Let's look at the interpretation of energy in spatially extended systems in more detail. Suppose we have N separate, completely uncoupled computers. From our earlier calculations, the energy of each computer i is simply $h\nu_i/2$, where ν_i is the rate at which computer i passes through a sequence of distinct configurations. Since the computers are completely uncoupled, the rate at which the entire collection of

computers goes through distinct states is

$$\nu_{\text{tot}} = \sum_i \nu_i \tag{16}$$

and thus if we call $h\nu_{\text{tot}}/2$ the energy of the entire system, we see that it is just the sum of the individual energies. For a fixed total energy, it doesn't matter how we divide it up among the various computers, the total number of distinct states per second depends only on the total energy.

In particular, if we only run one computer at a time, letting each run at the rate ν_{tot} for some interval, then the total energy of the system and the total rate of processing are unchanged. In other words, if we have no parallelism at all, and run the collection as a single serial computer, applying its processing power to only one place at a time, the overall energy is unchanged.

This leads us to the conclusion that we can serialize our description of parallel computers (Margolus, 1986, 1990), and we will still calculate the right energy! If only one spot at a time in a lattice is allowed to change, then the total energy tells us the maximum number of spots that can change, per unit time. If we double the size of the lattice but want to continue to update the entire lattice at the same rate, then the energy must also double.

7.2. Computers With Conservation Laws

If we have a computational system with a conservation law, then the maximum number of bits that can change per unit of time may be reduced. For example, if we have a simple single-speed lattice gas (Frisch *et al.*, 1986; Hardy *et al.*, 1976; Margolus, 2002) that conserves the number of ones and zeros, then if there are M ones in the system, there are at most $2M$ spots in the lattice that can change in one update of the entire lattice (M spots can change from 1 to 0, and M from 0 to 1). If some ones land on spots vacated by other ones, the number of changes is less. If fewer than half of the cells contain ones, then not all spots can change in a single update of the lattice. If we think of these ones as particles, then they have a period associated with them, which is the time that it takes a particle to move from one cell to the next. This is also the time that it takes to update the entire lattice, and so the frequency of lattice updates ν_u is also the frequency associated with these particles. If we associate a unit of energy $h\nu_u$ with each particle, then the maximum rate at which spots can change is

$$\nu_{\text{change}} = 2M\nu_u = 2E/h, \tag{17}$$

where $E = Mh\nu_u$. Thus even if we restrict ourselves to the portion of the energy associated with the particles, we still see that energy governs the maximum rate of evolution between distinct states according to Eq. (14).

8. QUANTUM COMPUTATION

Everything that we've said so far is really just standard QM. It is perfectly consistent to use this language to talk about statistical situations involving reversible circuitry—the quantum computing folks use such language all the time. As long as time evolution operators U simply permute states when expressed in the configuration basis, then nothing very strange is going on. This is, in fact, exactly where the probabilistic interpretation of quantum amplitudes comes from: from considering states such as those discussed above, which just describe statistical ensembles.

The only thing that QM adds is that U can, at least in principle, be any unitary operator—any operator that conserves probability. An ordinary reversible logic gate turns each distinct input configuration into a distinct result configuration. For example, a NOT gate turns $|0\rangle$ into $|1\rangle$. A distinctively quantum gate can turn a single configuration into a superposition of several configurations, or vice versa. For example, $|0\rangle$ might turn into $(|0\rangle - |1\rangle)/\sqrt{2}$ and vice versa. This latter example conserves total probability, but allows probabilities to appear and disappear.

Let's look at these examples in more detail. Consider U_τ that implements the NOT operation:

$$\begin{aligned} U_\tau|0\rangle &= |1\rangle, \\ U_\tau|1\rangle &= |0\rangle. \end{aligned} \tag{18}$$

What might $U_{\tau/2}$ do? Suppose we define U_t for continuous time t as

$$\begin{aligned} U_t|0\rangle &= \cos \pi t/2\tau|0\rangle - \sin \pi t/2\tau|1\rangle, \\ U_t|1\rangle &= \sin \pi t/2\tau|0\rangle + \cos \pi t/2\tau|1\rangle. \end{aligned} \tag{19}$$

Clearly the sum of the squares of the amplitudes is 1, so this definition conserves probability. In fact, this is just a rotation of the basis, and so the opposite rotation undoes it—this U_t is invertible. For $t = \tau$, this is just the NOT operation (with an extra minus sign that doesn't change the probability). For $t = \tau/2$, we get a kind of "square root of NOT" (Hayes, 1995):

$$\begin{aligned} U_{\tau/2}|0\rangle &= \frac{|0\rangle - |1\rangle}{\sqrt{2}}, \\ U_{\tau/2}|1\rangle &= \frac{|0\rangle + |1\rangle}{\sqrt{2}}. \end{aligned} \tag{20}$$

If we start off with a bit that is a zero (state $|0\rangle$), then after one application of $U_{\tau/2}$, we are in a configuration with equal probability of either value, $|0\rangle$ or $|1\rangle$. If we apply $U_{\tau/2}$ to this same bit a second time, then we arrive at the state $-|1\rangle$. This is a state with 100% probability of being a one. This seems somewhat at odds with our interpretation of the square-magnitudes of the amplitudes as probabilities, since

the probabilities come and go! Any interpretation in terms of the evolution of a completely described ensemble would not have this property.

By combining ordinary reversible logic with operations that act on single bits and make probabilities come and go, we can construct any unitary U (Barenco *et al.*, 1995). Thus all distinctively QM computation can be regarded as a combination of ordinary computation plus probability-changing single-bit operations such as that of Eq. (20).

9. THE POWER OF QM COMPUTATION

We should not get carried away by the strangeness of QM, and ascribe exaggerated capabilities to it. With enough work, the equations governing any QM system can be numerically integrated on an ordinary computer: QM doesn't change *which* things are computable and which are not. It may, however, change the *amount of effort* required to compute a desired result.

With an ordinary computer, we start it in some definite configuration, and we run it once to get a definite result. With a distinctively QM computer, even if we start it in a definite configuration and it finishes in a definite configuration, probabilities may arise and change during the course of the computation. To simulate this on an ordinary computer may require an enormous effort: we may need to keep track of the amplitudes of essentially all possible intermediate configurations in order to get a single result. If we could run the corresponding QM system just once, we would get the same result with much less work!

This observation, that distinctively QM computers are hard to simulate on ordinary computers, led Richard Feynman to speculate that perhaps such computers could do some computations faster than ordinary computers (Feynman, 1982; Lloyd, 1996). He suggested that an interesting class of such computations might be the simulation of physical systems where QM correlations are important. More recently, there have been proposals that distinctively QM computers could perform some interesting classical computations rapidly (Grover, 1996; Shor, 1994).

It is worth noting, in thinking about such proposals, that the distinction between ordinary computation and distinctively QM computation is basis dependent. If an ordinary computation is expressed in a basis other than the configuration basis, the time evolution will generally involve probabilities that appear and disappear. Conversely, a distinctively QM computation can be described in bases in which probabilities never change, or even in bases where, at regular time intervals, the system moves from basis state to basis state (Margolus and Levitin, 1998). How can QM computations have any extra power if there are bases in which they act like ordinary computations?

The answer is that, with more freedom to choose the time evolution operator U_τ, we may be able to speed up the computation. In practice, the only U_τ's available

are those we can piece together from the U's that are provided by Nature.[6] The extra component that QM provides is one which changes single-bit probabilities. Does this let us compute faster?

This question is closely related to the more fundamental question, What is the difference between a Quantum world and a Classical world? Might it be that we are simply analyzing our world in the wrong configuration basis, and so we see probabilities that appear and disappear? QM computation provides a direct challenge to this possibility: if there really are some classical computations that can be performed exponentially faster in a QM world than using just classical logic elements, then describing Nature in terms of a classical computational substratum is unnatural. But it may well turn out that, when compared to the right classical hardware and algorithms, QM doesn't actually let you solve problems faster.[7] It might also turn out that proposed QM computers aren't actually realizable, or large QM computers must be very slow for some reason. We'll find out as we try to build them.

10. ACTION OF A COMPUTATION

In our discussion above, we saw that for a classical computation, the energy gives the maximum rate at which the system can pass through a sequence of distinct (i.e., mutually orthogonal) states. This is true for any unitary evolution (Margolus and Levitin, 1998): for the fastest moving sequence of states, the maximum rate of orthogonal evolution is given by $2E/h$, where E is the energy of the system, taking the ground state energy as zero.

Putting this another way, $2Et/h$ is the greatest number of distinct states that a long unitary evolution can pass through in time t. Thus *action in time* (i.e, Et) counts the maximum number of distinct states for a system, given E and t. This is analogous to the semiclassical result, used frequently in Quantum Statistical Mechanics, that *action in space* (i.e., px) counts the maximum number of distinct states for a system, given a range of momenta of size p and a range of position of size x.

In different relativistic frames, we would expect these space and time counts of distinct states to be mixed together. Figure 3(a) shows a system (e.g., an atom) viewed from its rest frame. In this frame, the maximum number of changes that we could see in time t is going to be $2Et/h$. Now even if the atom isn't changing at all, if we move our head at some rate relative to this atom (Fig. 3(b)), we see a distinct state after some amount of motion. This change has nothing to do with anything that is going on internally in the atom—it has only to do with our relative

[6] These are all presumably special cases of some fundamental U_τ that we don't know.

[7] The factoring problem (Shor, 1994) may not be exponentially hard, and general database searches (Grover, 1996) can be sped up with parallel hardware.

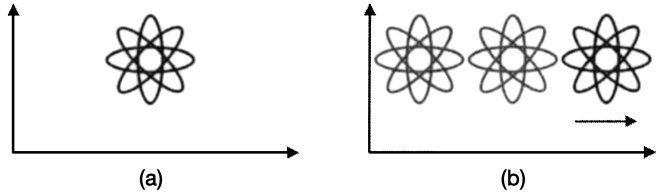

Fig. 3. When we are moving relative to a system, we see additional distinct states just because of our motion.

state of motion. Since $Et - px$ gives the value of Et in the rest frame, $2(Et - px)/h$ counts the maximum number of distinct changes in that frame.[8]

Connections between orthogonal evolution and action allow us to put bounds on the maximum performance of real computations. If we know how many distinct states the computation must pass through (i.e., how many bit-changing operations it must perform), we have a bound on the action resources needed for that computation. If the ideal algorithm abstracts away some implementation details, then there will be additional distinct states needed for any actual implementation, and so additional action—the ideal algorithm only gives a bound. This also associates a minimum energy with a given computation that proceeds at a particular rate: at least 3×10^{-34} J must be available for each operation per second. This energy doesn't have to be dissipated by the computation, it just has to be available for use by the computer.

11. INFORMATION MECHANICS

In Section 7.2, we discussed the example of a lattice gas that conserves the number M of ones on the lattice, and concluded that M can be interpreted as a form of energy, since it governs the maximum rate of change of spots on the lattice. In the same example, the number of ones M also governs how much information can be represented on the lattice, and so in this case energy governs both rate of computation and amount of memory.

This is a typical situation in statistical mechanics in cases where all forces are purely statistical. As an illustration, consider the well known and very simple model of a spring (Kubo, 1965) shown in Fig. 4. Here we have a one-dimensional chain consisting of N links, starting at the origin (dark circle) and ending some distance x away. Each link is a distance l long, and the joints connecting the links (shown as circles) have no configurational energy associated with which way the link is pointing. Each link extends the chain either one unit to the right, or one unit to the left. If p_+ is the fraction of the links that extend to the right, and p_- the

[8] By the same token, $(H - p\dot{x})\,dt$, which appears in action integrals, has an informational interpretation.

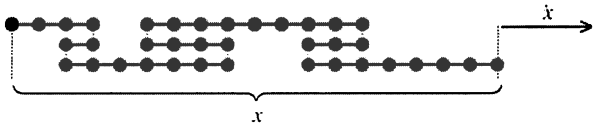

Fig. 4. A statistical model of a spring. The model is one-dimensional, with the extra dimension here used to show parts of the chain that are folded over other parts. The origin is the dark circle at the left. Each link extends the chain one unit rightwards or leftwards.

fraction that extend to the left, then the distance x is given by

$$x = Nl(p_+ - p_-). \tag{21}$$

We can define a dimensionless length variable u that represents the fraction of x associated with each link,

$$u = x/Nl. \tag{22}$$

Then p_+ and p_- can be rewritten in terms of u as

$$p_\pm = (1 \pm u)/2. \tag{23}$$

The entropy associated with a chain of length x is just the $\log_2$ of the number of microscopic configurations that give us a chain of that length,[9] and can be written approximately in terms of $p_\pm$ as

$$S_x = -N(p_+ \log_2 p_+ + p_- \log_2 p_-). \tag{24}$$

If there were no constraint on the total length of the chain, then each link could be freely chosen to point to the right or the left, and so $p_+ = p_- = 1/2$, $x = 0$, and the maximum entropy for this chain is

$$S_0 = N. \tag{25}$$

We will call the difference between the maximum and the actual entropy of the chain the *length constraint information*,

$$I_x = S_0 - S_x = \frac{1}{2}N\{(1+u)\log_2(1+u) + (1-u)\log_2(1-u)\}. \tag{26}$$

This is the number of bits of state information that are being used to remember the length constraint. Now, since we've assumed that there is no energy associated with the joints in our chain, all of the work that we do if we pull or push on the end of this chain is work done on the heat bath—we don't provide an explicit interaction mechanism with the heat bath, but simply assume this. Thus to extend

[9] We've chosen units in which Boltzmann's constant $k_B = \log_2 e$.

the chain from its equilibrium position to a length x requires energy

$$E_x = T(S_0 - S_x) = I_x T. \tag{27}$$

This relation between energy and entropy comes from thermodynamics and assumes that the process happens at constant temperature, slowly enough so that the system is always very nearly in equilibrium, and that no other energy enters or leaves the system.[10] For small displacements ($u \ll 1$), I_x is proportional to u^2, and so the potential energy E_x associated with a displacement x is proportional to x^2 (Hooke's Law).

This example says that, in a situation in which forces are statistical, E_x/T, the potential energy divided by the temperature, is simply the number of bits that the system is using to remember its macroscopic configurational constraint. If we attach a mass to this spring, and watch it exhibit simple harmonic motion, then we will see this constraint information flow back and forth between a configurational constraint, and a kinetic constraint. In a classical finite state model of the mass, this might translate into a constraint on the distribution of momenta of the components of the mass. Any stationary-action principle governing this kind of system is purely combinatorial, and has nothing to do with any special properties of QM.

12. CONCLUSION

Informational models have long been a cornerstone of Statistical Mechanics, but computational models of dynamics remain second-class citizens. I believe that the development and analysis of such models will help clarify fundamental issues in physics, and allow us to generalize concepts thought to be the unique province of physics. For Computer Science, ideal computations that map directly onto unitary time evolutions may provide useful bounds on the physical requisites of computation, such as time, volume, energy, heat, and power. I expect that there will be a profound interplay between the concepts and models of Physics and of Computer Science as computation continues to migrate to increasingly microscopic realms, and as our understanding of physics becomes more computational.

ACKNOWLEDGMENTS

Early exposure to John Conway's "Game of Life" Cellular Automation (Berlekamp *et al.*, 1982) first showed me that a computer could have a physics-like structure: Might simple computer models underly physics? Fredkin (1990, this issue), who was my PhD thesis advisor, was the one who first made me aware of the importance of reversibility and reversible computation in trying to make this connection. His many brilliant and original ideas in this area, including the classical

[10] This is essentially the definition of change of entropy in terms of heat energy added at constant temperature: $\Delta Q = T \Delta S$. Since the work is done on the heat bath, $\Delta Q = E_x$.

mechanical Billiard Ball Model of computation (Fredkin and Toffoli, 1982), were a wonderful source of insights, excitement, and inspiration. Toffoli (1982, 1984, 1990) has also enthusiastically shared his deep insights into connections between computational and informational ideas in physcis, and we collaborated for many years, playing with ideas and with parallel cellular automata hardware (Toffoli and Margolus, 1987).

REFERENCES

Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., Sleator, T., Smolin, J., and Weinfurter, H. (1995). Elementary gates for quantum computation. *Physical Review A* **52**(5), 3457–3467.

Bennett, C. H. (1982). The thermodynamics of computation—A review. *International Journal of Theoretical Physics* **21**, 905–940.

Bennett, C. H. (this issue). Quantum information. In *Proceedings of the Workshop on Digital Perspectives*.

Bennett, C. H. and Landauer, R. (1985). The fundamental physical limits of computation. *Scientific American* **253**(1), 38–46

Berlekamp, E., Conway, J., and Guy, R. (1982). *Winning Ways for Your Mathematical Plays, Vol.* 2, (Academic Press, New York).

Burks, A. (ed.) (1970). *Essays on Cellular Automata*, University of Illinois Press, Champaign, IL.

Farmer, D., Toffoli, T., and Wolfram, S. (eds.) (1984). *Cellular Automata*, North-Holland, Amsterdam; reprinted from *Physica D* **10**(1/2) (1984).

Feynman, R. P. (1982). Simulating physics with computers. *International Journal of Theoretical Physical* **21**, 467–488.

Fredkin, E. (1990). Digital mechanics: An informational process based on reversible universal CA. In *Cellular Automata: Theory and Experiment*, H. Gutowitz, ed. North-Holland, Amsterdam, pp. 254–270.

Fredkin, E. (this issue). Digital philosophy. In *Proceedings of the Workshop on Digital Perspectives*.

Fredkin, E., Landauer, R., and Toffoli, T. (eds.) (1982). Proceedings of the physics of computation conference. *International Journal of Theoretical Physics* **21**(3/4, 6/7, and 12).

Fredkin, E. and Toffoli, T. (1982). Conservative logic. *Internation Journal of Theoretical Physics* **21**, 219–253; reprinted in *Collision Based Computing*, A. Adamatzky, ed. Springer, Berlin, pp. 47–82.

Frisch, U., Hasslacher, B., and Pomeau, Y. (1986). Lattice-gas automata for the Navier-Stokes equation. *Physical Review Letters* **56**, 1505–1508.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC)*, pp. 212–218; and quant-ph/9605043.

Gutowitz, H. (ed.) (1990). *Cellular Automata: Theory and Experiment*, North Holland, Amsterdam; reprinted from *Physica D* **45**(1/3).

Hardy, J., de Pazzis, O., and Pomeau, Y. (1976). Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Physical Review A* **13**, 1949–1960.

Hayes, B. (1995). Computing science: The square root of NOT. *American Scientist* **83**(4), 304–308.

Kubo, R. (1965). Statistical Mechanics, North-Holland, Amsterdam.

Lloyd, S. (1996). Universal quantum simulators. *Science* **273**, 1073–1078.

Margolus, N. (1986). Quantum computation. In *New Techniques and Ideas in Quantum Measurement Theory*, D. Greenberger, ed., New York Academy of Sciences, New York, pp. 487–497.

Margolus, N. (1990). Parallel quantum computation. In *Complexity Entropy, and the Physics of Information*, W. Zurek, ed. Addison-Wesley, Reading, MA, 273–287.

Margolus, N. (1998). Crystalline computation. In *Feynman and Computation*, A. J. G. Hey, ed., Perseus Books, Reading, MA, pp. 267–305; comp-gas/9811002.

Margolus, N. (2002). Universal cellular automata based on the collisions of soft spheres. In *Collision Based Computing*, A. Adamatzky, ed. Springer, Berlin, pp. 107–134; Also to appear in *Constructive Cellular Automata*, C. Moore and D. Griffeath, eds.

Margolus, N. and Levitin, L. (1998). The maximum speed of dynamical evolution. *Physica D* **120**(1/2), 188–195; quant-ph/9710043.

Shor, P. W. (1994). Algorithms for quantum computation: Discrete log and factoring. In *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*, IEEE, Piscataway, NJ, pp. 124–134; and quant-ph/9508027.

Toffoli, T. (1982). Physics and computation. In *International Journal of Theoreratical Physics* **21**, 165–175.

Toffoli, T. (1984). Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. In *Cellular Automata*, D. Farmer, T. Toffoli, and S. Wolfram, eds., North-Holland, Amsterdam, pp. 117–127.

Toffoli, T. (1990). How cheap can mechanics' first principles be? In *Complexity, Entropy, and the Physics of Information*, W. Zurek, ed., Addison-Wesley, Reading, MA, pp. 301–318.

Toffoli, T. and Margolus, N. (1987). *Cellular Automata Machines—A New Environment for Modeling*, MIT Press, Cambridge, MA.

Zurek, W. (ed.) (1990). *Complexity, Entropy, and the Physics of Information*, Addison-Wesley, Reading, MA.