An Overview of Lattice-Gas Dynamics

Jeffrey Yepez

November 1997

Contents

1 Introduction **2**

2 Viscous Multiphase Fluids **7**

3 Ways to Simulate a Fluid **10**

4 Numerical Round-Off Error **12**

5 Molecular Dynamics **13**

6 Shrinking Bits **16**

7 Lattice Molecular Dynamics **18**

8 Artificially Discrete Microworlds **20**

9 Classical Lattice Gas **22**

10 Lattice-Gas Paradigm **23**

11 The CAM-8 Prototype **25**

12 Limitations and Drawbacks of Classical Lattice Gases **28**

13 Classical Lattice Gases on Quantum Computers **31**

14 Quantum Lattice Gases on Classical Computers **33**

15 Quantum Lattice Gases on Quantum Computers **34**

1 Introduction

One can argue it is the case that the fundamental nature of the physical world is that it is quantized in such a way that phasespace is granular[1], and one can observe that digital computation is discrete and granular too. Given these similarities, one might try to see just how far one can go in "connecting" the two. In this regard, Richard Feynman gave a talk entitled "Simulating Physics with Computers" in 1981: "I want to talk about the possibility that there is to be an *exact* simulation, that the computer will do *exactly* the same as nature. If this is to be proved..., then it's going to be necessary that *everything* that happens in a finite volume of space and time would have to be exactly analyzable with a finite number of logical operations. The present theory of physics is not that way, apparently. It allows space to go down into infinitesimal distances, wavelengths to get infinitely great, terms to be summed to infinite order, and so forth..." In this seminal talk and in subsequent papers [27, 28], Feynman discussed an interesting possibility: the possibility of constructing a quantum computer to simulate quantum mechanics.

As the fundamental computational element's size reduces to nanoscale ranges its behavior is governed by quantum mechanics. There is hope that in the future computation will be achieved with "quantum gates" [46, 42, 9, 33, 4]. Follow-

[1] To see this, count the number of possible energy levels for a particle in a cubical box of length side X. The particle's momentum components are quantized by periodic boundaries conditions so $p_i = \frac{hn_i}{X}$, where $i = (1, 2, 3)$ is an index over spatial directions and n_i are integers. Therefore to count the number of states $E \leq E_\circ$, we have

$$\frac{p_1^2}{2m} + \frac{p_2^2}{2m} + \frac{p_3^2}{2m} \leq \frac{P^2}{2m},$$

where $E_\circ = \frac{P^2}{2m}$. In terms of the quantum numbers this is

$$n_1^2 + n_2^2 + n_3^2 \leq \frac{P^2 X^2}{h^2}.$$

Defining an effective radius as $R \equiv \frac{PX}{h}$ it is clear that the number of energy levels for a particle in a box is equal to the count of the number of points in a sphere of radius R. This is just the volume of the accessible phasespace in units of Planck's constant. Therefore,

$$N(E \leq E_\circ) \sim \frac{4\pi}{3} R^3 = \frac{V_{\text{phasespace}}}{\delta V_{\text{phasespace}}},$$

where the volume of phasespace is $V_{\text{phasespace}} = \frac{4\pi}{3}(PX)^3$ and the smallest unit of phasespace is $\delta V_{\text{phasespace}} = h^3 \sim (\delta p \delta x)^3$. So according to quantum mechanics, phasespace is granular.

ing present-day electronic computer design philosophy, research has focused on quantum gate counterparts of well known universal reversible logic gates, for instance the two-input/two-output quantum XOR gate [4]. The prevailing expectation is to use the simplest universal quantum gates in networks to fashion arbitrary n-bit unitary operators. But the bits in such a "quantum computer" are quite different than the bits we are accustomed to in present-day conventional computers. A review of quantum computation has been provided by Ekert and Jozsa [25].

In quantum computing a two-level quantum object–termed a quantum bit or *qubit*–represents the smallest unit of information [27, 5, 23, 45, 8, 35]. Unlike a classical bit, a qubit, $\mid q\rangle$, may be in a superposition of the Boolean states $\mid 0\rangle$ and $\mid 1\rangle$ so that $\mid q\rangle = \alpha \mid 0\rangle + \beta \mid 1\rangle$. If one measures the value of the qubit there would be a certain amplitude, α, of it being in the *zero state*, $\mid 0\rangle$, and another amplitude, β, of finding it in the *one state*, $\mid 1\rangle$. The probabilities must add to unity: $\langle 0 \mid 0\rangle + \langle 1 \mid 1\rangle = \langle q \mid q\rangle$ so the complex coefficients are constrained by $\mid \alpha \mid^2 + \mid \beta \mid^2 = 1$. Examples of quantum objects used to represent qubits are two energy-level states of the fine structure splitting in the valence shell electron of a cesium atom held in a laser trap, or the z-component of the nuclear spin of an atom in a uniform external magnetic field. Light is used to initialize the individual spin states of the qubits (writing), then pairs of adjacent qubits interact via dipole-dipole coupling for example where the computing cycle is initiated by a particular sequence of light pulses, and finally light is used to measure the resulting individual spin states of the qubits (reading) [43]. This kind of controlled light-and-matter interaction is well known in nuclear magnetic resonance experiments where π-pulses are used to tip nuclear spins. For example, recently Cory *et al.* have employed the quantum number m_z of a nuclear spin of an atom in a molecule of a liquid placed in a strong external magnetic field to encode a single qubit and they used nuclear magnetic resonance to control its state and interaction with qubits in neighboring atoms within the same molecule [20].

Quantum computation aims to exploit the superposition of states as a prac-

tical means of parallel computing. This is termed *quantum parallelism* [23, 24]. After a decade of exploring Feynman's conjecture, it is now believed possible to indeed exploit quantum mechanics substitute quantum polynomial complexity for classical exponential complexity [61, 25, 63, 64, **?**]. Quantum computing relies on having interactions of a collection of qubits occurring in a controlled fashioned to achieve unprecedented parallelism not available in classical computing.

Shor's algorithm exemplifies how factoring can in principle be done on a quantum computer in a time that grows polynomially in the number of qubits used to encode a large composite number[2] [61]. It is difficult for a classical computer to factor a number a large composite number and this fact is the cornerstone upon which cryptographic algorithms are based.[3] Consequently, Shor's scheme for factoring numbers has stirred much interest in quantum computing.

An issue for quantum computing is isolating the qubits from the surrounding environment. Since interference effects among qubits are essential for the computation, uncontrolled coupling with the environment may destroy such effects. Quantum parallelism levies a high demand for coherence of the quantum computer's wavefunction to be realized by avoiding uncontrolled entanglement with the external world. Developing robust algorithms and scalable error correction techniques is considered crucial for the enterprise to continue [18, 8, 43, 25]. The tremendous difficulty of maintaining quantum coherence poses a problem that must be resolved before a quantum computer can be built. Because of the stringent demand for quantum coherence, prospects for any foreseeable quantum computers are focused on those containing only a very small number of

[2]Presently, there is no known classical factoring method in the polynomial complexity class, and furthermore, it is unproven that such a classical method does not exist. In Shor's algorithm, all factors are superposed in a quantum computer's "register" in polynomial time. A modulus operation is applied to all factors simultaneously producing a periodic function. The correct factor corresponds to the period of this function. A discrete Fourier transform is taken. Consequently, the peak in the power spectrum of the transformed data locates the factor. An amount of memory exponential in the number of bits of the composite number is not needed to perform the discrete transform. In this way a massively parallel search is accomplished in polynomial time.

[3]In 1994 a network of 1,600 computers around the world factored a 129-digit number in eight months, a 250-digit number would take two centuries on this network.

qubits.

In light of the possibilities and limitations of quantum mechanical computing, it is worthwhile to consider how one might profit from connections between quantum computing and physics. We shall show that lattice-based particle models implemented on hypothetical spatially fine-grained quantum computers exhibit, in the macroscopic scaling limit, an interesting "quasi-physical" dynamics that offers the computational physicist results not obtainable with conventional techniques.

Ever since Feynman's conjecture, many have imaged how a quantum computer might actually work. A good starting point is reversible computing [29]. Occasionally microscopic physics appears reversible—it is reversible only if there is no coupling to degrees of freedom such as photons or phonons that can escape to infinity, and that condition is rarely met. Since microscopic physics is sometimes well approximated as reversible, it is interesting to consider computational algorithms that are reversible. It is important to comprehend reversible algorithms useful for simulating reversible physics on a nanoscale device since these algorithms may serve as a guide for constructing such a device. The principle assumption is a quantum device itself undergoes reversible evolution as it progresses through its "computation". Furthermore, it is assumed that if irreversible evolution does occur, it is because the algorithm itself has caused this, not uncontrolled coupling to the external environment.

Even before quantum mechanical superposition of states, correlations, and entanglement become algorithmically important, the unitarity of the dynamical evolution becomes a significant benefit at very high logic densities where the dissipation of heat caused by irreversible computations is a menacing engineering issue [6, 7]. In nanoscale computing, one benefit of reversible algorithms is the avoidance of heat production. Since information is exactly preserved in a reversible algorithm, the entropy is constant throughout the course of the calculation, consequently since $dQ = TdS = 0$, no heat is produced.

For any reversible computation, one can describe the algorithm by a *computational evolution operator*, denoted $\hat{U}$, acting on the state data, $\mid \Psi\rangle$, which

is the configuration state of the computer memory. The new state data, $\mid \Psi' \rangle$, is generated at follows

$$\mid \Psi' \rangle = \hat{U} \mid \Psi \rangle. \tag{1}$$

By repeated application of $\hat{U}$ an ordered sequence of states is generated where each state is given a unique time label. If the first state is labeled by t then the next state is labeled by $t + \tau$, and the next by $t + 2\tau$, and so forth. With this understanding we write (1) as

$$\mid \Psi(t + \tau) \rangle = \hat{U} \mid \Psi(t) \rangle. \tag{2}$$

In this way the *computational time* advances incrementally in unit steps of duration τ. Of course the state of the quantum computer exists at all intermediate times, say at $t + \frac{\tau}{2}$, but for our purposes we need only use the state at intervals of the time step τ. The computer's quantum evolution is invertible by application of the adjoint of the evolution operator

$$\mid \Psi(t - \tau) \rangle = \hat{U}^{\dagger} \mid \Psi(t) \rangle. \tag{3}$$

This computational picture is consistent with the Heisenberg picture of quantum mechanics. For any reversible algorithm chosen, the task is to map the computational Hamiltonian of the algorithm on to the physical Hamiltonian of interacting qubits of the nanoscale device that is to implement the quantum computation.

What kind of physical simulation can be achieved on a fine-grained quantum computer? To understand the operation of a fine-grained quantum computer, it may be useful to understand first the operation of a fine-grained classical computer. For this reason, I first explore how a viscous fluid and a multiphase fluid can "live" in the confines of a computer simulation called a *classical lattice gas*, which has artificially discrete dynamics. In a classical lattice gas algorithm, the evolution operator is a permutation matrix with components being zero or one. The hope that simple discrete computational models such as a classical lattice gas can capture some of the behavior of fluids undergoing phase transitions has

contributed to my motivation to study of such artificial dynamical systems implemented on fine-grained computers; I argue in Volume that my hope has been realized.

Next, I explore how one might emulate a quantum fluid in an exact quantum computer simulation. A *quantum lattice gas* is a generalization of a classical lattice gas where quantum bits replace classical bits [74]. In a quantum lattice gas algorithm, the computational evolution operator is a general unitary matrix with complex components. The microscopic reversibility of the lattice-gas paradigm is compatible with the constraint that quantum mechanics requires unitary, and hence invertible, time evolution.

Quantum lattice gases are a realization of Feynman's conjecture: it is now known that quantum lattice gases can exhibit behavior quite similar to the many-body behavior described by the Schrödinger equation of nonrelativistic quantum mechanics [63, 64, **?**] or superfluid Helium II [76]. It appears to be a most interesting problem to quantify the similarity between the behaviors of the "artificial quantum fluid" and real quantum.

2 Viscous Multiphase Fluids

The analysis of a system of many particles is applied at three separate scales or physical regimes. The *microscopic scale* deals with the motions of the individual particles in the system. At this level using the metaphor that permeates this the lattice-gas subject, we may take everything to be discrete. A particle is a discrete microscopic object possessing a scalar mass, m, that moves through space with vector momentum, $m\vec{v}$. At this scale, consider space as being divided up into a collection of coordinatized volume elements. That is, each volume element of space has a unique coordinate $\vec{x}$, which might be the coordinate of its centroid. Each volume element is of size ℓ^3 which is much larger than the size of an individual particle. Suppose all the possible directions of momenta are denumerable so that they can be counted by the integers $a = 1, \ldots, B$, where B denotes the total number of possible directions in which a particle could be

moving through space.

Next, the *mesoscopic scale* deals with the expectation value of microscopic quantities obtained by averaging over an appropriate mesoscopic statistical ensemble. For example, the statistical mechanics of Boltzmann applies at the mesoscopic scale. Let $f_a(\vec{x}, t)$ denote the probability of finding a particle at time t moving in direction $\hat{e}_a$ with speed v_a located within the volume element with coordinate $\vec{x}$. $f_a(\vec{x}, t)$ is the particle distribution function and it obeys the Boltzmann equation, in Boltzmann's mesoscopic model.

Finally, the *macroscopic scale* deals with emergent hydrodynamic behavior of the system. At this scale, one characterizes the dynamical behavior of the system by partial differential equations of motion, one for each additive conserved quantity of the system (*viz.* mass, momentum, and energy). At the macroscopic scale the relevant quantities are continuous and the system behaves like a fluid. Define at the *macroscopic scale* two real valued quantities: the *mass density*, $\rho(\vec{x}, t)$, proportional to the number of particles at time t in the volume element centered at position coordinate $\vec{x}$

$$\rho(\vec{x}, t) = m \sum_{a=1}^{B} f_a(\vec{x}, t), \tag{4}$$

and the *momentum density*, $\rho(\vec{x}, t)\vec{v}(\vec{x}, t)$, proportional to the total momentum at time t in the volume element centered at position coordinate $\vec{x}$

$$\rho(\vec{x}, t)\vec{v}(\vec{x}, t) = \sum_{i=a}^{B} m v_a \hat{e}_a f_a(\vec{x}, t). \tag{5}$$

A quantity used to characterize fluid motion is the *characteristic length scale* of the flow, denoted L_f. It is on the order of the length of the inverse spatial gradient of the distribution function, $f_a/\partial_i f_a$, characterizing the size of the hydrodynamic fluctuation. Examples of the characteristic length scale for hydrodynamic flow is the size of one wavelength of the fluid's density field oscillation (a wavelength of sound) or the shear width of the fluid's velocity field. The mean free path length, λ, is the average distance a particle travels between collisions. The smallest possible mean free path length for a particle in a lattice-gas fluid

is on the order of the grain size of the lattice, ℓ. An important dimensionless quantity is the *Knudsen number*, denoted Kn, which is the ratio of the mean free path length to the characteristic length scale, $\lambda\partial \sim \frac{\lambda}{L_f}$. Another important dimensionless quantity is the *Mach number*, denoted M, which is the ratio of the characteristic velocity of the fluid flow to the speed of sound, $\frac{v}{c_s}$. The hydrodynamic description for a lattice-gas fluid is valid at small Knudsen numbers and small Mach numbers.

For any fluid where mass is conserved, which is the case for a lattice-gas fluid, there is a continuity equation that holds true at the macroscopic scale. To second order in the smallness this is the following

$$\partial_t \rho + \partial_i(\rho v_i) = \mathcal{O}(\varepsilon^3). \tag{6}$$

Here the shorthand notation for partial derivatives is used: $\partial_t \equiv \partial/\partial_t$ and $\partial_i \equiv \partial/\partial x_i$. Similarly, for any fluid where momentum is conserved, which is also the case for a lattice-gas fluid, there is a Navier-Stokes equation that holds true at the macroscopic scale. The Navier-Stokes equation for a viscous, incompressible fluid to second order in the smallness is the following

$$\partial_t(\rho v_i) - \partial_j(\rho v_i v_j) = -\partial_i P + \rho\nu\partial^2 v_i + \mathcal{O}(\varepsilon^3). \tag{7}$$

A lattice based procedure for deriving (6) and (7) from the Boltzmann equation for $f_a(\vec{x}, t)$ is given in Volume I. The very subtle limiting procedure is explained in the derivations.

In (7), ν is the *kinematic viscosity*, the transport coefficient for momentum diffusion. It gives a measure for the rate of decay of local shears in the fluid and determines how fast a perturbed fluid will relax from a locally anisotropic flow profile back to its isotropic steady state profile.

Furthermore in (7), P is the pressure of the fluid. In general the pressure is a function of the density and temperature, $P = P(\rho, T)$, this is termed the *equation of state*. The form of the equation of state arises from local collisional scattering and the microscopic mechanism underlying the interfluid force $-\partial_i P$ caused by nonlocal two-point momentum exchanges between particles.

The pressure has two contributing parts: it is the sum of a local ideal part and a nonlocal part

$$P = P^{\text{ideal}} + \mathcal{V}. \tag{8}$$

The ideal part of the pressure is directly proportional to the local density, $P^{\text{ideal}} = c_s(T)^2\rho$, where c_s is the sound wave speed at temperature T [72]. The quantity $\mathcal{V} = \mathcal{V}(\rho, T)$ appearing in (8) is called the *interaction energy density.* Its value depends on nonlocal interactions between different configurations of particles within the fluid system.

The interparticle force within the fluid due the interaction energy density $\mathcal{V}$ is

$$F_i = -\partial_i \mathcal{V}. \tag{9}$$

If the equation of state for a multiphase fluid has a flat region, there exists a *phase transition* between different overall organizational configurations of particles in the fluid. Fluid pressure below the ideal value arises from negative $\mathcal{V}$ which in turn is caused by interparticle binding forces between spatially separated particles. Given a strong enough interparticle binding force, the pressure can decrease with density over some limited range of densities—the slope of the equation of state curve would be negative, $\frac{\partial P}{\partial \rho} < 0$. In this situation, the fluid's compressibility would become negative and induce instability. To remain stable ($\frac{\partial P}{\partial \rho} = 0$), the fluid phase separates, for example into liquid and gas phases, and consequently the fluid's isothermal compressibility, defined by $\rho^{-1}(\partial\rho/\partial P)_T$, diverges. The response of the fluid density to small perturbations in the pressure is infinite. In other words, small perturbations cause large scale restructuring or reorganization of particles throughout the entire fluid system [62].

3 Ways to Simulate a Fluid

How precisely can one represent on a computer a physical system like the multiphase fluid system mentioned above? There are several ways to try to do such a fluid simulation on a computer. One way is to implement a "high level" numerical scheme (such as a finite-difference scheme, or spectral method, or

finite-element approach) that approximates the continuity equation (6) coupled to a nonideal Navier-Stokes equation (7). This is a valid approach only at small Knudsen numbers which means any density variation across an interfacial region must be smooth and slowing changing, or a special theory of such boundary layers must be supplied. Yet the nonideal pressure in (7) will cause the interfaces to sharpen with steep density variations. In the interfacial regions equations (6) and (7) alone cannot adequately provide a correct description of the fluid's behavior, including for example the phase separation, spinoidal decomposition, growth of drops dense fluid or bubbles of rarefied fluid form, and the coalescence of drop or bubbles driven by surface tension on thin interfaces. In the coexistence region of a liquid-gas fluid for example, the interfacial regions are usually so thin that gradient terms appearing in a partial differential equations become singular. So augmented partial differential equation are needed for an adequate high-level description.

An example of this is known as *model H* [36], a set of partial differential equations that models the behavior of the bulk flow with dynamic interfacial motions. The coupled equations are valid near the critical point where the fluid's pressure curve as a function of density has an inflection point, $\left.\frac{\partial P}{\partial \rho}\right|_{\rho=\rho_c} \sim 0$, which occurs at a particular value of the fluid density called the *critical density*. There is a Cahn-Hilliard diffusion equation for the fluid's order parameter, ψ,[4]

$$\partial_t \psi = \lambda_o \partial^2 \frac{\delta \mathcal{F}}{\delta \psi} - g_o \frac{\delta \mathcal{F}}{\delta j_i} \partial_i \psi, \tag{10}$$

where $\mathcal{F}$ is the fluid's free energy functional. This is written as

$$\mathcal{F} = \int d^D x \left[\frac{1}{2} (\nabla \psi)^2 + \mathcal{V}(\psi) + \frac{1}{2} \mathbf{j}^2 \right]. \tag{11}$$

The Cahn-Hilliard equation (10) is coupled to a Navier-Stokes equation through the momentum density, $\mathbf{j}$, appearing in the free energy functional (11). The

[4]Note that order parameter is conserved by continuity $\partial_t \psi = -\lambda_o \partial_i \mathcal{J}_i$ where the current is defined as $\mathcal{J}_i = -\partial_i (\delta \mathcal{F} / \delta \psi)$. Here the order parameter represents the quantity $\psi = \varepsilon - (\bar{\mu} + T\bar{\sigma})\rho$, where $n\varepsilon$ is the energy density, $\bar{\mu}$ is the chemical potential, and $\bar{\sigma}$ is the entropy per unit mass [36] and ψ is the grand potential density.

gradient squared term gives rise to surface tension and the interaction potential

$$\mathcal{V} \sim -\frac{\alpha}{2}\left(\frac{T-T_c}{T_c}\right)\psi^2 + \frac{\beta}{4}\psi^4 \tag{12}$$

drives the order parameter to zero or one (given suitable normalization of the parameters α and β) causing separation for $T < T_c$ where the parameters α and β are positive definite. The validity of this approach away from the critical point is uncertain.

Is there a way to model a multiphase fluid away from the critical point? One way familiar to me is to use a microscopically complete molecular dynamics approach. Before going on to discuss molecular dynamics, it is useful to be aware of a computational limitation that arises in high level partial differential formulations of physical systems: the occurrence of floating point numerical round-off error which is ubiquitous on classical computers, must be avoided in any algorithm implemented on a quantum computer to exploit the unitarity of quantum mechanical evolution.

4 Numerical Round-Off Error

The advent of digital computing in the second half of this century offers a paradigm that is distinctly granular in nature. In present day computing, discrete digital memory causes granularity in the numerical representation. Real numbers are used in most formulations of dynamical physical systems. Usually in computer simulations of dynamical physical systems, real valued quantities are represented by *floating point* numbers that have only a finite number of digits of precision. Floating point convention approximates a real number in exponential form by using two finite integers, one integer for the mantissa and the other for the exponent.[5] In modern dissipative computing floating-point errors is often quite menacing.

In computational schemes using a simple floating point protocal, the value of the most significant bits dominates the simulation's outcome under numer-

[5] The IEEE convention for 32-bit floating point uses 8-bits for exponent, 1-bit for sign, and 23-bits for mantissa. IEEE convention for 64-bit floating point uses 11-bits for exponent, 1-bit for sign, and 52-bits for mantissa.

ically stable regimes. This is a consequence of weighting the most significant bits exponentially more than the least significant bits. In floating point operations (*i.e* multiplication and division) there exist uncontrolled round-off errors in the least significant bits of the number. In unstable regimes, these small uncontrolled round-off errors in the least significant bits grow over the course of the numerical simulation and cause it to behave unphysically, even halting the simulation by overflow or underflow events. Any computational physicist is well acquainted with numerical instabilities associated with underflow or overflow in the range of some floating point variable.

The prevalence of numerical instabilities in floating point numerical schemes leads one to ask the question: Do there exist computational schemes that have no round-off error? Such computational schemes do exist and they can closely mimic the behavior of some physical systems, as we shall show.

5 Molecular Dynamics

One direct "low level" way to simulate on a computer the dynamical behavior of complex fluids is to model the actual *molecular dynamics* of all the particles in the fluid [1, 54]. The simulation of molecules undergoing interparticle interactions has the advantage of completeness in that all relevant physical processes at the kinetic scale can be captured. However the drawback of the traditional molecular dynamics approach is one of limited scale.

With the largest available supercomputers today one is capable of simulating the dynamics of hundreds of millions of particles. Yet the characteristic scale of a liquid simulation is quite small, on the order of a single micron in domain size. And with simulation volumes on the order of a cubic micron, a typical simulation run can mimic the behavior of this collection of molecules only for a short time span on the order of a few nanoseconds. Greater scales can be achieved by Monte Carlo techniques, but this gives information about equilibrium properties only and dispenses with all dynamical and nonequilibrium information.

A many-body system that behaves as a fluid is a concrete presentday example

of a physical system that can be approximated with a computational numerical scheme without any uncontrolled round-off error arising from floating point arithmetic.[6] Before describing a nonfloating point lattice gas numerical scheme for molecular dynamics, it is worthwhile to describe what molecular dynamics is in some detail.

Molecular dynamics is a large N-body problem where one successively iterates Newton's equations with a specified short-range interparticle potential. In a molecular dynamics code, the molecule either behaves like a hard sphere that bounces off other hard spheres or interacts with other particles via some continuous two-body potential, usually a Lennard-Jones 6-12 potential [1]. The Lennard-Jones potential function $u(r)$ models how individual atoms interact

$$U(r) = 4\mathcal{E}\left[\left(\frac{r_\circ}{r}\right)^{12} - \left(\frac{r_\circ}{r}\right)^{6}\right], \tag{13}$$

where r is the interatomic separtion, $\mathcal{E}$ is an energy parameter specifying the depth of the minimum of the the Lennard-Jones potential, and $r_\circ$ is a length parameter specifying the interatomic separation distance at which the Lennard-Jones potential is zero. Conceptually each molecule has a definite position within a perfectly smooth space, a continuum, and the molecule's momentum is any vector (arbitrary in both direction and magnitude).

In three dimensional simulations, there are 6N components of the position vector and momentum vector for N molecules: these components are taken to be real valued quantities. Therefore in traditional computer simulation codes, the position and momentum components are approximated by finite precision floating point numbers, and because of the limited numerical range of finite size

[6] Levesque and Verlet have recently presented a reversible molecular dynamics scheme that exactly conserves momentum [41]. Their scheme employs 60-bit integers to encode the positions of the molecules as they iterate Newton's equation of motion, which for the ith molecule is

$$\mathbf{r}_i(t+\delta t) = -\mathbf{r}_i(t-\delta t) + 2\mathbf{r}_i(t) + \frac{\delta t^2}{m} \sum_j \mathbf{f}_{ij}(t).$$

In effect they are constraining the molecules to move along a regular and periodic cubic spatial lattice. Floating point is used in the computation of the intermolecular forces, but rounded in a consistent fashion so that the sum of all forces is identically zero, $\sum_{ij} \mathbf{f}_{ij}(t) = 0$. This rounding does however induce uncontrolled errors in the total energy which in their scheme are manifest as fluctuations about the expected value.

floating point as mentioned above, the position and momentum components are consequently discretized. The "computational space" is *never* a perfectly smooth continuum.

Similarly, the interaction potential, conceptually a real valued quantity, typically is also discretized by floating-point arithmetic. Furthermore because of limitations in computational power, the interaction potential is usually cutoff beyond a short distance away from the molecule. So the floating point approximations of the real valued dynamical quantities induce representational discrepancies arising from numerical round-off. Nevertheless, if these discrepancies in the numerical simulation are kept small enough, the simulated dynamics remains stable, and a molecular dynamics code becomes a useful tool for studying many-body systems. Thus molecular dynamicists using floating point arithmetic [53] have created numerous useful physical modelling applications [2] and are continually finding new ones [1, 54].

To understand the effect of representational round-off error in traditional molecular dynamics simulation, consider a floating point number that encodes a component of the molecule's position coordinate. Flipping the least significant bit of this number represents a small displacement of that molecule along one of the axes. This distance is a kind of *grain length* since this displacement distance characterizes the scale of granularity of the computational space. Therefore, to limit the effect of this numerical discrepancy in the simulation, separation distances between molecules must be many orders of magnitude greater than the grain length. Next, consider a floating point number that encodes a component of the molecule's momentum vector. Flipping the least significant bit of this number represents a small impulse along one of the axes. Again, to limit the effect of this numerical discrepancy, the force of the interaction between molecules must cause accelerations many orders of magnitude greater than the smallest numerical momentum shift.

As a consequence of these types of numerical limitations, as a molecule traverses the distance of a mean-free path or as time elapses the duration of a mean-free time, many thousands of computational iterations are customarily

expended in a molecular dynamics code to ensure the simulation's validity. In other words, a large amount of computational power is expended each mean-free time of a molecular dynamics simulation to mitigate against round-off error and ensure overall numerical stability. Allocating computational power in this way limits the overall spatial and temporal scales achievable in the simulation.

6 Shrinking Bits

The informational extent of physical systems, such as the state of 10^{23} molecules in a cup of coffee, is quite extensive compared to the informational extent of a computer's memory, which today is currently about 10^9 bits in a desktop workstation. Furthermore, the computer's ability to process this information quickly[7] is again extremely limited when compared to the microscopic rate of change in physical systems.

This leads one to ask the following question: Will we ever have a computer big enough to capture all molecular dynamics completely, or in other words, is there a lower bound to the size of the physical embodiment of a single bit in a computation? One might expect a fundamental limit for bit densities to be the atomic densities of a solid (or liquid). The semiconductor industry is driven by its ability to pack more and more bits into a chip, typically a silicon-oxide substrate with a surface area of about a square centimeter. The data plotted in figure 1 clearly shows an exponential decrease in the areal size of a bit over the last fifty years, from 18,000 bits in the 1946 Eniac computer to about a trillion bits in today's biggest parallel supercomputer. It is clear there is an ongoing exponential reduction in bit size with its linear dimension halving approximately every 18 months. It appears a bit's size is heading towards the atom's size, and if the trend continues, DNA base pair densities (24 Angstrom feature size) will be achieved perhaps two decades from now. Computing at this small scale has been termed *nanoscale computing*.

Nanometer feature sizes (perhaps feasible around the year 2020) will allow

[7] Currently the overall bit change rate in a computational processing unit is about ten billion Hertz: around 100 bits clocked at 100 MHz.

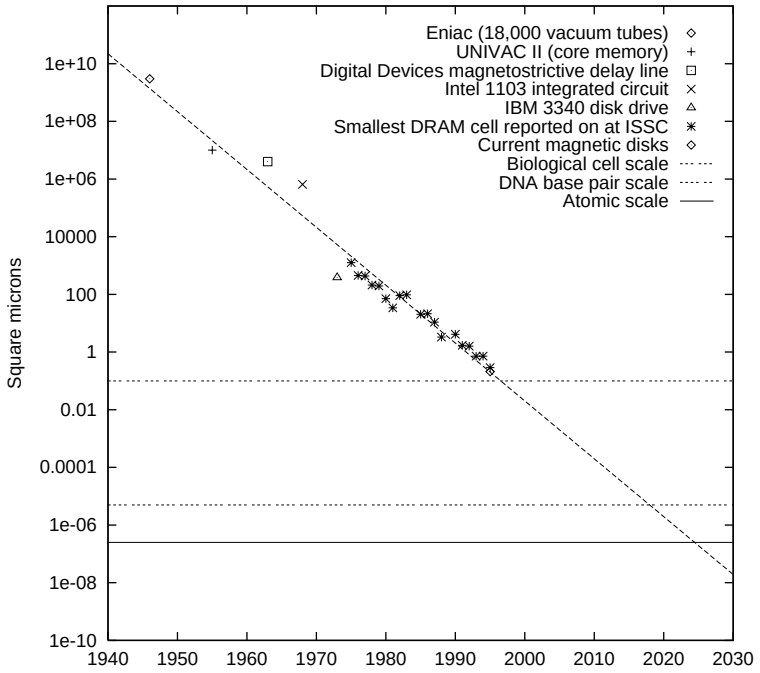

Figure 1: Exponential reduction in areal size of a bit for the last fifty years since the 1946 Eniac computer.

about 10^{16} qubits to be packed on a chip. Yet even 10^{16} qubits is still far short of Avagadro's number. So a molecular dynamicist around the year 2020 will still have insufficient computer memory to store all the microscopic positions and momenta of the molecules in a cup of coffee. So for the rest of our lives we will suffer the handicap of having far too few qubits to encode things completely.

7 Lattice Molecular Dynamics

So what can one do to simulate the turbulent flow of coffee as the milk is poured in? Clearly it is reasonable to get rid of useless detail in a computer simulation of the macroscopic behavior of a physical system. Of course almost all microscopic information is eliminated by formulating a set of governing partial differential equations that hold at macroscopic scales, yet as discussed above such high level formulations of the multiphase flow equations may not be reliable away from the critical point. We lack good model partial differential equations to capture all the relevant physical processes related to interfacial motion. Furthermore coding partial differential equations in modern computer languages engenders floating point approximations which may lead to computational instabilities.

While high level schemes are problematic, the traditional low level scheme, molecular dynamics, does not provide enough scale to capture large scale dynamical and nonequilibrium hydrodynamic behavior because much of the available computational power is used to squelch the effect of floating point round-off error. And in our lifetime it is likely that there will never be a computer quite big enough to model the required scales.

In this section we explore an alternative scheme that does not start with a macroscopic partial differential equation yet which eliminates much microscopic detail and avoids uncontrolled computational round-off error. We shall argue that it is possible to construct an artificially discrete microscopic dynamics that behaves at its macroscopic scale quite similarly to the macroscopic description of some interesting physical systems. Since computers are finite and discrete, it is desirable to consider an artificially discrete microscopic dynamics that is

maximally discretized. It is possible to capture important physical processes yet explore much larger spatial and temporal scales than molecular dynamics. This approach may be termed *lattice molecular dynamics*: it is a multienergy lattice gas method with a quite general long-range interaction. Early lattice gas methods were very limited in their energies and in their interactions.

It is known that interparticle potentials can be modeled by including a single anisotropic long-range interaction in the lattice gas dynamics for discrete momentum exchange between particles. The simplest model of this kind is the Kadanoff-Swift-Ising model [37]. A long-range interaction was used in a lattice gas scheme by Appert and Zaleski [3] in 1990 to cause an attractive force between particles giving rise to a nonthermal liquid-gas phase transition.[8] I consider a generalization of their approach by including repulsive forces between particles in addition to the attractive forces. This simple idea—using both *attractive and repulsive long-range interactions*—opens the way to many rich modeling possibilities [75, 71, 73].

The new possibility is to employ interactions transitions satisfying the principle of semi-detailed balance between multiple energy levels.

In the multienergy long-range lattice gas, the interaction energy density that arises from particular transitions between configurations of particles at locations x and x' is proportional to the product of two probabilities, $\mathcal{V} \propto \sum_{\langle xx' \rangle} \psi_{\text{emit}}(x')\psi_{\text{absorb}}(x)$, where the quantities ψ_{emit} and ψ_{absorb} are *emission* and *absorption* probabilities for momentum exchanges conveying an interparticle force $-\partial_i \mathcal{V}$. This approach is not without its drawbacks and limitations. The interaction range for the momentum exchange must be much smaller than any scale related to the dynamics of the interface region that exists between the two phases in order for the long-range lattice gas to simulate the correct macroscopic dynamics. For example, the interaction range must be much smaller than the radius of curvature of a drop or bubble, and much smaller than the wavelength

[8] A *nonthermal* lattice-gas is one with intensive quantities for the pressure and density, but no intensive quantity related to temperature. This is because, a nonthermal lattice-gas is one where all particles move at a single speed and where a particle's mass and momentum are uniquely defined, but its energy is not.

of a capillary wave or gravity wave travelling along the interface. I will try to point out other limitations as the formulation of the method is explained. Yet the benefit of this lattice gas approach is it achieves a larger scale than traditional molecular dynamics and is quite suitable for implementation on a fine-grained quantum computer.

8 Artificially Discrete Microworlds

Before introducing kinetic lattice gas models, it is worthwhile to first mention the most well known example of a maximally discretized "microworld," the Ising model of ferromagnets and antiferromagnets. Of all the simple models of many-body systems in physics, the Ising model is perhaps the most well studied, analytically and numerically.

The Ising model using classical up-down spins on a lattice is a simple model of ferro- and antiferromagnets or liquid-gas systems. There is a mapping between the thermodynamic variables for fluids and magnets—the order parameter $\rho - \rho_c$ is analogous to the magnetization; and, the response function, the negative compressibility, is analogous to magnetic susceptibility. Requiring only one classical bit of information per lattice site, the model captures the equilibrium thermodynamic behavior of these two phase systems in an elegant way. Figure 2 illustrates the behavior of the order parameter and magnetic susceptibility near the critical point for the onset of the order-disorder phase transition. These two quantities are plotted versus temperature centered about the critical temperature, $T_c = 2/\log(1+\sqrt{2}) = 2.2692$. The Ising Hamiltonian is

$$H = -J \sum_{\langle ij \rangle} s_i s_j,$$

where $\langle ij \rangle$ denotes the set of nearest neighbor bonds between spins, s_i and s_j, on the lattice. The energy of the spin-spin couple is modeled by J. The critical temperature, T_c, is expressed in units of the spin-spin coupling energy divided by the Boltzmann constant.

The Ising model manifests many universal properties: an order-disorder

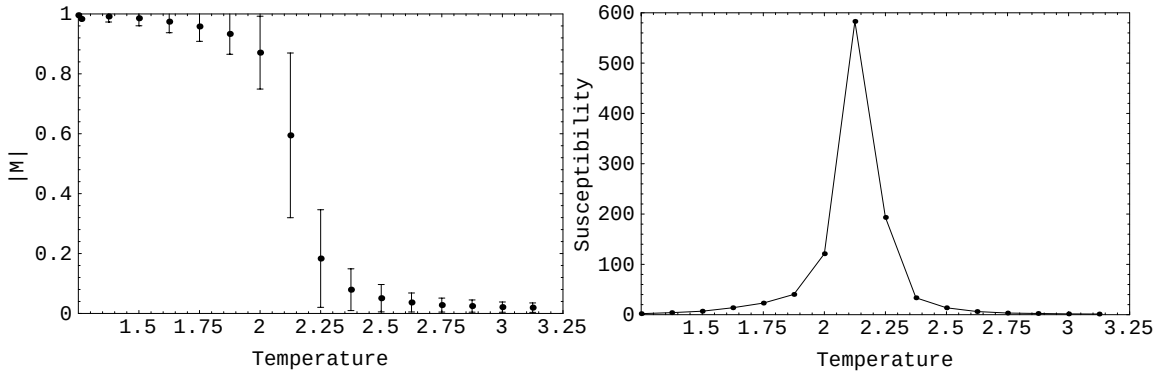

Figure 2: Total magnetization and susceptibility verses temperature is obtained on a 1024×1024 simulation using a fast Monte Carlo Metropolis algorithm with randomly chosen non-juxtaposed sites for parallel updating on the Connection Machine 2.

phase transition [11], scaling [10] that relates critical-point exponents [36], critical slowing down, and so forth. The numerical techniques applied to it appear to be endless: the Monte Carlo Metropolis algorithm, microcanonical cluster Monte Carlo[21], Ising cellular automata [70, 69, 21], deterministic heat-bath [32, 44], parallel Monte Carlo [56], multispin encoding [78], multigrid techniques [38], Monte Carlo renormalization group [39, 65], and so on. In order to capture the kinetic behavior of a liquid-gas fluid it is apropos to explore other simple models—following after the Ising model in spirit—that capture the physical kinetics of many-body systems, for example the behavior of a liquid-gas fluid undergoing the order-disorder phase transition.

9 Classical Lattice Gas

For the purpose of simplifying a classical molecular dynamics program so that it can be straightforwardly "coded" on a fine-grained quantum computer, consider a completely discrete version of things. In this simplest case, one would still like to correctly simulate the many-body system of N particles—that is, to capture all the relevant physical kinetics at the macroscopic "hydrodynamic" scale—yet one would attempt to achieve this using the most severely discretized microscopic behavior.

Each particle is assigned a definite position within a crystallographic lattice and time advances in discrete units. A particle is very restricted in the value for its momentum: it can only move along a lattice direction going from one site to a neighboring site, and so its velocity is quantized, $\vec{v} = \vec{e}c$, where $\vec{e}$ is a nearest-neighbor lattice vector and $c = \ell/\tau$ is the ratio of the lattice cell size to the size of the time-step. A classical particle occupies a point of the lattice, it resides at a single site at a given time. The information needed to encode a particle's existence is a single classical bit associated with that site. If the bit is *on*, the particle is there. If the bit is *off*, the particle is not there.

How many particles should be allowed to reside at one site of the lattice at any particular time? A minimalist would require that the maximum number of

particles that can simultaneously reside at a single lattice site should equal the maximum number of distinct momenta one is willing to keep track of. In the simplest scheme, there can be no more than one particle in each distinct local state.

As a particle in state α at some lattice site of the crystallographic space "hops" into an unoccupied state β at the same or a neighboring site, a digital bit is moved from α and into β. Data permutations between sites correspond to spatial translations and endow the bits with "kinetic energy." Data permutations at a particular site correspond to collisional interactions between bits. The critical computational work is placed in the collisional interactions, since it is there that the "decisions" are made as to whether or not a set of bits should collide and if so how.

10 Lattice-Gas Paradigm

This computational picture of lattice-gas dynamics is related to finite-difference methods for solving partial differential equations [67, 19]. But the lattice gas methodology embodies values beyond the practicality of finite-difference schemes for several reasons.

Two practical attributes of the lattice-gas paradigm are efficient massively parallel fine-grained processing and modeling of complex physical systems with stability properties different from those of other models. As another practical matter, most computational fluid dynamics codes are complicated and intricate in their approximations, whereas lattice gases are a quite simple conceptual expression of Navier-Stokes fluid flow.

To quote Frisch, lattice gases possess "bit democracy" with all bits having equal weight (an exception to this is the integer lattice gas). Bit democracy usually is not as efficient as the bit weighting found in standard floating point numerical methods. In a simple lattice gas, only a single bit is used to represent a particle, whereas in molecular dynamics a few hundred bits are used (six floating-point numbers for position and momentum). At small scales, molecular

dynamics is the appropriate modelling tool. But at large hydrodynamic scales and for quite complex fluids, lattice gases outstrip molecular dynamics and becomes the modelling tool of choice (provided no competing high level schemes are known—for complex fluids this is often the case [13]).

Lattice gases possess the theoretically attractive attributes of inherent simplicity and universality. Just as simple models in statistical mechanics, such as the Ising model mentioned above, shed light on equilibrium thermodynamics and equilibrium critical phenomena, so too do lattice gas constructs shed light on kinetics and dynamical phenomena [75]. Moreover, its inherent simplicity gives the lattice gas pedagogical value since many properties of macroscopic systems can be predicted through analysis of simple local microscopic properties. For example, the classical lattice gas construct provides a simple way to comprehend certain properties of fluid systems, such as the dependence of the shear viscosity on particle collision rates.

Lattice gases simulate physical systems while keeping mass, momentum, and energy exactly conserved. Exact modeling is valuable, particularly in cases where multiparticle correlations are essential to the system's behavior. Lattice gas simulations have verified theoretical predictions beyond the Boltzmann mean-field approximation of uncorrelated collisions: the phenomenon of long-time tails in the velocity autocorrelation function [2, 52, 26] has recently been observed in lattice gases [40, 16, 17].

Like their cellular automata cousins, lattice gases are local. The combination of simplicity and locality of lattice gas rules allows—in principle—nearly ideal logic density. Earlier in the introduction I tried to extrapolate what would be the highest logic density that one would expect two decades from now: qubits packed at nanometer scales. Because of the locality of lattice gas algorithms, there is the prospect of lattice gas architectures operating at such a high informational density.

Figure 3: MIT Laboratory for Computer Science cellular automata machine CAM-8. This 8 module prototype can evolve a D-dimensional cellular space with 32 million sites where each site has 16 bits of data with a site update rate of 200 million per second. The communication network is a Cartesian three-dimensional mesh. Crystallographic lattice geometries can be directly embedded into the CAM-8.

11 The CAM-8 Prototype

To better understand the lattice-gas paradigm as a possible computing architecture, a prototype machine has been constructed, called the cellular automata machine CAM-8 and is shown in figure 3. The CAM-8 architecture [48, 47] is the latest in a line of cellular automata machines developed by the Information Mechanics Group at MIT [66, 69, 49]. The CAM-8 architecture itself is a simple digital electronic realization of the lattice gas scheme, and in the early 1990's was tested against other parallel supercomputers and is optimal at performing lattice gas simulations [77]. Lattice gas data streaming and collisions are directly implemented in the architecture.

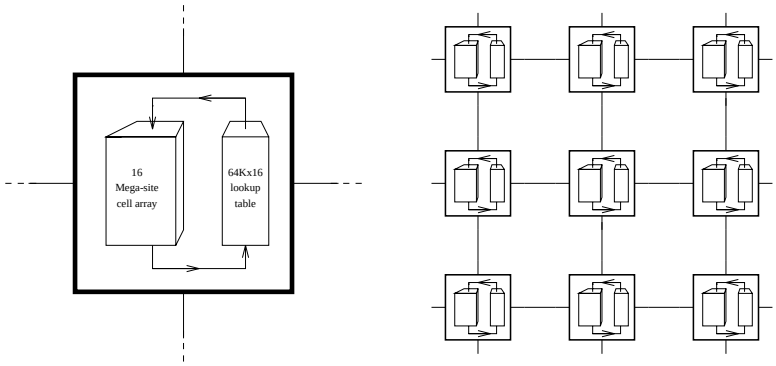

Figure 4: CAM-8 system diagram. (a) A single processing node, with DRAM site data flowing through an SRAM lookup table and back into DRAM. (b) Spatial array of CAM-8 nodes, with nearest-neighbor (mesh) interconnect (1 wire/bit-slice in each direction.

One can think of the discrete memory space within the CAM-8 as an artificial microworld. The discrete microworld paradigm with simple local rules governing the system's evolution made it quite straightforward to construct the fine-grained parallel CAM-8 out of elementary "chunks".[9] Figure 4 is a schematic diagram of a CAM-8 system. On the left is a single hardware module—the elementary "chunk" of the architecture. On the right is an indefinitely extendable array of modules (in the CAM-8 prototype the array is actually three-dimensional). A uniform spatial calculation is divided up evenly among these modules, with each module containing a volume of 16 million lattice sites. These sites are scanned in a sequential pipelined fashion. In the diagram, the solid lines between modules indicate a local *mesh* interconnection. These wires are used for spatial data movements.[10]

[9] An expanded machine, called the CAM-8-64 [68], in 1994 was designed to incorporate a billion sites using the standard CAM-8 module. A new design using RAMBus memory chips and field programmable gate arrays has recently been completed by Margolus and is two orders of magnitude faster than the CAM-8.

[10] There is also a tree network (not shown) connecting all modules to the front-end host, a SPARC workstation with a custom SBus interface card, that controls the CAM-8. It downloads a bit-mapped pattern as the initial condition for the simulations. It also sends a "step-list" to the CAM-8 to specify the sequence of kicks and scans that evolve the lattice gas in time. One can view the lattice gas simulation in real-time since a custom video module captures site data for display on a VGA monitor, a useful feature for lattice gas algorithm development, test and evaluation. The CAM-8 has built-in 25-bit event counters allowing real-time measurements without slowing the lattice gas evolution. This feature is used to do real-time coarse-grain block averaging of the occupation variables and to compute the components of

The CAM-8 uses custom VLSI chips to control data movement and commodity dynamic random access memory (DRAM) to store its state data. Each site of the lattice has 16 bits (or a multiple thereof). A 16-bit lattice site is also referred to as a *cell*. Each bit of a cell is part of an entire bit plane of the lattice, which is stored in a single DRAM chip. Therefore, a bit plane can be "translated" through the lattice arbitrarily by off-setting the pointer to the zeroth memory address in the DRAM chip. The translation vectors for the bit planes are termed *kicks*. The specification of the x,y, and z components of the kicks for each bit plane (or hyperplane) defines the lattice geometry. The kicks can be changed during the simulation. Thus, the data movement in the CAM-8 is quite general. Once the kicks are specified, the coding of the lattice gas streaming is completed. In effect, the kicks determine all the global permutations of the data.

The CAM-8 runs through its discrete dynamics with absolutely no round-off error so that in a lattice gas simulation all additive conserved quantities are kept exactly fixed throughout the course of a simulation. Its processors are ultimately simple, each able to act on only a small number of bits of information at a time. This is sufficient for a classical lattice gas algorithm that only permutes bits, never creating or destroying bits of information, just shuffling them about. Permutations achieve particle conserving reversible dynamics and are used in all classical lattice gas implementations on the CAM-8. Local permutations of data occur within the cells. These permutations are the computational metaphor for physical collisions between particles. The CAM-8 uses commodity static random access memory (SRAM) to store all the local state transitions, or local rules.[11]

the momentum vectors for each block. The amount of coarse-grained data is sufficiently small to be transferred back to the front-end host for graphical display as an evolving flow field within an X-window.

[11] All local permutations are implemented in look-up tables. All possible physical events with a certain input configuration and a certain output configuration are precomputed and stored in SRAM, for fast table look-up. The width of the CAM-8 look-up tables is 16-bits, or 64K entries. This is a reasonable width satisfying the opposing considerations of model complexity versus memory size limitations for the SRAM. Site permutations of data wider than 16-bits must be implemented in several successive table look-up passes. Since the look-up tables are double buffered, a scan of the space can be performed while a new look-up table is loaded for the next scan.

Within a CAM-8 classical lattice gas simulation, all information is exactly preserved in time, and as a consequence of this fact, at any point in a simulation one can decide to reverse the computation: the state of the artificial microworld evolves back to its initial state. Therefore the dynamics has a time-reversal invariance, or in other words, the algorithm is logically invertible. Because of algorithm reversibility, CAM-8 lattice gas simulations are unconditionally stable, since all transitions from any state to a new state occurs if the new state is a legitimate state, that is, one with the same number of particles and the same momentum. Unconditional stability in a numerical treatment is a highly valuable and desirable characteristic. Yet the CAM-8, which is a classical computer, is it not limited to performing only unitary operations on its 16-bit cell (*i.e.* permutations), it can do general mappings which are irreversible. Therefore, the CAM-8 dissipates heat like any conventional computer even though the Szilard entropy of the lattice gas is unchanged, but an optimal lattice gas computer would dissipate no heat as it processed through its simulation.

12 Limitations and Drawbacks of Classical Lattice Gases

In a classical lattice gas Boolean bits represent particles. In lattice gas machines such as the CAM-8 discussed in the previous section, *reversible computational dynamics* has been observed to give rise to hydrodynamic-like behavior quite similar to viscous and multiphase fluid motion observed in nature. Numerical measurements taken from classical lattice gas simulations are generally in excellent agreement with mean-field theory predictions [34, 3, 71] and, in the rare instance when this is not the case, with more exact field theoretic calculations [40, 17, 12]. Yet it is important to stress that in many cases classical lattice gases can behave in bizarre and clearly unphysical ways, albeit usually far away from equilibrium where theoretical constraints on the dynamics are violated.[12] In

[12] These bizarre behaviors are not signs of instabilities, but indicate that far away from equilibrium artifacts caused by the discreteness of the microscopic dynamics can arise at the macroscopic scale.

well understood regimes where all the necessary constraints are met (*i.e.* flows with small Knudsen numbers where the mean-free-path is much smaller than the characteristic scale of hydrodynamic gradients and where the flow speed is much slower than sound), usually the available effective spatial and temporal resolution within the fine-grained computer severely limits the usefulness of the simulation [51]. This is because lattice gas simulation is a form of classical molecular dynamics as explained above and we will always have a "shortage" of bits to simulate large-scale things including the finest details.

In a lattice gas at every time step as a bit hops a single lattice length, it undergoes a collision. So a mean-free-time elapses and a mean-free-path is traversed at every computational iteration by every particle. Although this is orders of magnitude more efficient than a classical molecular dynamics simulation where many iterations are expended per mean-free-time or mean-free-path, the classical lattice gas is like its classical molecular dynamics counterpart in that the available number of particles per computer simulation is still far too few (on the order of billions) in comparison with the vast numbers of particles in any natural situation (on the order of Avagadro's number) that it is trying to represent. So it is not surprising that classical lattice gases fail to adequately capture the fine details within large scale hydrodynamics motions, namely turbulence.

As well as limited by spatial and temporal resolution, classical lattice gases are plagued with noisy fluctuations [22] (these are somewhat related issues). Although these fluctuations have some positive advantages—for example they are akin to random fluctuation in many physical processes and an important mechanism whereby the lattice gas explores different metastable states [55]—, these fluctuations also have the negative aspect of effectively reducing the simulation's macroscopic scale. Assuming the dynamics is ergotic, to remove the noise in any measurement, one must either increase the spatial size of the lattice to allow for more coarse-grain averaging or else one must increase the number of sample runs with different initial conditions, a means of ensemble averaging. In either case, significant computational resources must be expended to remove noisy fluctuations instead of expending these resources on increasing

the simulation's size. This particular drawback of classical lattice gases has so far limited its application to solely academic uses.

Some possibilities have been explored to try to avoid the noisy fluctuations in classical lattice gas simulations. The lattice Boltzmann gas, a generalization of the classical lattice gas replacing Boolean bits with real floating point numbers, models the particle kinetics directly at the mesoscopic level [19]. There are fluctuations in the Boltzmann gas also, but they have smaller sizes. Although widely used nowadays and already applied to many important numerical applications [58, 59, 60, 57], the lattice Boltzmann method suffers from numerical instabilities typically encountered in finite-difference methods. The reason for this comes about by the method's lack of detailed balance, or even its lack of the weaker condition of semi-detailed balance, in its BGK collision operator.[13] Since it is essentially a first-order finite-difference method [50], the lattice Boltzmann method is not competitive with state-of-the-art and higher-order computational fluid dynamics methods, employing multiscaling or curvilinear adaptive meshing for example. Therefore, the conventional lattice Boltzmann method is not a satisfactory alternative to standard high-level numerical schemes.

The integer lattice gas, another generalization of the classical lattice gas, replaces each single Boolean bit with an integer [15]. The integer lattice gas still models the particle kinetics at the microscopic level, but has exponentially more local configurations available per lattice site than the classical lattice gas. This significantly reduces noise in the simulation. As a serendipitous benefit the integer lattice gases also possesses Galilean invariance for some particle densities whereas the classical lattice gas does not. Moreover, since the integer lattice gas retains detailed balance in its local collisions as well as computational reversibility, it is a model amenable to all the statistical mechanics one can muster.

[13] Mass and momentum are only conserved to within the precision of the floating-point representation. If the value of the single-particle distribution function at some site is close to either zero or one, it is possible owning to numerical round-off errors that the distribution function will become either negative or greater than one. When either of these situations arise, the latttice Boltzmann simulation will immediately become unstable and the values of the distribution function will diverge until an numerical underflow or overflow event occurs. Usually the BGK collision operator becomes unstable in a region with a high density gradient, for example at an interfacial boundary, or in region a with a high velocity shear.

Yet its drawbacks are two-fold. Firstly, even in the infinite integer limit, the transport coefficient for the kinematic viscosity remains very high, only slightly lower than its single bit counterpart. Therefore, there is no significant increase in the method's computational efficiency over the single bit case. Secondly, it is extremely costly to implement a collision operator for large integers. So in practice, it takes much effort to realize the theoretical advantages offered by an integer lattice gas, but as demonstrated in Volume I, integer lattice gases do work. Their intermediate statistics interpolates between Fermi-Dirac and Bose-Einstein statistics.

13 Classical Lattice Gases on Quantum Computers

Let us now return to the line of speculation begun earlier in the introduction regarding a quantum computer comprised of an array of qubits packed at near atomic densities—on the order of a nanometer separating adjacent qubits where quantum mechanical interactions such as dipole-dipole coupling between qubit pairs are exploited for computation. All operations are necessarily local involving only nearest neighbors of qubits within the computer's crystallographic lattice. An important architectural issue is defining a reasonable strategy for "processing" a collection of qubits to achieve useful *computational dynamics*.

I have already discussed at some length the simulation of classical lattice gas systems on classical computers such as the CAM-8. It is worthwhile to consider which quantum systems can be simulated on a classical computer–I'll discuss this shortly–and furthermore it is worthwhile to consider what practicality a quantum computer offers for classical simulations. The simplest classical lattice molecular dynamics simulation on a fine-grained quantum computer would be one for a Navier-Stokes classical fluid. So let us focus on this.

In the quantum lattice gas presented here many system configurations (here the term "system configurations" indicates single points in the $6N$-dimensional phasespace) can be simultaneously encoded in the computer's lattice of qubits

because of the possibility of superposition of quantum states (here the term "states" indicates a quantum wavefunction formed over the direct product manifold of N qubits). Quantum parallelism may be used to simultaneously encode many microscopic phasespace points, an ensemble of states of a classical N-particle system characterized by particular additive conserved quantities occurring in the model physical system. So in a single time step iteration an entire region of phasespace corresponding to a particular macroscopic situation can all be simultaneously evolved.[14] In a quantum lattice gas this is possible because of quantum superposition in the microscopic dynamics.[15] Therefore any measurement taken from the numerical quantum simulation, presumably at a coarse-grain level, then gives a value of the dynamical quantity.

As implied in the beginning of the introduction, it appears impractical to build a quantum computer with so many qubits where quantum coherence is mantained. This is indeed the case. Nevertheless, it is worthwhile to ask the following question: Can any useful computation be accomplished on a fine-grained quantum computer that does not require global coherence of the system wavefunction? The most surprising characteristic of the lattice gas simulation on a quantum computer (called a *quantum lattice gas with controlled decoherence*) is that it may give rise to Navier-Stokes fluid dynamics at the macroscopic scale without the need for global coherence of the system wavefunction. The basics of quantum lattice gas theory and the Navier-Stokes quantum lattice gas are treated in Volume III.

[14] In a fined-grained quantum computer with a million qubits say, the number of phasespace points that could be simultaneously evolved is amazingly large, $\sim 2^{1,000,000}$.

[15] In a classical lattice gas, Boolean commutation relations apply instead of quantum fermionic commutation relations. This does not significantly ease any analytical calculations because Boolean commutation relations causes almost as much complications as the fermionic anticommutation relations. Yet this simplification makes the computational simulation of large systems practical. The CAM-8 is a tangible realization of this fact.

14 Quantum Lattice Gases on Classical Computers

Since no quantum computers exist today, let us consider what quantum simulations can be carried out on a classical computer. Given a system with N qubits, there are 2^N basis kets in the number representation. The number of kets in what is termed the *p-particle sector* is equal to the binomial coefficient (N choose p). This is because of the Pascal triangle identity

$$\sum_{p=0}^{N} \left(\begin{array}{c} N \\ p \end{array} \right) = 2^N. \tag{14}$$

Suppose the quantum lattice gas wavefunction is constrained to reside in the 1-particle sector. The number of basis kets in this sub-space of the Hilbert manifold identically equals the number of qubits since (N choose 1) $= N$. That is, in the 1-particle sector of the quantum Hilbert space, there are N amplitudes, each a complex number. So the 1-particle sector of an N-qubit quantum computer can be represented on a classical computer with N complex numbers. While a classsical computer can only simulate the one-body problem using N complex amplitudes, a quantum computer in principle can simulate the full N-body problem using N qubits because of the exponential size of its Hilbert space. This clearly displays the advantage offered by a quantum computer. Yet even in the 1-particle sector, a fine-grained quantum computer is extremely useful since it could simulate, for example, the nonrelativistic Schrödinger equation [**?**] or any lattice molecular dynamics simulation of complex fluids discussed earlier in the introduction.

Since the Hilbert space of the quantum lattice gas grows exponentially in the number of qubits, to "fit things" into a classical computer one has only two choices: use the one-particle sector of a large lattice system or consider only simple models on small lattice clusters.[16] [17]

[16] The Boghosian-Levermore one dimensional lattice gas model for the solution of Burgers' equation is an example [14] of a useful and simple lattice gas model. With only two momentum states per site, left and right going particles, a present day classical computer could handle a quantum version of this lattice gas on a lattice with ten sites.

[17] It is possible to do a quantum Monte Carlo simulation of a Navier-Stokes quantum lattice

Using the 1-particle sector of a fined-grained quantum computer it is possible to choose a particular local and unitary evolution that gives rise to Bose condensation in the scaling limit. It is possible to construct a coupled lattice gas system, a quantum lattice gas and a classical lattice gas in mutual contact through "external" potentials. The coupled lattice gas system behaves like liquid He^4 where it is a superfluid at a finite temperature below the λ-point.

An interesting system is the Hubbard model with four spin states per site. In this case the quantum spins states are empty, up, down, and doubly occupied. A method of exactly solving the Hubbard model that reduces the Hamiltonian matrix of elements to block diagonal form with no block exceeding the size of 5×5 for the three and four site clusters. The idea is to use basic operators (*i.e.* creation and annihilation operators) to construct more complex operators that eventually aid in the problem's solutions. In the case of the Hubbard model, the total number operator, finite point group symmetry operators, the z-component of the total spin operator, and the total spin squared operator may be all constructed from the creation and annihilation operators, as can the Hamiltonian itself. Since these operators commute with the Hamiltonian, they are used to find representations where the Hamiltonian is block diagonalized.

15 Quantum Lattice Gases on Quantum Computers

Restricting the system to a small cluster size is one way in which the creation and annihilation operators are used to "exactly" simulate a multiparticle quantum system on a classical computer. It is also possible to restrict the dynamics to the 1-particle sector. But Feynman's original idea of quantum simulations on quantum computers is much more interesting and powerful. So finally, let us now consider this general computational situation. We know that for large clusters and large N-particle sectors, there is no way of solving things exactly on a classical computer. But suppose we have a quantum computer. Even a small

gas. This provides a mean for determining the expected shear viscosity of the Navier-Stokes quantum lattice gas.

quantum computer with only a few qubits, say 50, would allow us to simulate systems with clusters sizes we could not handle by any other available means.

Suppose the quantum computer's nanoscale architecture is fashioned according to the *lattice-gas paradigm* [31, 30] so that small groups of qubits are updated at a time by suitably chosen local quantum mechanical interactions—all the computational operations are strictly local as they are partitioned in both space and time. Suppose the bits in a classical lattice gas are replaced with qubits, and otherwise everything else is kept conceptually the same. The partitioning begins with an independent computation performed at each lattice site by a *collisional* unitary operator, denote $\hat{C}$. That is, $\hat{C}$ is block diagonal since it independently acts on each group of *on-site* qubits. Furthermore, within the on-site manifold, $\hat{C}$ is blocked over all the equivalence classes. Post collision interference of local configurations occurs after the application of $\hat{C}$. That is, there is the possibility of the superposition of outgoing configurations.

A *translational* permutation operator, denoted $\hat{S}$, exchanges qubits between neighboring sites of the lattice in such a fashion that every qubit translates through the crystallographic space as if it possessed a unit of momentum. In this way, a qubit encodes a particle with unit mass and unit momentum that undergoes collisional scattering with other qubits it happens to meet at any lattice site. Post streaming quantum entanglement globally occurs because $\hat{S}$ is not block diagonal in any representation in which $\hat{C}$ is block diagonal.

One computational time step is completed following a single application of the collisional unitary and translational permutation operators. So the microscopic transport equation is

$$\mid \Psi(t+\tau)\rangle = \hat{S}\hat{C} \mid \Psi(t)\rangle. \tag{15}$$

The characteristic feature of classical lattice gas computations is that all operators amount to conditional permutations of data. In classical lattice gas computers, such as the CAM-8, local permutations of site data is accomplished through a general matrix transformation in look-up table fashion and global permutation of site data is accomplished through pointer manipulations as ex-

plained above in §11. The underpinning of these kind of permutation operators, in a mathematical sense, conceptually reduces down to a single and very simple operation: *interchange*. A novel operator presented is the *interchange operator*. Using interchange operators I specify the quantum lattice gas streaming evolution operator.[18]

One may write down a 2-qubit interchange operator in terms of products of particle creation and annihilation operators. The interchange operator, $\hat{\chi}$, can be made unitary and then explicitly written in exponential form, $\hat{\chi} = \exp(-i\hat{N})$. If the evolution of the lattice gas is comprised of successive application of interchange operators over a partitioned lattice for a set of qubit pairs $\{\Gamma\}$, then a unitary streaming operator, $\hat{S}$, for the lattice gas is made by successive application of the interchange operator

$$\hat{S} = \prod_{\{\Gamma\}} \hat{\chi}_{\Gamma}. \tag{16}$$

In general, interchange operators are useful for expressing the streaming part of the unitary evolution operator of a quantum lattice gas system. It is straightforward to construct the unitary evolution streaming for a quantum lattice gas.

A very surprising is result regarding the quantum lattice gas is that the unitary collision operator, $\hat{C}$, which is block diagonalized over the equivalence classes of local configurations, can be *any* unitary matrix that mixes states within an equivalence class is sufficient. Particular choices of $\hat{C}$ optimally reduce the shear viscosity, but all are acceptable.

At this early stage in the exploration of quantum computing there does not yet exist much evidence indicating whether this is a reasonable strategy for evolving a large array of qubits [9, 43], nevertheless I explore the quantum lattice-gas paradigm because of its simplicity and because the theory and computation of classical lattice gas dynamics points the way to this new type of lattice based quantum computation.

[18] In the case of the Hubbard Hamiltonian, the interchange operators are used to exactly solve the model for very small clusters of sites.

References

[1] Farid F. Abraham. Computational statistical mechanics methodology, applications and supercomputing. *Advances in Physics*, 35(1):1–111, 1986.

[2] B.J. Alder and T.E. Wainwright. Velocity autocorrelations for hard spheres. *Physical Review Letters*, 18(23):988–990, 1967.

[3] Cécile Appert and Stéphane Zaleski. Lattice gas with a liquid-gas transition. *Physical Review Letters*, 64:1–4, 1990.

[4] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DeVincenzo, Norman H. Margolus, Peter W. Shor, Tycho Sleator, John Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457–3467, 1995.

[5] Paul Benioff. Quantum mechanical hamiltonian models of turing machines. *Journal of Statistical Physics*, 29(3):515–547, 1982.

[6] Charles H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 6:525–532, 1979.

[7] Charles H. Bennett. Thermodynamics of computation—a review. *International Journal of Theoretical Physics*, 21:219–253, 1982.

[8] Charles H. Bennett. Quantum information and computation. *Physics Today*, October:24–30, 1995.

[9] Michael Biafore. Cellular automata for nanometer-scale computation. *Physica D*, 70(3/4), 1994.

[10] Kurt Binder. Finite size scaling analysis of ising model block distribution functions. *Phys. B - Condensed Matter*, 43:119–140, 1981.

[11] Kurt Binder. Theory of first-order phase transitions. *Rep. Prog. Phys.*, 50:783–859, 1987.

[12] Bruce M. Boghosian. Correlations and renormalization in lattice gases. *Physical Review E*, 52(1), 1995. comp-gas/9403003.

[13] Bruce M. Boghosian, Peter V. Coveney, and Andrew N. Emerton. A lattice-gas model of microemulsions. *Proceedings of the Royal Society*, A(452):1221–1250, 1996.

[14] Bruce M. Boghosian and C. David Levermore. A cellular automaton for burger's equation. *Complex Systems*, 1:17–29, 1987.

[15] Bruce M. Boghosian, Jeffrey Yepez, Francis J. Alexander, and Norman H. Margolus. Integer lattice gas. *Physical Review E*, 55(4):4137–4147, 1997. comp-gas/9602001.

[16] R. Brito and M.H. Ernst. Lattice gases in slab geometries. *Physical Review A*, 44(12):8384–8687, 1991.

[17] R. Brito and M.H. Ernst. Ring kinetic theory for tagged-particle problems in lattice gases. *Physical Review A*, 46(2):875–887, 1992.

[18] A.R. Calderbank and Peter W. Shor. Good quantum error correcting codes exist. *LANL archive: quant-ph/9512032*, 1995.

[19] Hudong Chen, Shiyi Chen, and William H. Mattaeus. Recovery of the navier-stokes equations using a lattice-gas boltzmann method. *Physical Review A*, 45(8):R5339–R5342, 1992.

[20] David G. Cory, Amr F. Fahmy, and Timothy F. Havel. Ensemble quantum computing by nuclear magnetic resonance spectroscopy. Technical report tr-10-96, Harvard University Center for Research in Computing Technology, Aiken Computation Laboratory, 33 Oxford Street, Cambridge, MA 02138, December 1996.

[21] Michael Creutz. Microcanonical cluster monte carlo simulation. *Physical Review Letters*, 69(7):1002–1005, 1992.

[22] Jill P. Dahlburg, David Montgomery, and Gary D. Doolen. Noise and compressibility in lattice-gas fluids. *Physical Review A*, 36(5):2471–2474, 1987.

[23] David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society, London*, A400:97, 1985.

[24] David Deutsch and R. Jozsa. Rapid solutions of problems by quantum computation. *Proceedings of the Royal Society, London*, A439:553–558, 1992.

[25] Artur Ekert and Richard Jozsa. Quantum computation and shor's factoring algorithm. *Reviews of Modern Physics*, 68(3):733–753, 1996.

[26] M.H. Ernst, E.H. Hauge, and J.M.M van Leeuwen. Asymptotic time behavior of correlation functions. *Physical Review Letters*, 25(18):1254–1256, 1970.

[27] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467–488, 1982.

[28] Richard P. Feynman. Quantum mechanical computers. *Optics News*, 11(2):11–20, 1985.

[29] Edward Fredkin and Tommaso Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3/4):219–253, 1982.

[30] Uriel Frisch, Dominique d'Humières, Brosl Hasslacher, Pierre Lallemand, Yves Pomeau, and Jean-Pierre Rivet. Lattice gas hydrodynamics in two and three dimensions. *Complex Systems*, 1:649–707, 1987.

[31] Uriel Frisch, Brosl Hasslacher, and Yves Pomeau. Lattice-gas automata for the navier-stokes equation. *Physical Review Letters*, 56(14):1505–1508, 1986.

[32] B. Hede and H.J. Herrmann. Fast simulation of the ising model using cellular automata. *Journal of Physics A: Math. Gen.*, 24:L691–L697, 1991.

[33] Detlef Heitmann and Jorg P. Kotthaus. The spectroscopy of quantum dot arrays. *Physics Today*, 46(6):56–63, 1993.

[34] Michel Hénon. Viscosity of a lattice gas. In Gary D. Doolean, editor, *Lattice Gas Methods for Partial Differential Equations*, pages 179–207. Santa Fe Institute, Addison-Wesley Publishing Company, 1990.

[35] Anthony J.G. Hey and Robin W. Allen, editors. *Feynman Lectures on Computation*. The Advanced Book Program. Addison-Wesley Publishing Company, Inc., 1996.

[36] P.C. Hohenberg and B.I. Halperin. Theory of dynamic critical phenomena. *Reviews of Modern Physics*, 49(3):435–479, 1977.

[37] Leo P. Kadanoff and Jack Swift. Transport coefficients near the critical point: A master-equation approach. *Physical Review*, 165(1):310–322, 1967.

[38] Daniel Kandel, Eytan Domany, Dorit Ron, Achi Brandt, and Jr. Eugene Loh. Simulations without critical slowing down. *Physical Review Letters*, 60(16):1591–1594, 1988.

[39] Shang keng Ma. Renormalization group by monte carlo methods. *Physical Review Letters*, 37(8):461–464, 1976.

[40] T.R. Kirkpatrick and M.H. Ernst. Kinetic theory for lattice-gas cellular automata. *Physical Review A*, 44(12):8051–8061, 1991.

[41] D. Levesque and L. Verlet. Molecular dynamics and time reversibility. *Journal of Statistical Physics*, 72(3/4):519–537, 1993.

[42] Seth Lloyd. A technologically feasible quantum computer. Complex System Group T-13, Los Alamos National Laborarory, Preprint.

[43] Seth Lloyd. Quantum-mechanical computers. *Physics Today*, Oct:140–145, 1995.

[44] Allan B. MacIssac, D.L. Hunter, M.J. Corsten, and Naeem Jan. Determinism and thermodynamics: Ising cellular automata. *Physical Review A*, 43(6):3190–3193, 1991.

[45] Norman Margolus. Quantum computation. In Daniel Greenberger, editor, *New Techniques and Ideas in Quantum Measurement Theory*, pages 487–497. Annals of the New York Academy of Sciences, 1985. Vol. 480.

[46] Norman Margolus. Parallel quantum computation. In W.H. Zurek, editor, *Complexity, Entropy, and the Physics of Information,SFI Studies in the Sciences of Complexity, vol. VIII*, pages 273–287. Addison-Wesley, 1990.

[47] Norman Margolus. Cam-8: a computer architecture based on cellular automata. *American Mathematical Society*, 6, 1996. Fields Institute Communications.

[48] Norman Margolus and Tommaso Toffoli. Cellular automata machines. In Gary D. Doolean, editor, *Lattice Gas Methods for Partial Differential Equations*, pages 219–249. Santa Fe Institute, Addison-Wesley Publishing Company, 1990. The first 8-module CAM-8 prototype was operational in the fall of 1992.

[49] Norman Margolus, Tommaso Toffoli, and Gérand Vichniac. Cellular-automata supercomputers for fluid-dynamics modeling. *Physical Review Letters*, 56(16):1694–1696, 1986.

[50] Danial O. Martinez, William H. Matthaeus, Shiyi Chen, and Dave Montgomery. Comparison of spectral method and lattice bolzmann simulations of two-dimensional hydrodynamics. *Physics of Fluids*, 6(3):1285–1298, 1994.

[51] Steven A. Orszag and Victor Yakhot. Reynolds number scaling of cellular-automaton hydrodynamics. *Physical Review Letters*, 56(16):1691–1693, 1986.

[52] Yves Pomeau. A new kinetic theory for a dense classical gas. *Physics Letters*, 27A(9):601–602, 1968.

[53] A. Rahman. Correlations in the motion of atoms in liquid argon. *Physical Review*, 136(2A):405–410, 1964.

[54] D.C. Rapaport. Large-scale molecular dynamics simulation using vector and parallel computers. *Computer Physics Reports*, 9:3–53, 1988.

[55] Daniel H. Rothman and Stéphane Zaleski. Lattice-gas models of phase separation: interfaces, phase transitions, and multiphase flow. *Reviews of Modern Physics*, 1994.

[56] W. Schleier, G. Besold, and K. Heinz. Overcoming artificial spatial correlations in simulations of superstructure domain growth with parallel monte carlo algorithms. *Journal of Statistical Physics*, 66(3/4):1101–1122, 1992.

[57] Xiaowen Shan. Simulation of rayleigh-benard convection using lattice boltzmann method. *Physical Review E*, 55(3):2780–2788, 1997. Presented in 1996 at Sixth International Conference on Discrete Fluid Mechanics, Boston University, Massachusetts.

[58] Xiaowen Shan and Hudong Chen. Lattice boltzmann model for simulating flows with multiple phases and components. *Physical Review E*, 47(3):1815–1819, 1993.

[59] Xiaowen Shan and Hudong Chen. Simulation of nonideal gases and liquid-gas phase transitions by the lattice boltzmann equation. *Physical Review E*, 49(4):2941–2948, 1994.

[60] Xiaowen Shan and Gary Doolen. Multi-component lattice-boltzmann model with interparticle interaction. *Journal of Statistical Physics*, 81:379, 1995.

[61] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. Santa Fe, NM, IEEE Computer Society Press, 1994.

[62] H. Eugene Stanley. *Introduction to Phase Transitions and Critical Phenomena*. International series of monographs on physics. Oxford University Press, 1971.

[63] Sauro Succi. Lattice boltzmann equation for quantum mechanics. *Physica D*, 69:327–332, 1993.

[64] Sauro Succi. Numerical solution of the schrodinger equation using discrete kinetic theory. *Unpublished*, 1996.

[65] Robert H. Swendsen. Monte carlo renormalization group. *Physical Review Letters*, 42(14):859–861, 1979.

[66] Tommaso Toffoli. Cam: A high-performance cellular-automaton machine. *Physica*, 10D:195–204, 1984. A demonstration TM-gas experiment was part of the CAMForth software distribution.

[67] Tommaso Toffoli. Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. *Physica*, 10D:117–127, 1984.

[68] Tommaso Toffoli. Fine-grained parallel supercomputer. Technical report pl-tr-95-2013, Phillips Laboratory, Contract No. F196828-94-C-0068 Step Research 26 Athens Street, Cambridge, MA 02138, November 1994.

[69] Tommaso Toffoli and Norman Margolus. *Cellular Automata Machines*. MIT Press Series in Scientific Computation. The MIT Press, 1987.

[70] Gerard Y. Vichniac. Simulating physics with cellular automata. *Physica*, 10D:96–116, 1984.

[71] Jeffrey Yepez. Lattice-gas crystallization. *Journal of Statistical Physics*, 81(1/2):255–294, 1994.

[72] Jeffrey Yepez. Lattice gas dynamics: Volume 1 viscous fluids. Technical report pl-tr-96-2122(i), Air Force Research Laboratory, AFRL/VSBE Hanscom AFB, MA 01731, November 1996.

[73] Jeffrey Yepez. Lattice gas dynamics: Volume 2 multiphase fluids. Technical report pl-tr-96-2122(ii), Air Force Research Laboratory, AFRL/VSBE Hanscom AFB, MA 01731, November 1996. Currently in draft.

[74] Jeffrey Yepez. Lattice gas dynamics: Volume 3 quantum fluids. Technical report pl-tr-96-2122(iii), Air Force Research Laboratory, AFRL/VSBE Hanscom AFB, MA 01731, November 1996. Currently in draft.

[75] Jeffrey Yepez. A lattice-gas with long-range interactions coupled to a heat bath. *American Mathematical Society*, 6:261–274, 1996. Fields Institute Communications.

[76] Jeffrey Yepez. Lattice gas for superfluid helium ii. *International Journal of Modern Physics*, C, Accepted 1996. To appear 1998. Presented at the Sixth International Conference on Discrete Models for Fluid Mechanics, Boston University Center for Computational Science, August 1996.

[77] Jeffrey Yepez, Guy P. Seeley, and George Mou. Lattice-gas automata on parallel architectures. In *Proceedings of International Symposium on Parallel Computation*, pages 522–525, 1993. Beijing, China.

[78] M.Q. Zhang. A fast vectorized multispin coding algorithm for 3d monte carlo simulations using kawasaki spin-exchange dynamics. *Journal of Statistical Physics*, 56(5/6):939–950, 1989.