Complex Systems **10** (1996) 1–21

When Can Solitons Compute?

Mariusz H. Jakubowski
Ken Steiglitz
Deptartment of Computer Science,
Princeton University, Princeton, NJ, USA

Richard K. Squier
Deptartment of Computer Science,
Georgetown University, Washington, DC, USA

Abstract. The possibility of using soliton interactions in a one-dimensional bulk medium is explored as a basis for a new kind of computer. Such a structure is "gateless," that is, all computations are determined by an input stream of solitons. Intuitively, the key requirement for accomplishing this is that soliton collisions be *nonoblivious*; that is, solitons should transfer state information during collisions. All the well-known systems described by *integrable* partial differential equations (PDEs) such as the Korteweg–de Vries, sine-Gordon, cubic nonlinear Schrödinger, and perhaps *all* integrable systems, are oblivious when displacement or phase is used as state. A cellular automaton (CA) model is presented, the *oblivious soliton machine* (OSM), that captures the interaction of solitons in systems described by such integrable PDEs. We then prove that OSMs with either quiescent or periodic backgrounds can only do computation that requires time at most cubic in the input size; and thus, are far from being computation-universal. Next, a more general class of CA is defined, *soliton machines* (SMs), which describe systems with more complex interactions. It is shown that an SM with a *quiescent* background can have at least the computational power of a finite-tape Turing machine, whereas an SM with a *periodic* background can be universal. The search for useful nonintegrable (and nonoblivious) systems is challenging: We must rely on numerical solution, collisions may be at best only near-elastic, and collision elasticity and nonobliviousness may be antagonistic qualities. As a step in this direction, it is shown that the logarithmically nonlinear Schrödinger equation (log-NLS) supports quasi-solitons (gaussons) whose collisions are, in fact, very near-elastic and strongly nonoblivious. It is an open question whether there is a physical system that realizes a computation-universal soliton machine.

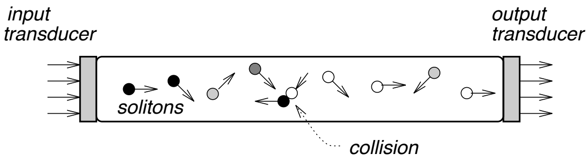

Figure 1: Computing with solitons in a bulk medium. Solitons are injected at the left of the diagram, computation takes place within the medium via the interaction of the pseudoparticles, and the results exit from the right of the diagram. The actual medium can be linear, planar, or three-dimensional.

1. Introduction

This paper is devoted to the question of whether effective computation can be performed by the interaction of solitons [23, 33] in a bulk medium. The resulting computational system would fulfill the promise of Toffoli's "programmable matter" [41], offering computation that is very close to the underlying physics, and therefore potentially providing ultrascale parallel processing.

The most immediate physical realization of such computation may be provided by solitons in an optical fiber [15, 20, 40], described by the cubic nonlinear Schrödinger equation. Other media are also possible, including Josephson junctions [34] and electrical transmission lines [21, 30], which support solitons governed by the sine-Gordon and Korteweg–de Vries equations, respectively.

We should emphasize that using optical solitons in this way is quite different from what is commonly termed "optical computing" [19, 20], which uses optical solitons to construct gates that could replace electronic gates, but which remains within the "lithographic" paradigm of laying out gates and wires. The idea presented here uses a completely homogeneous medium for computation, the entire computation is determined by an input stream of particles. A general version of the structure proposed is shown in Figure 1.

The idea of using solitons in a homogeneous medium for "gateless" computation goes back at least to [38], where solitons in a cellular automaton (CA)[1] are used to build a carry-ripple adder. A general model, the *particle machine* (PM), for computation using collisions of particles was described and studied in [36, 37]. This paper moves from the abstraction of CA to the physical realm represented by PDEs such as the nonlinear Schrödinger (NLS), Korteweg–de Vries (KdV), and sine-Gordon equations [33].

To use physical solitons for computation, we define restricted versions of the PM called *soliton machines* (SMs). Both PMs and SMs are one-

[1]These solitons arise in the mathematical framework of a CA [11, 12, 13, 29], and have an entirely different origin than the physically based solitons we consider here. However, CA-based and PDE-based solitons display remarkably similar behavior. As far as we know, the connection between CA solitons and PDE solitons is unexplained, though some authors [3, 28] have juxtaposed discussions of both systems.

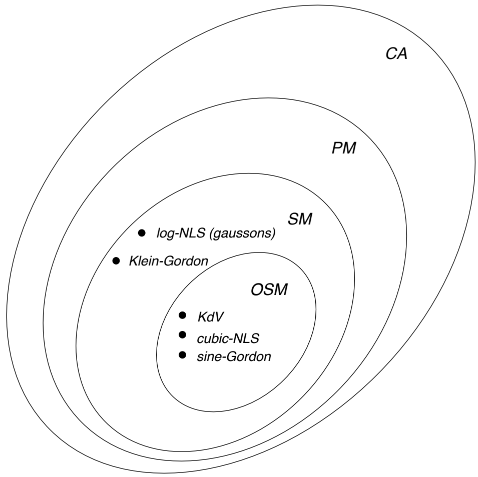

Figure 2: Hierarchy of computational systems in the world of CA. PMs are CA designed to model particle-supporting physical media. SMs are restricted PMs that model general soliton systems, including PDEs such as the Klein–Gordon and log-NLS equations. OSMs are SMs that model integrable soliton systems, such as the KdV, cubic-NLS, and sine-Gordon equations.

dimensional CA that model motion and collision of particles in a uniform medium. *Oblivious soliton machines* (OSMs) are SMs further restricted to model a class of integrable soliton systems. The hierarchy of the computational systems we consider is shown in Figure 2. In general, we abstract a physical system by modeling it first with PDEs, and then with CA, namely PMs and SMs, as shown in Figure 3.

We will discuss the computational power of the ideal machines we use to model physical systems. Being able to simulate a Turing machine, or another universal model, is neither necessary nor sufficient for being able to perform useful computation. For example, certain PMs can perform some very practical regular numerical computations, such as digital filtering, quite efficiently, and yet such PMs are not necessarily universal [36, 37]. Conversely, simulating a Turing machine is a very cumbersome and inefficient way to compute, and any practical application of physical phenomena to computing would require a more flexible computational environment. Nevertheless, universality serves as a guide to the inherent power of a particular machine model.

2. Particle machines

The PM model of computation, introduced and shown to be universal in [36], is an abstract framework for computing with particles. The PM is a general

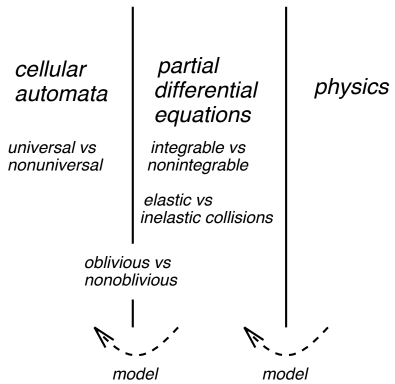

Figure 3: The three worlds considered in this paper. Notice that the property *oblivious* applies to both CA and soliton solutions of PDEs, whereas the properties *integrable* and *having elastic collisions* apply only to soliton systems.

model, not based on any specific physical system, but which tries to capture the properties of physical particles and particle-like phenomena.

Definition 1. A PM is a CA with an update rule designed to support the propagation and collision of logical *particles* in a one-dimensional homogeneous medium. Each particle has a distinct identity, which includes the particle's velocity. We think of each cell's state in a PM as a *binary occupancy vector*, in which each bit represents the presence or absence of one of n particle types. The state of cell i at time $t+1$ is determined by the states of cells in the *neighborhood* of cell i, where the neighborhood includes the $2r+1$ cells within a distance, or *radius*, r of cell i, including cell i. In a PM, the radius is equal to the maximum velocity of any particle, plus the maximum displacement that any particle can undergo during collision.

The one-dimensional medium of a PM supports particles propagating with constant velocities. Two or more particles may collide; a set of *collision rules* specifies which particles are created, which are destroyed, and which are unaffected in collisions. A PM begins with a finite *initial configuration* of particles and evolves in discrete time steps.

A PM, like a CA, can have a *periodic background*; that is, an infinite, periodic sequence of nonzero state values in the medium of the CA. Periodic backgrounds are sometimes used to add computational power to CA, as in [5]. To make this theoretical abstraction physically realizable in a PM, we can choose a specific cell to be the *terminus* of the PM, or logical end, located away from the region in which computation occurs. We can then inject a regular, periodic sequence of particles at this cell, thus simulating a periodic background.

PMs capture and abstract the behavior of particles in systems that may be used for computation. SMs, which are restricted PMs that we define later,

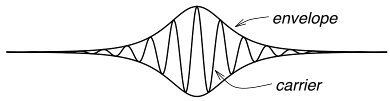

Figure 4: An envelope soliton.

bring the abstraction a step closer to physical reality by modeling systems governed by certain well-known PDEs. We now describe a class of these PDEs and systems.

3. Integrable soliton systems

3.1 Basics of solitons

Certain integrable[2] nonlinear PDEs give rise to *solitons*, or particle-like solitary waves that propagate without decay in homogeneous media and survive collisions with shape and velocity intact. Systems such as the Korteweg–de Vries (KdV), sine-Gordon (sG) and cubic nonlinear Schrödinger (cubic-NLS) equations describe the motion and interaction of solitons in shallow water, electrical transmission lines, optic fibers, and other materials [8, 33]. In recent years much effort has been expended on analyzing the properties of solitons for purposes such as high-speed communications and optical computing gates [19, 20, 40]. We examine issues involved in using solitons to implement SMs.

Nonintegrable systems also support soliton-like waves, whose more complex behavior we describe later. The integrable soliton-supporting equations that we consider in this section have exact soliton solutions, which may be obtained by the inverse scattering transform [3]. Nonintegrable equations, and integrable equations with arbitrary initial conditions, must in general be solved numerically.

Later we prove that a certain class of integrable PDEs can do only limited computation using SMs, we conjecture that this is true of all such equations. The simple behavior of integrable soliton systems makes them unlikely candidates for useful computing media.

3.2 Features of integrable solitons

Solitons arising from real-valued integrable PDEs, such as the KdV and sG equations, are uniquely identified by their constant velocities, which are determined by their amplitudes. Complex-valued integrable PDEs, such as the cubic-NLS equation, support *envelope* solitons, or wave packets consisting of *carrier waves* modulated by their surrounding *envelopes* (see Figure 4). In the cubic-NLS system, a carrier wave is characterized by an amplitude,

[2] The term *integrable*, referring to PDEs, is not used with perfect consistency throughout the literature. Here we use *integrable* to mean *solvable by the inverse scattering transform* [3].

a frequency, and a *phase* (a periodic value that changes according to the wave's *phase velocity*, a function of the wave's amplitude). An envelope soliton travels at its *group velocity*, which is determined by its amplitude and by the frequency of its carrier wave. We henceforth use the term *soliton* to refer to envelope solitons, since nonenvelope solitons are pulses without carrier waves or phases, and thus display simpler behavior less useful for our purposes.

The particle-like properties of solitons make them potentially suitable for implementing PM-like models. Solitons do not decay with time, and collide *elastically*;[3] that is, they retain their identities after collisions, undergoing only a phase shift (change) and a displacement in space. Both phase shift and spatial displacement are simple functions of the amplitudes and frequencies of the colliding solitons' carrier waves. Arbitrary numbers of solitons may collide simultaneously, but the cumulative phase shift and displacement of each soliton are obtained by summing the shifts and displacements that result from the soliton's pairwise collisions with all others; that is, phase shifts and displacements are additive. These phase shifts and displacements can be calculated easily, since exact multi-soliton solutions of well-known integrable equations can be obtained [16, 17, 18].

3.3 Breather solitons

Integrable PDEs such as the cubic-NLS and sG equations [4, 18, 32] support varied kinds of solitons, including *bound-state* solitons, or *breathers*, which consist of two or more equal-velocity solitons moving close together in perpetual collision. In the cubic-NLS equation, two equal-velocity solitons attract each other with a force that weakens exponentially as the distance between the solitons. These solitons form a breather by alternately colliding and separating, with the time between successive collisions, or *oscillation period* of the breather, increasing exponentially as the initial separation between the solitons. According to [40], "the interaction [between equal-velocity solitons] can effectively be avoided if the solitons are widely separated."

Our models ignore attraction between equal-velocity solitons. We leave it for future work to determine exactly what happens in collisions of breathers with solitons and with other breathers. However, preliminary numerical experiments suggest that such collisions result only in additive phase shifts and displacements of all colliding solitons. If this is true in general, then all results in this paper hold also for soliton systems in which breathers are allowed.

---

[3]In this paper, we use the term *elastic* to mean *nonradiating*, or *conserving total energy*. Solitons supported by integrable PDEs retain their respective energies after collisions, but later we show nonintegrable systems with (evidently) elastic collisions in which energy is exchanged between colliding solitons.

4. Oblivious soliton machines

The PM model is a convenient abstraction for computing with solitons. In practice, however, the soliton systems we have described are not suitable for implementing general PMs. Specifically, these integrable systems do not support the creation of new solitons or the destruction of existing solitons, and soliton state changes due to collisions are not arbitrary. Thus, we adopt as our model a restricted PM called an *oblivious soliton machine* (OSM). Like a PM, an OSM is a CA designed to support particles propagating through a homogeneous medium, but an OSM more closely models the integrable soliton systems under consideration.

4.1 The oblivious soliton machine model

An OSM is a PM in which each particle has a constant identity and a variable state that are both vectors of real numbers. The velocity of a particle is part of its identity. A typical state may consist of a *phase* and a *position relative to a galilean frame of reference*, whereas a typical identity may include an *amplitude* in addition to a velocity. No particles can be created or destroyed in collisions, and the identities of particles are preserved, much like the constant amplitudes and velocities of colliding solitons. A function of the *identities* (not states) of the colliding particles determines particle state changes.

Immediately after a collision, particles are *displaced*, much like the colliding solitons discussed earlier. Let P_{slow} and P_{fast} denote two particles such that the velocity of P_{slow} is (algebraically) less than the velocity of P_{fast}; that is, if V_{slow} and V_{fast} are signed integers representing the velocities of P_{slow} and P_{fast}, respectively, then $V_{\text{slow}} < V_{\text{fast}}$. In a two-way collision of P_{slow} and P_{fast}, P_{fast} is displaced by a positive integer amount, and P_{slow} by a negative integer amount. In collisions involving three or more particles, displacements are such that the relative order of the particles after the collision is the reverse of their order before the collision, and all particles are displaced into separate cells. Displacement amounts are functions of the identities of the colliding particles. In addition, we require that once two particles collide, the same particles can never collide again; this can be accomplished by spacing the particles properly, or by choosing particle velocities and displacements appropriately. This scheme models particle interaction in the integrable soliton systems described earlier.

Definition 2. An OSM consists of the following elements.

- A two-way infinite one-dimensional *medium* M.
- A finite set P of *particles*, each with one of a finite set of *velocities*.
- A finite set S of real-number vectors called *particle states*.
- A *collision function* C.
- A *post-collision displacement function* D.
- A finite *initial configuration* or *input* I.

M contains discrete *cells*, each of which can hold from 0 to $|P|$ particles. At most one particle of a given identity can occupy a cell. Each particle travels at a constant velocity and has a variable state that may change as a result of collisions with other particles in the medium. The initial configuration I is a finite section of M, and includes the *input* particles present at the beginning of the OSM computation. Up to $|P|$ particles can occupy each cell of I. The *input size* is defined as the length of I plus the number of input particles in I. During collisions, no particles are created or destroyed, and the identities of particles remain constant, but the states of particles change according to the function C. For each possible pair of identities of colliding particles, C specifies two new particle states. After two or more particles collide, they are displaced by amounts determined by the function D, as described previously. Any pair of particles can collide at most once. The machine begins with its initial configuration on either a quiescent or a periodic background, and evolves in discrete time steps.

4.2 Oblivious solition machines are not universal

We refer to OSMs as *oblivious* because the state changes in an OSM do not depend on the variable states of colliding particles, but only on their constant identities. *Oblivious* collisions in the OSM model correspond to *elastic* collisions in the integrable PDEs discussed here; however, it is an open question to the authors whether or not all elastic soliton collisions in all integrable systems are oblivious. The spatial displacements of OSM particles after collisions occur only in the constrained fashion described previously. The result of these properties is that OSMs cannot compute universally.

Theorem 1. *OSMs are not computation-universal, either with or without a periodic background. The maximum time that an OSM can spend performing useful computation is cubic in the size of the input.*

Proof. We show that the halting problem for OSMs is decidable. More specifically, we calculate a cubic upper bound (in terms of input size) on the amount of time taken by an OSM to do any computation.

To execute any algorithm using an OSM, we must encode the algorithm and its input as a finite sequence of particles in a finite-length initial configuration (input) I of an infinite homogeneous medium. We must also be able to decode the OSM state when the results of the algorithm are ready. Let N denote the number of particles in I, not counting the particles in a possibly periodic background (PB), and L the length of I. The input size $|I|$ of the OSM is then $N + L$.

We first examine the case of an OSM with a quiescent background. For such an OSM we can calculate upper bounds on the maximum number of particle collisions, and on the maximum time before each collision occurs. The product of these two values will give an upper bound on the maximum time that the OSM can spend performing useful computation.

- An upper bound on the number of particle collisions is $\binom{N}{2}$, since each particle can collide at most once with any other particle, by definition.
- An upper bound on the time before the collision of any two particles that do collide is

$$\frac{L + N|D_b| + N|D_f|}{|v_f| - |v_s|} \tag{1}$$

where D_b and D_f are the largest negative (backward) and positive (forward) displacements possible among the input particles, and $|v_f|$ and $|v_s|$ are the largest and smallest speeds, respectively, among the input particles.

Thus, an upper bound on the time that an OSM can perform useful computation is

$$\binom{N}{2} \frac{L + N|D_b| + N|D_f|}{|v_f| - |v_s|}. \tag{2}$$

Since $|I| = N + L$, $|v_f|$ and $|v_s|$ are nonnegative integers, and D_b and D_f are constants in a particular OSM, expression (2) is $O(|I|^3)$; that is, cubic in the input size.

In an OSM with a PB (PB-OSM), the PB particles can displace the input particles both left and right at regular intervals. Note that these periodic displacements cause the velocities of the input particles to change by constant amounts, which depend on the specific periodic configuration of PB particles. Thus, we can recalculate the velocities of the input particles, using the displacements effected by the PB particles. To find an upper bound on the time taken by a PB-OSM to do useful computation, we apply a similar argument as for the quiescent background, but with the newly calculated effective velocities.

Note that collisions in a PB-OSM can occur forever, but the collisions useful for computation can occur only within the time bound that we can calculate. To see this, observe that after the time given by this bound, the input particles of a PB-OSM will stay in a fixed relative order, unable to collide again with one another. Thus, each input particle either breaks away from the rest, as in the case with a quiescent background, or stays close to the others in a periodic configuration. In neither situation can the input particles do useful computation. ■

Corollary 1. *OSM-based computational systems governed by the KdV and sG equations are not universal, given that positions are used as state. OSM-based systems governed by the cubic-NLS equation are not universal, given that positions and phases are used as state.*

Conjecture 1. *All integrable systems using any choice of state are nonuniversal using the OSM model.*

5. Soliton machines

Intuitively, OSMs cannot compute universally because particles in an OSM do not transfer enough state information during collisions. We can make a simple modification to the OSM model so that universal computation becomes possible: We make the results of collisions depend on both the identities and *states* of colliding particles. In addition, we allow particle identities to change. We call the resulting model a *soliton machine* (SM). In the final section of this paper, we describe nonintegrable equations that support soliton-like waves that we believe may be capable of realizing the SM model.

5.1 The soliton machine model

Definition 3. A SM is defined in the same way as an OSM, with the following differences. Both the collision and displacement functions can depend on the identities and states of particles. The identities of particles can change during collisions, so that the collision function returns both the new states and the new identities of colliding particles.

Like an OSM, an SM is also a CA and a PM (see Figure 2). The only difference between an SM and a PM is that no particles can be created or destroyed in an SM. However, we can use a periodic background of particles in special *inert* or *blank* states, and simulate creation and destruction of particles by choosing collision rules so that particles go into, and out of, these states.

5.2 Universality of soliton machines

SMs with a quiescent background have at least the computational power of Turing machines (TMs) with finite tapes, as we will prove. The question of whether such SMs are universal is open, however. Still, these SMs are more powerful than any OSM, since OSMs can only do computation that requires at most cubic time, while problems exist that require more than cubic time on bounded-tape Turing machines.

The class of algorithms that a finite-tape TM can implement depends on the specific function that bounds the size of the TM's tape; for instance, TMs with tapes of length polynomial in the input size can do any problem in $PSPACE$. Although not universal, such TMs can do almost any problem of practical significance.

Theorem 2. *SMs with a quiescent background are at least as powerful as TMs with bounded tapes.*

Proof. We describe how SMs can simulate any finite-tape TM M. Let $B(N)$ denote the function that bounds the tape size of M, given an input of size N.

We construct an SM equivalent to M as follows. For each possible state of a cell of M, including the blank state, we introduce a distinct, stationary *state* particle. The finite tape then maps directly to a length-$B(N)$ section T of the medium of the SM.

To simulate the action of the tape head for M, we introduce two *head* particles, h_f and h_b, of velocities 1 and -1, corresponding to the right and left motions of the head, respectively. Before computation, we place h_f in the cell immediately to the left of T.

The SM begins in the initial configuration described, and operates as follows. The head particle h_f moves through T, colliding with the state particles in T. We choose collision rules such that collisions between state particles and either h_f or h_b simulate the transition function of M. Thus, a collision between h_f or h_b and a state particle s can change s to another state particle; in addition, h_f can change into h_b, and vice versa. This simulates M writing onto tape and changing head direction. ■

SMs with a periodic background are universal, since we can use such SMs to simulate a TM as described previously, but with a periodic background of blank-state particles. This background maps directly to the infinite blank portion of the tape of the TM. Thus, we have proved the following theorem.

Theorem 3. *SMs with a periodic background are computation-universal.*

5.3 Discussion

Theorems 1 through 3 suggest that we should look to nonintegrable systems for solitons that may support universal computation. It is an open question whether or not there exists such a soliton system. In what follows, we describe nonintegrable equations and explain the features that could enable them to encode a universal SM. Then we describe some preliminary experiments with a particular nonintegrable PDE, the log-NLS equation.

6. Nonintegrable soliton systems

Certain nonintegrable PDEs support soliton-like waves[4] with behavior more complex than that of integrable solitons. Examples include PDEs such as the Klein–Gordon [2] and log-NLS equations [6, 7, 25]. The solitons in these systems can change their velocities, as well as their phases, upon collisions, and new solitons may be created after collisions.

Soliton collisions in nonintegrable systems may be inelastic or *near-elastic*; that is, colliding solitons can dissipate their energy by producing varying amounts of *radiation* (see Figure 8), which erodes other solitons and may eventually lead to complete decay of useful information in the system. To our knowledge, it is an open question whether or not there exists a nonintegrable system with perfectly elastic, or nonradiating, collisions. It also appears to be an open question whether or not perfect elasticity implies obliviousness in any system. A system with collisions that are both perfectly elastic and nonoblivious would offer promise for realizing the SM model using solitons.

[4]In this section we refer to such waves as *solitons*, as is often done in the literature.

The system we describe next, the log-NLS equation, has very near-elastic, nonoblivious collisions, and may support perfectly elastic, nonoblivious collisions as well.

6.1 Gaussons in the logarithmically nonlinear Schrödinger system

The log-NLS equation, which supports solitons called *gaussons*, was proposed as a nonlinear model of wave mechanics [6, 7, 27]. Gaussons are wave packets with gaussian-shaped envelopes and sinusoidal carrier waves. They are analogous to the wavefunctions of linear wave (quantum) mechanics; that is, the square of the amplitude of a gausson at a given point x can be interpreted as the probability that the particle described by the gausson is at x.

6.1.1 The logarithmically nonlinear Schrödinger equation and single-gausson solutions

To study the behavior of gaussons, we consider the following form of the log-NLS equation:

$$-i\frac{\partial u}{\partial t} = \frac{1}{2}\frac{\partial^2 u}{\partial x^2} + b\ln(a|u|^2)u. \tag{3}$$

Here x is space, t is time, $u(x,t)$ is the complex amplitude of the wave described by the equation, and a and b are constants, with $a = e\sqrt{\pi/2b}$. This equation is nonintegrable, and analytical solutions that describe the motion and collision of multiple gaussons are not known. Single-gausson solutions, however, can be found. We use the following form of these solutions:

$$u(x,t) = \left(\sqrt{\frac{\pi}{2b}}\right)^{-\frac{1}{2}} e^{-\frac{iv^2t}{2}+ivx-b(x-vt)^2+i\phi_0}. \tag{4}$$

Here v is velocity, and ϕ_0 is initial phase. In solving equation (3) numerically to observe the movement and collisions of multiple gaussons, we plot two or more of these single-gausson solutions on a discrete spatial grid at time 0. We then apply a numerical method to obtain the state of the grid at successive time steps, plotting the results on a space-time graph, as in Figures 5 through 12.

6.1.2 Numerical experiments

Our numerical simulations of gausson collisions verify results in [27] in that they range from deeply inelastic to near-elastic, and perhaps perfectly elastic, depending on the velocities of the colliding gaussons. In [27] an approximate range of velocities (the *resonance region*) is given for which collisions are apparently inelastic; outside this region, collisions are reportedly elastic. We

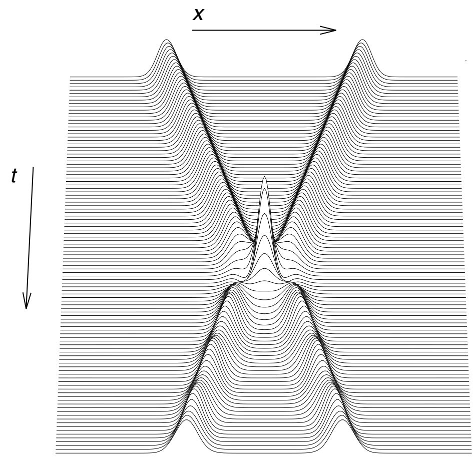

Figure 5: Gausson collisions in region 1. From left to right, velocities are 0.4 and -0.4; phases are both 0.

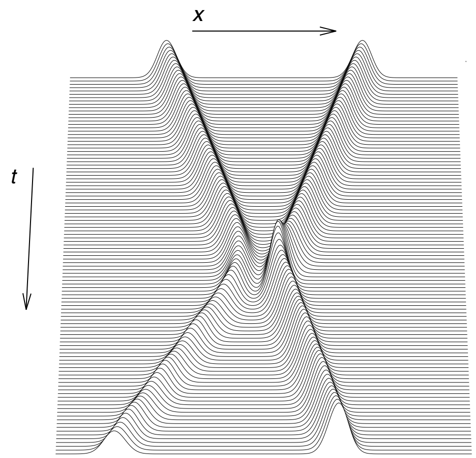

Figure 6: Gausson collisions in region 1. From left to right, velocities are 0.4 and -0.4; phases are 0 and 0.5π.

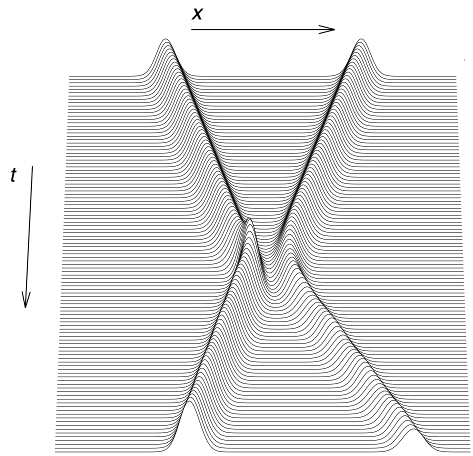

Figure 7: Gausson collisions in region 1. From left to right, velocities are 0.4 and -0.4; phases are 0.5π and 0.

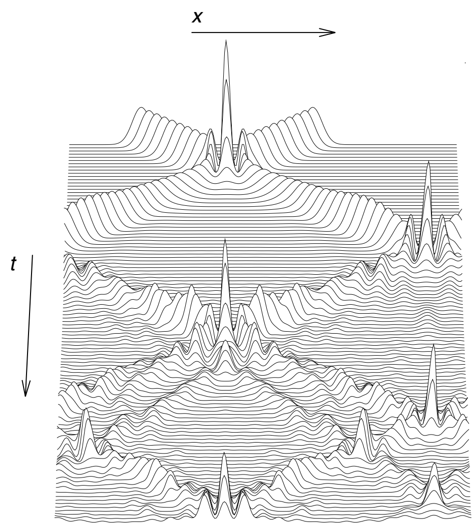

Figure 8: Gausson collisions in region 2. From left to right, velocities are 3.0 and -3.0; phases are both 0. A cylindrical coordinate system is used here, so that there is wrap around from the right to the left edge, and vice versa.

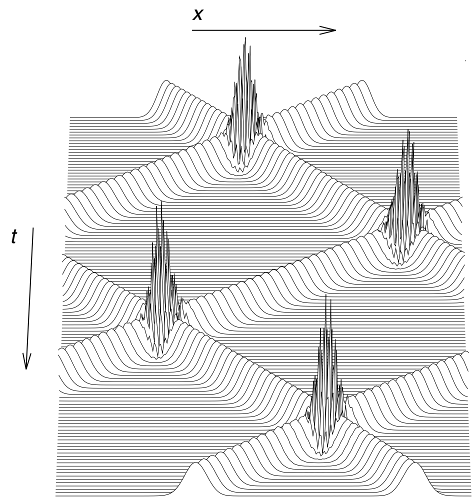

Figure 9: Gausson collisions in region 3. From left to right, velocities are 10.0 and −15.0; phases are both 0. A cylindrical coordinate system is used here, so that gaussons wrap around from right to left, and vice versa.

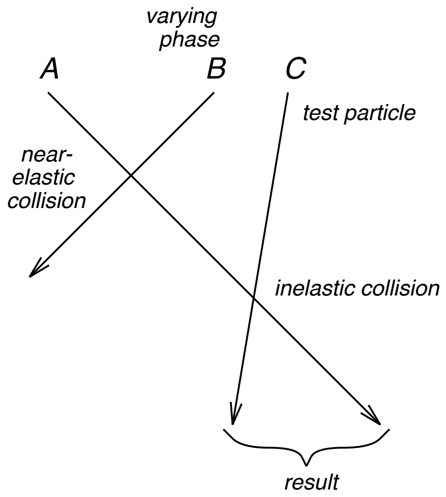

Figure 10: Testing for nonobliviousness using a near-elastic collision (AB) followed by an inelastic collision (AC). If the result of the AC collision depends on the initial phase of B, then the AB collision is nonoblivious.

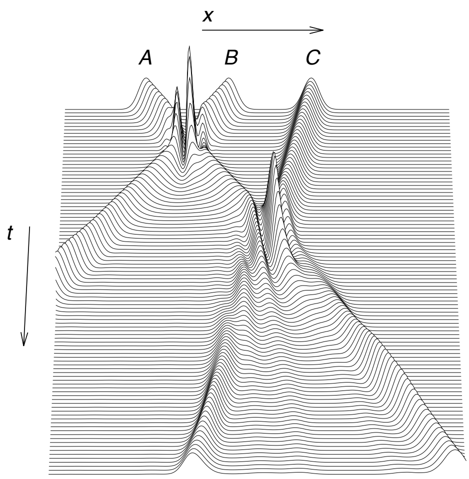

Figure 11: Testing for obliviousness of the collision between the leftmost (A) and center (B) gaussons in region 2. The center gausson's phase is 0.05π; the phase of the other two gaussons is 0. The collision is nonoblivious, since the results of the test collision between the leftmost and rightmost gaussons (A and C) differ from those in Figure 12.

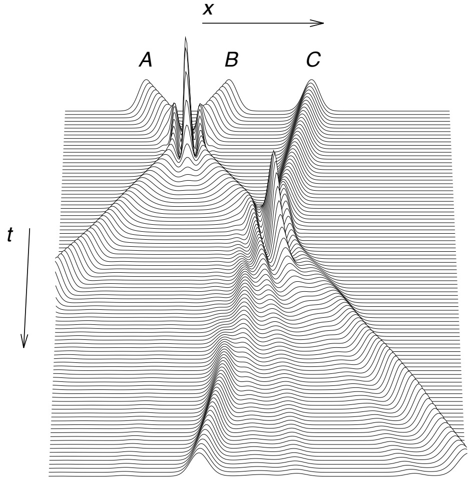

Figure 12: Testing for obliviousness of the collision between the leftmost (A) and center (B) gaussons in region 2. The center gausson's phase is 0.55π; the phase of the other two gaussons is 0. The collision is nonoblivious, since the results of the test collision between the leftmost and rightmost gaussons (A and C) differ from those in Figure 11.

confirmed these results, and investigated in more detail to find the following three distinct velocity regions in which gaussons behave very differently.

1. When $0 < |v| < 0.5$, gausson collisions are near-elastic, and possibly perfectly elastic, and clearly nonoblivious. We observed marked post-collision changes in both amplitude and velocity, which strongly depend on the phases of the colliding gaussons. These phenomena appear to be newly observed here. (See Figures 5, 6, and 7.)[5]

2. When $0.5 \leq |v| < 10$, collisions are nonelastic, and possibly near-elastic. The amount of radiation generated in collisions varies with the phases of the colliding gaussons, and in general decreases as v increases. (See Figures 8, 11, and 12.)

3. When $|v| \geq 10$, collisions are near-elastic, and possibly elastic, but apparently oblivious. They are similar to the collisions found in integrable soliton systems, such as the cubic-NLS equation. (See Figure 9.)

These ranges are approximate, and gausson behavior changes gradually from one to the next. The differences between our results and those in [27] (written circa 1978) are likely due to our more extensive numerical experimentation, given the faster computers available now. In our calculations we used the split Fourier method [9, 39] with a cylindrical (wrap-around) one-dimensional coordinate system; the authors of [27] used a finite difference scheme [14], which we also implemented, and which confirms the results of the split Fourier method. However, this finite difference method's treatment of boundary conditions makes a cylindrical coordinate system difficult to use.

Nonobliviousness in region 2 is not as easily determined by visual inspection as it is in region 1. Next, we describe a numerical experiment which demonstrates that there is a near-elastic gausson collision that is nonoblivious in region 2. Figure 10 shows the setup of this experiment. To show that the collision between two gaussons, A and B, is nonoblivious, we begin at time 0 with three gaussons, A, B, and C, in that order, on the x-axis. The velocities and initial distances among the three gaussons are set so that A and B collide first, followed by a collision of A and C. We observe the results of the AC collision for various initial phases of the carrier for B, keeping constant the initial phases of A and C. If we find two initial phases for B that lead to two different results of the AC collision, then we can conclude that the AB collisions were nonoblivious. Note that we require only that the results of the AC collisions be *different*; the AC collisions can be strongly inelastic, for the C particle is used only to probe the state of the A particle.

Figures 11 and 12 show an example of such an experiment. In both figures, the two leftmost gaussons move at a velocity of ± 3.25, the rightmost gausson has velocity -1, and all but the center gaussons have initial phase 0. The center gaussons in Figures 11 and 12 have phases 0.05π and 0.55π,

[5]All gausson figures are graphs of space versus time, with time increasing from top to bottom. The variable graphed is $|u|^2$, that is, the square of the gausson envelope.

respectively, which cause the different results after collisions. We conclude that the collision between A and B is nonoblivious. Note that we cannot determine nonobliviousness merely from the visual appearance of the AB collision, because what happens *during* a collision can depend on the phases of the colliding solitons, whether the collision is oblivious or not; it is the *post*-collision results that determine nonobliviousness.

6.2 Soliton stability and elasticity

The inelastic and near-elastic soliton collisions we observed in regions 1 and 2 are nonoblivious, thus leaving open the possibility of using them for computation in SMs. For example, to compute using gaussons, we can use an approach similar to the techniques in [38]. As with the CA solitons in [38], we might first create a database of pairwise collisions of gaussons by running a series of numerical experiments; we would then search the database for useful collisions to encode a specific computation. This approach was used in [38] to implement a solitonic ripple-carry adder.

One problem with such an approach is the potential connection between soliton stability and collision elasticity. We observed that inelastic collisions often resulted in *radiation ripples* emanating from collisions (Figure 8) and eventual disintegration of gaussons in a cylindrical one-dimensional system. In region 2, these ripples and the resulting instability may make the system unsuitable for sustained computation. The more inelastic the collisions, the quicker the system decayed. However, we do not know if stability and elasticity are necessarily correlated in general, nor do we know if elasticity and obliviousness (and thus lack of computation universality) are related. In fact, collisions of region-1 solitons in the log-NLS equation appear to be both elastic and strongly nonoblivious.

7. Summary and questions

We have explored certain well-known soliton systems, with the goal of using them for computation in a one-dimensional homogeneous bulk medium. We defined soliton machines (SMs) to model integrable and nonintegrable soliton systems, and found that a class of integrable PDEs cannot support universal computation under the OSM model. In addition, we proved that the SM model is universal in general, and suggested that gaussons in the log-NLS equation may be capable of realizing universal SMs.

Many open problems remain. Foremost among these is determining whether or not gaussons have behavior sufficiently complex and stable to implement a universal SM. We found three velocity regions in which gaussons have different behavior. Gaussons with low velocities (region 1) offer the most promise for realizing useful computation, since their collisions appear both elastic and nonoblivious. We may be able to use the phase-coding approach in [38] to implement useful computation with gausson interactions. Collisions of gaussons with higher velocities (regions 2 and 3) appear in general to be either oblivious or radiating, though for some combinations of velocities and phases, these collisions are nonoblivious and very near-elastic. The search for answers is complicated by the necessity of numerical solution of the log-NLS equation.

Even if we were to show that the log-NLS equation can be used for universal computation, we would still be left with a gap: We know of no physical realization of this equation. But other nonintegrable nonlinear PDEs also offer possibilities for implementing SMs, and many of these do correspond to real physical systems. For example, the Klein–Gordon equation [2], the NLS equation with additional terms to model optical fiber loss and dispersion, and the coupled NLS equation for birefringent optical fibers [20, 40] all support soliton collisions with complex behavior potentially useful for encoding SMs. Optical solitons that arise from these more complicated equations exhibit gausson-like behavior, and are easily realizable in physical fibers; thus, such optical solitons may be particularly useful as practical means of computing using SMs. *Near-integrable* equations [24], or slightly altered versions of integrable equations, could also offer possibilities for implementing general SMs.

In addition, we may consider using solitons in two or three dimensions [10, 31]. Gaussons, for example, exist in any number of dimensions, and display behavior similar to that in one dimension. The added degrees of freedom of movement in two or more dimensions may enable implementation of universal systems such as the billiard ball computation model [26] or lattice gas models [35].

References

[1] M. J. Ablowitz and P. A. Clarkson, *Solitons, Nonlinear Evolution Equations, and Inverse Scattering* (Cambridge University Press, Cambridge, UK, 1991).

[2] M. J. Ablowitz, M. D. Kruskal, and J. F. Ladik, "Solitary Wave Collisions," *SIAM Journal of Applied Mathematics*, **36** (1979) 429–437.

[3] M. J. Ablowitz and H. Segur, *Solitons and the Inverse Scattering Transform* (SIAM, Philadelphia, PA, 1981).

[4] S. A. Akhmanov, V. A. Vysloukh, and A. S. Chirkin, *Optics of Femtosecond Laser Pulses* (American Institute of Physics, New York, 1992).

[5] J. Albert and K. Culik II, "A simple Universal Cellular Automaton and Its One-way and Totalistic Version," *Complex Systems*, **1** (1987) 1–16.

[6] I. Białynicki-Birula and J. Mycielski, "Nonlinear Wave Mechanics," *Annals of Physics*, **100** (1976) 62–93.

[7] I. Białynicki-Birula and J. Mycielski, "Gaussons: Solitons of the Logarithmic Schrödinger Equation," *Physica Scripta*, **20** (1979) 539–544.

[8] P. G. Drazin and R. S. Johnson, *Solitons: An Introduction* (Cambridge University Press, Cambridge, UK, 1989).

[9] J. A. Fleck, J. R. Morris, and M. D. Feit, "Time-dependent Propagation of High Energy Laser Beams through the Atmosphere," *Applied Physics*, **10** (1976) 129–160.

[10] N. C. Freeman, "Soliton Interaction in Two Dimensions," *Advances in Applied Mechanics*, **20** (1980) 1–37.

[11] A. S. Fokas, E. Papadopoulou, and Y. Saridakis, "Particles in Soliton Cellular Automata," *Complex Systems*, **3** (1989) 615–633.

[12] A. S. Fokas, E. Papadopoulou, and Y. Saridakis, "Coherent Structures in Cellular Automata," *Physics Letters*, **147A** (1990) 369–379.

[13] C. H. Goldberg, "Parity Filter Automata," *Complex Systems*, **2** (1988) 91–141.

[14] A. Goldberg, H. M. Schey, and J. L. Schwartz, "Computer-Generated Motion Pictures of One-dimensional Quantum-mechanical Transmission and Reflection Phenomena," *American Journal of Physics*, **35** (1967) 177–187.

[15] H. A. Haus, "Optical Fiber Solitons, their Properties and Uses," *Proceedings of the IEEE*, **81** (1993) 970–983.

[16] R. Hirota, "Exact Solution of the Korteweg–de Vries Equation for Multiple Collisions of Solitons," *Physical Review Letters*, **27** (1971) 1192.

[17] R. Hirota, "Exact Solution of the Modified Korteweg–de Vries Equation for Multiple Collisions of Solitons," *Journal of the Physical Society of Japan*, **33** (1972) 1456–1458.

[18] R. Hirota, "Exact Solution of the Sine–Gordon Equation for Multiple Collisions of Solitons," *Journal of the Physical Society of Japan*, **33** (1972) 1459–1463.

[19] M. N. Islam and C. E. Soccolich, "Billiard-ball Soliton Interaction Gates," *Optics Letters*, **16** (1991) 1490–1492.

[20] M. N. Islam, *Ultrafast Fiber Switching Devices and Systems* (Cambridge University Press, Cambridge, UK, 1992).

[21] N. Islam, J.P. Singh, and K. Steiglitz, "Soliton Phase Shifts in a Dissipative Lattice," *Journal of Applied Physics*, **62** (1987) 689–693.

[22] K. Lindgren and M. G. Nordahl, "Universal Computation in Simple One-dimensional Cellular Automata," *Complex Systems*, **4** (1990) 299–318.

[23] P. S. Lomdahl, "What Is a Soliton?" *Los Alamos Science*, **10** (1984) 27–31.

[24] V. G. Makhankov, "Computer and Solitons," *Physica Scripta*, **20** (1979) 558–562.

[25] V. G. Makhankov, *Soliton Phenomenology* (Kluwer Academic Publishers, Norwell, MA, 1990).

[26] N. Margolus, "Physics-like Models of Computation," *Physica*, **10D** (1984) 81–95.

[27] J. Oficjalski and I. Białynicki-Birula, "Collisions of Gaussons," *Acta Physica Polonica*, **B9** (1978) 759–775.

[28] P. J. Olver and D. H. Sattinger (editors), *Solitons in Physics, Mathematics, and Nonlinear Optics* (Springer-Verlag, New York, NY, 1990).

[29] J. K. Park, K. Steiglitz, and W. P. Thurston, "Soliton-like Behavior in Automata," *Physica*, **19D** (1986) 423–432.

[30] G. E. Peterson, "Electrical Transmission Lines as Models for Soliton Propagation in Materials: Elementary Aspects of Video Solitons," *AT&T Bell Laboratories Technical Journal*, **63** (1984) 901–919.

[31] J. Satsuma, "N-Soliton Solution of the Two-dimensional Korteweg–de Vries Equation," *Journal of the Physical Society of Japan*, **40** (1976) 286–290.

[32] A. C. Scott, "The Sine–Gordon Equation," *AMS Lectures in Applied Mathematics*, **15** (1974) 211–213.

[33] A. C. Scott, F. Y. F. Chu, and D. W. McLaughlin, "The Soliton: A New Concept in Applied Science," *Proceedings of the IEEE*, **61** (1973) 1443–1483.

[34] S. S. Shen, *A Course on Nonlinear Waves* (Kluwer Academic Publishers, Norwell, MA, 1993).

[35] R. Squier and K. Steiglitz, "Two-dimensional FHP Lattice Gasses Are Computation Universal," *Complex Systems*, **7** (1993) 297–307.

[36] R. Squier and K. Steiglitz, "Programmable Parallel Arithmetic in Cellular Automata Using a Particle Model," *Complex Systems*, **8** (1994) 311–323.

[37] R. Squier, K. Steiglitz, and M. H. Jakubowski, "Implementation of Parallel Arithmetic in a Cellular Automaton," 1995 International Conference on Application Specific Array Processors, Strasbourg, France, July 24–26, 1995.

[38] K. Steiglitz, I. Kamal, and A. Watson, "Embedding Computation in One-dimensional Automata by Phase Coding Solitons," *IEEE Transactions on Computers*, **37** (1988) 138–145.

[39] T. R. Taha and M. J. Ablowitz, "Analytical and Numerical Aspects of Certain Nonlinear Evolution Equations," *Journal of Computational Physics*, **55** (1984) 192–202.

[40] J. R. Taylor (editor), *Optical Solitons: Theory and Experiment* (Cambridge University Press, Cambridge, UK, 1992).

[41] T. Toffoli, "Fine-grained Models and Massively Parallel Architectures: The Case for Programmable Matter," 1995 SIAM Conference on Parallel Processing for Scientific Computing, February 1995.