Complex Systems **1** (1987) 17–30

A Cellular Automaton for Burgers' Equation

Bruce M. Boghosian
Thinking Machines Corporation,
245 First Street, Cambridge, MA 02142-1214, USA

C. David Levermore
Lawrence Livermore National Laboratory,
University of California
Livermore, CA 94550, USA

Abstract. We study the approximation of solutions to the Burgers' equation,

$$\frac{\partial n}{\partial t} + c\frac{\partial}{\partial x}\left(n - \frac{n^2}{2}\right) = \nu\frac{\partial^2 n}{\partial x^2} \qquad (1)$$

by spatially averaging a probabilistic cellular automaton motivated by random walks on a line. The automaton consists of moving "particles" on a one-dimensional periodic lattice with speed one and in a random direction subject to the exclusion principle that at most one particle may move in a given direction from a given lattice site, at a given time. The exclusion principle gives rise to the nonlinearity in Eq. (1) and introduces correlations between the particles which must be estimated to obtain statistical bounds on the error. These bounds are obtained in two steps. The first is showing that the ensemble average of the automaton is a stable explicit finite differencing scheme of Eq. (1) over the lattice with a second order convergence in the lattice spacing. The numerical diffusion of this scheme plays an important role in relating the automaton rules to Eq. (1). The next step is showing that the spatial averaging of a single evolution of the automaton converges to the spatial averaging of the ensemble as $1/\sqrt{M}$ where M is the number of lattice sites averaged. Simulations are presented and discussed.

1. Introduction

Recently it has been proposed to use cellular automata on large lattices for obtaining solutions to partial differential equations, in particular the incompressible Navier-Stokes equations [1]. Such automata have rules with locally conserved (or nearly conserved) quantities which, when averaged

over microscopic configurations, give macroscopic behavior which is hopefully described by the PDE's. The analyses justifying such hopes have been largely formal.

In this paper we study as a model an automaton for solving the Burgers' equation which is simple enough to analyze. We are not proposing this automaton as an effective method for computing solutions to Burgers' equation, but it does allow us to study in a relatively simple context some of the same issues that arise in the application of cellular automaton techniques to solving the incompressible Navier-Stokes equations [1].

Our probabilistic cellular automaton is motivated by considering random walks on a one dimensional lattice (in this presentation, we shall restrict the discussion to one spatial dimension, although all that we do has obvious higher dimensional analogs). All "particles" on the lattice move exactly one lattice site to either the right or the left in one time step. Between two neighboring lattice sites we associate a physical distance Δx, and between two successive steps of the random walk we associate a physical time Δt. It is well-known [2] that the density of such a system of particles executing an uncorrelated, unbiased random walk obeys the diffusion equation,

$$\frac{\partial n}{\partial t} = \nu \frac{\partial^2 n}{\partial x^2} \tag{1.1}$$

where $\nu = (\Delta x)^2/2\Delta t$ is the diffusion coefficient.

Next consider an uncorrelated random walk that is biased so that the probability of a step to the right is $(1+\overline{\alpha})/2$, and the probability of a step to the left is $(1-\overline{\alpha})/2$. It is also well-known [2] that such biasing leads to linear advection in the direction of the bias. That is, the density of the system obeys

$$\frac{\partial n}{\partial t} + c\frac{\partial n}{\partial x} = \nu \frac{\partial^2 n}{\partial x^2} \tag{1.2}$$

where $c = \overline{\alpha}\Delta x/\Delta t$ is the linear advection coefficient, and $\nu = (\Delta x)^2/2\Delta t$ is the diffusion coefficient.

2. The Cellular Automaton

To model systems of the sort described in the Introduction by a cellular automaton, it is most convenient to impose the Fermi exclusion rule that no two particles occupying the same site may be moving in the same direction. That way, the state of each site is uniquely specified by two bits of information: the right bit, which is one if there is a rightward moving particle present and zero otherwise, and the left bit, which is one if there is a leftward moving particle present and zero otherwise. Thus each site has four possible states labelled by the four binary numbers from 00 to 11. Each step of the automaton has two substeps: in the first, *the collision substep*, the particles change their direction randomly at the present lattice site (either with or without bias) subject to the exclusion principle; in the

second, *the advection substep*, the particles move to the neighboring lattice site in their new direction.

This exclusion rule induces a correlation between particles that are two lattice sites apart. To see this, suppose that three successive lattice sites have states 01, 00, and 10, respectively, after the collision substep of a time step. Then, after the collision substep of the next time step, it follows that the middle lattice site *must* have state 11. Thus the evolution of one particle is affected by that of another particle two sites away, so they are correlated. We shall now show that this correlation naturally gives rise to the nonlinear term in Eq. (1).

Following the nomenclature developed in the introduction, we denote the right bit at lattice site k and time step l by $b_0(k,l)$, and the left bit at lattice site k and time step l by $b_1(k,l)$. After the collision substep of a time step, we denote the new states by $b'_0(k,l)$ and $b'_1(k,l)$. These are given by the truth table,

$b_1(k,l)$	$b_0(k,l)$	$b'_1(k,l)$	$b'_0(k,l)$
0	0	0	0
0	1	$(1-\alpha(k,l))/2$	$(1+\alpha(k,l))/2$
1	0	$(1-\alpha(k,l))/2$	$(1+\alpha(k,l))/2$
1	1	1	1

where $\alpha(k,l)$ is either 1 or -1 with mean $\overline{\alpha}$. The rule in the above table may be written in the form

$$b'_0(k,l) = \frac{1+\alpha(k,l)}{2} b_0(k,l) \vee b_1(k,l) + \frac{1-\alpha(k,l)}{2} b_0(k,l) \wedge b_1(k,l) \quad (2.1)$$

$$b'_1(k,l) = \frac{1-\alpha(k,l)}{2} b_0(k,l) \vee b_1(k,l) + \frac{1+\alpha(k,l)}{2} b_0(k,l) \wedge b_1(k,l). \quad (2.2)$$

Here, $\vee$ denotes the *inclusive or* operation, and $\wedge$ denotes the *and* operation on a pair of bits.

In the advection substep, the particles move to the neighboring lattice site in their new direction. The rule for this is easily seen to be

$$b_0(k+1,l+1) = b'_0(k,l) \quad (2.3)$$

$$b_1(k-1,l+1) = b'_1(k,l). \quad (2.4)$$

By composing the rules for the above two substeps, we arrive at the rule for one full time step of the cellular automaton

$$b_0(k+1,l+1) = \frac{1+\alpha(k,l)}{2} b_0(k,l) \vee b_1(k,l) + \frac{1-\alpha(k,l)}{2} b_0(k,l) \wedge b_1(k,l) \quad (2.5)$$

$$b_1(k-1,l+1) = \frac{1-\alpha(k,l)}{2} b_0(k,l) \vee b_1(k,l) + \frac{1+\alpha(k,l)}{2} b_0(k,l) \wedge b_1(k,l). \tag{2.6}$$

Now if b and b' are bits, there is a well-known algebraic representation for the $\vee$ and $\wedge$ operations: $b \vee b' = b + b' - bb'$ and $b \wedge b' = bb'$. Using this, the cellular automaton rule can be written in the algebraic form

$$b_0(k+1,l+1) = \frac{1+\alpha(k,l)}{2} (b_0(k,l) + b_1(k,l)) - \alpha(k,l) b_0(k,l) b_1(k,l) \tag{2.7}$$

$$b_1(k-1,l+1) = \frac{1-\alpha(k,l)}{2} (b_0(k,l) + b_1(k,l)) + \alpha(k,l) b_0(k,l) b_1(k,l). \tag{2.8}$$

Note that the nonlinear terms in Eqs. (7) and (8) owe their origin to the exclusion principle.

3. The Ensemble Average

We now turn our attention to *ensemble averages* of the automaton described in the last section; that is, we envision applying the above-described cellular automaton rule to a large set of systems, with possibly different initial conditions. For example, we might perform a large number of simulations of the automaton on a computer, using a grid of fixed size, with the initial conditions $b_i(k,0)$ chosen randomly from some known distribution. Then, $\overline{b_i}(k,l)$ denotes the value of the ith bit at position k and time step l *averaged over all the simulations.* Henceforth, we shall consistently use overbars to denote ensemble averages.

A word should be said about the random numbers, $\alpha(k,l)$. Throughout this work, we shall assume that they are generated by a "perfect" random number generator. That is, we assume that

$$\overline{\alpha(k,l)} = \overline{\alpha} \tag{3.1}$$

and

$$\overline{\alpha(k,l)\alpha(k',l')} = \delta_{kk'}\delta_{ll'} + (1 - \delta_{kk'}\delta_{ll'})\overline{\alpha}^2. \tag{3.2}$$

Then, since $b_i(k,l)$ depends on past random numbers, $\alpha(k',l')$ with $l' < l$, we can do things like

$$\overline{\alpha(k,l) b_i(k,l)} = \overline{\alpha} \cdot \overline{b_i(k,l)}, \tag{3.3}$$

etc. A study of exactly how "perfect" a random number generator has to be in order to validate our results would be interesting, but is beyond the scope of the present paper. Note that one can regard the cellular automaton defined by Eqs. (7) and (8) as a *stochastic cellular automaton* thanks to the inclusion of the random $\alpha(k,l)$'s; or, if one prefers, one can regard the random number generator as part of the rule, in which case it is a perfectly *deterministic cellular automaton.*

In our simulations, we produced the random bits using a simple cellular automaton due to Wolfram [3] that generates bits with a high degree of randomness. To get random bits from Wolfram's automaton, one applies it to a finite string of bits (we used 59 bits) with periodic boundary conditions, and samples the values at one site as a function of time. To bias the mean of the random bits one can generate more than one unbiased random bit per site and then apply logical operations to them; for example, when two strings of unbiased random bits are combined using the "and" operation the result is a string of random bits with mean 0.25, and when they are combined using the "inclusive or" operation the result is a string of random bits with mean 0.75.

We now take the ensemble average of Eqs. (7) and (8) to get

$$\overline{b}_0(k+1,l+1) = \frac{1+\overline{\alpha}}{2}\left(\overline{b}_0(k,l) + \overline{b}_1(k,l)\right) - \overline{\alpha}\cdot\overline{b_0(k,l)b_1(k,l)} \tag{3.4}$$

$$\overline{b}_1(k-1,l+1) = \frac{1-\overline{\alpha}}{2}\left(\overline{b}_0(k,l) + \overline{b}_1(k,l)\right) + \overline{\alpha}\cdot\overline{b_0(k,l)b_1(k,l)}. \tag{3.5}$$

This may be written

$$\begin{aligned}\overline{b}_0(k+1,l+1) &= \frac{1+\overline{\alpha}}{2}\left(\overline{b}_0(k,l) + \overline{b}_1(k,l)\right)\\ &\quad - \overline{\alpha}\cdot\left(\overline{b}_0(k,l)\,\overline{b}_1(k,l) + C_{01}(k,l;k,l)\right)\end{aligned} \tag{3.6}$$

$$\begin{aligned}\overline{b}_1(k-1,l+1) &= \frac{1-\overline{\alpha}}{2}\left(\overline{b}_0(k,l) + \overline{b}_1(k,l)\right)\\ &\quad + \overline{\alpha}\cdot\left(\overline{b}_0(k,l)\,\overline{b}_1(k,l) + C_{01}(k,l;k,l)\right),\end{aligned} \tag{3.7}$$

where we have defined the *covariance*

$$C_{ij}(k,l;k',l') \equiv \overline{(b_i(k,l) - \overline{b}_i(k,l))(b_j(k',l') - \overline{b}_j(k',l'))}. \tag{3.8}$$

We now begin to establish the relationship between ensemble-averaged quantities and solutions of the Burgers' equation, Eq. (1), by introducing new quantities which are more directly related to those solutions. Define

$$\overline{b} \equiv \overline{b}_0 + \overline{b}_1 \tag{3.9}$$

$$\overline{v} \equiv \left(\overline{b}_0 - \overline{b}_1\right)/\Delta x, \tag{3.10}$$

where Δx is the lattice spacing. Thus

$$\overline{b}_0 = \frac{1}{2}\left(\overline{b} + \overline{v}\Delta x\right) \tag{3.11}$$

$$\overline{b}_1 = \frac{1}{2}\left(\overline{b} - \overline{v}\Delta x\right). \tag{3.12}$$

Substituting, we can find the update rules for $\overline{b}$ and $\overline{v}$,

$$\begin{aligned}\overline{b}(k,l+1) &= \frac{1+\overline{\alpha}}{2}\overline{b}(k-1,l)+\frac{1-\overline{\alpha}}{2}\overline{b}(k+1,l)\\ &\quad+\frac{\overline{\alpha}}{4}[\overline{b}^2(k+1,l)-\overline{b}^2(k-1,l)\\ &\quad-(\Delta x)^2\left(\overline{v}^2(k+1,l)-\overline{v}^2(k-1,l)\right)]\\ &\quad+\overline{\alpha}\,[C_{01}(k+1,l;k+1,l)-C_{01}(k-1,l;k-1,l)] \qquad (3.13)\end{aligned}$$

$$\begin{aligned}\overline{v}(k,l+1) &= -\frac{\overline{b}(k+1,l)-\overline{b}(k-1,l)}{2\Delta x}+\frac{\overline{\alpha}}{\Delta x}\frac{\overline{b}(k-1,l)+\overline{b}(k+1,l)}{2}\\ &\quad+\frac{\overline{\alpha}}{4\Delta x}[-\overline{b}^2(k+1,l)-\overline{b}^2(k-1,l)\\ &\quad+(\Delta x)^2\left(\overline{v}^2(k+1,l)+\overline{v}^2(k-1,l)\right)]\\ &\quad+\frac{\overline{\alpha}}{\Delta x}\,[C_{01}(k+1,l;k+1,l)+C_{01}(k-1,l;k-1,l)] \qquad (3.14)\end{aligned}$$

Now suppose that $n(x,t)$ is the *exact* solution of Eq. (1), and define

$$w(x,t)\equiv\frac{c}{2\nu}\left(n(x,t)-\frac{1}{2}n^2(x,t)\right)-\frac{\partial}{\partial x}n(x,t). \qquad (3.15)$$

Discretize these by defining

$$\hat{n}(k,l)\equiv n(k\Delta x,l\Delta t) \qquad (3.16)$$

$$\hat{w}(k,l)\equiv w(k\Delta x,l\Delta t). \qquad (3.17)$$

Then by Taylor expanding and using the fact that $n(x,t)$ solves Eq. (1), we find

$$\begin{aligned}\hat{n}(k,l+1) &= \frac{1+\overline{\alpha}}{2}\hat{n}(k-1,l)+\frac{1-\overline{\alpha}}{2}\hat{n}(k+1,l)\\ &\quad+\frac{\overline{\alpha}}{4}[\hat{n}^2(k+1,l)-\hat{n}^2(k-1,l)\\ &\quad-(\Delta x)^2\left(\hat{w}^2(k+1,l)-\hat{w}^2(k-1,l)\right)]\\ &\quad+O\left((\Delta x)^4\right) \qquad (3.18)\end{aligned}$$

$$\begin{aligned}\hat{w}(k,l+1) &= -\frac{\hat{n}(k+1,l)-\hat{n}(k-1,l)}{2\Delta x}+\frac{\overline{\alpha}}{\Delta x}\frac{\hat{n}(k-1,l)+\hat{n}(k+1,l)}{2}\\ &\quad+\frac{\overline{\alpha}}{4\Delta x}[-\hat{n}^2(k+1,l)-\hat{n}^2(k-1,l)\\ &\quad+(\Delta x)^2\left(\hat{w}^2(k+1,l)+\hat{w}^2(k-1,l)\right)]\\ &\quad+O\left((\Delta x)^2\right). \qquad (3.19)\end{aligned}$$

Note that these are very similar in form to the update equations for $\bar{b}$ and $\bar{v}$, Eqs. (13) and (14). The covariances, C_{01}, that appeared in the former equations have been replaced by truncation errors from the Taylor expansion in the latter equations.

Thus we can define the *errors*,

$$e(k,l) \equiv \bar{b}(k,l) - \hat{n}(k,l) \tag{3.20}$$

$$f(k,l) \equiv \bar{v}(k,l) - \hat{w}(k,l). \tag{3.21}$$

The update rule for the errors is then

$$\begin{aligned} e(k,l+1) \;=\; & \frac{1+\bar{\alpha}}{2} e(k-1,l) + \frac{1-\bar{\alpha}}{2} e(k+1,l) \\ & + \frac{\bar{\alpha}}{4}\left[\left(\bar{b}(k+1,l) + \hat{n}(k+1,l)\right) e(k+1,l)\right. \\ & \qquad \left. - \left(\bar{b}(k-1,l) + \hat{n}(k-1,l)\right) e(k-1,l)\right] \\ & + \bar{\alpha}\left[C_{01}(k+1,l;k+1,l) - C_{01}(k-1,l;k-1,l)\right] \\ & + \mathcal{O}\left((\Delta x)^4\right) \end{aligned} \tag{3.22}$$

$$\begin{aligned} f(k,l+1) \;=\; & -\frac{e(k+1,l) - e(k-1,l)}{2\Delta x} + \frac{\bar{\alpha}}{\Delta x} \frac{e(k-1,l) + e(k+1,l)}{2} \\ & + \frac{\bar{\alpha}}{4\Delta x}\left[-\left(\bar{b}(k+1,l) + \hat{n}(k+1,l)\right) e(k+1,l)\right. \\ & \qquad \left. - \left(\bar{b}(k-1,l) + \hat{n}(k-1,l)\right) e(k-1,l)\right] \\ & + \frac{\bar{\alpha}}{\Delta x}\left[C_{01}(k+1,l;k+1,l) + C_{01}(k-1,l;k-1,l)\right] \\ & + \mathcal{O}\left((\Delta x)^2\right). \end{aligned} \tag{3.23}$$

Note that the evolution equation for $e(k,l)$ has decoupled from that of $f(k,l)$, so that it suffices to consider Eq. (22) alone. If we ignore the correlation (but *not* the truncation) terms, then we have the matrix equation

$$e(j,l+1) = \sum_k L(j,k;l) e(k,l) + \mathcal{O}\left((\Delta x)^4\right) \tag{3.24}$$

where $L(j,k;l)$ has positive elements, and columns that sum to unity. If we use the $\mathcal{L}_1$ norm,

$$\|\mathbf{e}(l)\| \equiv \frac{\Delta x}{L} \sum_j |e(j,l)|, \tag{3.25}$$

then we may write

$$\|\mathbf{e}(l+1)\| \;=\; \frac{\Delta x}{L} \sum_j |e(j,l+1)|$$

$$\begin{aligned} &\leq \frac{\Delta x}{L}\sum_j \left|\sum_k L(j,k;l)e(k,l)\right| + O\left((\Delta x)^4\right) \\ &\leq \frac{\Delta x}{L}\sum_{jk} L(j,k;l)|e(k,l)| + K(\Delta x)^4 \\ &= \frac{\Delta x}{L}\sum_k |e(k,l)| + K(\Delta x)^4 \\ &= \|\mathbf{e}(l)\| + K(\Delta x)^4 \end{aligned} \tag{3.26}$$

where K is some constant. Here we have used the triangle inequality, the positivity of the $L(j,k;l)$, and the fact that the columns of $L(j,k;l)$ sum to unity. Then, supposing $\mathbf{e}(0) = 0$, we see that iteration for $O\left((\Delta x)^{-2}\right)$ generations (times of order unity) will still yield $\|\mathbf{e}(l)\| = O\left((\Delta x)^2\right)$.

Now fix a subinterval (x_1, x_2) of the spatial domain of $n(x,t)$, and fix a time t_0. Let $\Delta x \to 0$ such that $x_1 = k\Delta x$, $x_2 = (k+M)\Delta x$, and $t_0 = l\Delta t$. Then by basic quadrature estimates

$$\frac{1}{M}\sum_{i=0}^{M-1} \hat{n}(k+i,l) = \frac{1}{x_2 - x_1}\int_{x_1}^{x_2} n(x,t_0)dx + O\left((\Delta x)^2\right), \tag{3.27}$$

while by our basic $\mathcal{L}_1$ error estimate we have

$$\frac{1}{M}\sum_{i=0}^{M-1} \bar{b}(k+i,l) = \frac{1}{M}\sum_{i=0}^{M-1} \hat{n}(k+i,l) + O\left((\Delta x)^2\right). \tag{3.28}$$

Here we have used the fact that $M\Delta x/L = (x_2 - x_1)/L$ is fixed. Combining these results gives

$$\frac{1}{M}\sum_{i=0}^{M-1} \bar{b}(k+i,l) = \frac{1}{x_2 - x_1}\int_{x_1}^{x_2} n(x,t_0)dx + O\left((\Delta x)^2\right), \tag{3.29}$$

which is our final convergence result for the ensemble average.

Thus, we have shown that the ensemble average of the cellular automaton simulates a stable, second-order accurate, fully-explicit differencing scheme for the Burgers' equation. Note that the proof of this is very similar in form to demonstrations of stability and accuracy for finite difference approximations. The neglect of the correlations is the weakest link in the chain of reasoning, and it will be discussed further in future work.

4. Bounding the Covariance for the Diffusion Equation

For the diffusion equation ($\overline{\alpha} = 0$), we can obtain an upper bound on the covariance, C_{ij}. To get dynamical equations for the covariance, we write

$$C_{ij}(k,l+1;k',l+1) \equiv \overline{(b_i(k,l+1) - \bar{b}_i(k,l+1))(b_j(k',l+1) - \bar{b}_j(k',l+1))} \tag{4.1}$$

and use Eqs. (7) and (8) to express the right- hand side in terms of quantities at time step l. For $k \neq k'$ or $i = j$, we get

$$\begin{aligned}
C_{00}(k+1,l+1;k'+1,l+1) &= \frac{1}{4}(C_{00}(k,l;k',l) + C_{01}(k,l;k',l) \\
&\quad + C_{10}(k,l;k',l) + C_{11}(k,l;k',l)) \\
C_{01}(k+1,l+1;k'-1,l+1) &= \frac{1}{4}(C_{00}(k,l;k',l) + C_{01}(k,l;k',l) \\
&\quad + C_{10}(k,l;k',l) + C_{11}(k,l;k',l)) \\
C_{10}(k-1,l+1;k'+1,l+1) &= \frac{1}{4}(C_{00}(k,l;k',l) + C_{01}(k,l;k',l) \\
&\quad + C_{10}(k,l;k',l) + C_{11}(k,l;k',l)) \\
C_{11}(k-1,l+1;k'-1,l+1) &= \frac{1}{4}(C_{00}(k,l;k',l) + C_{01}(k,l;k',l) \\
&\quad + C_{10}(k,l;k',l) + C_{11}(k,l;k',l)) \qquad (4.2)
\end{aligned}$$

When $k = k'$ and $i \neq j$, however, we get

$$\begin{aligned}
C_{01}(k+1,l+1;k-1,l+1) &= C_{01}(k,l;k,l) - \frac{1}{4}\left(\bar{b}_1(k,l) - \bar{b}_0(k,l)\right)^2 \\
C_{10}(k-1,l+1;k+1,l+1) &= C_{10}(k,l;k,l) - \frac{1}{4}\left(\bar{b}_0(k,l) - \bar{b}_1(k,l)\right)^2 . \qquad (4.3)
\end{aligned}$$

Note that Eqs. (2) are homogeneous in the covariances, while Eqs. (3) contain forcing terms on the right hand side. Because these forcing terms are negative definite, we can use induction on l to conclude that

$$C_{ij}(k,l;k',l) \leq 0 \qquad (4.4)$$

if $k \neq k'$ or $i \neq j$.

Suppose that we use spatial averaging to estimate the density n at a given gridpoint k. For example, we could average over M gridpoints to get the density

$$n_s \equiv \frac{1}{M}\sum_{i=0}^{M-1}[b_0(k+i,l) + b_1(k+i,l)]. \qquad (4.5)$$

We would like to compare this with the ensemble-averaged version of the same thing,

$$n_e \equiv \frac{1}{M}\sum_{i=0}^{M-1}[\bar{b}_0(k+i,l) + \bar{b}_1(k+i,l)], \qquad (4.6)$$

at the same gridpoint, k. We find

$$\begin{aligned}
\overline{(n_s - n_e)^2} &= \frac{1}{M^2}\sum_{i,j=0}^{M-1}(C_{00}(k+i,l;k+j,l) + C_{01}(k+i,l;k+j,l) \\
&\quad + C_{10}(k+i,l;k+j,l) + C_{11}(k+i,l;k+j,l))
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{1}{M^2}\sum_{i=0}^{M-1}\left(C_{00}(k+i,l;k+i,l)+C_{11}(k+i,l;k+i,l)\right)\\
&= \frac{1}{M^2}\sum_{i=0}^{M-1}\left[\bar{b}_0(k+i,l)-\left(\bar{b}_0(k+i,l)\right)^2\right.\\
&\quad \left.+\bar{b}_1(k+i,l)-\left(\bar{b}_1(k+i,l)\right)^2\right]\\
&\leq \frac{1}{M}n_e\left(1-\frac{n_e}{2}\right)\\
&\leq \frac{1}{2M}=\frac{\Delta x}{2(x_2-x_1)}
\end{aligned}
\tag{4.7}
$$

where we have used the Schwarz inequality in the final step. Thus, the spatial averaging of a single evolution of the automaton converges to the spatial averaging of the ensemble as $1/\sqrt{M}$. The simulations presented in the next section were carried out in precisely this fashion; the displayed results are spatial averages for a single evolution of the automaton.

5. Simulations

The equation simulated by the above-described automaton is

$$
\frac{\partial n}{\partial t}+c\frac{\partial}{\partial x}\left(n-\frac{n^2}{2}\right)=\nu\frac{\partial^2 n}{\partial x^2},
\tag{5.1}
$$

where $0<n<2$. To maximize the signal-to-noise ratio, it is best to operate with $n\sim 1$. Then, the transformation

$$
u=c(n-1)
\tag{5.2}
$$

may be applied to the result, so that u obeys the Burgers' equation in standard form,

$$
\frac{\partial u}{\partial t}-u\frac{\partial u}{\partial x}=\nu\frac{\partial^2 u}{\partial x^2}.
\tag{5.3}
$$

Note that $-c<u<c$, so the parameter c should be chosen greater than $\sup|u|$ to insure that $0<n<2$.

We have used the automaton on a Connection Machine [4] computer to simulate the solution to Burgers' equation with periodic boundary conditions on a spatial domain of unit length, and with initial condition

$$
n(x,0)=n_a+n_b\cos(2\pi x).
\tag{5.4}
$$

The exact solution to this problem may be found by application of the Cole-Hopf transformation. It is

$$
n=n_a+\frac{2\nu}{c\psi}\frac{\partial\psi}{\partial x},
\tag{5.5}
$$

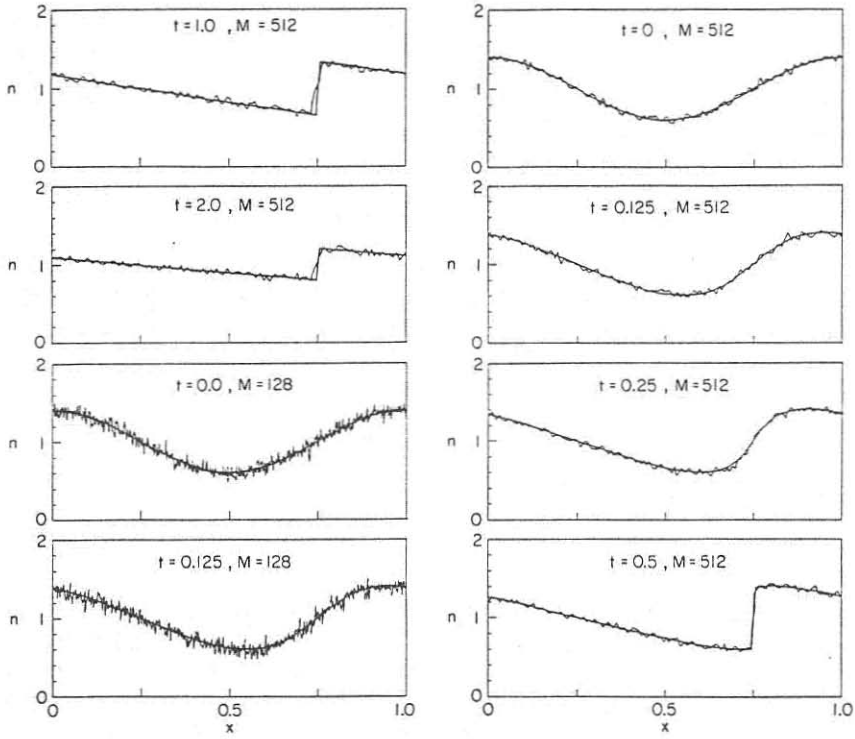

Figure 1: Simulations of the cellular automaton model as a function of time t. The solid curve gives the exact solution to Burgers' equation. The development of a shock is evident. Averages were made over M lattice sites.

where

$$\psi \equiv I_0(z) + 2\sum_{\ell=1}^{\infty}(-1)^{[\ell/2]} I_\ell(z) f_\ell(2\pi\ell x + \nu_\ell t)\exp(-\mu_\ell t), \tag{5.6}$$

where in turn

$$z \equiv \frac{cn_b}{4\pi\nu} \tag{5.7}$$

$$\mu_\ell \equiv \nu(2\pi\ell)^2 \tag{5.8}$$

$$\nu_\ell \equiv c(n_a - 1)(2\pi\ell), \tag{5.9}$$

the I_ℓ's are modified Bessel functions, and f_ℓ denotes the sine (cosine) function when ℓ is odd (even):

$$f_\ell \equiv \begin{cases} \sin & \text{if } \ell \text{ is odd} \\ \cos & \text{if } \ell \text{ is even.} \end{cases} \tag{5.10}$$

In the results presented below, we took initial conditions with $n_a = 1.0$ and $n_b = 0.4$. We took the diffusion coefficient to be $\nu = 2^{-15}$, and the advection velocity to be $c = 1$. We used $2^{16} = 65536$ spatial gridpoints, so $\Delta x = 2^{-16}$. Then the bias was given by

$$\overline{\alpha} = \frac{c\Delta x}{2\nu} = 0.25, \tag{5.11}$$

and the time step was given by

$$\Delta t = \frac{(\Delta x)^2}{2\nu} = 2^{-18}. \tag{5.12}$$

Note that the characteristic time for shock formation is $t_s = (2\pi c n_b)^{-1} \approx 0.398$, and that $2^{18} = 262144$ automaton time steps correspond to $t = 1$. Below we plot the results for several different values of t, with $M = 2^9 = 512$ so that there are $65536/512 = 128$ points plotted on each of these graphs. The last two plots, however, were made with $M = 128$; note that the amplitude of the noise in these plots is roughly twice that in the other plots, as expected from Eq. (7). The smooth curves are the exact answer, as given by Eq. (5).

6. Conclusion

We have motivated, developed, and analyzed a cellular automaton for the simulation of Burgers' equation. As stated at the outset of this paper, we are not proposing that this technique be used as an effective method for computing solutions to Burgers' equation, but rather that it be used to study in a relatively simple context important issues about stability and accuracy that arise in the application of cellular automaton techniques to solving the incompressible Navier-Stokes equations [1]. The argument used to show that the cellular automaton does indeed yield an approximation to

the solution of the partial differential equation is the same for both cases; including the neglect of the correlations. The general procedure used to get the solution from the cellular automaton is the same in both cases: the spatial average of the cellular automaton is used to approximate the spatial average of the ensemble average of the automaton.

Acknowledgements

The work of BMB was performed in part at Lawrence Livermore National Laboratory under DOE contract number W-7405-ENG-48. The work of CDL was performed in part at Lawrence Livermore National Laboratory under DOE contract number W-7405-ENG-48, and in part at Brown University.

References

[1] See for example, U. Frisch, B. Hasslacher, Y. Pomeau, "Lattice Gas Automata for the Navier-Stokes Equation," *Physical Review Letters*, **56** (1986) 1505; and S. Wolfram, "Cellular Automaton Fluids 1: Basic Theory," *Journal of Statistical Physics*, **45** (1986) 471-526, and references contained therein.

[2] W. Feller, *An Introduction to Probability Theory and its Applications*, Volume I, Chapter 14, Section 6 (J. Wiley, 1970).

[3] S. Wolfram, "Random Sequence Generation by Cellular Automata," *Advances in Applied Mathematics*, **7** (1986) 123. See, in particular, eqn. (3.1a).

[4] D. Hillis, *The Connection Machine*, (M.I.T. Press, 1985).