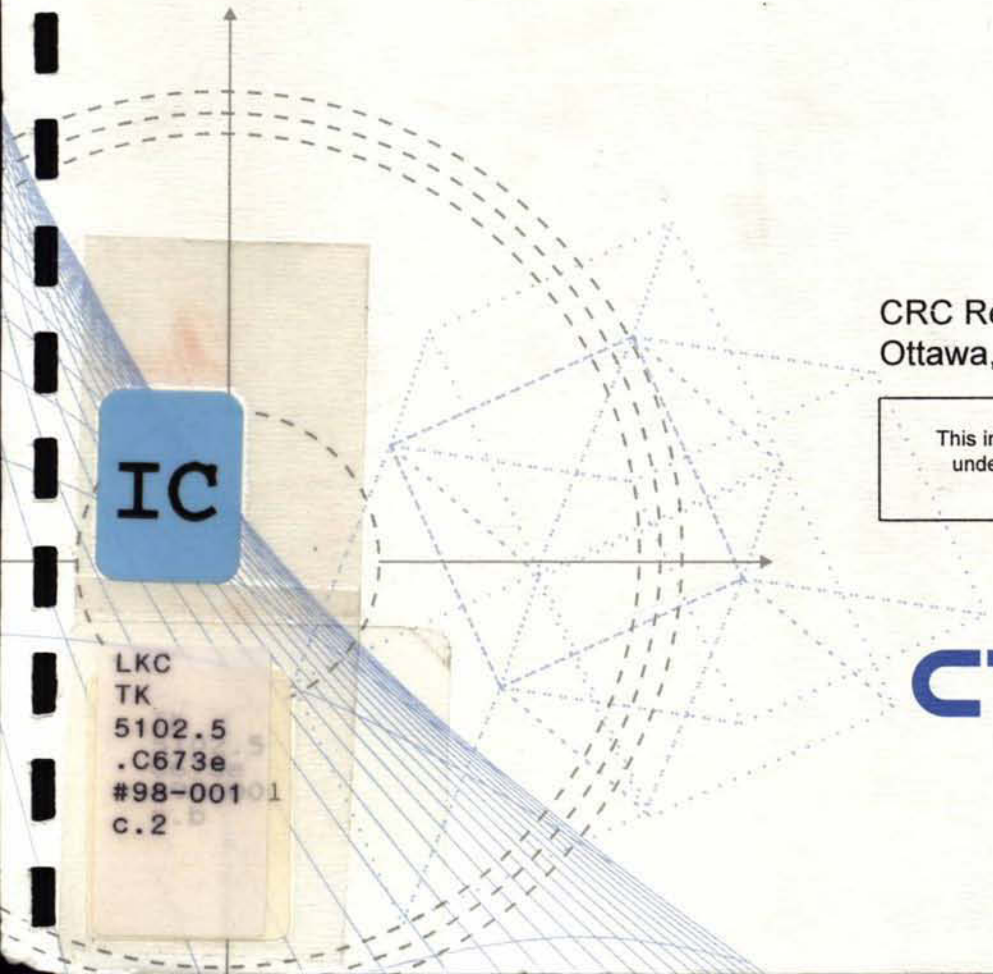
Integer Lattice Gas Automata for the Numerical Solution of Electromagnetic Problems

Joanne R. Treurniet & N.R. Simons

Antenna and Integrated Electronics



CRC Report No. CRC-RP-98-001
Ottawa, January 1998

CAUTION
This information is provided with the express understanding that proprietary and patent rights will be protected

Communications Research Centre
Centre de recherches sur les communications

Abstract

Integer lattice gas methods are gaining in popularity due to their speed, their inherent lack of computational error, and the possible use of non-floating point computer architectures. In this study, we attempt to apply these advantages to electromagnetic problems. This report will provide an introduction to lattice gas automata and transmission line matrix methods, then show how the two can be combined to simulate electromagnetic wave behaviour. Application of the model to the TE modes of a 2-dimensional waveguide will be presented and limitations of the model will be discussed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Many problems in electromagnetics cannot be solved analytically. We must therefore rely on numerical methods to obtain the solutions to Maxwell's equations. Currently, finite element [1], finite difference [2] and transmission line matrix (TLM) [3] methods are used, all of which are discrete in space and time and continuous in dependent variable.

Improvements are always sought over traditional computational methods, and the above are not excepted. The need for a floating point processor necessarily introduces error (round-off and truncation) [17], a problem which may be eliminated by the use of integer arithmetic, which at the same time shortens the computation time.

The method proposed here to eliminate this need is a combination of the traditional TLM method with the entirely discrete Cellular Automata (CA) methods [12][13][14][16]. The motivation behind this marriage lies in the shared TLM and CA concept of "packets" moving from node to node on a lattice and interacting by a set of "rules".

In this report it is shown that the future of cellular automata methods in electromagnetic studies is promising.

1.2 Structure of This Report

In this report, the results of a study into the application of integer lattice gas methods to a simple two-dimensional waveguide are presented. The chapters to follow are organized in the following manner. The TLM method is introduced in Chapter 2 and the dispersion relations related to the discretization of the TLM method are derived in Chapter 3. In Chapter 4, cellular automata are defined, followed by a description of lattice gas models and their usual applications. The TLM method is combined with an integer lattice gas model in Chapter 5, where lattice gas collision rules based on the TLM algorithm are presented. The general theory of waveguides is given in Chapter 6 and an outline of the program in Chapter 7, to provide a basis for the results of the study in Chapter 8.

Chapter 2

TLM Method for Electromagnetics

Maxwell's equations must in some cases be solved numerically:

$$\begin{aligned}\nabla \times H &= \epsilon\dot{E} + \sigma E \\ \nabla \times E &= -\mu\dot{H} \\ \nabla \cdot E &= \rho/\epsilon \\ \nabla \cdot H &= 0\end{aligned}$$

One of the methods applied to solve these problems is the Transmission Line Matrix (TLM) Method. This chapter will concentrate on the derivation of the TLM method. It will be shown that the TLM method is mathematically equivalent to Maxwell's equations.

2.1 Background

The TLM method has its roots in the Newtonian idea of a corpuscular form of light and in Huygen's Principle [4]: every point on a wavefront is a spherical source.

These ideas are combined in the TLM method, originally proposed by P. B. Johns in 1974 [5]. It can be regarded as a physical model of Huygen's principle applied to voltages and currents, where the voltages are thought of as moving in packets of energy (or, having the form of an impulse). A lattice is used for these impulses to travel upon and at each node they are scattered. The impulse then becomes a source for the surrounding nodes, which in turn become sources propagating through the mesh. This reflects Huygen's principle.

2.2 Derivation of the TLM Algorithm

The circuit shown in Figure 2.1 was used to model Maxwell's equations in two dimensions and was known as a "network analyzer" [4] (not to be confused with a network analyzer test setup used in modern microwave reflection and transmission measurements). These circuits were built to solve for the electromagnetic properties of irregular two-dimensional objects by taking on the same boundary shape.

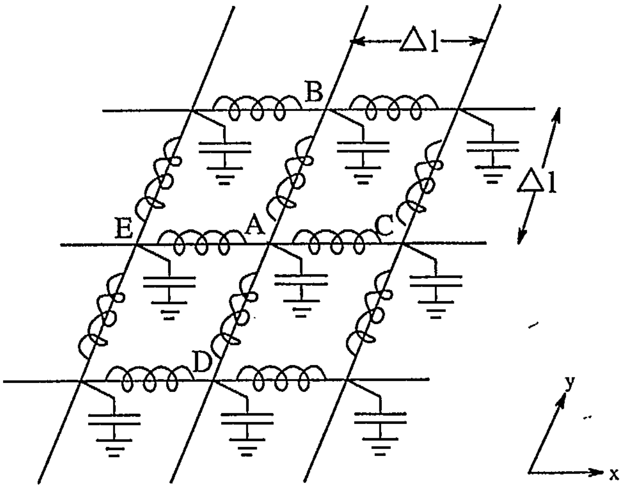


Figure 2.1: Network analyzer

The voltage drop across an inductance is given by

$$V = L\frac{\partial i}{\partial t}$$

where L is the inductance and i is the current through the inductor. Assigning values of $L\Delta l$ to the inductances and $2C\Delta l$ to the capacitances in where V_C is the voltage a node C (similarly for A) and Δl is the distance between nodes. This may also be written as

$$\frac{V(x+\Delta l)-V(x)}{\Delta l} = -L\frac{\partial i_x}{\partial t} \tag{2.1}$$

Similarly,

$$\frac{V(y+\Delta l)-V(y)}{\Delta l} = -L\frac{\partial i_y}{\partial t} \tag{2.2}$$

Kirchoff's current law states that the current flowing into a node is equal to the current flowing out. In Figure 2.1, the current flowing into the node A is $i_{EA}+i_{DA}+i_{CA}+i_{BA}$, and the current flowing out is i_C, the current through the capacitor. The current through a capacitor is equal to $C\frac{\partial V}{\partial t}$, hence

$$i_{EA}+i_{DA}+i_{CA}+i_{BA} = 2C\Delta l\frac{\partial V_A}{\partial t} \tag{2.3}$$

Expanding this in terms of current in the x direction (i_x) and y direction (i_y),

$$-\left[\frac{i_x(x+\frac{\Delta l}{2})-i_x(x-\frac{\Delta l}{2})}{\Delta l}\right]-\left[\frac{i_y(y+\frac{\Delta l}{2})-i_y(y-\frac{\Delta l}{2})}{\Delta l}\right] = 2C\frac{\partial V_A}{\partial t} \tag{2.4}$$

(recall that the capacitance is assigned a value of $2C\Delta l$). Letting $\Delta l, \Delta t \to 0$ in equations (2.1), (2.2) and (2.4), one obtains the set of equations

$$\begin{aligned}
\frac{\partial V}{\partial x} &= -L\frac{\partial i_x}{\partial t}\\
\frac{\partial V}{\partial y} &= -L\frac{\partial i_y}{\partial t}\\
-\frac{\partial i_x}{\partial x}-\frac{\partial i_y}{\partial y} &= 2C\frac{\partial V}{\partial t}
\end{aligned}$$

which are strikingly similar to Maxwell's 2 dimensional TE (Transverse Electric, *i.e.* $E_z = 0$) propagation equations [6]:

$$\begin{aligned}
\frac{\partial E_z}{\partial x} &= \mu\frac{\partial H_y}{\partial t}\\
\frac{\partial E_z}{\partial y} &= -\mu\frac{\partial H_x}{\partial t}\\
\frac{\partial H_y}{\partial x}-\frac{\partial H_x}{\partial y} &= \epsilon\frac{\partial E_z}{\partial t}
\end{aligned}$$

where E and H are the electric and magnetic fields, respectively. Hence with the substitutions $V \Longleftrightarrow E_z$, $i_x \Longleftrightarrow -H_y$, $i_y \Longleftrightarrow H_x$, $\mu \Longleftrightarrow L$, $\epsilon \Longleftrightarrow 2C$, the equations are analogous.

2.2.1 Transmission Lines

A transmission line is ideally only meant to introduce a delay into a system without affecting the voltage, *i.e.* $V_0(t) = V_i(t-\tau)$. It may be represented as

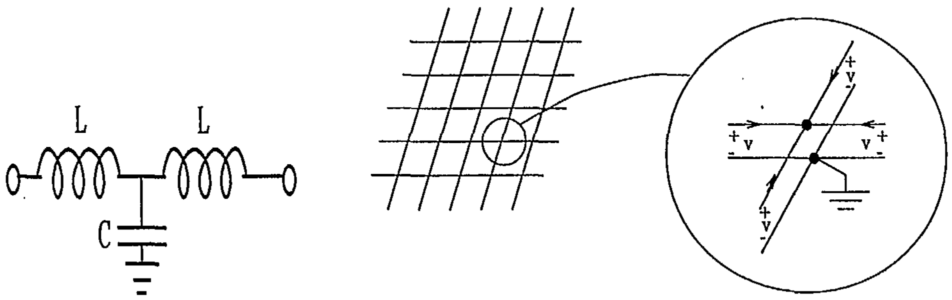
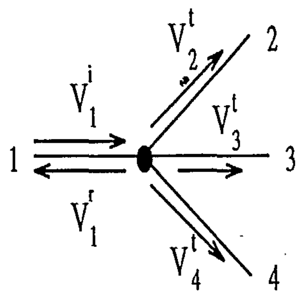


Figure 2.2: A transmission line node.

shown on the left of Figure 2.2, which is the same as a node on the network analyzer. Hence the network analyzer may be thought of as a "transmission line matrix" or TLM. The right diagram shows a real transmission line, which consists of two lines per node, one of which is grounded.

2.2.2 TLM for a Square Lattice

Say an impulse voltage enters a node at position 1 on a square lattice, and the impedance of each line is the same, z_0. The total impedance, by Kirchoff's Laws [7], is found as three impedances in parallel:



$$\begin{aligned} z_{in} &= z_0 \\ z_{out} &= z_0 \parallel z_0 \parallel z_0 \\ &= \frac{1}{3} z_0 \end{aligned}$$

The voltage reflection coefficient [6] is then

$$\Gamma = \frac{z_{out} - z_{in}}{z_{out} + z_{in}} = -\frac{1}{2}$$

If we were to normalize the input voltage such that V_i=1, we would get V_r=-1/2 and V_{T2}=V_{T3}=V_{T3}=1/2 since the transmission coefficient is given by $T = 1 + \Gamma$.

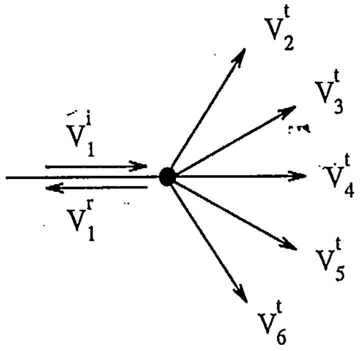
A set of equations may be set up in matrix form:

$$\begin{pmatrix} V_1^r \\ V_2^r \\ V_3^r \\ V_4^r \end{pmatrix} = \frac{1}{2} \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \begin{pmatrix} V_1^i \\ V_2^i \\ V_3^i \\ V_4^i \end{pmatrix} \tag{2.5}$$

This gives the formula for the evolution of a voltage impulse, since all reflected and transmitted amplitudes become incident voltages at adjacent nodes at the next time step.

2.2.3 TLM for a Hexagonal Lattice

For a hexagonal lattice, an impulse voltage again enters a node at position 1 with the impedance of each line at z_0. There are 5 directions for transmission and one for reflection. The total impedance is



$$\begin{aligned} z_{in} &= z_0 \\ z_{out} &= z_0 \parallel z_0 \parallel z_0 \parallel z_0 \parallel z_0 \\ &= \frac{1}{5} z_0 \end{aligned}$$

The reflection coefficient is then

$$\Gamma = \frac{z_{out} - z_{in}}{z_{out} + z_{in}} = -\frac{2}{3}$$

and the transmission coefficient is

$$T = 1 + \Gamma = \frac{1}{3} \tag{2.6}$$

Thus the unit impulse will be reflected with an amplitude of -2/3 and have 5 transmitted voltages of 1/3.

In matrix form,

$$\begin{pmatrix} V_1^r \\ V_2^r \\ V_3^r \\ V_4^r \\ V_5^r \\ V_6^r \end{pmatrix} = \frac{1}{3} \begin{bmatrix} -2 & 1 & 1 & 1 & 1 & 1 \\ 1 & -2 & 1 & 1 & 1 & 1 \\ 1 & 1 & -2 & 1 & 1 & 1 \\ 1 & 1 & 1 & -2 & 1 & 1 \\ 1 & 1 & 1 & 1 & -2 & 1 \\ 1 & 1 & 1 & 1 & 1 & -2 \end{bmatrix} \begin{pmatrix} V_1^i \\ V_2^i \\ V_3^i \\ V_4^i \\ V_5^i \\ V_6^i \end{pmatrix} \tag{2.7}$$

Equation (2.7) is the TLM method applied to one node on the lattice. The equation is simultaneously applied to every node on the lattice in one time step.

The TLM model approximates wave behaviour in the limit of small Δl and Δt. This discretization introduces an error known as the dispersion relation, which can be derived analytically.

Chapter 3

TLM Dispersion Relations

Numerical dispersion is defined as the variation of a propagating wave's wavelength with frequency. The dispersion relation describes the fundamental manner in which plane waves propagate through a mesh [2]. It illustrates the effect of using a lattice approximation by providing an analytical relationship between the wavelength of the simulation in discrete space (dependent on Δl and Δt) and the frequency in real space. Hence it provides a means to determine the numerical phase constant which may be compared to the exact physical phase constant to determine the amount of velocity error. It can be shown [9] that for wavelengths a few multiples of the mesh size, Δl, the numerical phase velocity is approximately constant in all directions on a hexagonal lattice, whereas it is definitely not so for a rectangular lattice.

The following derivation was presented in [8]. In this method, quantum mechanical notation for the propagation of waves is used. This derivation will use two dimensions only; the first section shows Krumpholz and Russer's derivation for the 2D square lattice, while the second section shows the derivation for a 2D hexagonal lattice.

Unfortunately, this method cannot be applied to a case such as the one we are using, where only a portion of the states can undergo a TLM operation (see Chapter 5). In this case the field becomes non-local, *i.e.* in order to determine the field strength at a site, one must know the field strength at all other sites on the lattice [10].

3.1 2D Dispersion in Free Space

The two-dimensional scalar wave equation is given by

$$\frac{\partial^2 u}{\partial^2 t} = c^2 \nabla^2 u \tag{3.1}$$

where $u = u(x, y, t)$. If the solution for a 2D propagating wave

$$\vec{u}(x, y, t) = \mathrm{e}^{j(\omega t - k\cos\phi x - k\sin\phi y)}$$

is substituted in (3.1), where ϕ is the angle which defines the direction of propagation, the variation in the wavelength λ with it's frequency f can be found (or equivalently, the variation of wavenumber $k = 2\pi/\lambda$ with angular frequency $\omega = 2\pi f$).

$$\begin{aligned} (\omega)^2 &= c^2(k^2(\cos^2\phi + \sin^2\phi)) \\ \omega &= \pm ck \\ f &= \frac{c}{\lambda} \end{aligned}$$

Thus the wave propagation velocity in 2 dimensions is

$$v_p = \frac{\omega}{k} = \pm c \tag{3.2}$$

$$f = \frac{c}{\lambda} \tag{3.3}$$

which is the relationship between wavelength and frequency (dispersion relation) in free space.

3.2 TLM Dispersion on a 2D Square Lattice

In discrete time and space, let (m, n) represent the location of a node and k represent the time step. Let the incident wave amplitudes be represented by the state $| a >$ and the scattered wave amplitudes by the state $| b >$:

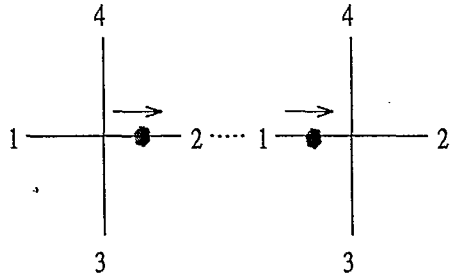
$$\begin{aligned} | a > &= \sum_{k,m,n=-\infty}^{+\infty} {}_k\mathbf{a}_{m,n} \, | k; m, n > \\ | b > &= \sum_{k,m,n=-\infty}^{+\infty} {}_k\mathbf{b}_{m,n} \, | k; m, n > \end{aligned}$$

where $x = m\Delta l$, $y = n\Delta l$ and $t = k\Delta t$.

The instantaneous wave propagation between nodes may be described by

$$|\, a >= \Gamma \,|\, b > \tag{3.4}$$

where Γ is the transfer or advection operator. Recall that at any node on a two dimensional lattice, the TLM method starts with incident wave amplitudes and outputs reflected wave amplitudes. This affects the transfer matrix by reversing the indices:



$$\begin{aligned} {}_k\mathbf{a}_{1,m,n} &= {}_k\mathbf{b}_{2,m-1,n} \\ {}_k\mathbf{a}_{2,m,n} &= {}_k\mathbf{b}_{1,m+1,n} \\ {}_k\mathbf{a}_{3,m,n} &= {}_k\mathbf{b}_{4,m,n-1} \\ {}_k\mathbf{a}_{4,m,n} &= {}_k\mathbf{b}_{3,m,n+1} \end{aligned} \tag{3.5}$$

For example, an incident wave at position 1 at (m,n) came from a wave scattered at position 2 at (m-1,n).

The transfer matrix operator which satisfies Equations (3.5) is then

$$\Gamma = \begin{bmatrix} 0 & X & 0 & 0 \\ X^\dagger & 0 & 0 & 0 \\ 0 & 0 & 0 & Y \\ 0 & 0 & Y^\dagger & 0 \end{bmatrix} \tag{3.6}$$

where X is the shift operator defined by

$$\begin{aligned} X \,|\, k; m, n > &= \,|\, k; m+1, n > \\ X^\dagger \,|\, k; m, n > &= \,|\, k; m-1, n > \end{aligned}$$

and similarly for Y.

The equation

$$|\, b >= TS \,|\, a > \tag{3.7}$$

describes the simultaneous scattering at all mesh nodes. Here S is the TLM scattering matrix described by Equation (2.5),

$$S = \frac{1}{2} \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \tag{3.8}$$

and T is the time increment operator,

$$T\,|\,k; m, n >= \,|\,k+1; m, n > \qquad (3.9)$$

Combining (3.4) and (3.7) gives

$$(\Gamma TS - 1)\,|\,a >= 0 \qquad (3.10)$$

To calculate the complete field state in the frequency domain, the base vectors

$$|\,\Omega >= \sum_{k=-\infty}^{\infty} \mathrm{e}^{jk\Omega}\,|\,k > \qquad (3{,}11)$$

are introduced. Here, $|\,k >$ and $|\,\Omega >$ are sets of orthonormal basis functions. Ω is a normalized frequency, $\Omega = 2\pi\Delta t f$. Taking the scalar product of $|\,a >$ and $|\,\Omega >$, one obtains

$$\begin{aligned} |\,a(\Omega) > &= \;<\Omega\,|\,a >= \sum_{k=-\infty}^{\infty} \mathrm{e}^{-jk\Omega} < k\,|\,a > \\ &= \sum_{k=-\infty}^{\infty} \mathrm{e}^{-jk\Omega}\,|\,a(k) > \qquad (3.12) \end{aligned}$$

Note that $|\,a(\Omega) >$ and $|\,a(k) >$ are connected by a Fourier series.

Multiplying equation (3.10) by $|\,\Omega >$:

$$<\Omega\,|\,\Gamma TS\,|\,a > - <\Omega\,|\,a >= 0$$

Only the time operator affects Ω; Γ and S operate on positions.

$$\begin{aligned} \Gamma S <\Omega\,|\,T\,|\,a > - <\Omega\,|\,a > &= 0 \\ \Gamma S \left(\sum_{k=-\infty}^{\infty} \mathrm{e}^{-j(k+1)\Omega} < k+1\,|\,a > \right) - <\Omega\,|\,a > &= 0 \\ \Gamma S \mathrm{e}^{-j\Omega} \left(\sum_{k=-\infty}^{\infty} \mathrm{e}^{-jk\Omega} < k\,|\,a > \right) - <\Omega\,|\,a > &= 0 \end{aligned}$$

The last equation results since k ranges from $-\infty$ to $+\infty$. Using the relations defined in (3.12),

$$\Gamma S \mathrm{e}^{-j\Omega}\,|\,a(\Omega) > - \,|\,a(\Omega) >= 0$$

or

$$\Gamma S \,|\, a(\Omega) >= e^{j\Omega} \,|\, a(\Omega) >$$

which from linear algebra shows the state in frequency space as the eigenvector and $e^{j\Omega}$ as its corresponding set of eigenvalues. Finding the roots of the characteristic polynomial

$$f(\lambda) = |\lambda I_4 - \Gamma S| = 0 \qquad (3.13)$$

gives the solution of eigenvalues

$$\begin{aligned} \lambda_1 &= C + \sqrt{C^2 - 1} \\ \lambda_2 &= C - \sqrt{C^2 - 1} \\ \lambda_3 &= 1 \\ \lambda_4 &= -1 \end{aligned} \qquad (3.14)$$

where $C = \frac{1}{4}(X + X^\dagger + Y + Y^\dagger)$. λ_3 corresponds to the stationary solution, $\Omega = 0$. λ_4 corresponds to an oscillating spurious solution with $\Omega = \pi$.

Since λ_1 and λ_2 are composed of operators, an expectation value for the eigenvalues must be found. At this point the base vectors

$$|\, \chi, \eta >= \sum_{m,n=-\infty}^{\infty} e^{j(\chi m + \eta n)} \,|\, m, n > \qquad (3.15)$$

are introduced, where $\chi = \Delta l k_x$, $\eta = \Delta l k_y$ (normalized wave vector components). Then

$$< \chi, \eta \,|\, a(\Omega) >= \sum_{m,n=-\infty}^{\infty} e^{-j(\chi m + \eta n)} < m, n \,|\, a(\Omega) > \qquad (3.16)$$

again related by Fourier series. Since $|\, \chi, \eta >$ and $|\, a(\Omega) >$ are both representations of $|\, a >$ in different domains, the expectation value of the eigenvalue can be found by using these states.

$$< \chi, \eta \,|\, \Gamma S \,|\, a(\Omega) >=< \chi, \eta \,|\, \lambda \,|\, a(\Omega) >=< \chi, \eta \,|\, e^{j\Omega} \,|\, a(\Omega) >$$

First the expectation value of C is calculated:

$$< \chi, \eta \,|\, C \,|\, a(\Omega) > \;=\; \frac{1}{4} < \chi, \eta \,|\, (X + X^\dagger + Y + Y^\dagger) \,|\, a(\Omega) >$$

$$\begin{aligned} &= \frac{1}{4}\sum_{m,n=-\infty}^{\infty} \mathrm{e}^{-j(\chi(m+1)+\eta n)} < m+1, n \,|\, a(\Omega) > \\ &+\frac{1}{4}\sum_{m,n=-\infty}^{\infty} \mathrm{e}^{-j(\chi(m-1)+\eta n)} < m-1, n \,|\, a(\Omega) > \\ &+\frac{1}{4}\sum_{m,n=-\infty}^{\infty} \mathrm{e}^{-j(\chi m+\eta(n+1))} < m, n+1 \,|\, a(\Omega) > \\ &+\frac{1}{4}\sum_{m,n=-\infty}^{\infty} \mathrm{e}^{-j(\chi m+\eta(n-1))} < m, n-1 \,|\, a(\Omega) > \end{aligned}$$

expanding from (3.15). Using (3.16),

$$\begin{aligned} < \chi, \eta \,|\, C \,|\, a(\Omega) > &= \frac{1}{4}(\mathrm{e}^{-j\chi} + \mathrm{e}^{j\chi} + \mathrm{e}^{-j\eta} \\ &\quad +\mathrm{e}^{j\eta}) < \chi, \eta \,|\, a(\Omega) > \\ &= \frac{1}{2}(\cos\chi + \cos\eta) < \chi, \eta \,|\, a(\Omega) > \end{aligned}$$

It is evident that the expectation value of C is bounded between -1 and +1. Since we are interested in the real part of the solution, we can neglect the part of the eigenvalues (Equations (3.14)) that are under the square root since their expectation values will be complex. Using

$$\Gamma S \,|\, a(\Omega) >= \lambda \,|\, a(\Omega) >= \mathrm{e}^{j\Omega} \,|\, a(\Omega) >$$

and

$$\mathrm{e}^{j\Omega} = \cos\Omega + j\sin\Omega$$

one obtains the dispersion relation

$$\cos\Omega = \frac{1}{2}(\cos\chi + \cos\eta) \qquad (3.17)$$

Using trigonometric identities

$$\begin{aligned} \chi &= \Delta l k_x = \Delta l k \cos\phi \\ \eta &= \Delta l k_y = \Delta l k \sin\phi \\ \Omega &= 2\pi\Delta t f = \omega\Delta t \qquad (3.18) \end{aligned}$$

where ϕ is the angle of wave propagation, and substituting back, one obtains the traditional form of the dispersion equation for two dimensional TLM on a square lattice [4]:

$$\sin^2\left(\frac{\omega\Delta t}{2}\right) = \frac{1}{2}\left(\sin^2\left(\frac{k\Delta l}{2}\cos\phi\right) + \sin^2\left(\frac{k\Delta l}{2}\sin\phi\right)\right) \tag{3.19}$$

To illustrate the relationship given by the above equation, let Δl and Δt approach zero, as in the case of free space:

$$\begin{aligned}\left(\frac{\omega\Delta t}{2}\right)^2 &= \frac{1}{2}\left(\left(\frac{k\Delta l}{2}\cos\phi\right)^2 + \left(\frac{k\Delta l}{2}\sin\phi\right)^2\right)\\ &= \frac{1}{2}\left(\frac{k\Delta l}{2}\right)^2\\ (f\Delta t)^2 &= \frac{1}{2}\left(\frac{\Delta l}{\lambda}\right)^2\end{aligned}$$

or

$$\lambda f = \frac{1}{\sqrt{2}}\frac{\Delta l}{\Delta t} \tag{3.20}$$

This matches the dispersion relation for two-dimensional free space as shown in (Section 3.1) with c defined as

$$c = \frac{1}{\sqrt{2}}\frac{\Delta l}{\Delta t}$$

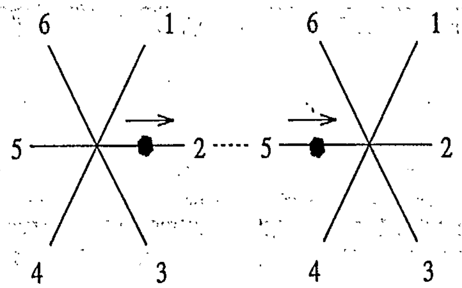
which is the speed of propagation in a 2D TLM mesh [4].

3.3 TLM Dispersion on a 2D Hexagonal Lattice

Using the same notation as in the previous section, the instantaneous wave propagation between nodes is again described by

$$|\,a >= \Gamma\,|\,b >$$

where Γ is the transfer or advection operator. The transfer matrix now takes on a different form:



$$\begin{aligned}
{}_k\mathbf{a}_{1,m,n} &= {}_k\mathbf{b}_{4,m-1/2,n-\sqrt{3}/2}\\
{}_k\mathbf{a}_{2,m,n} &= {}_k\mathbf{b}_{5,m-1,n}\\
{}_k\mathbf{a}_{3,m,n} &= {}_k\mathbf{b}_{6,m-1/2,n+\sqrt{3}/2}\\
{}_k\mathbf{a}_{4,m,n} &= {}_k\mathbf{b}_{1,m+1/2,n+\sqrt{3}/2}\\
{}_k\mathbf{a}_{5,m,n} &= {}_k\mathbf{b}_{2,m+1,n}\\
{}_k\mathbf{a}_{6,m,n} &= {}_k\mathbf{b}_{3,m+1/2,n-\sqrt{3}/2}
\end{aligned}$$

For example, a wave scattered from (m-1,n) at position 2 becomes the incident wave at position 5 at (m,n). Defining the operators

$$\begin{aligned}
X\,|\,k;m,n> &= |\,k;m+1,n>\\
X^\dagger\,|\,k;m,n> &= |\,k;m-1,n>\\
Y_1\,|\,k;m,n> &= |\,k;m+1/2,n+\sqrt{3}/2>\\
Y_1^\dagger\,|\,k;m,n> &= |\,k;m-1/2,n-\sqrt{3}/2>\\
Y_2\,|\,k;m,n> &= |\,k;m+1/2,n-\sqrt{3}/2>\\
Y_2^\dagger\,|\,k;m,n> &= |\,k;m-1/2,n+\sqrt{3}/2>
\end{aligned}$$

one obtains the form of Γ applicable to the system:

$$\Gamma = \begin{bmatrix} 0 & 0 & 0 & Y_1 & 0 & 0\\ 0 & 0 & 0 & 0 & X & 0\\ 0 & 0 & 0 & 0 & 0 & Y_2\\ Y_1^\dagger & 0 & 0 & 0 & 0 & 0\\ 0 & X^\dagger & 0 & 0 & 0 & 0\\ 0 & 0 & Y_2^\dagger & 0 & 0 & 0 \end{bmatrix} \qquad (3.21)$$

The TLM operator for a hexagonal lattice is given by Equation (2.7):

$$S = \frac{1}{3}\begin{bmatrix} -2 & 1 & 1 & 1 & 1 & 1\\ 1 & -2 & 1 & 1 & 1 & 1\\ 1 & 1 & -2 & 1 & 1 & 1\\ 1 & 1 & 1 & -2 & 1 & 1\\ 1 & 1 & 1 & 1 & -2 & 1\\ 1 & 1 & 1 & 1 & 1 & -2 \end{bmatrix} \qquad (3.22)$$

and the time increment operator is as previously defined (3.9).

Following the same method, the eigenvalue equation remains as

$$\Gamma S\,|\,a(\Omega) > = e^{j\Omega}\,|\,a(\Omega) >$$

for which the eigenvalues are

$$\begin{aligned}\lambda_1 &= C + \sqrt{C^2-1}\\ \lambda_2 &= C - \sqrt{C^2-1}\\ \lambda_3 &= 1\\ \lambda_4 &= -1\end{aligned}$$

with $C = \frac{1}{6}(X + X^\dagger + Y_1 + Y_1^\dagger + Y_2 + Y_2^\dagger)$.

Once again we can ignore the complex part of the eigenvalue.

$$\begin{aligned}<\chi,\eta\,|\,C\,|\,a(\Omega)> &= \frac{1}{6} < \chi,\eta\,|\,(X + X^\dagger + Y_1 + Y_1^\dagger + Y_2 + Y_2^\dagger)\,|\,a(\Omega) >\\ &= \frac{1}{6}\sum_{m,n=-\infty}^{\infty} e^{-j(\chi(m+1)+\eta n)} < m+1, n\,|\,a(\Omega) >\\ &\quad +\frac{1}{6}\sum_{m,n=-\infty}^{\infty} e^{-j(\chi(m-1)+\eta n)} < m-1, n\,|\,a(\Omega) >\\ &\quad +\frac{1}{6}\sum_{m,n=-\infty}^{\infty} e^{-j(\chi(m+1/2)+\eta(n+\sqrt{3}/2))} < m+1/2, n+\sqrt{3}/2\,|\,a(\Omega) >\\ &\quad +\frac{1}{6}\sum_{m,n=-\infty}^{\infty} e^{-j(\chi(m-1/2)+\eta(n-\sqrt{3}/2))} < m-1/2, n-\sqrt{3}/2\,|\,a(\Omega) >\\ &\quad +\frac{1}{6}\sum_{m,n=-\infty}^{\infty} e^{-j(\chi(m+1/2)+\eta(n-\sqrt{3}/2))} < m+1/2, n-\sqrt{3}/2\,|\,a(\Omega) >\\ &\quad +\frac{1}{6}\sum_{m,n=-\infty}^{\infty} e^{-j(\chi(m-1/2)+\eta(n+\sqrt{3}/2))} < m-1/2, n+\sqrt{3}/2\,|\,a(\Omega) >\\ &= \frac{1}{6}(e^{-j\chi} + e^{j\chi} + e^{-j\chi/2-j\sqrt{3}\eta/2} + e^{j\chi/2+j\sqrt{3}\eta/2}\\ &\quad + e^{-j\chi/2+j\sqrt{3}\eta/2} + e^{j\chi/2-j\sqrt{3}\eta/2}) < \chi,\eta\,|\,a(\Omega) >\\ &= \frac{1}{3}\left(\cos\chi + 2\cos\frac{\chi}{2}\cos\frac{\sqrt{3}\eta}{2}\right) < \chi,\eta\,|\,a(\Omega) >\end{aligned}$$

Then

$$\cos\Omega = \frac{1}{3}\left(\cos\chi + 2\cos\frac{\chi}{2}\cos\frac{\sqrt{3}\eta}{2}\right) \qquad (3.23)$$

Using the identities

$$2\cos\theta_1\cos\theta_2 = \cos(\theta_1+\theta_2)+\cos(\theta_1-\theta_2)$$

and

$$\cos\theta = 1-2\sin^2\theta/2$$

$$3\sin^2\frac{\Omega}{2} = \sin^2\frac{\chi}{2}+\sin^2\left(\frac{\chi}{4}-\frac{\sqrt{3}\eta}{4}\right)+\sin^2\left(\frac{\chi}{4}+\frac{\sqrt{3}\eta}{4}\right) \qquad (3.24)$$

or, substituting back for Ω, χ and η, the dispersion relation for TLM on a hexagonal lattice is given by

$$\begin{aligned} 3\sin^2\frac{\omega\Delta t}{2} &= \sin^2\left(\frac{k\Delta l}{2}\cos\phi\right)+\sin^2\left(\frac{k\Delta l}{2}\left(\frac{1}{2}\cos\phi-\frac{\sqrt{3}}{2}\sin\phi\right)\right) \\ &\quad +\sin^2\left(\frac{k\Delta l}{2}\left(\frac{1}{2}\cos\phi+\frac{\sqrt{3}}{2}\sin\phi\right)\right) \qquad (3.25)\end{aligned}$$

This agrees with the result obtained in [18]. Again letting $\Delta l, \Delta t \to 0$, one obtains the two-dimensional dispersion relation for free space (Section 3.1).

3.4 Why a Dispersion Relation Can't Be Found for Partial Streaming

The goal of this theory was to determine the effect of changing the form of the collision operator to be partly a TLM operator and partly an operator which leaves a state unaffected (S_{stream}), *i.e.*

$$S = kS_{TLM}+(1-k)S_{\text{stream}} \qquad (3.26)$$

for use with the rules shown in Chapter 5. When the eigenvalues are solved for this system, they are large and complicated. On further analysis using finite difference techniques, it can be shown that the field becomes non-local for this case [10]. A solution has not yet been found.

Chapter 4

Cellular Automata

By definition,

> A *cellular automaton* ...is a theoretical model of a parallel computer, subject to various restrictions to make a formal investigation of its computing powers tractable. All versions of the model have the following properties: Each is an interconnection of identical cells, where a cell is a model of a computer with finite memory – i.e. a finite-state machine. Each cell computes an output from inputs it receives from a finite set of cells forming its neighbourhood, and possibly from an external source.
>
> All cells compute one output simultaneously and each cell computes an output after each time step. The output of a cell is distributed to its neighbourhood and possibly to an external receiver. [11]

In other words, cellular automata are dynamic systems that are discrete in space, time and dependent variable [12]. This differs from partial differential equations, which have all variables continuous, and finite difference, finite element and TLM schemes, which are discrete in space and time but continuous in dependent variable.

Discretization of space means that a lattice is used and discretization of time that the process occurs in time steps (Δt). Discretization of the dependent variable means that it may be represented by a finite number of bits at a lattice site. This discretization has the advantage of not introducing round-off error into the calculations, as well as eliminating the need for a floating point processor, hence speeding up the calculations.

The dynamics of a cellular automaton are specified by a "rule" that determines what the state of the system will be at time step $t + \Delta t$ based on its state and the state of its neighbours at time t.

4.1 Lattice Gas Models

Lattice gases are special types of cellular automata (CA) for which there is some association between the bits at a site and that site's neighbours on the lattice [12]. These bits may be thought of as "parcels" of matter travelling about the lattice. The systems which have been successfully modeled using Lattice Gas Automata (LGA) are dissipative systems with conserved quantities of the form [12]

$$\frac{\partial \rho_i}{\partial t} + \nabla \cdot J_i = 0$$

The Navier-Stokes equation, which is commonly used in fluid dynamics, falls into this category:

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\nabla p + \nu \nabla^2 \vec{u} \tag{4.1}$$

where $\vec{u}$ is the flow velocity, p is the pressure and ν is the viscosity.

In these models, the state at a node may be described in a bit representation: if a node has one of it's directions occupied, the bit for that direction is turned "on" (set to 1); if it is unoccupied, that bit is turned "off" (set to 0).

Consider a 2-dimensional square lattice. The minimum number of bits required to describe the state of a site is 4 (one per direction). If a site has a direction occupied, the bit representing that direction is turned on (set to 1) otherwise it remains off (set to 0). The bits can be thought of as particles travelling about the mesh. In many simulations it is profitable to allow a site to have more than one particle travelling in any direction, as well as sometimes having rest particles.

4.2 Evolution of a Lattice Gas

A time step is composed of 2 stages. The first is known as the "streaming" (or "advection") phase, in which all particles on the lattice simultaneously move (*i.e.* a north-bound particle moves one step north). The second phase is known as a "collision" phase. It is in this phase that the nature of the simulation is determined. Collisions between particles are governed by a set of rules which determine the model.

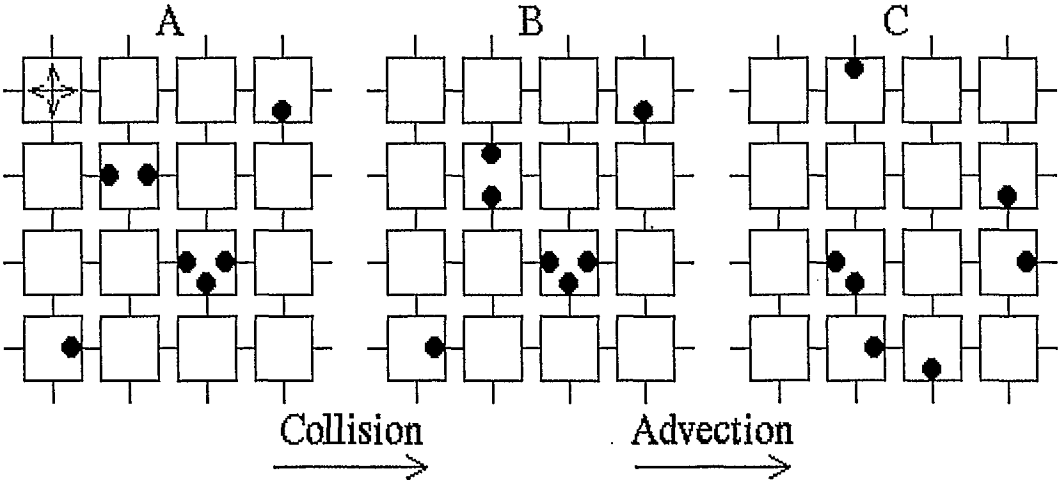


Figure 4.1: The collision and advection stages for the HPP model.

Mass (*i.e.* number of particles) and momentum conservation is a minimum requirement in defining a set of rules (for most systems).

4.2.1 The HPP Model

The Hardy, de Pazzis and Pomeau (HPP) model is the simplest case of a lattice gas automaton [12]. It takes place on a square lattice, therefore having four directions of motion, and has a maximum of one particle per direction per site on the lattice.

The collision rules for the HPP model conserve mass and momentum: a head-on collision of only two particles results in a configuration at right angles to the original travelling away from the center of the site. Figure 4.2 shows the rule.

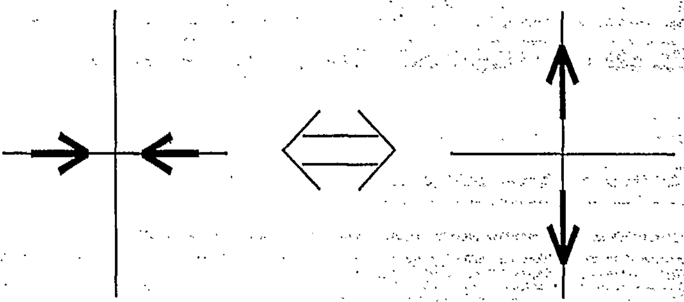


Figure 4.2: The HPP collision rule.

4.2.2 The FHP Model

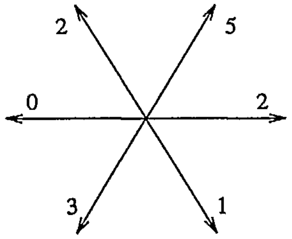
The FHP (Frisch, Hasslacher and Pomeau) models take place on a hexagonal lattice [13]. Each node has 6 neighbours, hence 6 directions to choose from. In the simplest of these models, a maximum of one particle is allowed in any given direction and any collision conserving mass and momentum is allowed.

In some versions, rest particles are allowed, which may also be involved in collisions.

4.3 Integer Lattice Gases

The Integer Lattice Gas Model (ILGA) allows multiple bits to represent the number of particles in a given direction. As an example,

67157 =



The above state, set on a hexagonal lattice site, has allocated 3 bits to represent the number of particles at a site, hence requiring an 18 bit integer to store the state at the site. Each direction may hold any number of particles between 0 and 7: it has 5 particles travelling in direction 1, 2 in direction 2, 1 in direction 3, 3 in direction 4, none in direction 5 and 2 travelling in direction 6. The integer representing this state is, in binary with the least significant bit first,

$$|101|010|100|110|000|010|$$

which has a value of 67157. The maximum integer value that a state can hold is given by $2^{6M}-1$; in this case $M=3$, which gives 262143 for the completely occupied state. A site for this case can take on any value ranging from 0 to 262143.

Chapter 5

Lattice Gas Collision Rules

As stated in Section 4.2, the collision stage of ILGA requires a set of "collision rules". The rules are arbitrary as long as the applicable conservation laws are satisfied; however, the rules involving transmission line matrix theory make more intuitive sense as we are attempting to model an electromagnetic wave and the TLM method is already used for electromagnetic problems.

The sections to follow give the seven rules used in this study to determine the validity of the TLM method as an integer lattice gas collision rule.

5.1 Rule 0

For any given state, if a collision is to take place it must satisfy mass and momentum conservation. This is true for all of the rules; this rule is therefore the fundamental rule:

$$\begin{aligned} \sum_{i=1}^{6} N_i &= \sum_{i=1}^{6} N_i' \\ \vec{p} &= \vec{p'} \end{aligned} \tag{5.1}$$

where N_i is the number of particles present at a site in direction i and $\vec{p}$ is the total momentum of the state. For a given inital state, all other states are searched and those which satisfy mass and momentum conservation are stored. These states then form an "equivalence class", meaning that if initially in a state in this group, the outcome is randomly chosen from the same group. This particular use of an equivalence class also implies that the collision rule satisfies semi-detailed balance. The significance of this is explained in [12].

To use this rule, a set of look-up tables are generated. Groups of states for which their total mass and momentum are the same exist, so when a state within the group undergoes a collision one of the possible outcomes is chosen randomly so as not to bias the outcome. As the states are written to a file, a base index is stored to keep track of which group a state belongs to, as well as the number of states in that group. If the number of states in a group is 1, no collision is possible and only streaming occurs.

5.2 Rule 1

This rule enforces the conservation laws for mass, momentum and "power". The mass and momentum conservation laws are the same as for rule 0. Power conservation is enforced by regarding the particles as packages of voltage and using

$$P = \frac{V^2}{R}$$

With the resistance normalized to 1, power conservation is simply that the sums of the number of particles squared must be the same before and after:

$$\begin{aligned} \sum_{i=1}^{6} N_i &= \sum_{i=1}^{6} N_i' \\ \vec{p} &= \vec{p'} \\ \sum_{i=1}^{6} N_i^2 &= \sum_{i=1}^{6} N_i'^{\,2} \end{aligned} \tag{5.2}$$

The look-up tables are generated in the same manner as in Rule 0. Collisions for which Equations (5.2) are not exactly satisfied are rejected and the states are streamed.

5.3 Rule 2a

Rule 2 makes use of the conventional TLM (transmission line matrix) method used for solving electromagnetic fields (Section 2.2).

Due to notational differences between TLM theory and the lattice gas model, shown in Figure 5.1, the equations must be translated to the language of the lattice gas model since TLM notation shows *position* and LGA notation shows *direction.*

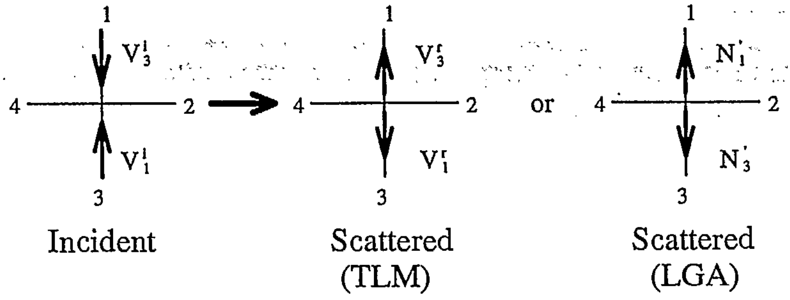


Figure 5.1: Notational difference between TLM and LGA.

What is defined in TLM as V_1^i translates in the language of the lattice gas to N_3, V_1^r translates to N_3', where N_i is the number of particles in direction i, proportional to the voltage in that direction. This results in an equation for the lattice gas TLM rule:

$$\begin{pmatrix} N_1' \\ N_2' \\ N_3' \\ N_4' \end{pmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \\ -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \end{bmatrix} \begin{pmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{pmatrix} \tag{5.3}$$

For a hexagonal lattice, the 6-directional lattice gas TLM rule becomes

$$\begin{pmatrix} N_1' \\ N_2' \\ N_3' \\ N_4' \\ N_5' \\ N_6' \end{pmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 & -2 & 1 & 1 \\ 1 & 1 & 1 & 1 & -2 & 1 \\ 1 & 1 & 1 & 1 & 1 & -2 \\ -2 & 1 & 1 & 1 & 1 & 1 \\ 1 & -2 & 1 & 1 & 1 & 1 \\ 1 & 1 & -2 & 1 & 1 & 1 \end{bmatrix} \begin{pmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \end{pmatrix} \tag{5.4}$$

Note that since this is an integer-based model, truncation error must be accounted for. To do this, the number of particles in each direction is divided by 3, the quotient stored in $[N]_Q$ and the remainder stored in $[N]_R$. The TLM operation is carried out only on those numbers of particles that are divisible by 3 (the quotient) and the remainder is added back on after, essentially resulting in a streaming operation on the remainder bits.

$$[N]^{t+1} = C_{TLM}(\frac{1}{3}[N]^t_Q) + T_S[N]^t_R \tag{5.5}$$

Here T_S is the identity matrix I_6 and C_{TLM} is

$$C_{TLM} = \begin{bmatrix} 1 & 1 & 1 & -2 & 1 & 1 \\ 1 & 1 & 1 & 1 & -2 & 1 \\ 1 & 1 & 1 & 1 & 1 & -2 \\ -2 & 1 & 1 & 1 & 1 & 1 \\ 1 & -2 & 1 & 1 & 1 & 1 \\ 1 & 1 & -2 & 1 & 1 & 1 \end{bmatrix} \tag{5.6}$$

Note that conservation of mass must still be obeyed for this method, no negative numbers of particles are allowed, and this number must not exceed $2^M - 1$. If any of these should occur, the transition is rejected and all particles are streamed.

5.4 Rule 2b

In Rule 2a (5.5), there is a choice in the order of operation on the quotient state $[N]_Q$. There, the state was first split into quotient and remainder, then the quotient was divided by 3 and operated on by C_{TLM}. Rule 2b reverses the order of operation by operating C_{TLM} on the complete state, then splitting the output into quotient and remainder, dividing the quotient by 3 and streaming the remainder:

$$[N]^{t+1} = \frac{1}{3}(C_{TLM}[N]^t)_Q + T_S(C_{TLM}[N]^t)_R \tag{5.7}$$

This rule only works when $(C_{TLM}[N]^t)_R$ is exactly zero, since conservation of mass is violated otherwise. The 1/3 term is a necessary part of the TLM operation to satisfy conservation laws. The rule then becomes

$$[N]^{t+1} = \frac{1}{3}C_{TLM}[N]^t \tag{5.8}$$

Again, any final states violating conservation laws or existing outside of the range of possible values are rejected and the entire state is streamed instead.

5.5 Rule 3a

Rule 3a is variation of rule 2, with the following alteration: if the TLM operation fails, the original state is altered by a fractional amount α and the TLM operator is re-tried.

$$\begin{aligned} [N'] &= \alpha[N] \quad ; \quad \alpha = \frac{k}{2^M - 1}, \quad k = 0, 1, \ldots, 2^M - 1 \\ [N''] &= [N] - [N'] \end{aligned}$$

$[N'']$ is then split into its quotient and remainder parts and the final state becomes

$$[N]^{t+1} = C_{TLM}(\frac{1}{3}[N'']^t_Q) + T_S[N'']^t_R + T_S[N']^t \tag{5.9}$$

If the operation does not satisfy the requirements for any value of α, all particles are streamed for that state.

What this is doing, in essence, can be considered as follows: if the state will not undergo a TLM operation on its quotient part, a small amount is shaved off of the number of particles in all directions, increasing the number of particles streamed and decreasing the overall number in the quotient. The TLM operation is then re-tried. The amount shaved off is increased and the process repeated until a portion of the state is transformed, or until the state is finally entirely streamed.

5.6 Rule 3b

Rule 3b has the same variation on Rule 3a as Rule 2b has on Rule 2a, giving

$$[N]^{t+1} = \frac{1}{3}(C_{TLM}[N'']^t)_Q + T_S(C_{TLM}[N'']^t)_R + T_S[N']^t \tag{5.10}$$

As for Rule 2b, the remainder term $(C_{TLM}[N''])_R$ must be zero to satisfy mass conservation. The rule then becomes

$$[N]^{t+1} = \frac{1}{3}(C_{TLM}[N'']^t)_Q + T_S[N']^t \tag{5.11}$$

Any final states violating conservation laws or existing outside of the range of possible values are rejected and the entire state is streamed instead.

5.7 Rule 3c

This rule maximizes the number of possible TLM transitions by performing the TLM operation on all possible permutations and combinations of multiples of 3.

First these states, containing only multiples of 3, are generated and stored according to the total number of particles in the state. Then, starting with the group which contains the most particles and working down, the TLM operation is performed on each state. If no direction of an output state exceeds the maximum number of particles allowed, remainder particles are added back on sequentially to form both an input state and an output state, such that they still satisfy all conservation laws and bounds. For each input/output combination, the transition containing the most particles is the one accepted. It should be noted that there may be more than one option with the same number of particles involved in the TLM operation; the one obtained first is the one accepted.

Chapter 6

Waveguides

In this chapter the equations of propagation for a 2 dimensional rectangular waveguide with perfect conducting boundaries will be presented, as derived from Maxwell's equations, yielding the critical values for frequency and wavelength [6] for the TE mode case. The equations derived here will be used to compare to test cases.

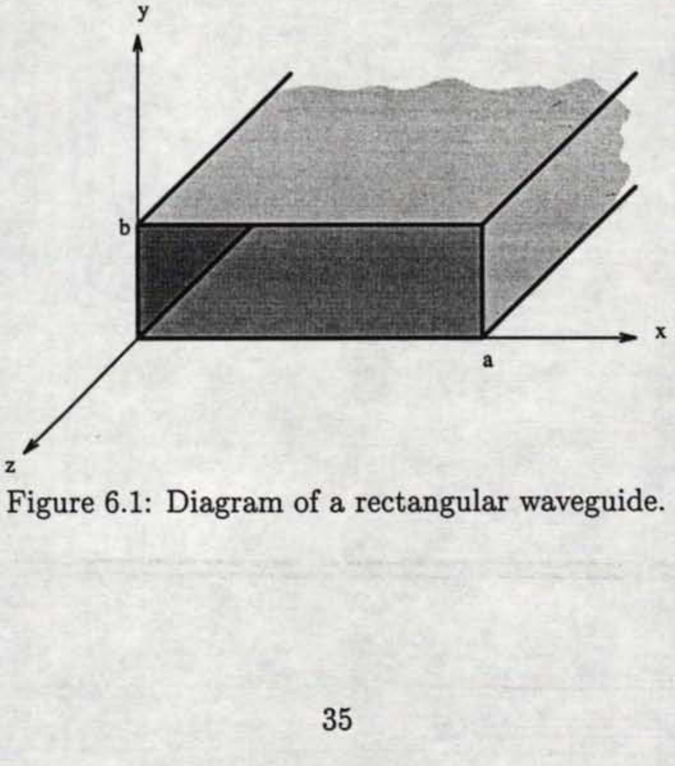


Figure 6.1: Diagram of a rectangular waveguide.

The following set of equations applies to the TE case:

$$
\begin{aligned}
H_x &= \frac{\gamma}{h^2}(CB\sin Bx\cos Ay)\mathrm{e}^{-\gamma z} \\
H_y &= \frac{\gamma}{h^2}(CA\cos Bx\sin Ay)\mathrm{e}^{-\gamma z} \\
H_z &= (C\cos Bx\cos Ay)\mathrm{e}^{-\gamma z} \\
E_x &= j\frac{\omega\mu}{h^2}(CA\cos Bx\sin Ay)\mathrm{e}^{-\gamma z} \\
E_y &= -j\frac{\omega\mu}{h^2}(CB\sin Bx\cos Ay)\mathrm{e}^{-\gamma z} \\
E_z &= 0
\end{aligned}
$$

6.1 Simulation Expectations

This section details what results are expected from the simulation for a square waveguide with dimensions $a \times b$ initialized to a density (field strength) corresponding to various TE modes.,

6.1.1 Frequency

In deriving the TE mode solution, the relationship $h^2 = A^2 + B^2$ was defined. Equating this to the previous value of $h^2 = \gamma^2 + \omega^2\mu\epsilon$ gives the relationship

$$\gamma = \alpha + j\beta = \sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2 - \omega^2\mu\epsilon} \tag{6.1}$$

From this it can be seen that for small ω, γ will be a real number, hence β is zero, resulting in no wave propagation in the z direction. For large ω, γ is purely imaginary and α is zero, allowing the wave to propagate without decay. For perfectly conducting walls α is zero for all frequencies above the cutoff frequency ω_c:

$$\omega_c = \frac{1}{\sqrt{\mu\epsilon}}\sqrt{\left(\frac{m\pi}{a}\right)^2 + \left(\frac{n\pi}{b}\right)^2} \tag{6.2}$$

The cutoff wavelength is

$$\lambda_c = \frac{2}{\sqrt{(m/a)^2 + (n/b)^2}} \tag{6.3}$$

This is the wavelength above which wave propagation cannot occur.

Since wavelength and frequency are related by $f = v/\lambda$ and for these simulations the size of the grid is fixed at $a = b$ and the velocity in 2 dimensions is defined as $v = \Delta l/\sqrt{2}\Delta t$ (Section 3.1), the frequency expected is a value close to

$$f_c = \frac{\sqrt{m^2+n^2}}{2\sqrt{2}a\Delta t} \tag{6.4}$$

with slight variations due to the dispersion (recall that the dispersion cannot be calculated analytically for most of the collision rules used in these simulations).

6.1.2 Viscosity

When modeling electromagnetic waves with this method, the portion of particles streamed due to the 1/3 term in the TLM operator introduces an error in the form of a damping of the wave amplitude. In fluid dynamics, this damping is common and is known to be caused by the presence of viscosity. Obviously, this viscosity term should not be present in our calculations and we therefore want to minimize it.

It can be shown [14] that a solution to the Navier-Stokes equation gives the variation in density of particles with time as

$$\delta\rho \propto \sin(\omega t)e^{k^2\nu t} \tag{6.5}$$

where ν is the average of the shear and bulk viscosities. If the viscosity is zero, we obtain perfect wave behaviour. Hence the validity of the model as applied to electromagnetics is dependent on showing that the viscosity can be eliminated in some manner.

An analytical value for the viscosity can be obtained [15][16] providing that the collision rules obey semi-detailed balance [13]. The TLM-like rules used here do not obey semi-detailed balance, hence the results obtained from simulation cannot be compared to analytical results. The behaviour remains as expected, however, obeying Equation (6.5).

Chapter 7

The Program

This chapter contains a description of the algorithm used for an integer lattice gas on a hexagonal lattice with reflective boundaries. A flow chart is shown in Figure 7.1.

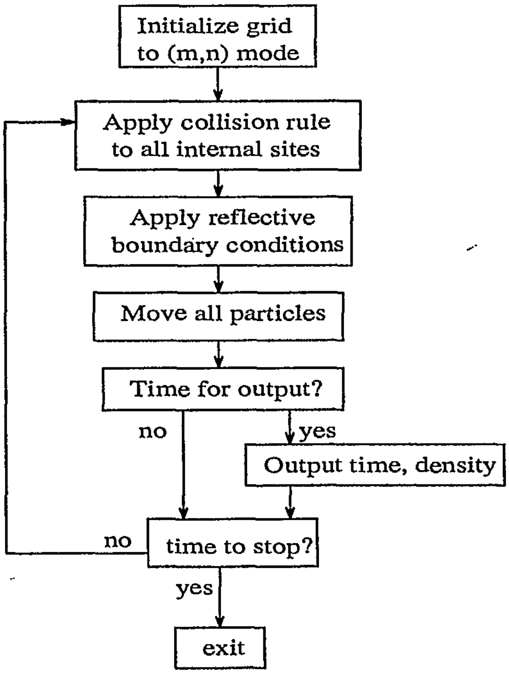

Initialize grid to (m,n) mode
Apply collision rule to all internal sites
Apply reflective boundary conditions
Move all particles
Time for output?
no
yes
Output time, density
no
time to stop?
yes
exit

Figure 7.1: Flow diagram of the algorithm.

7.1 The Hexagonal Lattice

For computational purposes, a hexagonal lattice may still use square lattice coordinates, as shown in Figure 7.2, considering movement from odd and even rows as two different cases.

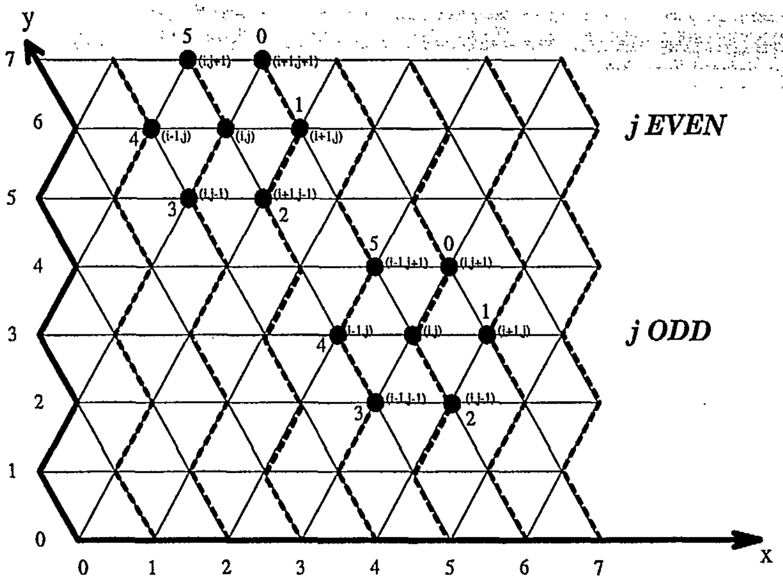


Figure 7.2: The hexagonal lattice.

7.2 Initialization

Initially, the lattice is filled with particles in a binomial distribution. As an example, consider the case in which we desire the lattice to be 50% filled and we have 3 bits per direction to work with. For every bit in the state, a random number between 0 and 1 is generated and if this is greater than 0.5, the bit is filled. Then for each direction, the probability of any number of particles from 0 to 7 is equal at $(\frac{1}{2})^3 = \frac{1}{8}$.

Note that in this case, since we are using reflective boundaries, sites along the boundaries must be skipped since some directions are forbidden for those sites (see Figure 7.2). This must be accounted for to obtain an accurate 50% fill. For M bits per direction on an n×n lattice:

- On the top and bottom rows, excluding corner points, 2 of the 6 directions are forbidden, allowing a maximum of $2 \times 4(n-2)(2^M-1)$ particles on the top and bottom rows.
- On the sides, excluding corner points, a maximum of $2 \times (3(n/2-1)(2^M-1) + 5(n/2-1)(2^M-1))$ particles are allowed in total.
- In the corner points, $2 \times (3(2^M-1) + 2(2^M-1))$ particles are allowed.

The maximum number of particles a lattice can hold is then the sum of these values and the maximum number of particles that the inner sites can hold, $6(n-2)^2(2^M-1)$, giving the desired 50% fill as $(3n-1)(n-1)(2^M-1)$ particles. The boundary sites of the lattice must be filled separately from the inner sites to achieve an accurate 50% fill.

Since we want to look at wave properties, the initialization scheme changes to reflect the mode of the wave desired for study. As shown in Chapter 6, the equation of a TE wave in a (m,n) mode is given by

$$H_z^0(x,y) = C\cos(\frac{m\pi x}{a})\cos(\frac{n\pi y}{b}) \tag{7.1}$$

We want the density of particles at a site to reflect the amplitude of the wave at the (x,y) coordinate. To accomplish this, the probability p of filling a bit is perturbed by approximately 10% of the amplitude of the initial wave. *i.e.* if a uniform random number is larger than

$$p + 0.1\cos\left(\frac{m\pi x}{a}\right)\cos\left(\frac{n\pi y}{b}\right)$$

a bit is "turned on". In these simulations, $p = 0.5$. Note that the model only applies for small perturbations.

7.3 Collision and Advection

For each node on the lattice, excluding the boundaries, the state after collision is determined. Then the boundary sites are checked to ensure that no particles exist in directions that are forbidden. Reflective boundaries are enforced. The particles are then moved. This process is repeated for as many time steps as desired.

7.4 Output

At intervals, the average density is calculated by averaging over a small region of space, about 10% of the grid, in an area near a maximum at initialization. The time step and average density of particles on these sites is output to a file.

Further data which could be obtained at this stage includes distributions of numbers of particles per direction per site and spatial densities.

Chapter 8

Results

8.1 Equilibria

The equilibrium distributions of the number of particles in any direction were found as a function of time, for 3 bits per direction and 100 time steps. To justify the shape of each equilibrium distribution, a binning procedure was used directly on the rule tables, where loss or gain of particles was recorded. For example, if a state originally has 2 directions which contain 5 particles, but its final state has only 1 direction which contains 5 particles, bin[5] will lose 1. If a state originally has 1 direction with 3 particles but winds up with 3 directions with 3 particles, bin[3] will gain 2. These statistics give an idea of the general trend of the rule relating to this type of distribution. Figures 8.1 and 8.2 show the equilibrium distribution for each rule (excluding Rule 3c) and the binned statistics.

	Rule 0	Rule 1	Rule 2a
0	0	0	-6570
1	0	0	618
2	0	0	12894
3	0	0	444
4	0	0	-5706
5	0	0	-9882
6	0	0	12918
7	0	0	-4716

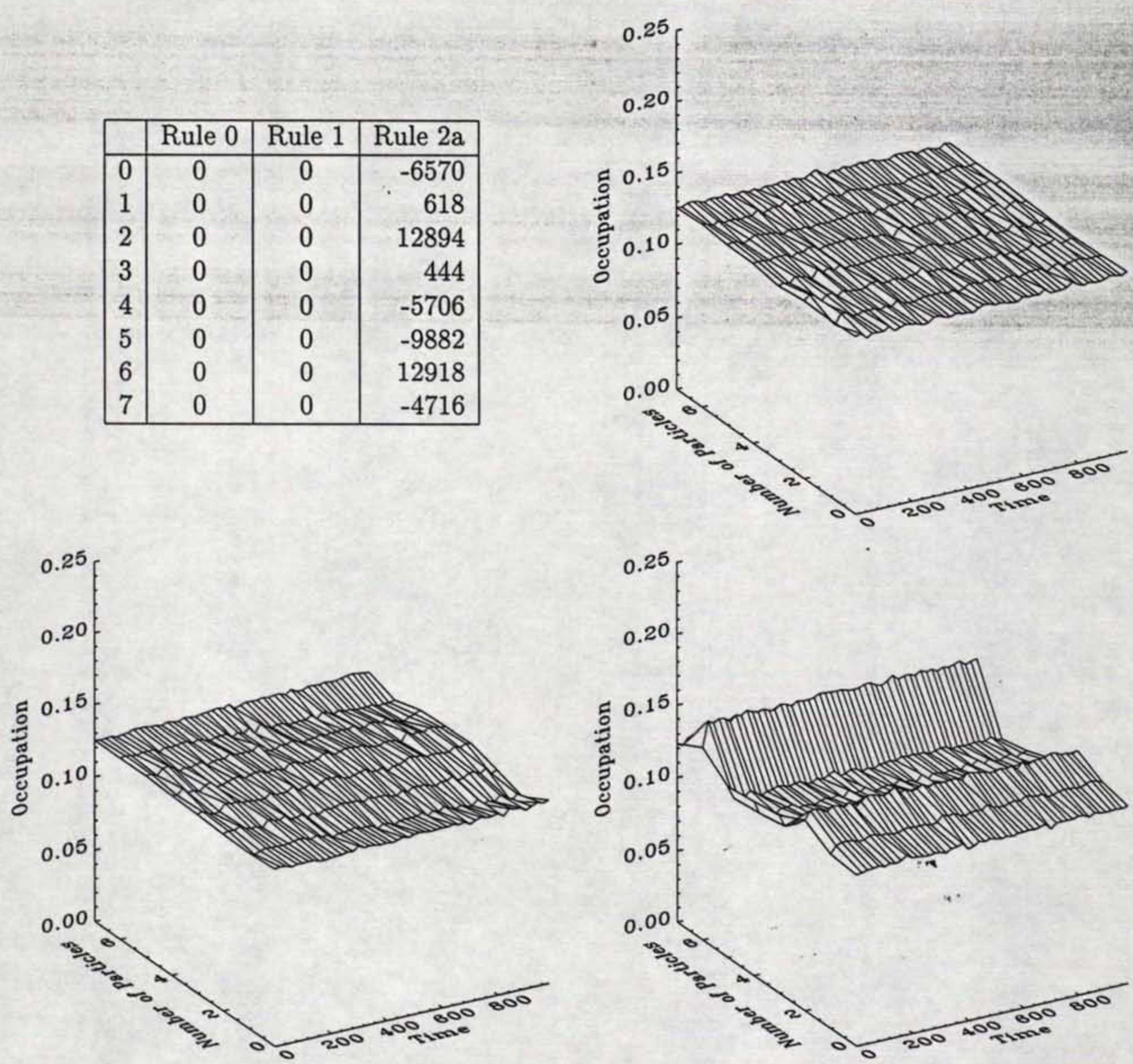
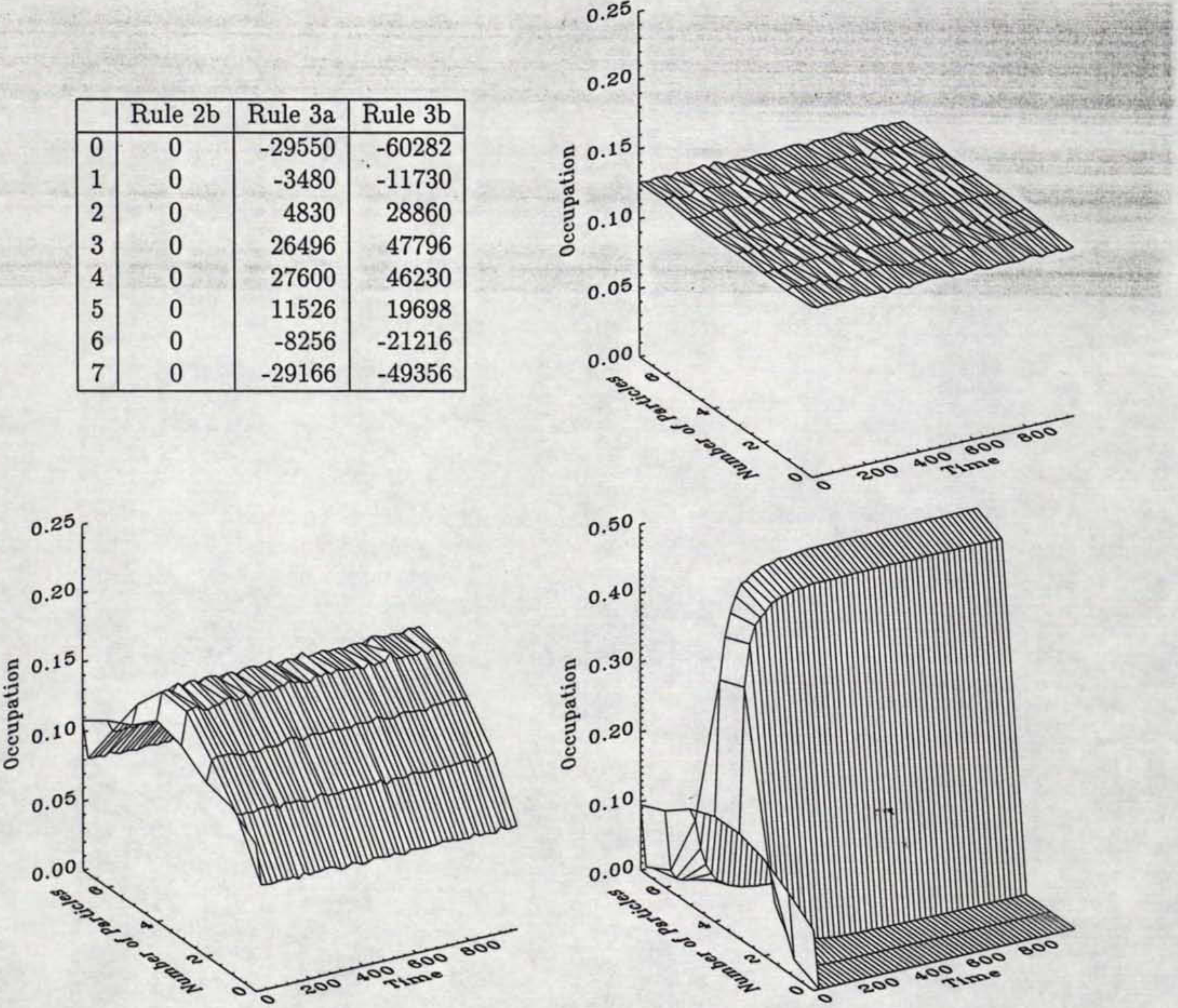


Figure 8.1: The upper right diagram shows the equilibrium distribution of particles for Rule 0, the bottom left for Rule 1, and the bottom right for Rule 2a. The table shows the statistics of expected particles lost or gained, as described in the text. Note how the trends agree.



	Rule 2b	Rule 3a	Rule 3b
0	0	-29550	-60282
1	0	-3480	-11730
2	0	4830	28860
3	0	26496	47796
4	0	27600	46230
5	0	11526	19698
6	0	-8256	-21216
7	0	-29166	-49356

Figure 8.2: The upper right diagram shows the equilibrium distribution of particles for Rule 2b, the bottom left for Rule 3a, and the bottom right for Rule 3b. The table shows the statistics of expected particles lost or gained, as described in the text. Note how the trends agree.

8.2 Rule Statistics

The simulations were carried out using look-up tables for up to 4 bits per direction. From these tables, some statistics about each rule were obtained and are shown in Table 8.1. The "table" values quoted are the values averaged over all states in the table. For example, the total change in energy (power$_{out}$-power$_{in}$) is averaged over all states and does not necessarily reflect the change in energy that would occur in a simulation. These statistics do, however, show the general trends in the rules. The "actual" values are the values averaged over time and space to convergence in a simulation excited to a (1,0) mode. "Weighted" values were obtained by weighting the percentage not streamed and change in energy with the probability of attaining a state, *i.e.* $P(0|3|3|6|4|2) = P(0)P(3)^2P(6)P(4)P(2)$. The individual equilibrium probabilities were found in the previous section.

The weighted values may vary due to fluctuations in the equilibrium distributions, however minor. It would be beneficial to find an analytic equilibrium distribution to generate accurate weighted statistics. This may be found through a transformation of the statistics of net loss or gain of particles. Semi-detailed balance was not calculated for more than 3 bits per direction since the trends are consistent and do not justify the computation time.

Rule 3c does not work for more than 3 bits per direction as the lookup table generation becomes biased to favour some directions of motion over others. Only the results for 2 and 3 bits per direction are presented in this report.

Rule	Bits	% that change	%ns (table)	%ns (weighted)	%ns (actual)	dE (table)	dE (actual)	SDB
0	1	31.25	31.25	17.19	12.93±.03	0	0	Y
	2	90.14	90.14	71.97	69.20±.024	0	.035±.002	Y
	3	99.19	99.19	94.84	94.73±.01	-.000047	-.282±.005	Y
1	1	31.25	31.25	17.19	12.93±.03	0	0	Y
	2	68.41	68.41	42.02	34.99±.04	0	0	Y
	3	86.36	86.36	61.95	57.71±.03	0	0	Y
2a	2	17.43	7.74	7.74	6.864±.008	-.0938	-.034±.001	N
	3	42.38	30.65	28.25	28.44±.02	-.18	-.131±.005	N
	4	51.35	46.91	36.14	38.48±.02	-.41	-2.01±.02	—
	5	54.04	50.46	45.50	41.7±.7	-.2737	.0028±.0007	—
2b	2	18.75	18.75	20.88	18.18±.02	0	0	Y
	3	18.46	18.46	18.43	17.48±.02	0	0	Y
	4	18.37	18.37	18.70	17.38±.02	0	0	—
3a	2	17.43	7.74	7.74	6.864±.008	-.0938	-.034±.001	N
	3	80.00	48.70	55.44	51.58±.02	-2.83	-.351±.005	N
	4	95.44	73.23	83.55	81.376±.007	-13.33	-2.19±.01	—
	5	98.49	98.44	98.00	92.62±.14	-9.63	-.0004±.0003	—
3b	2	47.02	39.88	19.31	18.2±.6	-.80	-.000031±.000002	N
	3	83.70	68.82	23.23	17.6±.7	-5.4	-.00047±.00003	N
	4	94.95	79.47	26.41	16.1±.6	-19.8	-.0020±.0001	—
3c	2	25.42	16.25	—	—	-1.44	—	N
	3	84.49	62.66	—	—	-8.16	—	N

Table 8.1: Rule table statistics. *% that change* is the number of states that undergo a TLM-like collision. *%ns* is the percentage of particles that undergo a TLM-like collision. *table* denotes the statistics drawn directly from the table. *actual* denotes statistics taken from within a simulation excited to a (1,0) mode. *weighted* denotes a weighted average based on the equilibrium distributions shown in the previous section. *dE* is the average ΔP. *SDB* shows whether the rule table satisfies semi-detailed balance. The statistics for Rule 3c were not completed since the table generator was faulty.

8.3 Noise

For each of the 7 rules, 5 simulations of different seeds for the Numerical Recipes "ran3" random number generator [17] were carried out on a 64×64 hexagonal lattice initialized by the four modes (m,n) = (0,1), (1,0), (1,1) and (2,1). These were repeated for up to four bits per direction, where memory allowed. Figure 8.3 shows the five trials of density versus time step for a (1,0) mode using Rule 0 (mass and momentum conservation only) with 3 bits per direction.

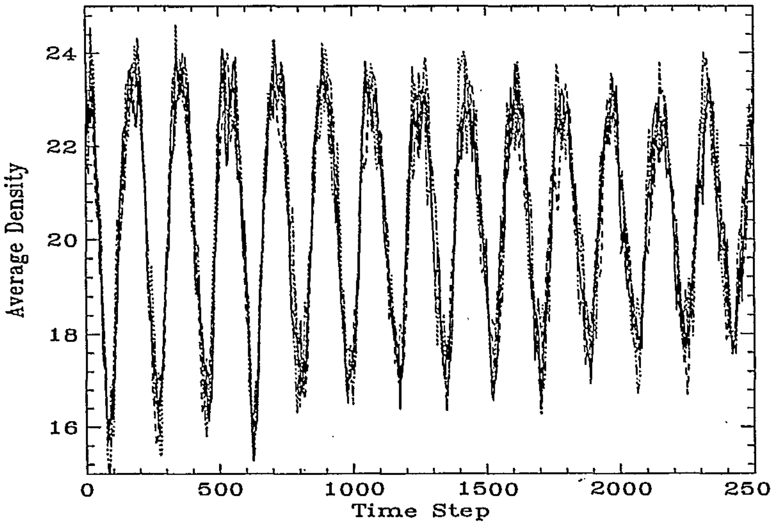


Figure 8.3: 5 trials for Rule 0, 3 bits per direction, initialized to a (1,0) mode. The y-axis is the average density in the sample space.

The results are noisy due to the small grid used; an increase in grid size to 150×150 smooths out the curves considerably but requires twice the computation time.

8.4 Curve Fits

To obtain a viscosity from these curves, an average of the five data sets was taken and the result fit to

$$y(t) = A\sin(\omega t + \phi)\mathrm{e}^{-Bt} + C$$

using the Marquardt-Levenberg subroutines from Numerical Recipes [17]. The error on the each data point was taken as the standard deviation or 0.5, whichever was larger. Tables 8.2, 8.3, 8.4 and 8.5 show the fitting parameters for each mode, rule and number of bits per direction.

Rule	Bits	A	ω	B	C
0	1	0.48 ± 0.08	0.0402 ± 0.0002	0.0007 ± 0.0002	3.24 ± 0.01
	2	1.58 ± 0.06	0.04056 ± 0.00002	0.00029 ± 0.00002	8.98 ± 0.01
	3	3.82 ± 0.06	0.040917 ± 0.000006	0.000253 ± 0.000006	20.48 ± 0.01
1	2	1.57 ± 0.08	0.04116 ± 0.00005	0.00065 ± 0.00005	9.51 ± 0.01
	3	3.62 ± 0.08	0.04099 ± 0.00002	0.00052 ± 0.00002	21.35 ± 0.01
2a	2	1.61 ± 0.08	0.04108 ± 0.00004	0.00063 ± 0.00004	8.738 ± 0.01
	3	3.78 ± 0.07	0.04098 ± 0.00001	0.00038 ± 0.00001	20.45 ± 0.01
	4	8.21 ± 0.09	0.041008 ± 0.000007	0.000501 ± 0.000008	43.90 ± 0.01
2b	2	1.67 ± 0.09	0.04108 ± 0.00006	0.00083 ± 0.00007	8.74 ± 0.01
	3	3.8 ± 0.1	0.04114 ± 0.00006	0.00121 ± 0.00006	20.45 ± 0.01
	4	8.0 ± 0.2	0.04124 ± 0.00005	0.00138 ± 0.00005	43.76 ± 0.02
3a	2	1.61 ± 0.08	0.04108 ± 0.00004	0.00063 ± 0.00004	8.74 ± 0.01
	3	3.77 ± 0.04	0.040967 ± 0.000002	0.000063 ± 0.000002	20.44 ± 0.01
	4	8.37 ± 0.06	0.040967 ± 0.000001	0.000030 ± 0.000001	43.74 ± 0.02
3b	2	1.65 ± 0.02	0.040872 ± 0.000005	0.000381 ± 0.000004	8.984 ± 0.002
	3	3.75 ± 0.02	0.040438 ± 0.000005	0.000533 ± 0.000004	20.990 ± 0.003
	4	13.63 ± 0.07	0.040840 ± 0.000002	0.000464 ± 0.000002	44.899 ± 0.006
3c	2	1.64 ± 0.07	0.04104 ± 0.00002	0.00041 ± 0.00002	9.07 ± 0.01
	3	3.37 ± 0.04	0.040217 ± 0.000003	0.000064 ± 0.000003	20.41 ± 0.01

Table 8.2: Fit parameters for all rules for the (0,1) mode.

The fitted curves (solid lines) for each rule are shown in Figures 8.4–8.10, using the (1,0) mode and 3 bits per direction. The raw data shown (dashed lines) is the average over the 5 data sets obtained for different random number generator seeds. The fits are reasonably good for most of the rules. Rule 3b

Rule	Bits	A	ω	B	C
0	1	0.42 ± 0.04	0.03630 ± 0.00002	0.00007 ± 0.00002	3.24 ± 0.01
	2	1.50 ± 0.04	0.035632 ± 0.000007	0.000118 ± 0.000007	9.01 ± 0.01
	3	3.74 ± 0.05	0.035060 ± 0.000005	0.000205 ± 0.000005	20.46 ± 0.01
1	2	1.39 ± 0.04	0.035604 ± 0.000009	0.000161 ± 0.000009	9.53 ± 0.01
	3	3.55 ± 0.06	0.035331 ± 0.000006	0.000256 ± 0.000006	21.32 ± 0.01
2a	2	1.60 ± 0.07	0.03499 ± 0.00003	0.00048 ± 0.00003	8.76 ± 0.01
	3	3.74 ± 0.06	0.034977 ± 0.000007	0.000296 ± 0.000007	20.42 ± 0.01
	4	7.98 ± 0.07	0.034969 ± 0.000004	0.000338 ± 0.000004	43.89 ± 0.01
2b	2	1.69 ± 0.09	0.03497 ± 0.00006	0.00076 ± 0.00006	8.77 ± 0.01
	3	3.8 ± 0.1	0.03495 ± 0.00004	0.00101 ± 0.00004	20.42 ± 0.01
	4	7.8 ± 0.2	0.03493 ± 0.00004	0.00107 ± 0.00004	43.78 ± 0.02
3a	2	1.60 ± 0.07	0.03499 ± 0.00003	0.00048 ± 0.00003	8.76 ± 0.01
	3	3.72 ± 0.04	0.034968 ± 0.000002	0.000049 ± 0.000002	20.42 ± 0.01
	4	8.13 ± 0.05	0.034979 ± 0.000001	0.000029 ± 0.000001	43.80 ± 0.02
3b	2	1.48 ± 0.01	0.035026 ± 0.000004	0.000296 ± 0.000003	9.042 ± 0.002
	3	3.19 ± 0.02	0.035191 ± 0.000004	0.000449 ± 0.000004	21.053 ± 0.003
	4	12.14 ± 0.07	0.035129 ± 0.000003	0.000512 ± 0.000003	45.266 ± 0.005
3c	2	1.56 ± 0.06	0.03499 ± 0.00002	0.00033 ± 0.00002	9.10 ± 0.01
	3	3.57 ± 0.04	0.035253 ± 0.000002	0.000049 ± 0.000002	20.38 ± 0.01

Table 8.3: Fit parameters for all rules for the (1,0) mode.

Rule	Bits	A	ω	B	C
0	1	0.18 ± 0.08	0.0538 ± 0.0004	0.0007 ± 0.0004	3.23 ± 0.01
	2	1.23 ± 0.07	0.05382 ± 0.00004	0.00046 ± 0.00004	8.97 ± 0.01
	3	3.49 ± 0.07	0.05389 ± 0.00001	0.00039 ± 0.00001	20.40 ± 0.02
1	2	0.8 ± 0.1	0.0543 ± 0.0002	0.0010 ± 0.0002	9.50 ± 0.01
	3	2.7 ± 0.1	0.05413 ± 0.00004	0.00081 ± 0.00004	21.25 ± 0.01
2a	2	1.5 ± 0.1	0.0539 ± 0.0001	0.0010 ± 0.0001	8.73 ± 0.01
	3	3.49 ± 0.09	0.05386 ± 0.00002	0.00057 ± 0.00002	20.35 ± 0.01
	4	8.0 ± 0.1	0.05386 ± 0.00002	0.00099 ± 0.00002	43.74 ± 0.01
2b	2	1.4 ± 0.1	0.0540 ± 0.0001	0.0013 ± 0.0002	8.73 ± 0.01
	3	3.7 ± 0.2	0.0544 ± 0.0002	0.0023 ± 0.0002	20.36 ± 0.01
	4	8.2 ± 0.3	0.0541 ± 0.0001	0.0025 ± 0.0001	43.62 ± 0.02
3a	2	1.5 ± 0.1	0.0539 ± 0.0001	0.0010 ± 0.0001	8.73 ± 0.01
	3	3.50 ± 0.04	0.053860 ± 0.000003	0.000098 ± 0.000003	20.36 ± 0.01
	4	7.71 ± 0.05	0.053857 ± 0.000001	0.000027 ± 0.000001	43.59 ± 0.02
3b	2	1.45 ± 0.02	0.05386 ± 0.00001	0.00063 ± 0.00001	8.969 ± 0.002
	3	3.50 ± 0.03	0.053827 ± 0.000006	0.000636 ± 0.000005	20.899 ± 0.003
	4	10.66 ± 0.08	0.053761 ± 0.000003	0.000587 ± 0.000004	44.795 ± 0.005
3c	2	1.58 ± 0.09	0.05392 ± 0.00006	0.00073 ± 0.00006	9.06 ± 0.01
	3	3.51 ± 0.04	0.053663 ± 0.000003	0.000100 ± 0.000003	20.33 ± 0.01

Table 8.4: Fit parameters for the (1,1) mode.

Rule	Bits	A	ω	B	C
0	1	0.35 ± 0.14	0.081 ± 0.001	0.002 ± 0.001	3.23 ± 0.01
	2	1.2 ± 0.1	0.0811 ± 0.0002	0.0012 ± 0.0002	8.96 ± 0.01
	3	2.4 ± 0.1	0.08089 ± 0.00006	0.00092 ± 0.00006	20.41 ± 0.01
1	2	1.0 ± 0.1	0.0799 ± 0.0003	0.0019 ± 0.0004	9.49 ± 0.01
	3	2.5 ± 0.2	0.0806 ± 0.0002	0.0019 ± 0.0002	21.25 ± 0.01
2a	2	0.9 ± 0.1	0.0808 ± 0.0004	0.0018 ± 0.0004	8.71 ± 0.01
	3	2.5 ± 0.1	0.0805 ± 0.0001	0.0014 ± 0.0001	20.37 ± 0.01
	4	4.7 ± 0.2	0.08056 ± 0.00007	0.00155 ± 0.00007	43.83 ± 0.01
2b	2	1.0 ± 0.2	0.0816 ± 0.0008	0.0030 ± 0.0008	8.71 ± 0.01
	3	1.9 ± 0.2	0.0818 ± 0.0006	0.0037 ± 0.0006	20.36 ± 0.01
	4	4.8 ± 0.5	0.0826 ± 0.0005	0.0043 ± 0.0005	43.67 ± 0.02
3a	2	0.9 ± 0.1	0.0808 ± 0.0004	0.0018 ± 0.0004	8.71 ± 0.01
	3	2.64 ± 0.05	0.080928 ± 0.000006	0.000199 ± 0.000006	20.37 ± 0.01
	4	6.40 ± 0.06	0.080940 ± 0.000002	0.000069 ± 0.000002	43.66 ± 0.02
3b	2	1.27 ± 0.04	0.08105 ± 0.00005	0.00138 ± 0.00005	8.9560 ± 0.0002
	3	3.06 ± 0.05	0.08154 ± 0.00002	0.00141 ± 0.00002	20.928 ± 0.003
	4	9.5 ± 0.1	0.08131 ± 0.00002	0.00125 ± 0.00002	44.829 ± 0.006
3c	2	0.9 ± 0.1	0.0809 ± 0.0002	0.0012 ± 0.0002	9.05 ± 0.01
	3	3.06 ± 0.07	0.08116 ± 0.00002	0.00048 ± 0.00002	20.34 ± 0.01

Table 8.5: Fit parameters for the (2,1) mode.

does not fit well for it's first 1000 steps, likely due to the relatively long time it takes to reach it's equilibrium distribution.

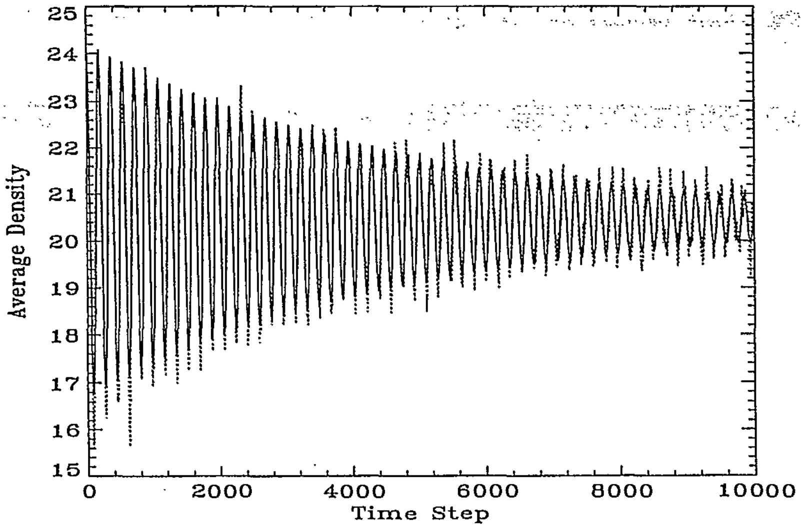


Figure 8.4: Averaged data and fitted curve for Rule 0, 3 bits per direction, initialized to a (1,0) mode.

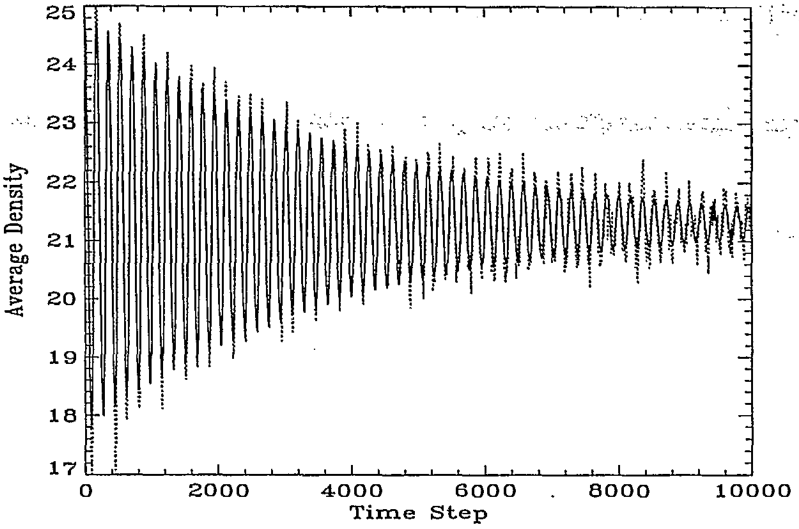


Figure 8.5: Averaged data and fitted curve for Rule 1, 3 bits per direction, initialized to a (1,0) mode.

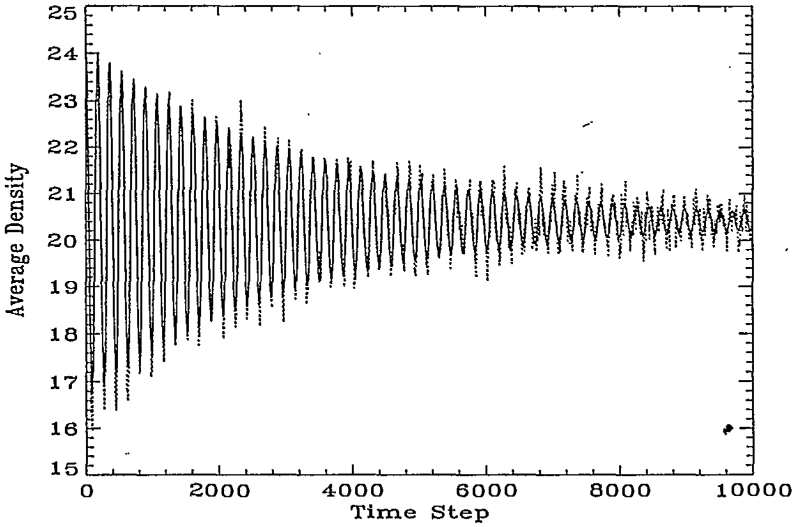


Figure 8.6: Averaged data and fitted curve for Rule 2a, 3 bits per direction, initialized to a (1,0) mode.

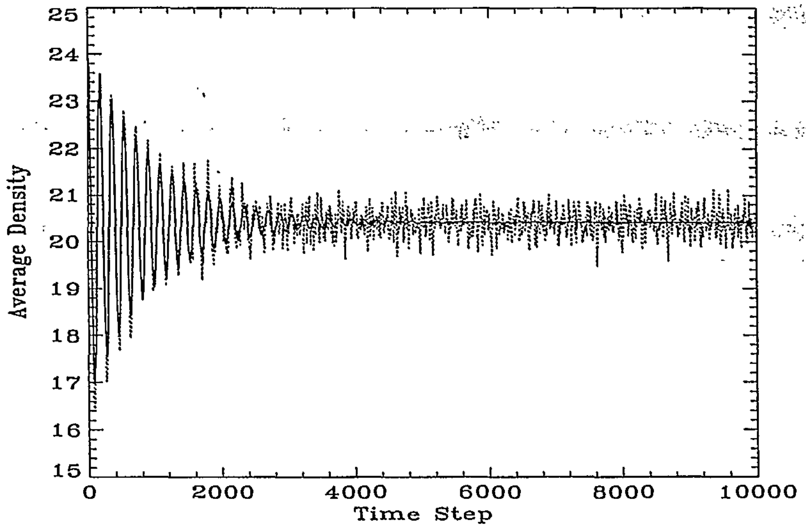


Figure 8.7: Averaged data and fitted curve for Rule 2b, 3 bits per direction, initialized to a (1,0) mode.

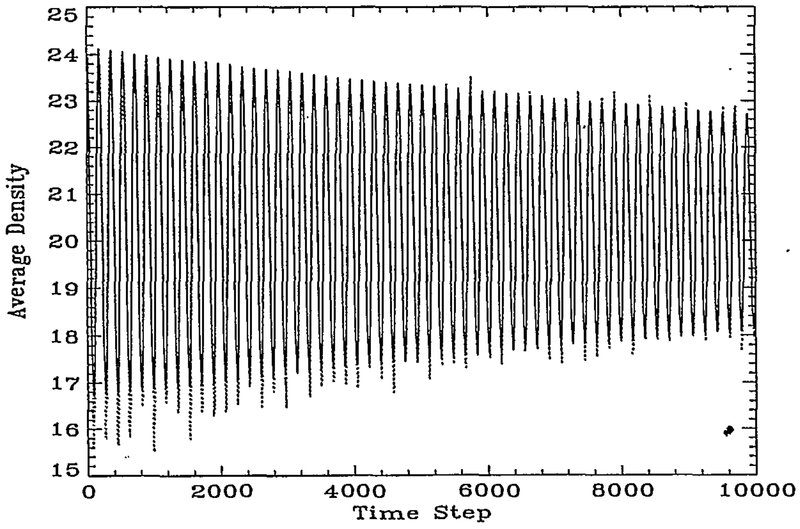


Figure 8.8: Averaged data and fitted curve for Rule 3a, 3 bits per direction, initialized to a (1,0) mode.

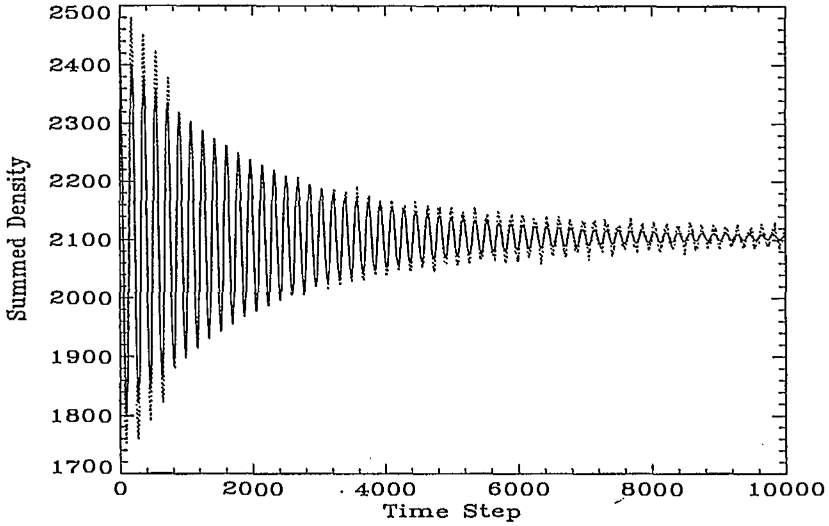


Figure 8.9: Averaged data and fitted curve for Rule 3b, 3 bits per direction, initialized to a (1,0) mode. Note that the y-axis here represents the summed density over all sites in the sample space.

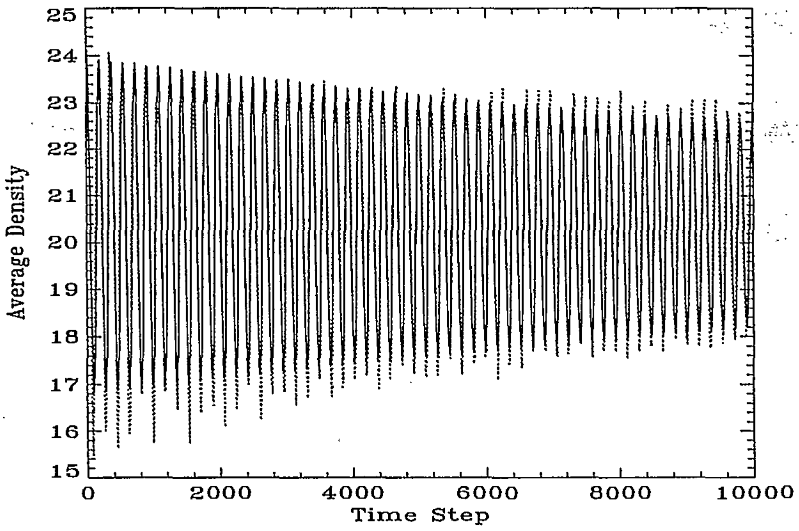


Figure 8.10: Averaged data and fitted curve for Rule 3c, 3 bits per direction, initialized to a (1,0) mode.

8.5 Modes

The results shown in Figures 8.11–8.14 are for Rule 3a, 3 bits per direction. They contain the averaged raw data and the fitted curve for density versus time step.

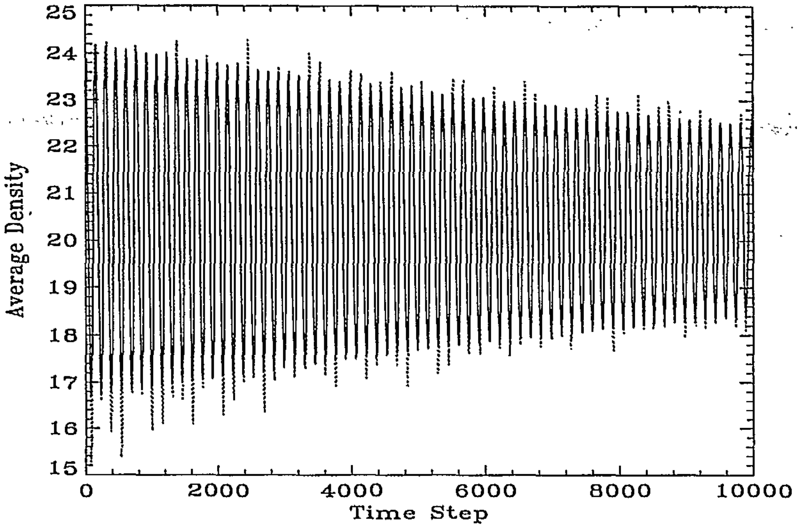


Figure 8.11: Averaged data and fitted curve for Rule 3a, 3 bits per direction, initialized to a (0,1) mode.

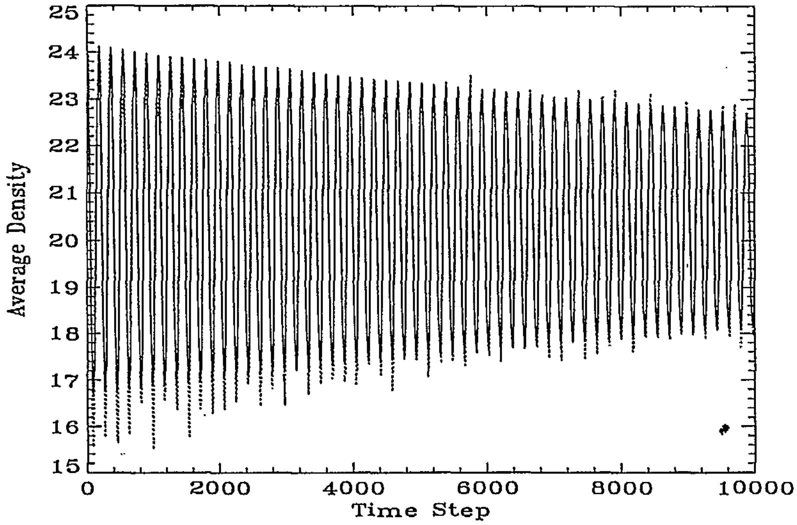


Figure 8.12: Averaged data and fitted curve for Rule 3a, 3 bits per direction, initialized to a (1,0) mode.

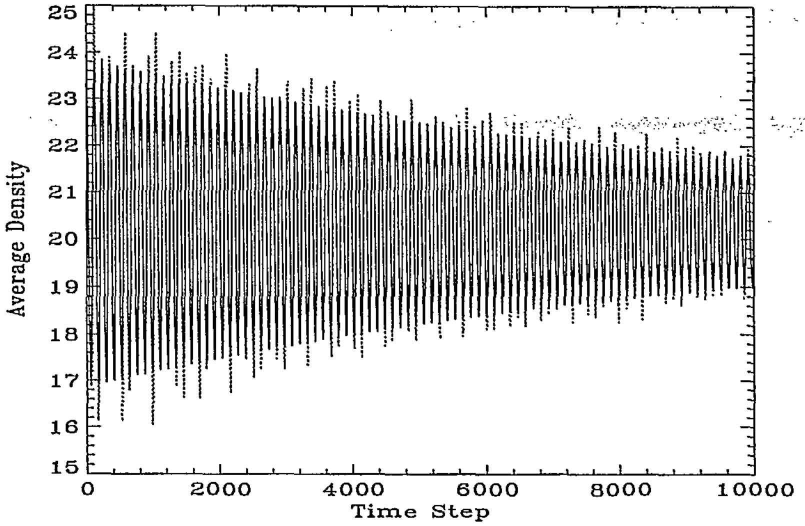


Figure 8.13: Averaged data and fitted curve for Rule 3a, 3 bits per direction, initialized to a (1,1) mode.

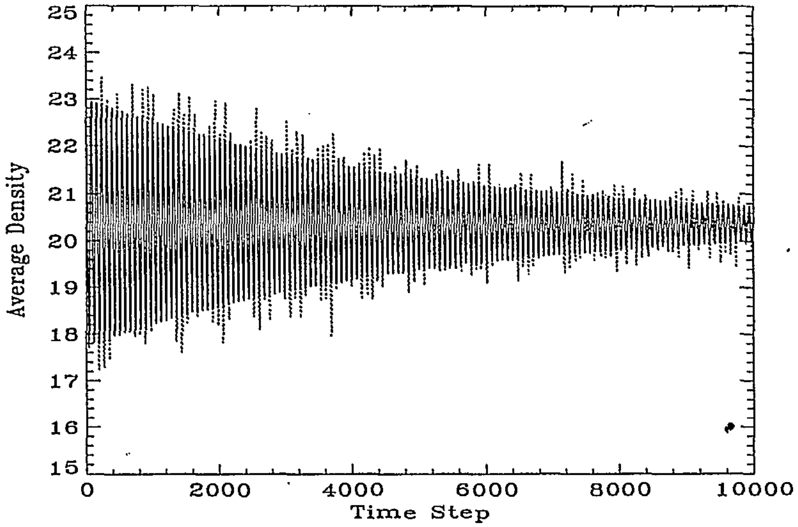


Figure 8.14: Averaged data and fitted curve for Rule 3a, 3 bits per direction, initialized to a (2,1) mode.

8.6 Bits

The 4 plots to follow show the averaged data for Rule 3a in a (1,0) mode for 2, 3, 4 and 5 bits per direction, Figures 8.15, 8.16, 8.17 and 8.18, respectively. As expected from percent of particles streamed in the table of look-up table statistics (Table 8.1), the damping decreases as the number of bits increases.

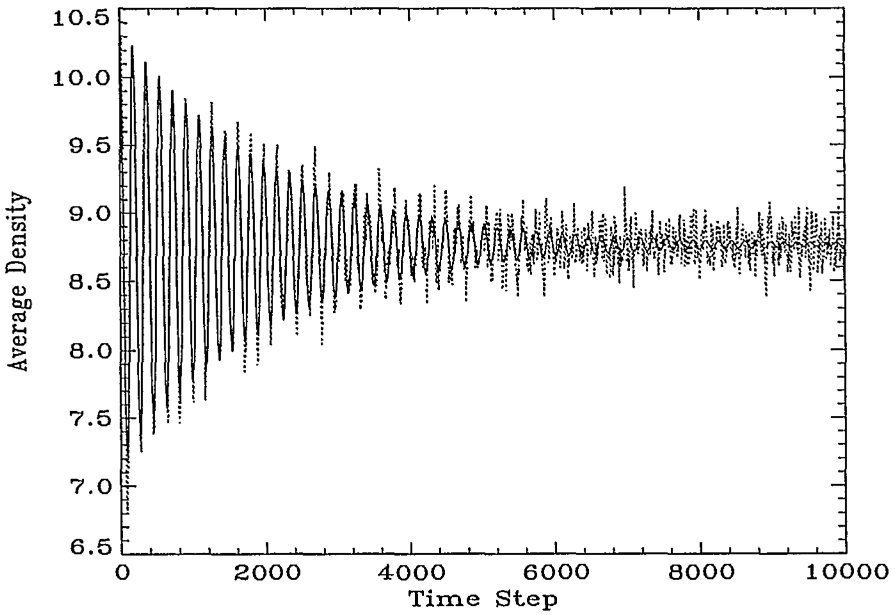


Figure 8.15: Averaged data and fitted curve for Rule 3a, 2 bits per direction, initialized to a (1,0) mode.

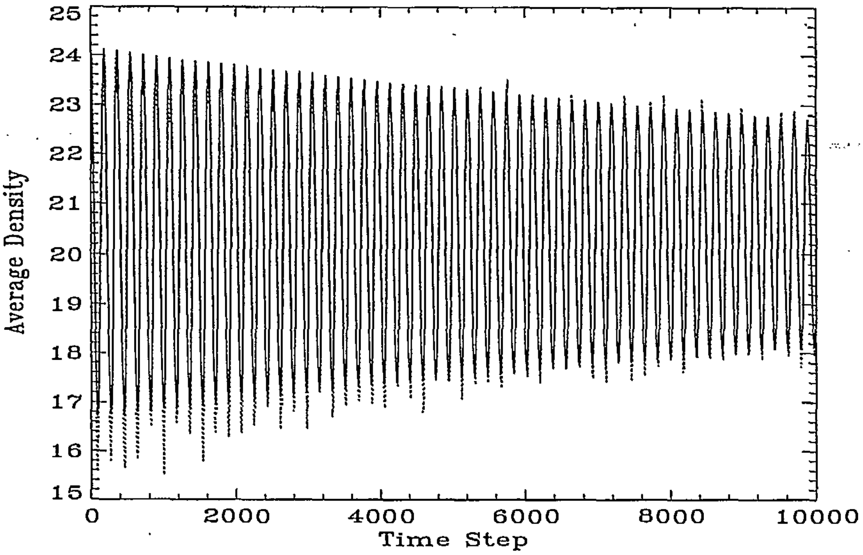


Figure 8.16: Averaged data and fitted curve for Rule 3a, 3 bits per direction, initialized to a (1,0) mode.

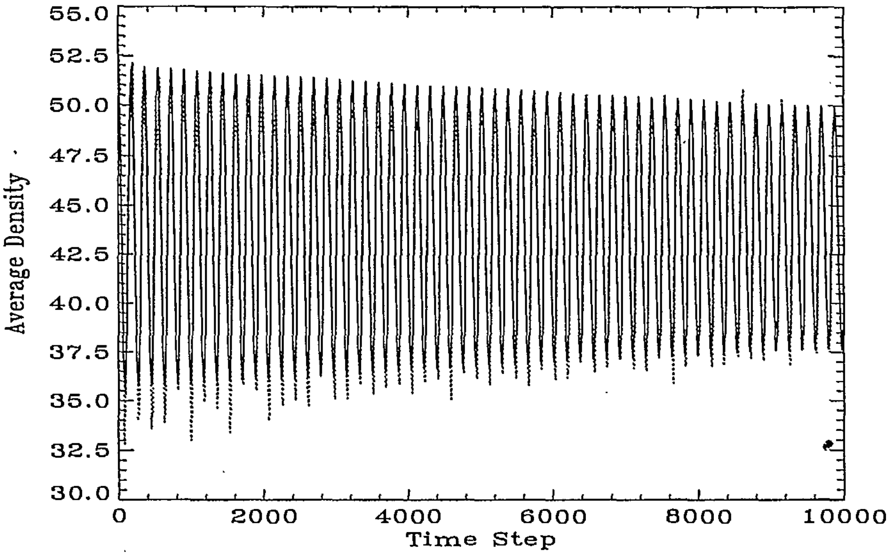


Figure 8.17: Averaged data and fitted curve for Rule 3a, 4 bits per direction, initialized to a (1,0) mode.

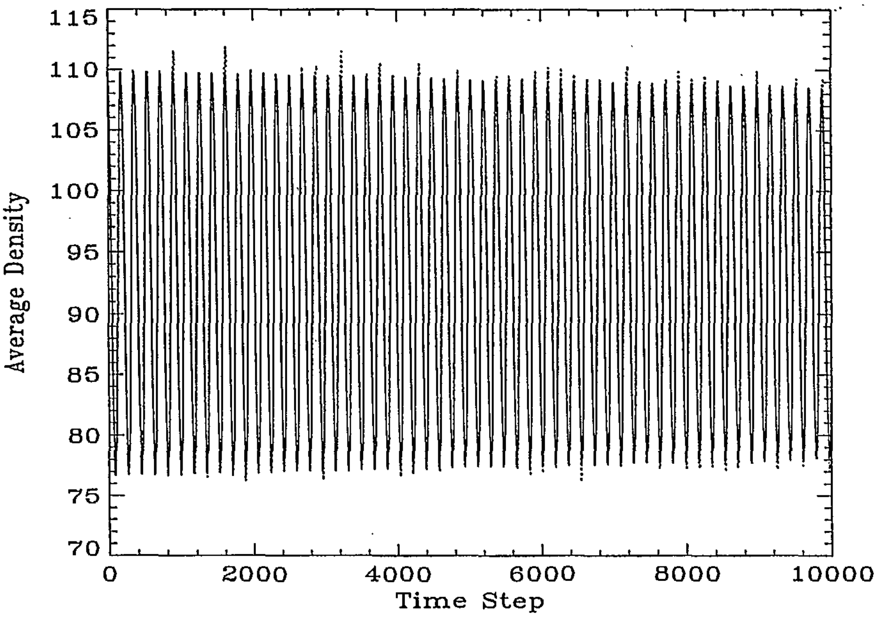


Figure 8.18: Averaged data and fitted curve for Rule 3a, 5 bits per direction, initialized to a (1,0) mode.

8.7 Viscosity

Figures 8.19, 8.20, 8.21 and 8.22 show the viscosities calculated for each rule for the (0,1), (1,0), (1,1) and (2,1) modes respectively. The values are tabulated in Table 8.6. Note that for all of the modes, Rule 3a consistently shows the lowest viscosity and Rule 2b the highest.

Rule	Bits	ν_{01}	ν_{10}	ν_{11}	ν_{21}
0	1	0.21±0.05	0.029±0.008	0.12±0.08	0.15±0.08
	2	0.091±0.005	0.049±0.003	0.082±0.006	0.09±0.01
	3	0.079±0.002	0.085±0.002	0.068±0.002	0.071±0.004
1	1	0.21±0.05	0.029±0.008	0.12±0.08	0.15±0.08
	2	0.20±0.02	0.067±0.004	0.18±0.03	0.14±0.03
	3	0.161±0.005	0.106±0.003	0.143± 0.007	0.14 ± 0.01
2a	2	0.20 ± 0.01	0.20± 0.01	0.18± 0.02	0.14 ± 0.03
	3	0.120 ± 0.003	0.123±0.003	0.102± 0.003	0.108 ± 0.008
	4	0.156 ± 0.002	0.140±0.002	0.175± 0.004	0.120 ±0.006
	5	N/A	0.1369±0.0022	N/A	N/A
2b	2	0.26 ± 0.02	0.32 ± 0.02	0.23 ± 0.03	0.24 ± 0.06
	3	0.38 ± 0.02	0.42 ± 0.02	0.41 ± 0.03	0.29 ± 0.05
	4	0.43±0.02	0.44± 0.02	0.44 ± 0.02	0.33 ± 0.04
3a	2	0.20±0.01	0.20±0.02	0.18±0.02	0.14± 0.03
	3	0.0197±0.0007	0.0204±0.0009	0.0175±0.0005	0.0154±0.0005
	4	0.0093±0.0004	0.0122±0.0005	0.0048±0.0002	0.0054±0.0002
	5	N/A	0.0036±0.0002	N/A	N/A
3b	2	0.118±0.001	0.123±0.001	0.112±0.002	0.107 ±0.004
	3	0.186 ± 0.005	0.286±0.009	0.193±0.007	0.14 ± 0.01
	4	0.1444 ±0.0007	0.213±0.001	0.1044±0.0006	0.097±0.001
3c	2	0.129±0.007	0.137±0.008	0.15±0.01	0.11±0.02
	3	0.0199±0.0009	0.0205±0.0009	0.0208±0.0006	0.040±0.001

Table 8.6: Viscosity for all rules and all modes. (N/A ≡ not attempted)

Results were obtained for Rule 3a, 3 bits per direction for (2,0) and (4,2) modes for comparison to the (1,0) and (2,1) modes, respectively, on a 128×128 lattice. For these cases, $\nu_{20} = 0.024 \pm 0.001$ and $\nu_{42} = 0.017 \pm 0.002$, showing good agreement with the (1,0) amd (2,1) results of $\nu_{10} = 0.0204 \pm 0.0009$ and $\nu_{21} = 0.0174 \pm 0.0007$. A (3,0) mode gives $\nu_{30} = 0.026 \pm 0.007$. Note that the error increases considerably for higher modes.

There is some small variation in viscosity with angle of propagation. These results show that the viscosity is anisotropic.

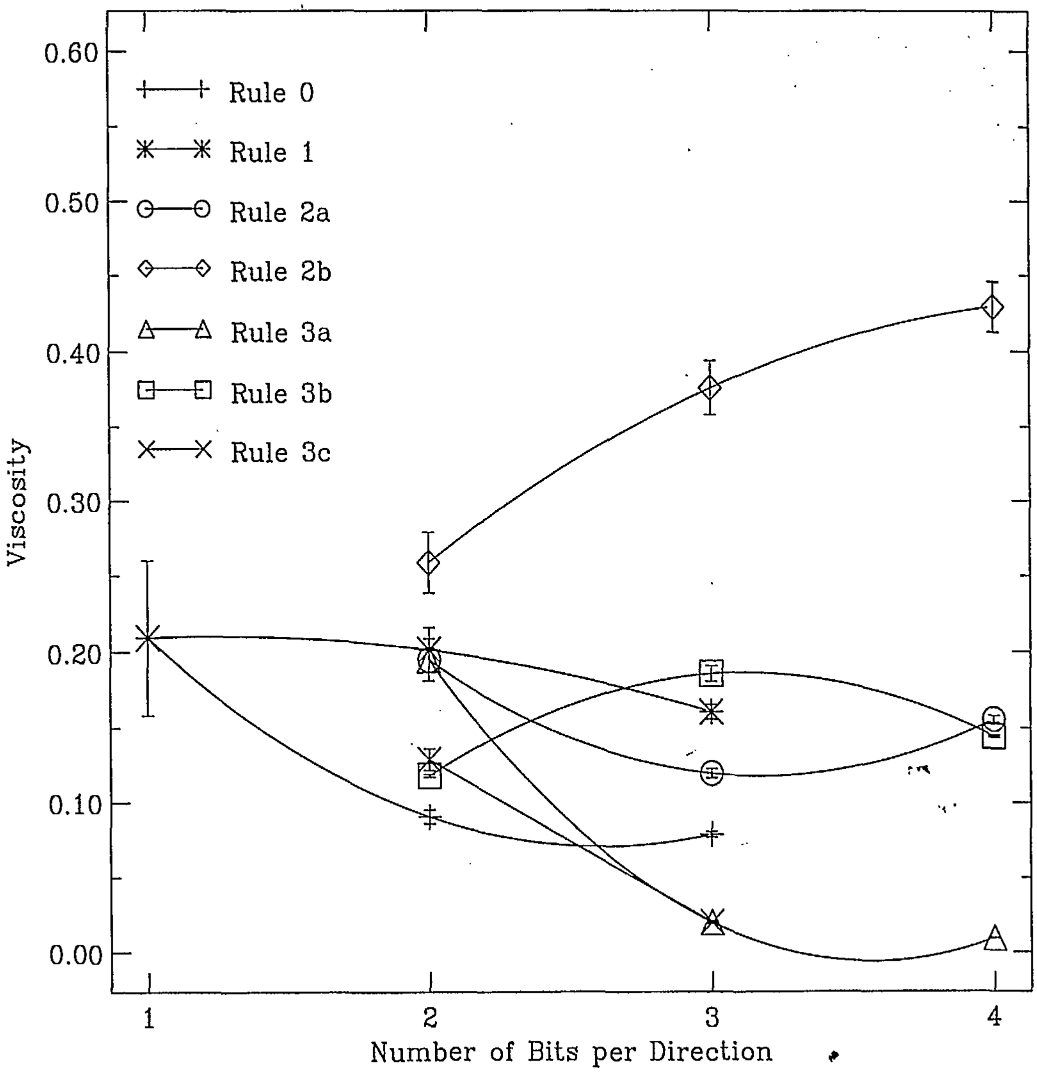


Figure 8.19: Viscosity as a function of number of bits per direction for the seven rules for a (0,1) mode.

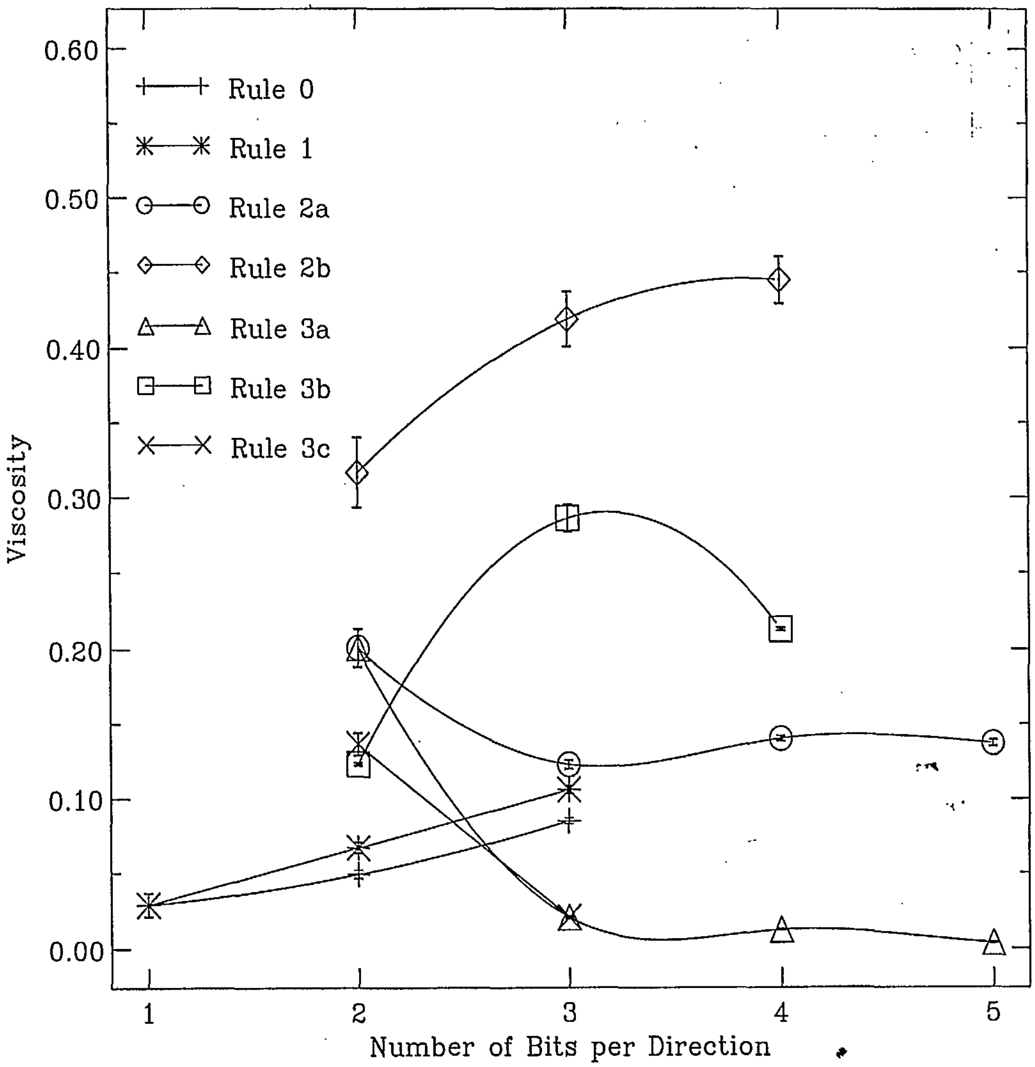


Figure 8.20: Viscosity as a function of number of bits per direction for the seven rules for a (1,0) mode.

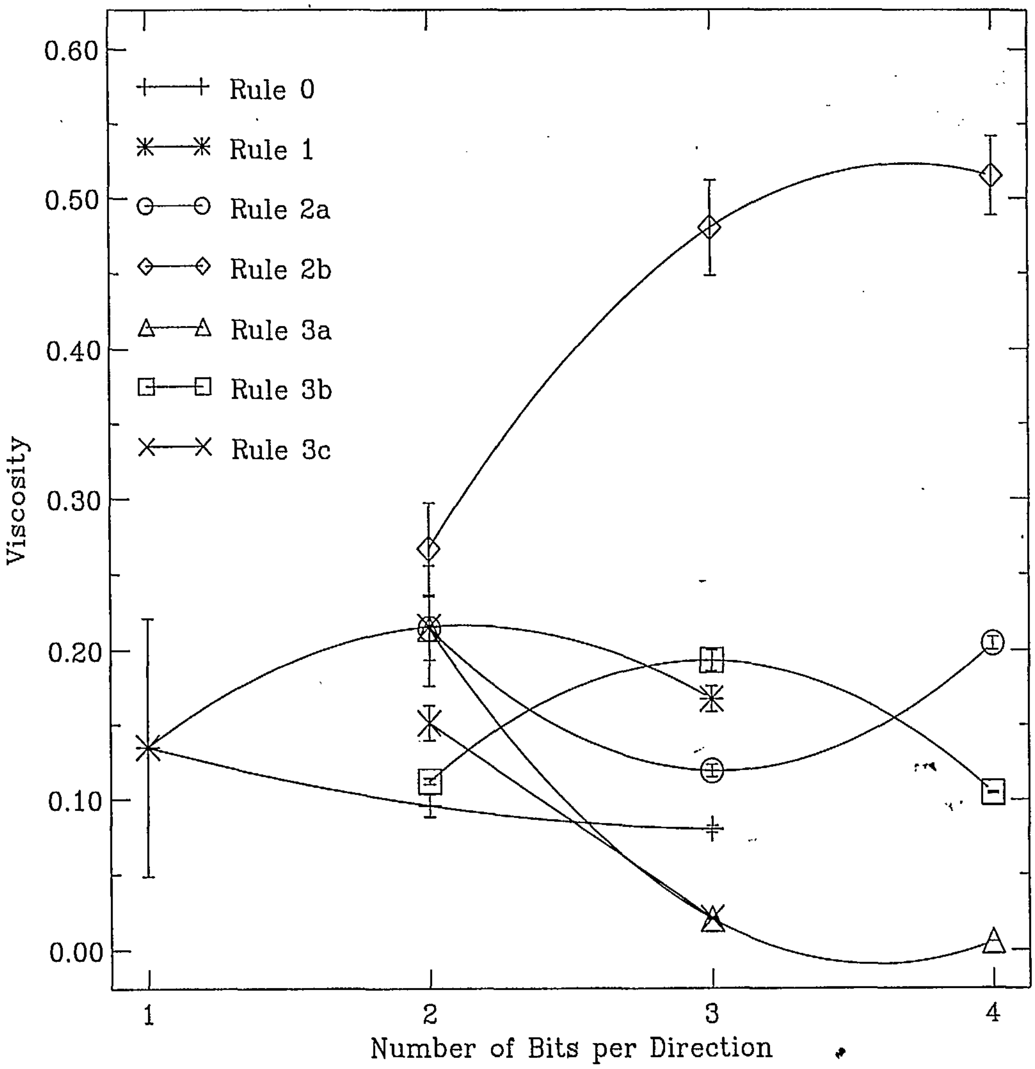


Figure 8.21: Viscosity as a function of number of bits per direction for the seven rules for a (1,1) mode.

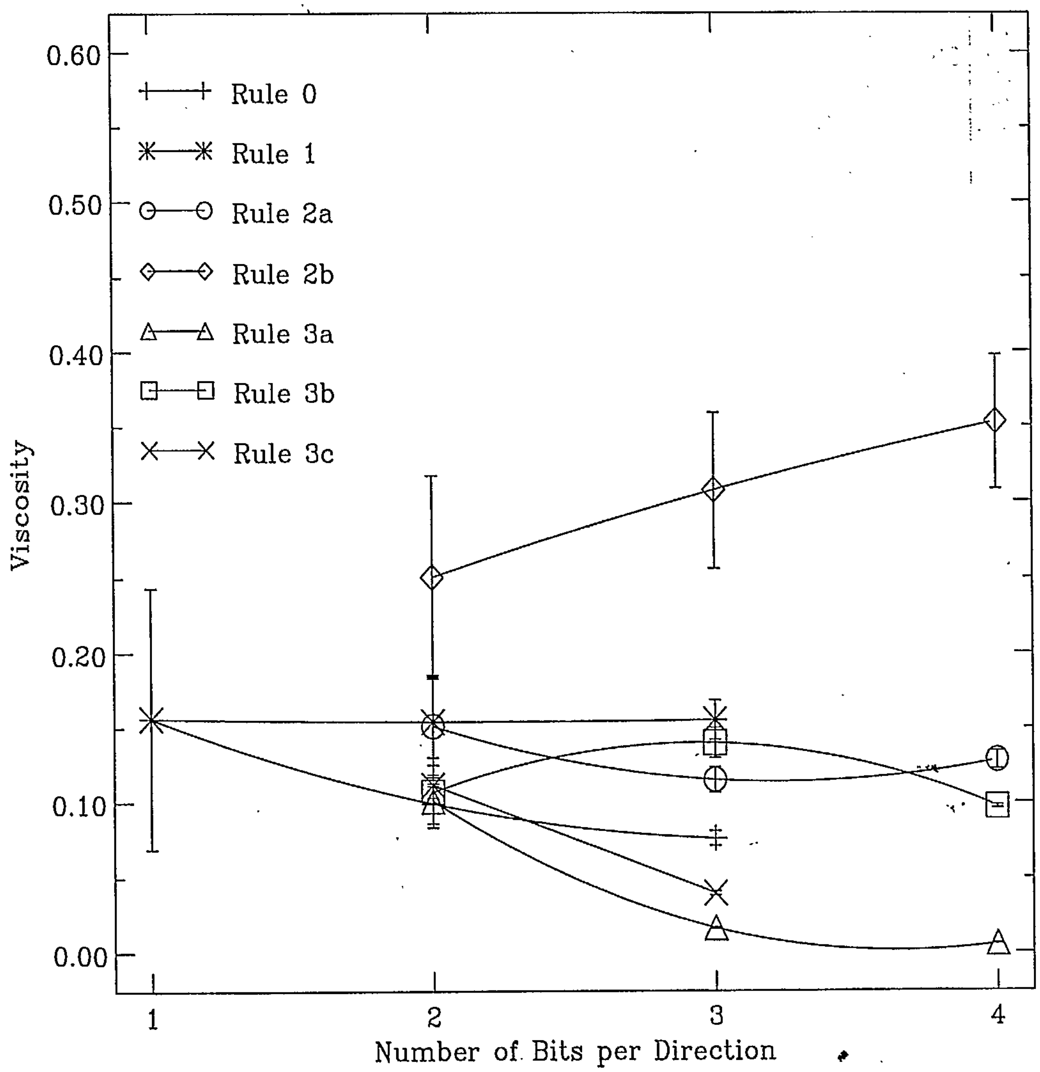


Figure 8.22: Viscosity as a function of number of bits per direction for the seven rules for a (2,1) mode.

Chapter 9

Discussion

The viscosity curves for all 4 modes, although taking on different numerical values, follow the same trends. The best rule for all modes investigated is Rule 3a.

At 5 bits per direction, the viscosity for Rule 3a is (0.0036 ± 0.0002) for a (1,0) mode. Comparing the 3 bits per direction Rule 3a result of (0.0204±0.0009) to the Rule 0 (mass and momentum conservation only) result of (0.085±0.002) for the (1,0) mode, it is evident that the viscosity has been significantly reduced.

From these Rule 3a results, it is evident that the viscosity is approaching 0 as the number of bits increases. The obstacle which must be overcome to achieve even smaller values is that of memory. For the TLM rules, a look-up table for 4 bits per direction requires $4 \times 6 \times (2^{24} - 1)$ bytes, or 67 MB of RAM. Increasing to 5 bits per direction, the memory required becomes $5 \times 6 \times (2^{30} - 1)$ bytes or approximately 32 GB of RAM. This is, of course, not feasible. The collision rule for each state was determined within the simulation at each step to obtain these results, which significantly slowed the simulation. Speed, however, is one of the benefits of using this method, so one is not inclined to sacrifice it. Another option is to perform a collision on the remainder part of the state, *e.g.* apply Rule 0 to the remainder part, to reduce viscosity. Further investigation into these possibilities is required.

To give an idea of the speed of these simulations, 1 trial for a 64×64 grid size takes approximately 7 minutes to run 10000 steps on a Sun station with 512 MB of RAM.

In Section 8.7, the anisotropy of the viscosity was mentioned. This is an interesting characteristic which will also require further investigation to determine its cause.

To fully validate the model, analytical results for the dispersion relations and the viscosity must be obtained for the low-bit cases for comparison to the results of the model.

In conclusion, we have shown that the viscosity introduced into the simulation can be drastically reduced by applying the TLM-modified rules to integer lattice gas automata when the number of bits per direction is equal to 5. The results of this study are promising for the application of cellular automata methods to electromagnetic problems.

Bibliography

[1] *Finite Elements for Wave Electromagnetics: Methods and Techniques,* P.P.Silvester, G. Pelosi, Ed., IEEE Press, New York, NY, 1994

[2] A. Taflove, *Computational Electrodynamics: The Finite-Difference Time-Domain Method* Artech House, Inc., 1995

[3] C. Christopoulos, *The TLM Method*, IEEE Press, Piscataway, NJ, 1985

[4] T. Itoh, *Numerical Techniques for Microwave and Millimeter-wave Passive Structures*, Wiley, New York, NY, 1989

[5] P. B. Johns, *Radio Electron. Eng.* 44 657 (1974)

[6] E. C. Jordan, K. G. Balmain, *Electromagnetic Waves and Radiating Systems*, Prentice-Hall Inc., New Jersey, 1968

[7] R. A. Serway, *Physics: for scientists and engineers, 2nd Ed.*, Saunders College, Philadelphia, Pa., 1986

[8] M. Krumpholz, P. Russer, "On the Dispersion in TLM and FDTD", *IEEE Transactions on Microwave Theory and Techniques* 42, 1275 (1994)

[9] N. R. Simons, "Development and Application of Differential-Equation Based Numerical Techniques to Electromagnetic Scattering and Radiation Problems", PhD Thesis, University of Manitoba, 1994

[10] R. U. Landau, *Quantum Mechanics II: A Second Course in Quantum Theory* John Wiley & Sons, Inc., New York, NY, 1996

[11] A. Ralston, E. D. Reilly *Encyclopedia of Computer Science, 3rd Ed.*, Van Nostrand Reinhold, New York, NY, 1993

[12] B. Boghosian, "Lattice Gases", *1989 Lectures in Complex Systems, SFI Studies in the Sciences of Complexity, Lect Vol. II*, Erica Jen, Ed., Addison-Wesley, 1990

[13] U. Frisch, D. d'Humières, B. Hasslacher, Pierre Lallemand, Y. Pomeau, J.-P. Rivet , "Lattice Gas Hydrodynamics in Two and Three Dimensions" *Complex Systems* **1** 649 (1987)

[14] D. d'Humières, P. Lallemand, "Numerical Simulations of Hydrodynamics with Lattice Gas Automata in Two Dimensions" *Complex Systems* **1** 599 (1987)

[15] M. Hénon, "Viscosity of a Lattice Gas" *Complex Systems* **1** 763 (1987)

[16] B. Boghosian, J. Yepez, F. J. Alexander, N. H. Margolus, "Integer Lattice Gases" (preprint).

[17] *Numerical recipes in C: The Art of Scientific Computing, 2nd Ed.*, W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Cambridge University Press, Cambridge, England, 1992

[18] N. R. S. Simons, A. A. Sebak, "New transmission-line matrix node for two-dimensional electromagnetic field problems", *Can. J. Phys.* **69**, 1388 (1991)