# CONSERVATIVE PERMUTATION MODELS OF TWO-DIMENSIONAL FLUID FLOW

A thesis presented in partial fulfilment of the requirements for the degree of

Master of Science

in

Mathematics

at

Massey University

Paul Geoffrey Turner

1997

# Abstract

At a meeting of Massey University Mathematics Department staff and students on 20 March 1997, a project to model fluid flow by constructing successive permutations of lattice cells was discussed. Several assumptions about the fluid being modelled were necessary to make the problem manageable, the chief ones being that the fluid would be two dimensional, ideal and incompressible.

Two square lattice models were developed, one Eulerian and the other more Lagrangian, in which the lattice cells were each initially assigned a value of a vorticity function. The time evolution of these models consisted of finding a permutation of the cells that was "close" to the fluid flow, then permuting the fixed initial vorticity values according to this lattice map. Justification of this method followed from consequences of the assumptions, including advection of vorticities, and the invertibility and area-preserving nature of the fluid flow map.

Reliance was placed on two previously published papers: one containing a result guaranteeing that such permutations are possible in certain circumstances, and the other providing the key to their practical construction. An essential algorithm in the latter paper relies on a theorem concerning the selection of a set of distinct elements from several sets.

Also as a result of the assumptions, the enstrophy, total vorticity and kinetic energy of such hypothetical fluid flow is conserved, although tests neither conclusively confirmed nor denied all conserving properties of the models.

# Acknowledgements

I wish to express my gratitude to my supervisor, Dr.Robert McLachlan who parted the waters of many a confused sea for me, to my mother for her endurance, and to the students and other staff members who frequently gave of their knowledge and too-short time when I became stuck. Without the assistance of these people I would still be stuck.

# Contents

# List of Figures

# Chapter 1

# Introduction

On 20 March 1997 Massey University Mathematics Department staff and interested students discussed a proposed model to mimic the flow of a fluid in two dimensions. Certain assumptions about the fluid would allow an area-preserving map to be found close to the flow map, followed by construction of a nearby bijection of lattice points to approximate the flow. The chief difference between this model and most of those that have already been developed is that it would use permutations as well as partial differential equations.

Before going any further some notation used in the following pages is presented. Vector quantities will be displayed in a bolder format than scalars.

$t$ Time: either a constant interval $\Delta t$ or an instant depending on context.

$\boldsymbol{\omega}$ Vorticity, which in two dimensions has the component form $(0, 0, \omega)$.

$\boldsymbol{V}$ Velocity, which has components $(u, v, w)$.

$\psi$ The "stream function" of a flow.

Rather than devoting space to a description of fluid mechanics, I refer you to (Chorin, A.J. and Marsden, J.E. [1]) which appears in the Bibliography.

Certain assumptions about fluids made them more manageable:

**Assumption 1**. From a macroscopic viewpoint the fluid is continuous and, in a sufficiently small volume, uniform. This natural supposition justifies the treatment of any chosen point as if it were occupied by a fluid particle contributing to flow activity.

**Assumption 2**. Any external force such as gravity acting on a fluid is zero.

**Assumption 3**. The fluid is two dimensional. Vorticity which is defined as

$$\boldsymbol{\omega} = \nabla \times \boldsymbol{V} \tag{1.1}$$

becomes

$$\omega = \nabla \times (u, v, w) \tag{1.2}$$

$$= (0, 0, \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}) \tag{1.3}$$

since in two dimensions the velocity component $w = 0$ and $z$ derivatives are zero. In two dimensions, vorticity is a scalar.

**Assumption 4**. Although mass density $\rho$ can vary spatially in circumstances such as stratification we will assume that it is constant throughout and therefore can be assumed to be equal to unity.

**Assumption 5**. The fluid is incompressible so that in two dimensions the flow $\phi$ preserves area. The "equation of continuity" for fluid dynamics can be expressed as

$$\frac{D\rho}{Dt} + \rho(\nabla \cdot \boldsymbol{V}) = 0. \tag{1.4}$$

Thus:

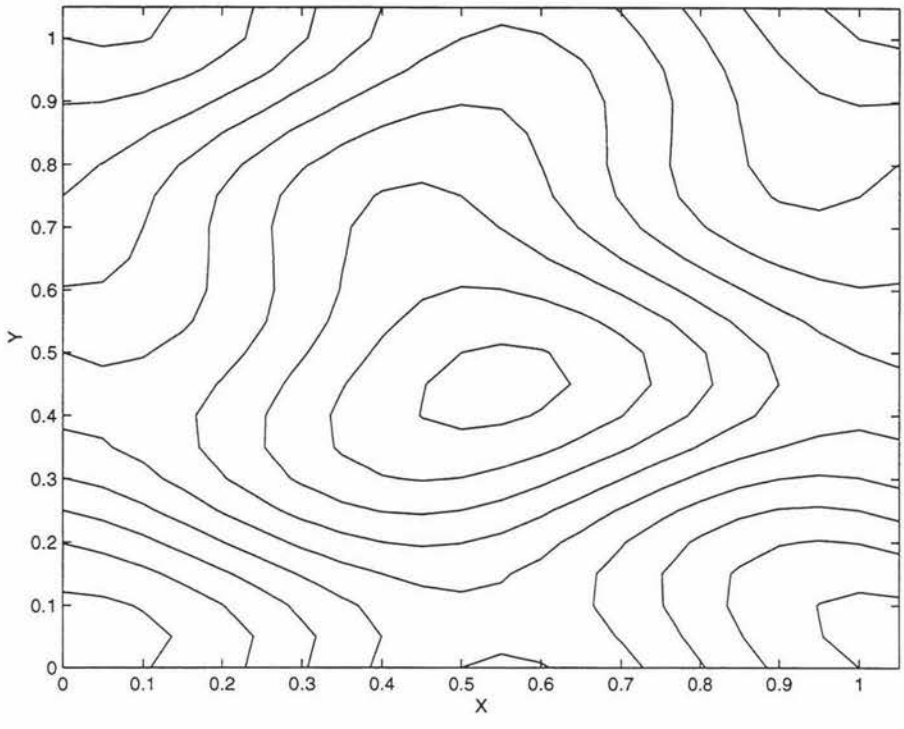


Figure 1.1: Contours of equal vorticity

a) the divergence $\nabla \cdot \boldsymbol{V}$ of the velocity field can be taken to be zero, and

b) the mass density is constant following the fluid.

$0 = \nabla \cdot \boldsymbol{V} = \partial u/\partial x + \partial v/\partial y$ implies that there exists a "stream function" $\psi$ such that

$$(u, v) = (\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x}). \tag{1.5}$$

Consequently if $\psi$ is known, the velocity field $\boldsymbol{V}$ can be calculated.

Substituting for $(u, v)$ in

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \text{ (see (1.3) on page 2)} \tag{1.6}$$

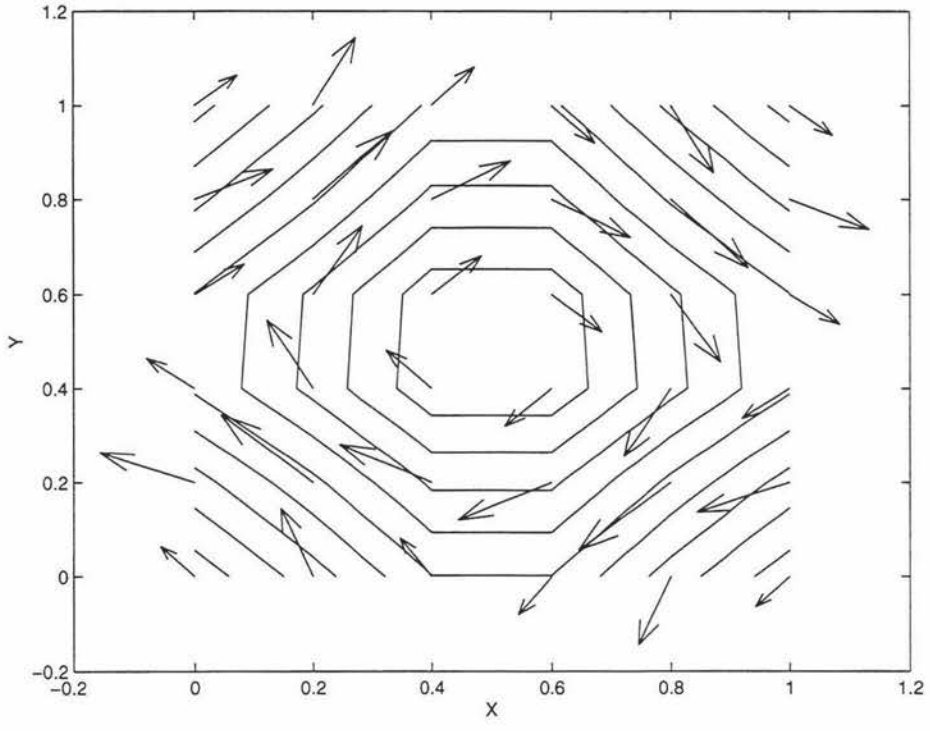


Figure 1.2: Flow through lattice sites indicated by velocity vectors and streamlines, step 1.

gives

$$\omega = -(\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2}) \tag{1.7}$$

$$= -\nabla^2 \psi. \tag{1.8}$$

This enables the stream function $\psi$ to be deduced from a known vorticity function $\omega$.

**Assumption 6**. The fluid is inviscid. Equivalently, the fluid lacks internal tangential or shear forces. Any force acting on the boundary of a region within the fluid cannot induce an alteration in the rotation of the region, so that the vorticity, being equal to twice the angular velocity at a point,

remains constant. Total vorticity is conserved.

Furthermore, vorticity is constant following the flow (a phenomenon known as advection of a passive scalar). For consider Euler's equation of motion for an incompressible, inviscid, homogeneous fluid when external forces are ignored,

$$\frac{\partial \boldsymbol{V}}{\partial t} + \boldsymbol{V}\cdot\nabla\boldsymbol{V} = -\nabla p \tag{1.9}$$

where $p$ is the fluid pressure, whose temporal evolution is unknown. Taking the curl of this equation yields the vorticity equation

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = (\boldsymbol{\omega}\cdot\nabla\boldsymbol{V}) - (\boldsymbol{V}\cdot\nabla)\boldsymbol{\omega} \tag{1.10}$$

(Olver,P.J. [1]) which again is for the three dimensional case. But as this also describes two dimensional flow when $\boldsymbol{\omega}$ is a scalar $\omega$, and $z$ derivatives are zero, it follows that

$$\boldsymbol{\omega}\cdot\nabla\boldsymbol{V} = (\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y})\frac{\partial \boldsymbol{V}}{\partial z} \tag{1.11}$$

$$= 0. \tag{1.12}$$

That is, $\partial\omega/\partial t = -\boldsymbol{V}\cdot\nabla\omega$. These terms occur in the "particle derivative" of $\omega$

$$\frac{D\omega}{Dt} = \frac{\partial\omega}{\partial t} + (\boldsymbol{V}\cdot\nabla)\omega \tag{1.13}$$

which describes the rate of change of $\omega$ due to all causes. Consequently

$$D\omega/Dt = 0. \tag{1.14}$$

This allows vorticities to be used as markers of particle positions, since where there is a particle there is a vorticity and both are carried along together.

According to these assumptions, the partial differential equations (pde's) (1.14), (1.5) and (1.8) define the Hamiltonian system that was to be modelled, namely

$$\frac{\partial \omega}{\partial t} = \{\omega, H\} \tag{1.15}$$

where the Hamiltonian function

$$H = \frac{1}{2}\int (u^2 + v^2) dx dy \tag{1.16}$$

$$= \frac{1}{2}\int \psi \omega dx dy \tag{1.17}$$

is the kinetic energy of the fluid. The Poisson bracket $\{\omega, \psi\}$, defined by

$$\{F, G\} = \int \frac{\partial F}{\partial \omega}(\omega_x \partial_y - \omega_y \partial_x)\frac{\partial G}{\partial \omega} \tag{1.18}$$

is the Lie-Poisson bracket associated with the Lie algebra of incompressible two-dimensional vector fields, where $F$ and $G$ are any smooth, real-valued functions.

For such a system, any function F evolves according to

$$\dot{F} = \{F, H\}. \tag{1.19}$$

Therefore,

$$\dot{H} = \{H, H\} \tag{1.20}$$

$$= -\{H, H\} \text{ (by antisymmetry)} \tag{1.21}$$

$$= 0 \tag{1.22}$$

(Olver,P.J. [1]).

The fluid flow map $\phi$: $(x(0), y(0)) \rightarrow (x(t), y(t))$, which describes particle trajectories is defined by the solution of the ordinary differential equations

(ode's)

$$\dot{x} = u(x, y, t) \tag{1.23}$$

$$\dot{y} = v(x, y, t) \tag{1.24}$$

and because $\nabla \cdot \boldsymbol{V} = 0$, $\phi$ preserves area, or $\det D\phi = 1$. $\phi$ is also invertible since it is a one-to-one mapping of particles onto their images, so any model of $\phi$ should be a bijection. Hence:

1. energy, $H$ is preserved,

2. because $\frac{D\omega}{Dt} = 0$ and $\phi$ is invertible, individual vorticity values are advected according to

$$\omega(\phi(\boldsymbol{x}, t), t) = \omega(\boldsymbol{x}, 0) \tag{1.25}$$

where $\boldsymbol{x} = (x, y)$, and

$$\omega(\boldsymbol{x}, t) = \omega(\phi^{-1}(\boldsymbol{x}, t), 0). \tag{1.26}$$

That is, the area-preserving fluid flow map $\phi$ permutes vorticities. In particular, the vorticity flow map $\omega(t) = \omega(0) \circ \phi$ is a Poisson map,

3. the Casimir functions of the Poisson bracket $\{.,.\}$ are the functions

$$C = \int f(\omega) dx dy \tag{1.27}$$

where $f(\omega)$ is any smooth function of $\omega$.

Casimir functions C are characterised by $\{C, F\} = 0$ for any real-valued function $F$, so by (1.19), $\dot{C} = 0$. This gives an infinite family of conserved

quantities and is equivalent to the rearrangement of vorticities by (1.25) and (1.26).

For example, the "enstrophy"

$$\int \omega^2 dxdy \tag{1.28}$$

is preserved in time.

A numerical scheme was sought that would preserve some analogues of 2. and 3.

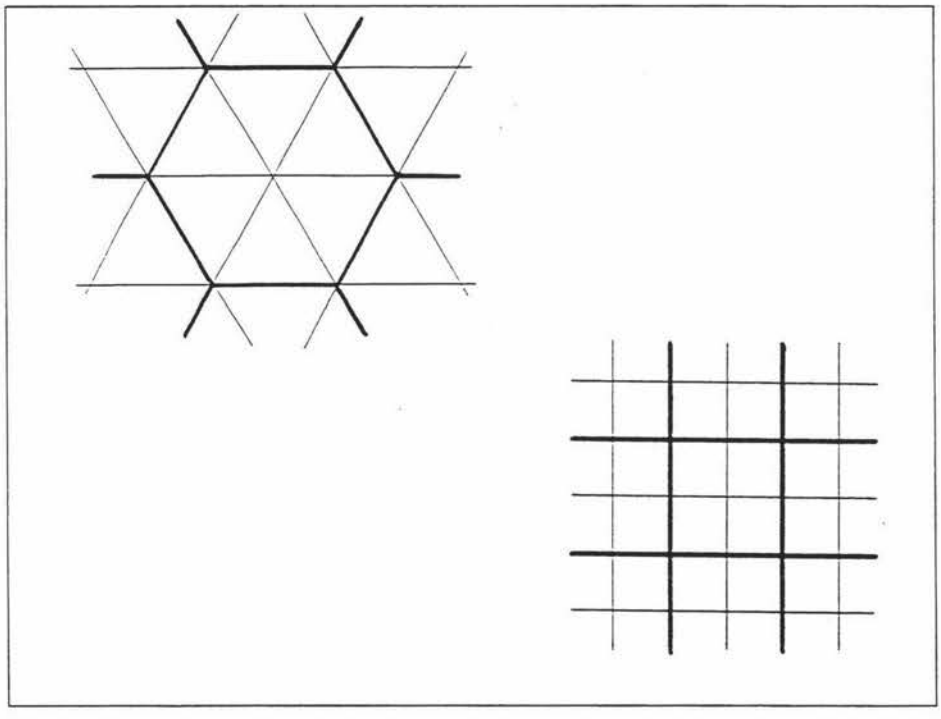

Figure 1.3: Triangular/hexagonal, and square dual regular tessellations.

Traditionally, discrete computations use either Eulerian or Lagrangian methods, or even a combination of these. Eulerian techniques are based on the determination of the velocity $\boldsymbol{V}(\boldsymbol{x}, t)$ of a fluid particle passing through

a point $\boldsymbol{x}$ at a time $t$, whereas the Lagrangian option considers the trajectory that each particular fluid particle follows through space (Feistauer,M. [1]).

An Eulerian approach was considered that would involve numerical finite difference formulae to solve differential equations. These replace derivatives by finite difference approximations, and an example is the "second order central difference formula"

$$\frac{\partial \omega}{\partial t}(x,y) \approx \frac{\omega(x+h,y) - \omega(x-h,y)}{2h} \tag{1.29}$$

which improves in accuracy as the finite spatial interval $h$ diminishes toward zero. Such spatial discretizations do not, in general, provide a set of Hamiltonian ordinary differential equations, and in fact this is believed not to be possible for the two dimensional Euler equations. However, judicious choice of the finite difference *can* lead to some conserving properties, and one of the permutation models developed used such an Eulerian scheme to preserve discrete analogues of energy and enstrophy.

As Eulerian finite difference methods do not support Poisson mappings and do not offer any area-preserving analogues, attention turned to Lagrangian options which have no difficulty with advection and preservation of vorticity. Nevertheless, Lagrangian methods which involve updating particle positions with coordinate pairs $(x(t), y(t))$ involve twice as many degrees of freedom as the Eulerian options concerned with updating a single variable at known, fixed sites and, like these, do not necessarily support area preservation.

To obtain a strictly discrete analogue of both area preservation and the Casimir functions, models were considered whose foundation would be a fixed tessellation of polygons or cells (see figure 1.3 on page 8 above). Irregular

tessellations were rejected because the assumption that a flow preserves area implies that it can only be represented by permuting the cells of a partition if those cells have equal area. It is well known that the only regular tessellations are equivalent to either triangular or square tessellations, and in fact the only other such tessellation is hexagonal (Coxeter,H.S.M. [1]). A regular tessellation consisting of congruent cells can alternatively be regarded as a regular lattice of intersecting lines (Coxeter,H.S.M. [1]).

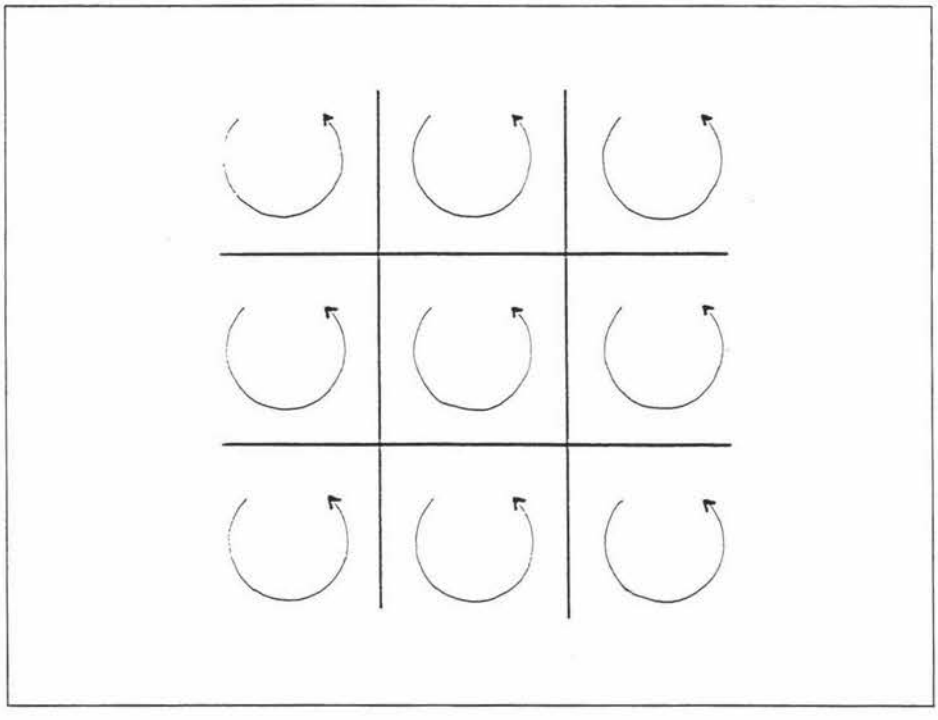

Figure 1.4: Periodic array boundaries indicated by congruent flow cells

As is common practice, the chosen lattice was constructed from a finite array with certain boundary conditions. Two types of boundary conditions were considered: "wall" and periodic (see figure 1.4 above). Inviscid flow experiences only tangential flow at a boundary wall, so that the wall coincides

with a streamline and the stream function $\psi$ has zero value there. Imposing a wall boundary condition on a flow may seem more artificial than using periodic boundary conditions which assume no "barriers", and in fact both the permutation models that were developed had periodic boundaries.

Both these models of a known fluid flow map $\phi$ relied on three key consequences of the assumptions:

a. $\phi$, being invertible, was modelled by a bijection or permutation,

b. as $\phi$ preserves area, the permutation was performed on equal-area cells of a lattice, and

c. vorticities would be permuted by this lattice map with the intention of reflecting their advection.

That is, $\phi$ would be replaced by a "nearby" permutation of the equal area lattice cells (and of the fixed vorticities initially assigned to them), so forming a discrete analogue of $\frac{\partial \omega}{\partial t} = \{\omega, H\}$. In this way the entire set of vorticity values would be preserved and a fully discrete (in space, time and vorticity) analogue of the Casimirs would be created. This was reminiscent of discrete models called "automata networks" which are surveyed in chapter 2. Following that, chapters 3 and 4 describe the construction of two particular conserving permutation models, whose behaviour is examined in chapter 5. The conclusion presents the main results and possibilities.

# Chapter 2

# Models from the Past

The history of what might be called discrete or cellular machines began with the "Turing machine", which was invented in the mid 1930's by A.M.Turing to make precise the concept of "algorithm". A Turing machine consists of a finite automaton or machine with a "head" for reading and writing to a "tape" of cells, each capable of storing a single symbol. Such tape is usually one dimensional and infinite in both senses of a direction but the dimension can be any positive integer. The automaton and tape can move relative to each other so that the automaton reads or writes to each cell of the tape sequentially (not synchronously). From the program, or data initially recorded on the tape, the automaton constructs an answer and then prints it on cells following those containing the program.

The "automata network", an extension of the concept of the Turing machine, was inspired by a suggestion that discrete machines could model biological phenomena. It has:

1. a uniform cellular space,

2. an associated family of finite-state machines,

3. local transition functions.

One type of automata network is the "cellular automaton" (or CA), which is an automata network exhibiting

4. parallel rather than sequential evolution,

5. a neighbourhood that has the same definition for each cell.

More formal descriptions such as that in (Garzon,M. [1]) are available but as yet there is no consensus to choose just one of them as definitive. In the cellular space of a CA, the state of every cell is updated by a common set of rules (local transition functions). Notice then that if the cellular space of a CA were finite then the neighbourhoods of the marginal cells would differ from those of the other cells so that the same set of rules could not apply to all the cells. This is why the cellular spaces of CA are effectively infinite.

CA were discovered in the 1940's independently by Stanislaw M.Ulam (who suggested them as models of growth, self-reproduction and biological evolution) and Konrad Zuse (who was a German engineer hiding in the mountains from the Nazis during that period), and their possibilities were explored by John von Neumann during his attempts to model aspects of reproduction and other biological phenomena (Wolfram, S. [1]). The idea that led to the discovery of CA, with the associated notion of motion or change at the sites of a lattice came partly from Cayley digraphs (a directed graph whose vertices represent group elements and are joined by directed arcs representing generators) which were introduced by A.Cayley over a century ago to describe groups (Garzon, M. [1]).

In 1943 McCulloch and Pitts showed that a CA can simulate a Turing machine, despite the serial computation of the latter (Toffoli, T. and Margolus, N. [1]). Toffoli later showed that any reversible CA is capable of universal computation, meaning it can simulate any computer, and so reversible CA are of particular interest.

But research into CA themselves was initiated when von Neumann wondered whether there exists a cellular space (his name for a cellular automaton) on which a universal computer-constructor (a self-replicating automaton) could be exhibited (Stewart, I. [1]). In the early 1950's von Neumann was inspired by this concept to prophetically imagine a robot swimming in a sea filled with spare parts which it collected to make a replica of itself, very like DNA the replication mechanism of which was actually demonstrated by Kornberg and colleagues only later in 1956. In 1958 John Myhill took an intermediate step toward abstraction and generality by devising an infinite half plane tessellated like a chess board with a conveyor belt running along the finite edge. Machine components moved discretely from square to square and were augmented by other components grasped by the automaton from the conveyor belt. In this way the original finite automaton could conceivably replicate itself.

Von Neumann idealised this arrangement by substituting propagations, or waves of patterns of state values for the kinetic automata of Myhill and at last succeeded in creating a self-reproducing CA by specifying appropriate transition functions ("rules") and initial configurations. This CA was self-reproducing in the sense that an initial pattern could become two or more (Packard,N.H. and Wolfram,S. [1]; Bagnoli,F. [1]). At the time it was not

obvious that this could be done as there is a proven theorem called the "Garden of Eden" theorem which states roughly that some configurations (namely Garden of Eden configurations, which have no ancestors) cannot evolve from others.

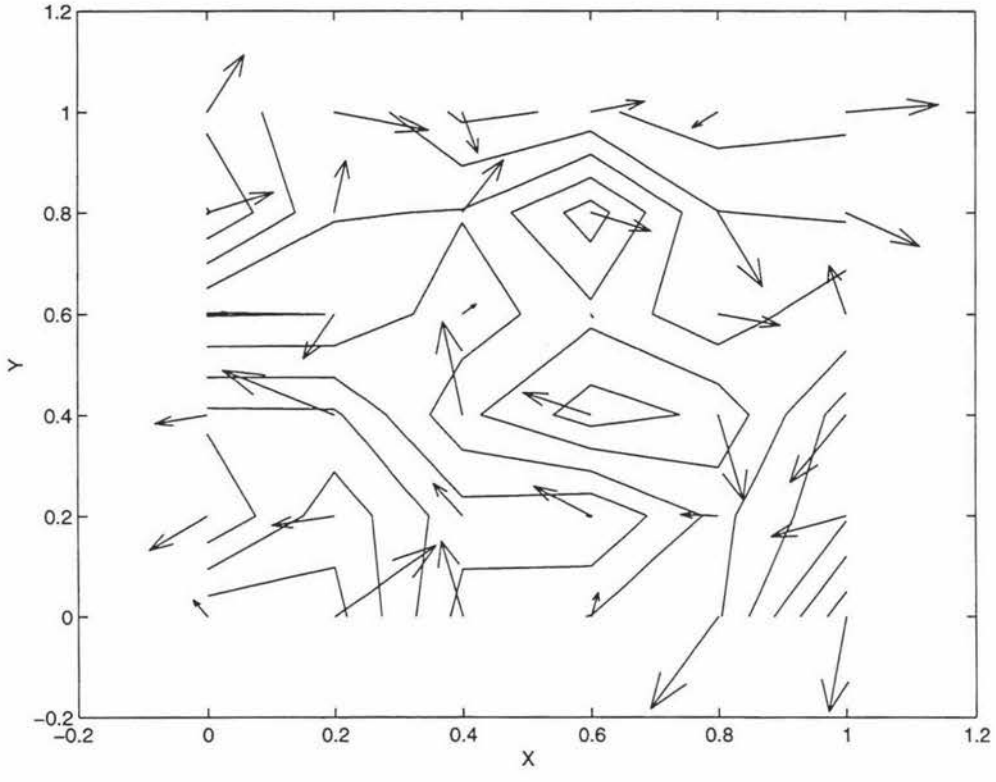


Figure 2.1: Flow through lattice sites indicated by velocity vectors and streamlines, step 2.

CA have been used to study "self-organisation" in discrete systems (continuous systems are explored by the science of "cybernetics", the science of systems of control and communication in animals and machines). Toffoli was using them in the early 1970's to investigate adaptation and optimisation, when he became convinced of the need for parallel computing architecture

and developed the CAM-6 plug-in module (followed by later versions) for IBM personal computers, so allowing them to operate as synchronous rather than as sequential computers.

In the 1980's Stephen Wolfram's studies of computation universality, classification and physical analogies led him to discover that even with very simple rules a CA can produce very complex behaviour. From this Wolfram began developing what he called the "science of complexity" and in 1985 he posed twenty problems concerning CA, ranging from their classification to their computation limitations, that had at that time yet to be solved. Now, the availability of more powerful computers enables the study of CA themselves as autonomous entities, rather than in their other role as transducers that merely output modified input data. Indeed, Edward Fredkin postulated that our real universe is itself a cellular automaton. According to his suggestion the problem for physics is simply to discover the rules by which this CA operates.

The global dynamics of a CA can be visualised in the "phase space" of the transition function (the program of rules). This phase space consists of a digraph in which the possible configurations of a CA have shrunk to vertices connected by directed arcs representing the actions of the transition function. According to this concept, a configuration $c$ is said to be "temporally periodic" with period $p$ if the outcome of $p$ iterations of the transition function $f$ on $c$ is $c$, so that in phase space the orbit of the configuration passes through a finite number of vertices before returning to its starting point. If a configuration does not return then it is "aperiodic". This results in phase space looking like a lot of loops, and points with lines radiating away

from them, forming a so-called "forest of unicyclic graphs". An "attractor" is a set of state values through which a net cycles between two occurrences of a configuration, and is either a limit point (one state) or a limit cycle (more than one state). Some CA support attractors called "solitons": localised structures which can interact with one another while retaining their own identity. Two dimensional CA can generate "dendritic" patterns (linear features that grow) and "gliders" which are periodically repeating patterns that slide through space. There are also "blinkers" which repeatedly reproduce themselves, "guns" which periodically discharge gliders and the dazzling "star ships" which leave a trail of blinkers in their wake. Some examples of CA follow.

1. In 1970 the Cambridge mathematician John Horton Conway invented the "Game of Life", a CA which was designed to simulate social evolution in two dimensional living systems, and which has been subsequently extended by Vitanyi and others to more realistic scenarios. Here it is used to illustrate the operation of a typical set of local rules. These form a "look-up" table consulted by each and every cell simultaneously when their state values are to be revised. In the Game of Life the neighbourhood of each cell consists of itself (labelled $C$) and its eight adjacent neighbours (each marked $N$) — the "Moore neighbourhood". Conway introduced the program consisting of the rules:

    a. If a cell $C$ is living and at least four of its neighbours $N$ are also living, then when the state of the cell $C$ is updated, it becomes dead.

    b. If the cell is dead and exactly three of its neighbours $N$ are living

then when the state of the cell $C$ is updated it becomes a living cell.

c. For all other configurations of the cell and its neighbours no change occurs when the state of the cell is updated.

In operation, the Game of Life exhibits dynamic, self-organising behaviour in which clusters of cells with an identical state value form and disperse in a lifelike manner, but although configuration patterns move progressively across adjacent neighbourhoods all this begins with initially local communication within neighbourhoods. At each iteration there is no exchange of information between cells at sites more remote than those within adjacent neighbourhoods.

2. The "partitioning cellular automaton" has a partitioning of its cells into finite, disjoint and uniformly arranged blocks. It has a block rule to update whole blocks rather than individual cells in the blocks, and the partition into blocks is changed from one step to the next so as to introduce some overlap — otherwise the CA would consist of a collection of independent subsystems. The "Margolus neighbourhood" is an example of such a partitioning and uses just two partitioning grids — called even and odd. This ensures interaction between any cell and different blocks from step to step.

3. A stochastic CA has been used to model the occurrence of solar flares. Solar flares are caused when magnetic energy builds up in the sun before being released, when the associated magnetic field becomes unstable. Each such instability may beget further instabilities at neighbouring

sites in the sun, and hence the analogy with CA. Conflicting in detail with observations, this model nevertheless shows remarkable qualitative agreement with the distribution of these solar events (MacKinnon and others [1]).

4. Biological cells contain a cytoskeleton of protein cylinders called microtubules. These were only discovered in the 1970's because when samples were prepared for examination by electron microscopes the fixative chemicals that were used dissolved the microtubules. Apart from giving the cell rigidity and shape microtubules have other functions such as forming sites ("centrioles") for the initiation of cell division, and motile cells use them in the form of paddling cilia and flagella for locomotion. Single celled animals such as paramecia are capable of learning to negotiate a microscopic maze despite their lack of a multicelled brain, so in order to learn they must have some sort of memory and learning mechanism. Hameroff suggested that a study of microtubules could explain this when he noted that protein molecules on the walls of a microtubule vibrate, causing interaction between one another so that meaningful waves of information travel along the microtubule surface. Brain cells are known to have unique forms of microtubule organisation and Hameroff thought that within each neuron cell the network of these microtubules forms a cellular automaton which is capable of generating thoughts, besides retaining memories as standing waves on the surfaces of the microtubules. In this way it is conceivable that CA could even be implicated in the foundation of consciousness (McCrone,J. [1]).

Other automata networks range from artificial neural networks with cells

located at the intersections of arbitrary digraphs (useful for cognitive modelling), to Stuart Kauffman's random net, proposed in 1969, that would have parallel iteration and Boolean automata but randomly chosen transition functions and random connections or in other words random neighbourhoods. The most successful application of automata networks in the natural sciences to date are the lattice gas automata (LGA), the main distinguishing features of which consist of identical particles (of mass) which move iteratively from site to site of a regular lattice (see figure 1.3 on page 8). The outcome of any collisions at the sites is governed by a set of rules. LGA have encouraged many variants including the FHP model of 1986 by Frisch, Hasslacher and Pomeau which had a hexagonal lattice (McNamara,G.R. [1]).

Until that year, even though Zuse had recognised the connections between CA and partial differential equations in 1970 (Rothman,D.H. and Zaleski,S. [1]) and there followed much speculation about the abilities of cellular and other automata to model a partial differential equation and actual physical phenomena, no concrete example of such an association was known. The FHP model demonstrated that a discrete model of simplified molecular dynamics can asymptotically simulate the irreversible Navier-Stokes equations for two-dimensional, macroscopic, homogeneous and incompressible fluid flow. These are:

$$\frac{\partial \boldsymbol{V}}{\partial t} + (\boldsymbol{V}\cdot\nabla\boldsymbol{V}) = -\nabla p + \mu\nabla^2\boldsymbol{V} \tag{2.1}$$

and

$$\nabla\cdot\boldsymbol{V} = 0 \tag{2.2}$$

(Chorin, A.J. and Marsden,J.E. [1]; Toffoli and others [1]; Rothman,D.H.

and Zaleski, S.[1]) with certain necessary boundary conditions, where $p$ is pressure, and $\mu$ is the "coefficient of kinematic viscosity". Since these become the Euler equations for ideal, incompressible fluid flow when $\mu = 0$, the chances of successfully simulating such flow with an automata network model do not seem too slim. A conserving permutation model was not however developed from an FHP model because the latter has mass poles rather than vorticity poles.

Could a CA do the job? Natural information processing systems resemble CA more than they do serial-processing computers. In particular, fundamental physical phenomena tend to be reversible such as the microscopic molecular dynamics of fluids which can be modelled by reversible CA. At the other end of the scale macroscopic systems, like most CA are irreversible (Wolfram,S. [2]).

Much could be said in favour of CA models but the main objection to developing a CA model of fluid flow was prompted by its local mode of communication. The model to be constructed was to operate by selecting permutations of the sites of an array, and there was no guarantee that sites would be mapped by the permutation either to adjacent sites or to "nearby" sites. Still, the preference for an automata network model indicated that something of the lattice variety incorporating partial differential equations rather than local rules was needed.

# Chapter 3

# A First Permutation Model

A model with Lagrangian features was tried first, not least because of its abilities regarding vorticity. Its square lattice consisted of arrays with periodic boundaries, where each square array had $N$ sites along each edge for some positive integer $N$. Construction of this model, known from now on as Model A, consisted firstly of finding a bijection $P$ of the lattice sites so that $P$ was close to the area-preserving fluid flow map $\phi$.

Whether or not a permutation exists depends on a previously published result:

> Let $T$ be a continuous invertible measure preserving transformation mapping the $n$-dimensional torus $S$ onto itself. Divide each generator of $S$ into $M$ equal parts, thus dividing $S$ into $M^n$ congruent $n$-dimensional cubes. Then there exists a transformation $T_M$ which permutes these cubes, and approximates $T$.

in which I have changed some of the notation to avoid later conflict (Lax,P.D. [1]).

For our purposes, $T$ can be interpreted as the flow map $\phi$, and $S$ as the

two-dimensional space partitioned by the $M^n$ congruent cells of its tessellation. By the assumptions explained in the Introduction this implies that a permutation $T_M$ does exist.

As Model A had Lagrangian features, it was necessary to calculate the velocity field at each iteration in order to follow the flow. This can be found from a given vorticity function $\boldsymbol{\omega}$ by solving the equation (1.8) on page 4 for $\psi$. This can be accomplished by introducing the DFT or Discrete Fourier Transform as follows.

In two dimensions, the stream function can be expressed in terms of its DFT as

$$\psi = \sum_k \sum_l e^{i(kx+ly)} \widehat{F}(\psi) \tag{3.1}$$

where $\widehat{F}(\psi)$ is the DFT (calculated using the Fast Fourier Transform algorithm), and so

$$\frac{\partial \psi}{\partial x} = ik\psi. \tag{3.2}$$

Further differentiation gives

$$\frac{\partial^2 \psi}{\partial x^2} = -k^2\psi \tag{3.3}$$

and similarly

$$\frac{\partial^2 \psi}{\partial y^2} = -l^2\psi. \tag{3.4}$$

Taking the DFT of these for any given value of $k$,

$$\widehat{F}(\frac{\partial^2 \psi}{\partial x^2}) = \widehat{F}(-k^2\psi) \tag{3.5}$$

$$= -k^2 \widehat{F}(\psi) \tag{3.6}$$

by the linearity property of the DFT. Similarly

$$\widehat{F}(\frac{\partial^2\psi}{\partial y^2}) = \widehat{F}(-l^2\psi) \tag{3.7}$$

$$= -l^2\widehat{F}(\psi). \tag{3.8}$$

Now

$$\nabla^2\psi = \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} \tag{3.9}$$

so

$$\widehat{F}(\nabla^2\psi) = -k^2\widehat{F}(\psi) - l^2\widehat{F}(\psi) \tag{3.10}$$

$$= -(k^2 + l^2)\widehat{F}(\psi). \tag{3.11}$$

Since

$$-\omega = \nabla^2\psi, \tag{3.12}$$

$$-\widehat{F}(\omega) = -(k^2 + l^2)\widehat{F}(\psi) \tag{3.13}$$

and

$$\widehat{F}(\psi) = \frac{\widehat{F}(\omega)}{(k^2 + l^2)} \tag{3.14}$$

where $k^2 + l^2$ is nonzero whenever $\widehat{F}(\psi)$ is nonzero. In this last equation, $\psi$ is the inverse DFT of the right hand side, leading to the velocity field

$$\boldsymbol{V} = (\frac{\partial\psi}{\partial y}, -\frac{\partial\psi}{\partial x}). \tag{3.15}$$

However, because of the problems electronic computers have with accurate differentiation the practical procedure followed was a little different to this. $\widehat{F}(\psi)$ was found much as already described, but instead of first finding the

inverse DFT of $\widehat{F}(\psi)$ and then taking partial derivatives use was made of the formulae

$$u = \frac{\partial \psi}{\partial y} \tag{3.16}$$

$$= \widehat{F}^{-1}(il\widehat{F}(\psi)) \tag{3.17}$$

and

$$v = -\frac{\partial \psi}{\partial x} \tag{3.18}$$

$$= -\widehat{F}^{-1}(ik\widehat{F}(\psi)) \tag{3.19}$$

as derived above (see equation 3.2).

Once the velocity field was approximately known, images of the lattice sites could be obtained, but in most circumstances these images would not coincide with lattice sites. Consideration was given to a double shear:

$$x' = x + f(y) \tag{3.20}$$

followed by

$$y' = y + g(x') \tag{3.21}$$

for some functions $f$ and $g$ that would result in an approximation of the flow map $\phi$. The Jacobian matrix of the map that this represents is

$$D_{(x,y)}\phi = \begin{bmatrix} 1 & f' \\ g' & (1 + g'f') \end{bmatrix}. \tag{3.22}$$

Taking the determinant of the Jacobian,

$$det(D_{(x,y)}\phi) = 1 + g'f' - g'f' \tag{3.23}$$

$$= 1 \tag{3.24}$$

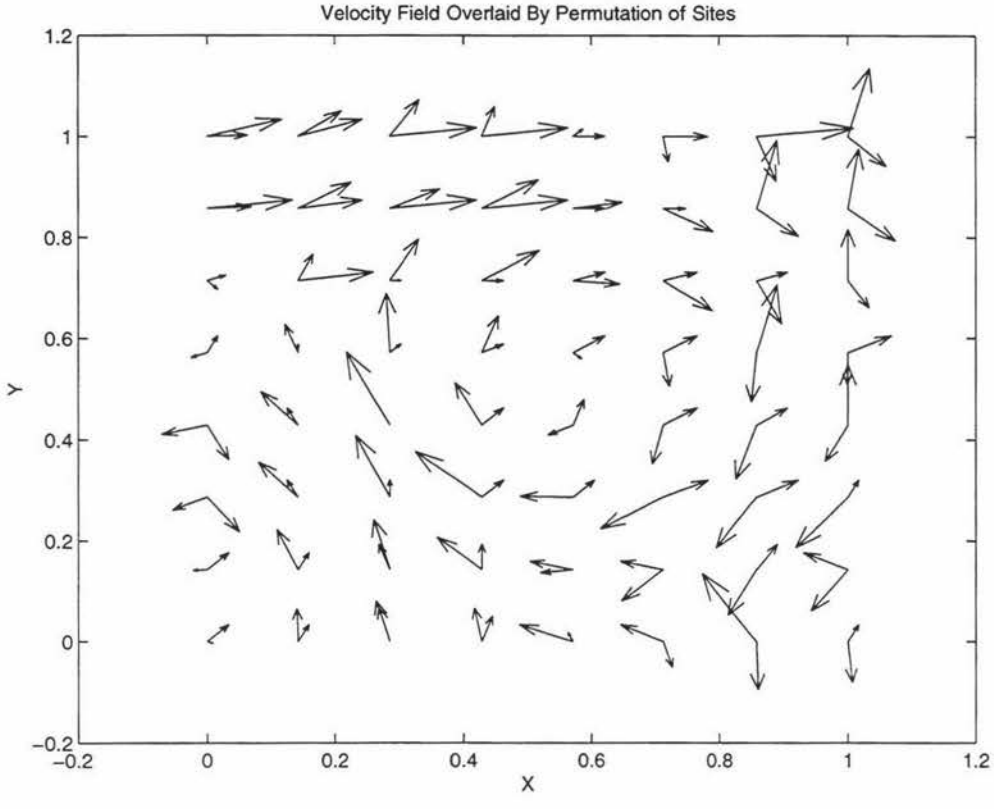


Figure 3.1: Velocity and displacement vectors show difference between flow and permutation at each site.

which means that area is preserved as required. However, it was essential to truncate the images to the appropriate sites if a permutation of the sites was to be a good approximation of the flow (see figure 3.1 above).

A solution to both the problem of permutation and of truncation was devised essentially by ascertaining a group of "candidate" sites close to the image of each lattice site and then choosing just one candidate to uniquely represent the position of the image. To obtain these sets of candidate sites, a second lattice of finer mesh (the "finemesh lattice") was superimposed on the original lattice (the "coarsemesh lattice") so that each coarsemesh lattice site

was also a finemesh lattice site. Each coarsemesh lattice site $C$ thus occupied a square neighbourhood containing a finite number of finemesh lattice sites but no other coarsemesh lattice sites.

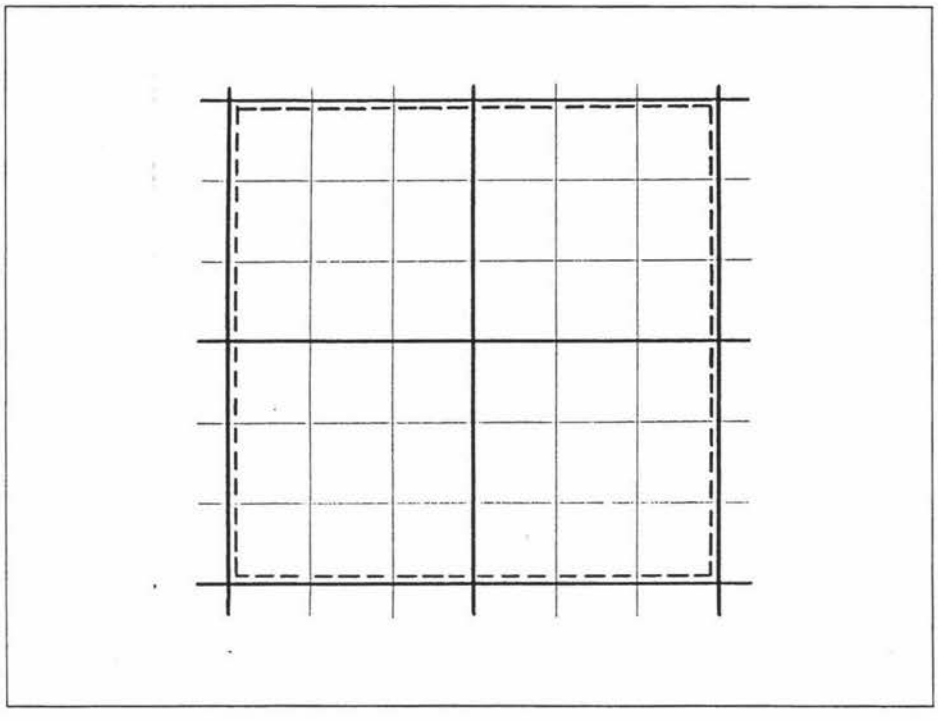

Figure 3.2: The neighbourhood (inside dashed boundary) of a coarsemesh site.

The continuous flow mapped all sites in a neighbourhood of $C$ to a neighbourhood of the image of $C$. These images were then truncated to coarsemesh sites, so forming a candidate set $S$ for $C$.

Choosing a distinct element from $S$ would be equivalent to selecting a site, different from all the others chosen, to uniquely match $C$, although this would mean assuming that vorticities were transported from site to site. If the selections could be accomplished so that all the representatives were

distinct, and so that every coarsemesh site was the image of at least one coarsemesh site (in fact it must be the image either of some other coarsemesh site, or of itself) then the selection would be a bijection or in other words a permutation.

At this point the unanswered question loomed: "What circumstances guarantee that enough distinct elements can be chosen from several sets to form a set of distinct representatives of the sets?" In fact it turns out that in some cases no set of distinct representatives exists, but even so this difficulty can in principle be overcome essentially by increasing the number of elements in each set, or in our case by using a lattice of finer mesh. After all, creating more elements in each candidate set would tend to increase the probability that each set has a distinct representative. With this in mind it was necessary to produce an algorithm for selecting such a permutation.

A prototype of the algorithm was suggested by a paper dealing with just this problem (Kloeden, P.E. and Mustard, J. [1]). In their paper, Kloeden and Mustard propose a method for selecting a unique representative element from each of several given sets of elements. The method depends upon a theorem by Phillip Hall:

> Let $I$ be a finite set of indices, $I = \{1, 2, ..., n\}$. For each $i \epsilon I$, let $S_i$ be a subset of a set $S$. A necessary and sufficient condition for the existence of distinct representatives $X_i$, $i = 1, 2, ..., n$, $x_i \epsilon S_i$, $x_i \neq x_j$, when $i \neq j$, is condition $C$: For every $k = 1, ..., n$ and choice of $k$ distinct indices $1_1, ..., i_k$, the subsets $S_{i_1}, ..., S_{i_k}$ contain between them at least $k$ distinct elements.

(Hall, M. [1]).

Consequently, if it is known that such a set of distinct representatives exists, then for any value of $k$ from 1 to $n$, if $k$ of the $n$ sets are chosen in

*any* way at all it can be taken that there will certainly be at least $k$ distinct elements among them. Conversely, if it turns out that any $k$ sets selected have fewer than $k$ distinct elements in their union, then the $n$ sets do not contain a set of distinct representatives and in such a case it is impossible to construct either a set of distinct representatives or a permutation of lattice sites without making adjustments.

Before discussing how it works, here is the algorithm.

**Distinct Representatives Algorithm**

1. Label the sets: given $n$ "candidate" sets, label the sets in any order from 1 to $n$.

2. Initial selection: select any one element from each set to construct a column vector so that the ith element of the vector represents the ith candidate set.

3. Substitute for duplicates: working from the first element toward the $n$th element of the vector, or equivalently from the representative of the first set toward the representative of the $n$th set, attempt to eliminate duplications among the representative vector elements (preceding the representative under current examination) by substitution from the set it represents.

4. Ascertain current "block": if the $n$th set is reached and is allocated a distinct representative then display the completed set of distinct representatives and stop. Otherwise a set, $k$ say, will be reached which has no element different from the representatives of sets 1 to $k-1$, and the sets 1 to $k$ constitute a set which will be referred to as a "block". At this stage one of the selected representatives that also occurs in the kth set must, by substitution of some previously unselected element be made available to uniquely represent the $k$th set.

5. Find free element if "$j \in k$": working from the first set toward the $(k-1)$th set, if the $j$th set say is represented by an element of set $k$ and set $j$ has an element that is not currently representing any set of the block of $k$ sets (that is, a "free" element) then the free element is substituted for the $j$th representative which is then available to uniquely represent the $k$th set, and the algorithm either branches to step 3 (but this time working from set $k+1$ of the next block, and checking back to set 1 that proposed substitutes are not duplicated), or if the set of distinct representatives is complete, displays it and stops. If the $j$th set has no free element, then sets $j+1$ to $k-1$ are considered.

6. Find free element if "$j \notin k$": if set $j$ is not represented by an element of set $k$, and if representative $j$ also occurs in set $l$ say of any of the sets 1 to $k-1$ that are represented by elements of set $k$, then determine whether there is a free element in set $j$. If there is, let the representative of set $l$ represent set $k$, let the representative of set $j$ represent set $l$ and make the free element the new representative of set $j$. If this takes place, the algorithm either branches to step 3 or displays the completed set of distinct representatives and stops. If representative $j$ does not occur in a set represented by an element of set $k$ then the algorithm considers sets $j+1$ to $k-1$ in step 5.

7. No free elements?: if no free element is found anywhere then it is impossible to construct a set of distinct representatives and the algorithm stops.

How does it work? The key is to substitute some free element that exists in the union of the sets 1 to $k-1$ for one of the representative elements that also occurs in the set at the end of each block (set $k$) so that it can become a distinct representative of that set. According to Hall's theorem and assuming that a set of distinct representatives exists, since only $k-1$ distinct elements are at first located to represent $k$ sets and all the elements in set $k$ are already being used as representatives, there remains at least one other free element to be found among the sets 1 to $k-1$. Consequently the search for a free element will be futile only if the assumption is incorrect, and also of course a representative of set $k$ exists among the elements of a set of $n$ distinct representatives.

If a free element occurs in a set represented by a member of set $k$, then

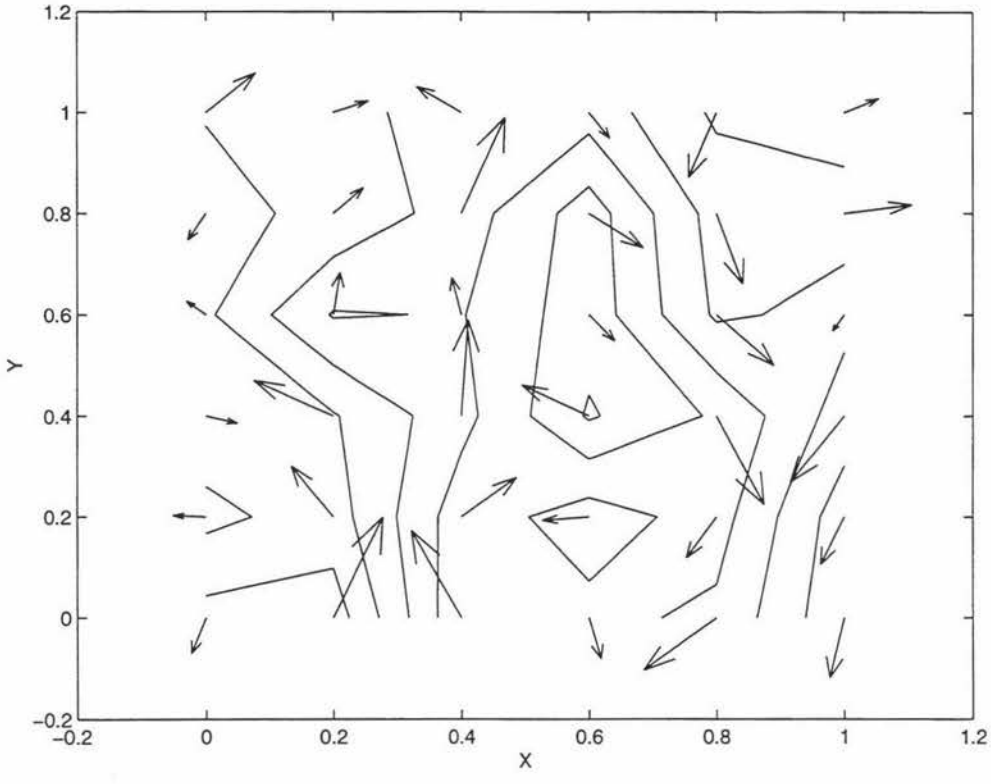


Figure 3.3: Flow through lattice sites indicated by velocity vectors and streamlines, step 3.

substitution immediately completes the representative allocations for the current block. But if the only free elements are found in sets with other representatives, then substitution frees such representatives, at least one of which must also occur in a set represented by a set $k$ element since otherwise the assumption that a distinct representative of set $k$ exists would be false.

And if there are no free elements available for substitution, none of the set $k$ elements can be freed to represent set $k$ so that there is in fact no complete set of distinct representatives. This completes the description of the distinct representatives selection algorithm.

Discussion of the success or otherwise of Model A is best postponed until it can be compared with that of Model B which comes next.

# Chapter 4

# An Eulerian Permutation Model

Initial tests of Model A showed it to be slow in operation, particularly when large arrays were involved, so efforts were channelled into another model, called Model B, which had been largely created even before work on Model A began. A new, faster permutation process was added to the existing Model B but the assumptions about the fluid remained the same as for Model A.

Model B had a single square lattice consisting of arrays with periodic boundaries (see figure 1.4 on page 10). Unlike Model A, each boundary consisted of two adjacent and parallel rows of sites, the "inner" of which was copied to the "outer" row of the opposite boundary, and likewise for the inner row of the opposite boundary. This double copying was made necessary by the finite difference numerical approximation technique used to update the vorticity at each site, and ensured that the boundaries remained periodic.

The flow, represented as before by permuting values of the vorticity function, was approximated by calculating a time differential rate of change of

vorticity at each site, based upon the flow behaviour in a volume encompassed by neighbouring sites.

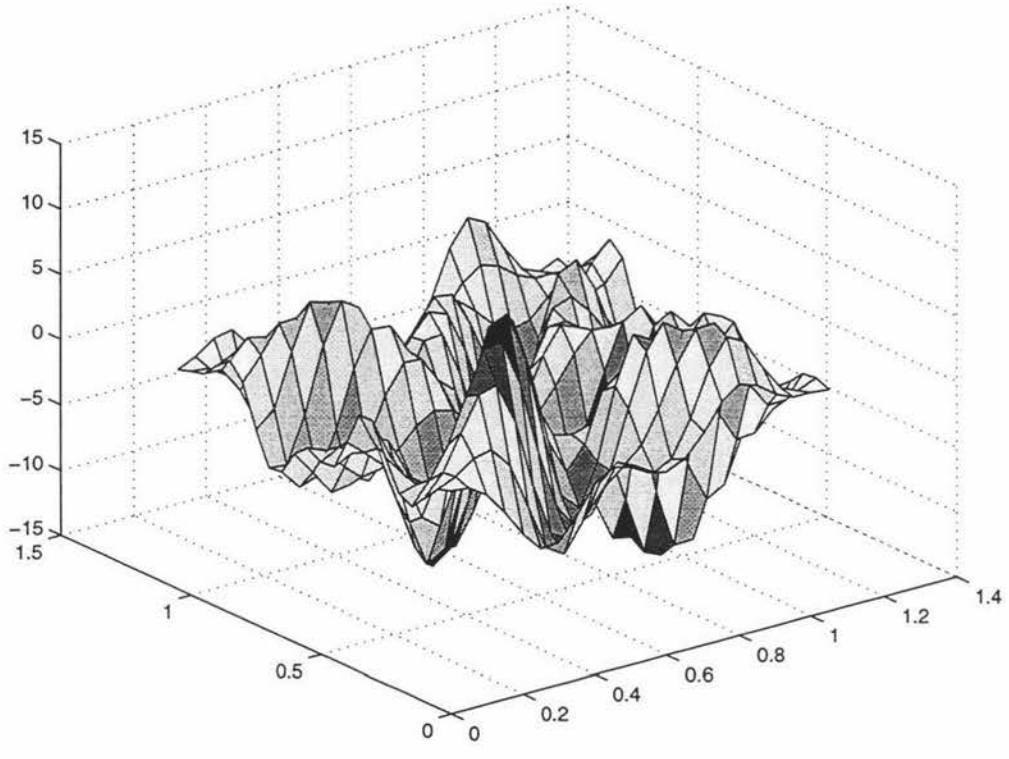


Figure 4.1: Vorticity "surface" indicating vorticity gradients.

Recalling the discussion in the Introduction,

$$\boldsymbol{V} = (\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x}) \text{ (see 1.5 on page 3)} \tag{4.1}$$

which implies

$$(\boldsymbol{V}\cdot\nabla)\omega = \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y} \tag{4.2}$$

$$= J(\omega, \psi), \tag{4.3}$$

the Jacobian determinant of $\psi$ and $\omega$. Therefore,

$$\frac{D\omega}{Dt} = \frac{\partial \omega}{\partial t} + J(\omega, \psi) \text{ (see (1.13) on page 5)} \tag{4.4}$$

and since $D\omega/Dt = 0$, it is immediate that $\partial\omega/\partial t = -J(\omega, \psi)$. According to the second order central difference approximation,

$$\frac{\partial\omega}{\partial t} \approx \frac{(\omega(t+\Delta t) - \omega(t-\Delta t))}{2\Delta t} \tag{4.5}$$

for the time increment $\Delta t$ (compare (1.29) on page 9), so

$$\frac{(\omega(t+\Delta t) - \omega(t-\Delta t))}{2\Delta t} \approx -J(\omega, \psi). \tag{4.6}$$

Transposing terms,

$$\omega(t+\Delta t) \approx \omega(t-\Delta t) - 2\Delta t J(\omega, \psi) \tag{4.7}$$

so in order to update vorticities by this formula it is required to find at least an approximation of $J$.

Let the minimum separation of sites in directions parallel to the lattice lines be a unit distance. A well-known procedure for finding the approximate value of $J$ at a central site $(i, j)$ surrounded by the eight square lattice sites adjacent to it involves firstly finding the second order central difference approximations of the terms $\frac{\partial\omega}{\partial x}$, $\frac{\partial\omega}{\partial y}$, $\frac{\partial\psi}{\partial x}$ and $\frac{\partial\psi}{\partial y}$ at the central site, using the four geometrically nearest of the eight neighbouring sites. That is, partial derivatives at $(i, j)$ are:

$$\frac{\partial\psi}{\partial y} \mid (i,j) \approx \psi_{ij+1} - \psi_{ij-1}$$

divided by the volume (distance), 2, between the two points $(i, j+1)$ and $(i, j-1)$,

$$\frac{\partial\omega}{\partial x} \mid (i,j) \approx \omega_{i+1j} - \omega_{i-1j}$$

divided by the volume between these two new points,

$$\frac{\partial \psi}{\partial x} \mid (i,j) \approx \psi_{i+1j} - \psi_{i-1j}$$

divided by the volume between these points, and

$$\frac{\partial \omega}{\partial y} \mid (i,j) \approx \omega_{ij+1} - \omega_{ij-1}$$

divided by the volume between these points.

Therefore,

$$J(\psi,\omega) \mid (i,j) = \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y} \mid (i,j) \tag{4.8}$$

is approximately equivalent to a summation of the "$\psi\omega$" terms
$+\psi_{ij+1}\omega_{i+1j}$, $-\psi_{ij+1}\omega_{i-1j}$, $-\psi_{ij-1}\omega_{i+1j}$, $+\psi_{ij-1}\omega_{i-1j}$, $-\psi_{i+1j}\omega_{ij+1}$,
$+\psi_{i+1j}\omega_{ij-1}$, $+\psi_{i-1j}\omega_{ij+1}$ and $-\psi_{i-1j}\omega_{ij-1}$, evaluated at the indicated points on the perimeter of the square region defined by the eight neighbours of $(i,j)$, when the sum is divided by the volume (area) $2^2$ of the region within the perimeter.

Unfortunately, if the approximation is based on just four sites neighbouring each central site, the model does not conserve energy and vorticity to a satisfactory approximation. Greater accuracy can be achieved by including more of the neighbouring sites, and further "$\psi\omega$" terms in the summation to acquire enhanced information on the behaviour of the flow near the central

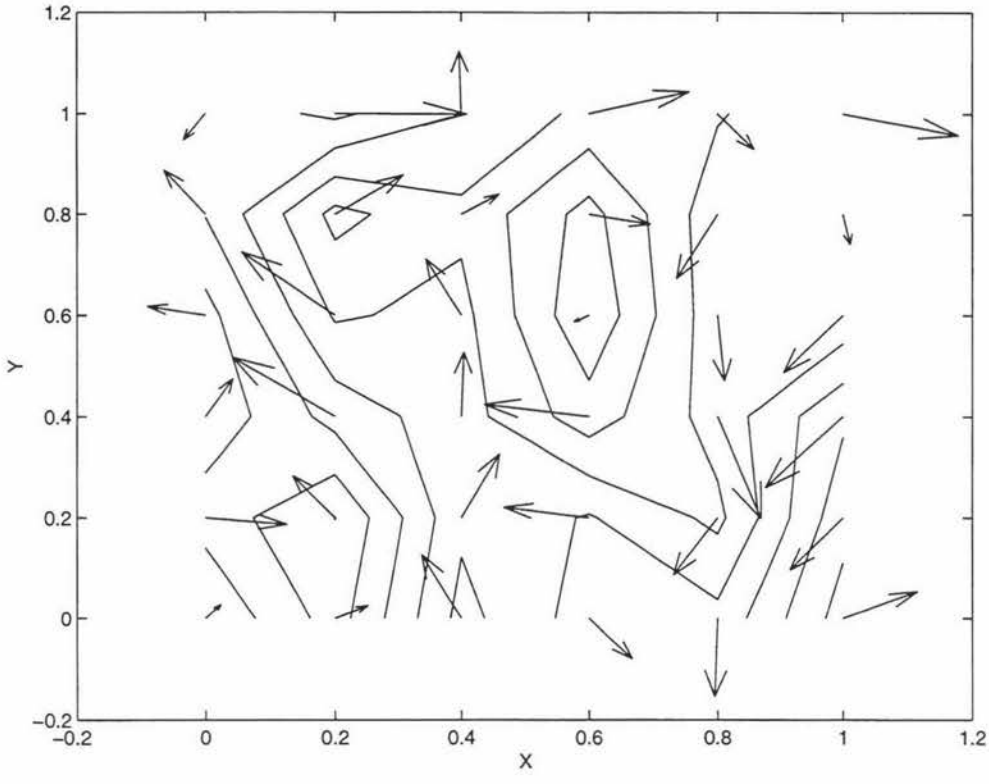


Figure 4.2: Flow through lattice sites indicated by velocity vectors and streamlines, step 4.

site, and if all eight of the neighbour sites are used then similar computations to the above more nearly achieve conservation. However, since it was intended to approximate a fluid flow by means of a bijection, a permutation of values representing particles at lattice sites had at this stage yet to be found.

As previously, the lattice sites were permuted to obtain the desired effect but the permutation mechanism was different. Starting with an initial value of the vorticity function, it was updated through the method just explained. Two vectors $P$ and $Q$ were then created: $P$ of the initial vorticities and $Q$

of the updated vorticities. Next the elements in the two vectors were sorted while an index for each monitored the original positions of the elements in the unsorted vectors, and hence of their original positions in the plane. The $j$th elements in each sorted vector could be expected to have similar magnitudes unless vorticity gradients were steep (see figure 4.1 on page 35).

Mapping the $j$th initial vorticity to the $j$th updated vorticity in the sorted vectors would have the effect of permuting the lattice sites, through the index which related sites to sorted vorticities.

In practice, the speed of Model B was increased by matching initial vorticites to the latest updated vorticities only after every few iterations rather than at every iteration. The energy and other monitoring variables were updated at every iteration by Eulerian calculations as were the vorticities, but the vorticities were periodically restored to their original values by the permutation, though with a possibly different spatial configuration, every few iterations during the permutation process. Another benefit of performing the bijection less often was that a slow flow could develop sufficiently for the identity permutation to be avoided.

Probably the major drawback of this method is that two updated vorticities can be nearly equal despite being located at sites far apart, encouraging one of the two corresponding initial vorticities to be matched to the less realistically representative option of the two updated vorticities, with the result that the flow can be portrayed incorrectly. There is relatively little chance of this happening however since the assumed continuity of a flow favours progressive change in vorticity magnitude between nearby points, with few if any sudden inflations or diminutions, and similarly there should be gradual

changes in time. When, as in such a case, the vorticity profile is relatively flat and nearly horizontal one would expect a vorticity at any site to have magnitudes similar to those of the updated vorticities in a "near" vicinity of the site. Only if vorticity gradients are very steep is the likelihood of mismatching significantly increased, because ridges and ravines in the vorticity landscape favour more frequent instances of similar vorticity values separated by both small and large distances. As would be expected from the transport of invariant vorticities the bijection maps sites occupied by initial vorticities to sites occupied by updated vorticities of similar magnitude, but a wary eye must be propped ajar to catch curious flows.

# Chapter 5

# Explorations

The behaviour of the two constructed permutation models was investigated to see if

a) they coped well with various parameter settings, and

b) they conserved quantities as required by the assumptions.

**a) parameter settings:**

Neither Model A nor Model B should indicate any activity if it is to accurately represent the flow of a fluid that is at rest. In this situation it might be expected to choose the identity permutation.

The finemesh lattice of Model A must be of strictly finer mesh than the coarsemesh lattice, yet its mesh size must clearly be greater than zero. Thus there is no guarantee that Model A will choose the identity permutation unless remedial steps are taken, and this can lead to a paradoxically nonzero representation of a zero velocity field. It is precisely when the velocity field is

zero that the candidate sets of Model A have the maximum possible size: nine elements for a square lattice, or seven if a triangular lattice is used, and when the fluid is stationary the likelihood of choosing the identity permutation is minimal. Model B whose permutation selection process relies on matching vorticity values rather than site coordinates, and is therefore immediately aware of a zero velocity field, does not have this problem.

In order to prevent manifestations of this contradiction, the computer programme for Model A recognises a zero velocity field, displays its configuration and then simply bypasses the permutation selection process to avoid erroneous modification.

Conversely, if the time step $t$ is too small the fluid may travel such a short distance that neither Model may detect the motion, and erroneously select the identity permutation. For a sufficiently small value of $t$, the candidate set for each coarsemesh site of Model A will certainly include the coordinates of that site, which suggests that something greater than or equal to a critical minimum value should be assigned to $t$ if the identity permutation is to be avoided.

Let the neighbourhood of a square coarsemesh lattice site be the square area containing the site and which almost touches the eight adjacent sites (see figure 3.2 on page 27). If the images of all the finemesh sites in the neighbourhood are more than halfway to the boundary of the neighbourhood from the central site, then none of these images can be truncated by numerical rounding to the central site. If all the images are to be more than halfway with certainty, then the least distance moved by the fluid in time interval $t$ must be greater than three quarters of the greatest possible distance across

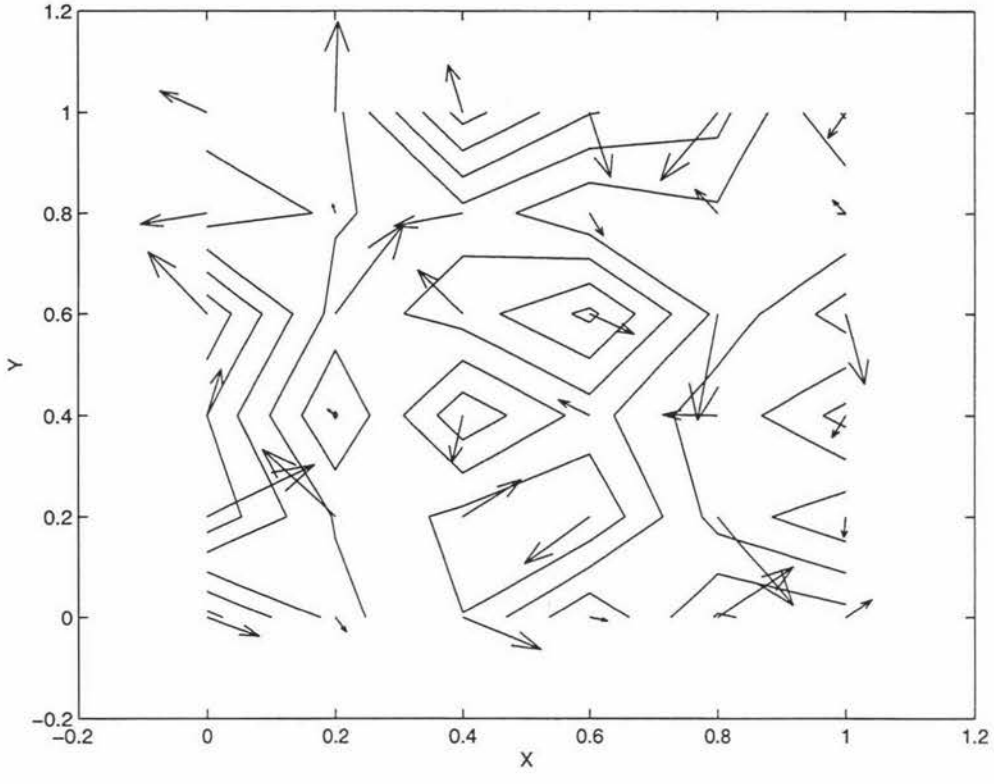


Figure 5.1: Flow through lattice sites indicated by velocity vectors and streamlines, step 5.

the neighbourhood, namely its diagonal, $D$. The assumption that at least this minimum distance is traversed by the fluid in time $t$ at a speed $V$ is equivalent to the statement $tV >= 0.75D$, or

$$t >= \frac{0.75D}{V} \tag{5.1}$$

if $V$ is non zero, as can be assumed as otherwise the above paradox avoidance routine would ensure that Model A would not continue. Accordingly, Model A presents a recommendation to its human operator suggesting a minimum value of $t$ based on this calculation.

The question may be asked whether, for either Model in fact, there is some maximum value that the time interval $t$ should not exceed. Both use square arrays, each edge of which has length $L$, and periodic boundary conditions. Translations parallel to the lattice lines are therefore carried out modulo $L$ and have absolute magnitude less than $L$, which defines the maximum useful value of $t$. That is, when operating Model A it is sufficient to choose $t$ so that

$$t < min\{\frac{L}{|u|}, \frac{L}{|v|}\} \tag{5.2}$$

for a nonzero velocity field.

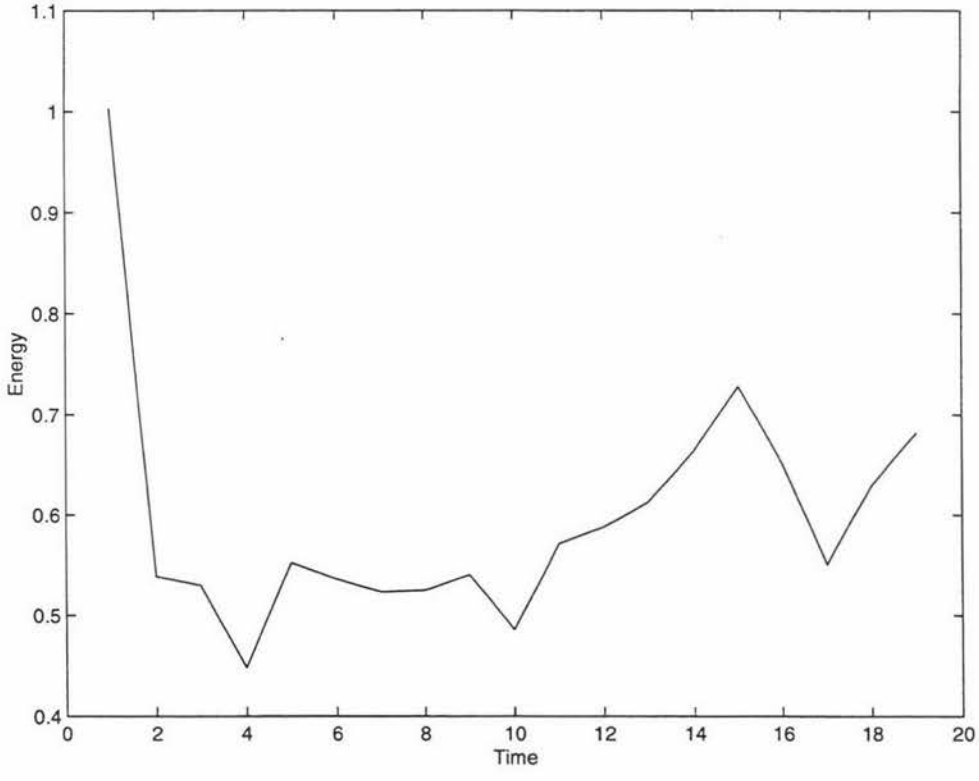


Figure 5.2: Model A energy

By the second order central difference numerical approximation method of Model B (see (4.7) on page 36), a zero value of $\Delta t$ corresponds to no change in the vorticity value at each site, so that the bijection maps each lattice site to itself and absence of flow activity is accurately noted. Any nonzero value of $\Delta t$ in Model B can result in selection of the identity permutation when there is a nonzero velocity field, depending on the nature of the flow, but this is unlikely unless $\Delta t$ is close to zero when there will be little difference between an initial and updated vorticity at each site which is therefore likely to be mapped to itself.

What of the spatial increment $h$ where $h$ is the minimum distance between two of the finemesh lattice sites of Model A? In the limit as $h$ approaches zero, space becomes dense with lattice sites; a situation alien to an electronic computer so $h > 0$. In practice, a nonzero value was assigned to $h$ by specifying a nonzero number $N$ of sites along the edge of each array. Enough finemesh and coarsemesh sites, and thus enough distinct elements, had to be available to Model A for it to be able to select a set of distinct representatives of the candidate sets, and four finemesh sites on a lattice line between two coarsemesh sites was found to be sufficient for the initial iteration at least.

Even then, when Model A had been running for a short while, it often encountered a point in time at which it was unable to construct a permutation due to an insufficient number or distribution of sites among the candidate sets, when the program had be initiated again with a finer finemesh lattice. This had the effect of increasing the probability that more distinct sites would be included in each candidate set, but even if this was successful the same problem could recur when the model had been operating for a longer

time. At times Model A succeeded in constructing a permutation which was, however the identity transformation in which event, if the Model was allowed to continue, all future permutations were most likely to be the identity.

If a point in time was reached at which every coarsemesh site was mapped to itself, it was highly likely that the velocity field had become weak and no longer transported particles far enough to allow construction of a non-identity permutation. Sudden increases in velocity were similarly improbable so that this situation most often continued indefinitely. Depending upon how long the Model was required to run, this could be a great inconvenience and again the only solution offered by Model A was to refine the finemesh.

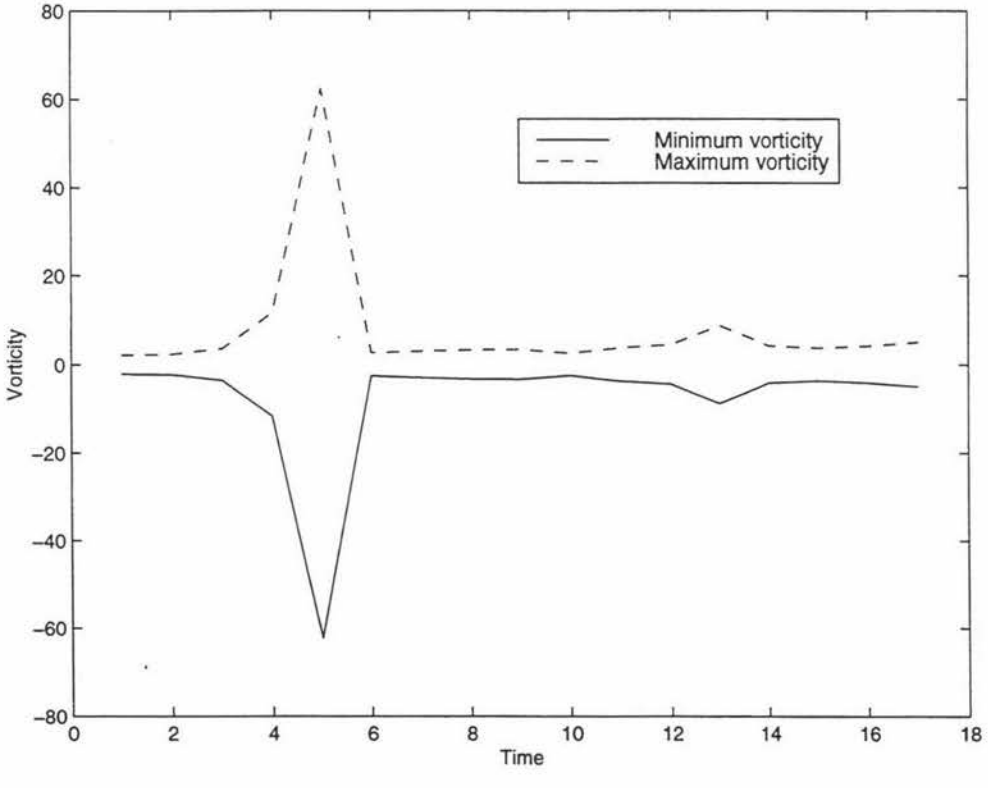


Figure 5.3: Model B maximum and minimum vorticity

Diminishing the mesh size of the fine lattice meant dealing with more elements, as did increasing the number $N$ of coarsemesh sites along each edge of the repeated array, and more time was required to process more elements. The images of the increased number of finemesh sites were truncated to coarsemesh sites: the usual one step process which did not seriously affect the performance of Model A. However, increasing $N$ slowed Model A considerably because of the larger number of comparisons between elements that the permutation selection algorithm often had to do several times over. Enlarging $N$ by two from six to eight for example may not seem much, yet for a square lattice the extra 28 sites and their associated candidate sets of up to nine elements each could brake progress to a crawl. Apart from the tedium, the dynamic picture of movement was easily lost from one iteration to the next, which was a pity as the accuracy of Model A tended to be enhanced by increasing $N$.

Although the run time of Model B increased noticeably when $N$ was enlarged, this effect was not so drastic as for Model A owing to the lesser number of operations performed by Model B in sorting and matching vorticities. Apart from its slowness, Model A was prone to collapse in the sense of either being unable to construct a permutation (due to an insufficiently broad distribution or an insufficient number of distinct elements across candidate sets), or of being able to select only the identity permutation (sluggish flow). Frequently parameters (notably the time interval and the density of the finemesh grid) had to be reset to increase the number of iterations performed by Model A.

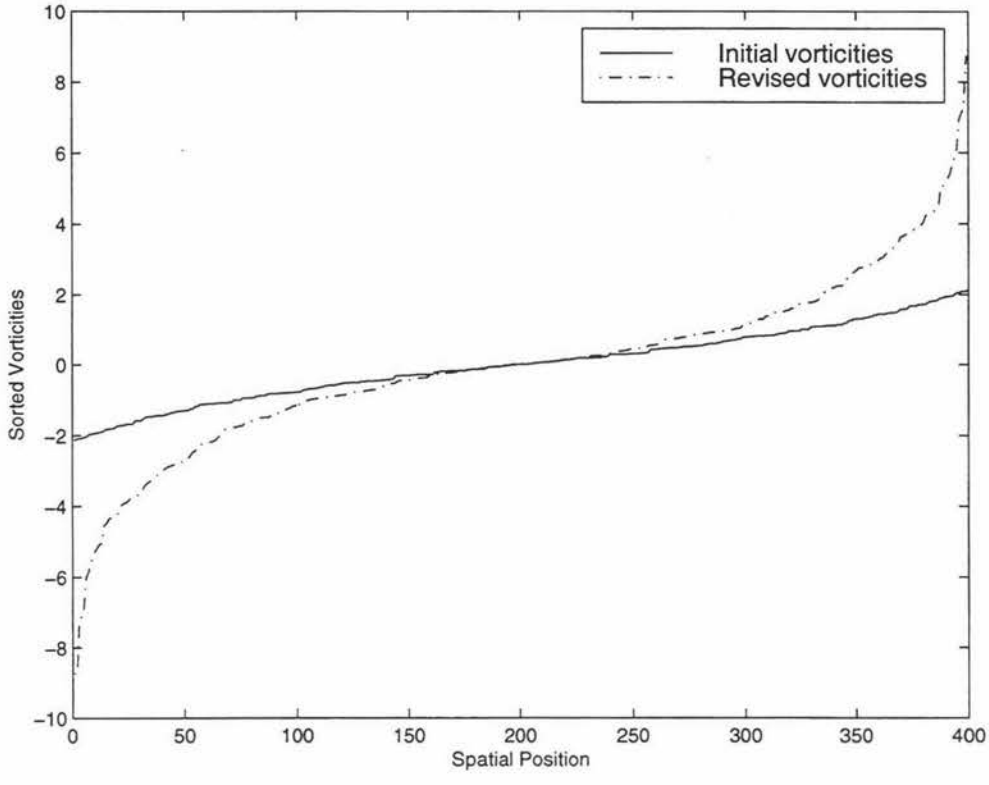


Figure 5.4: Model B consecutive distributions of sorted vorticities

**b) conservation:**

Whenever Model A did achieve a relatively large number of iterations, plots of enstrophy against time were invariably horizontal, with the computer displaying a message "axes limit range too small" and an additional comment that it would do its best to complete the plot in an environment of so little variation that it could not tell how to label the enstrophy axis with more than one value. Unlike Model B, Model A does not modify vorticity values so it is not surprising that Model A conserved vorticity. That Model A does not update vorticities can be seen by plotting first vorticities at one iteration and then vorticities at a subsequent iteration. The two graphs are the same.

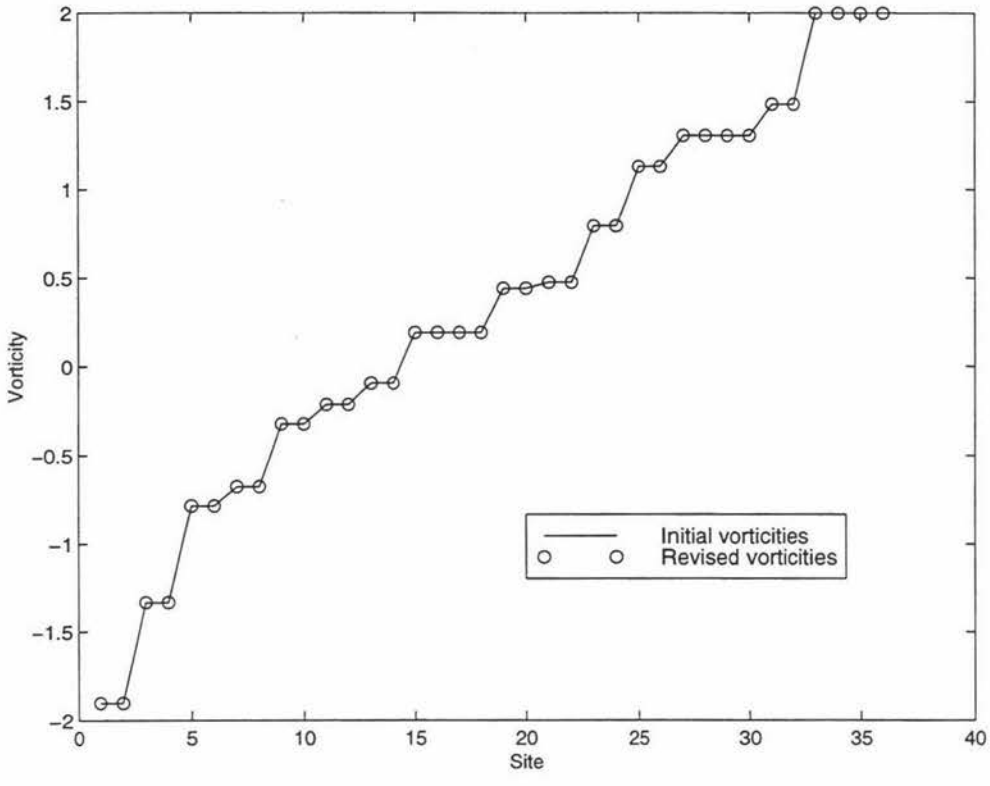


Figure 5.5: Model A consecutive distributions of vorticities

Plots of energy

$$E = \sum_i \sum_j \psi_{ij}\omega_{ij} \tag{5.3}$$

against time, were less conclusive for Model A due to error and the short span of time involved (see figure 5.2 on page 44).

Generally, tests of Model B suggested that total vorticity was conserved despite a tendency for significant increases in enstrophy. Although the fluid flow conserves enstrophy, any error in the Eulerian calculations of vorticity would be magnified by the enstrophy formula (see (1.28) on page 8 and figure 5.6 on page 50). Conservation of vorticity was indicated by plots of maximum and minimum vorticity, changes in which appeared to be mutually

compensatory at each iteration (see figure 5.3 on page 46).

In addition, plots of the distribution of vorticities at two consecutive iterations typically differed (unlike similar plots for Model A) but the total vorticity, depicted by the area beneath the plot for each iteration, remained constant (see figure 5.4 on page 48).

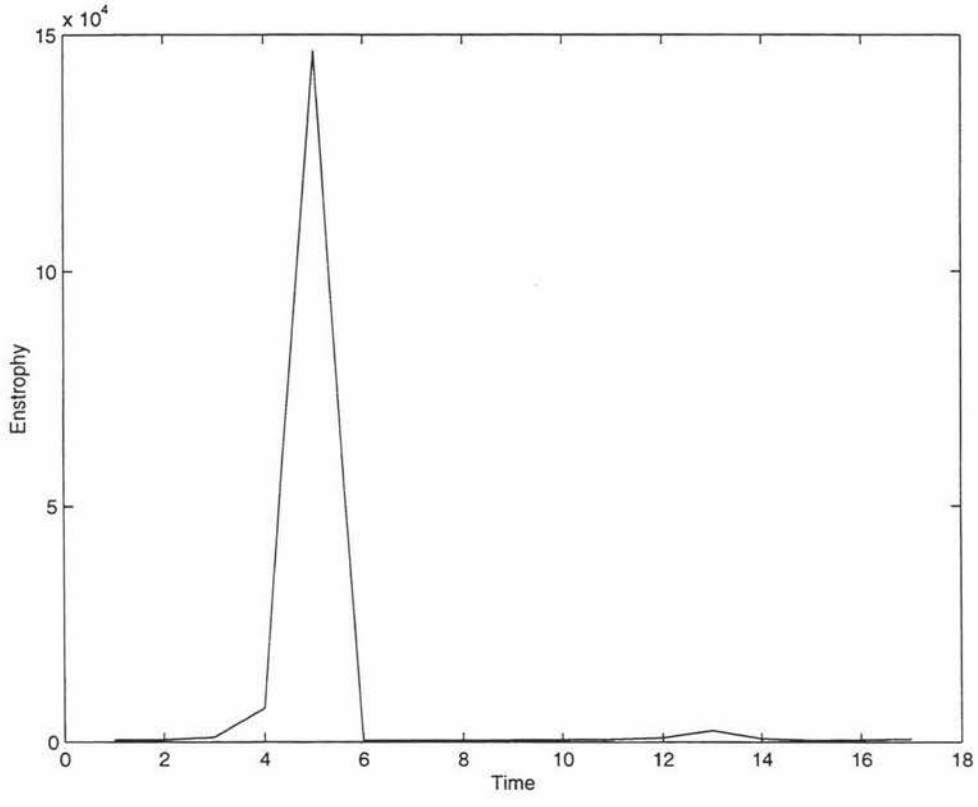


Figure 5.6: Model B enstrophy

Initial vorticity values were restored whenever the permutation was invoked, though of course their spatial configuration may then have been different to the original. Apart from the need to restore the initial vorticities so that the permutation could indicate their advection, if Eulerian calculations were allowed to continue without this restoration then the accumulating er-

ror, manifested by enstrophy fluctuations, could become so large that the computer would not be able to continue with meaningful calculations.

Model B conserved total vorticity better than it preserved energy, which rapidly became sufficiently large to yield meaningless results until the original vorticity values were restored. Again, this was probably due to compounding error. However, with frequent restoration of the initial vorticity values the total energy was artificially encouraged not to inflate indefinitely, despite its dependence on the stream function which was also regularly updated (see (1.17) on page 6, (5.3) on page 49 and figure 5.7 below).

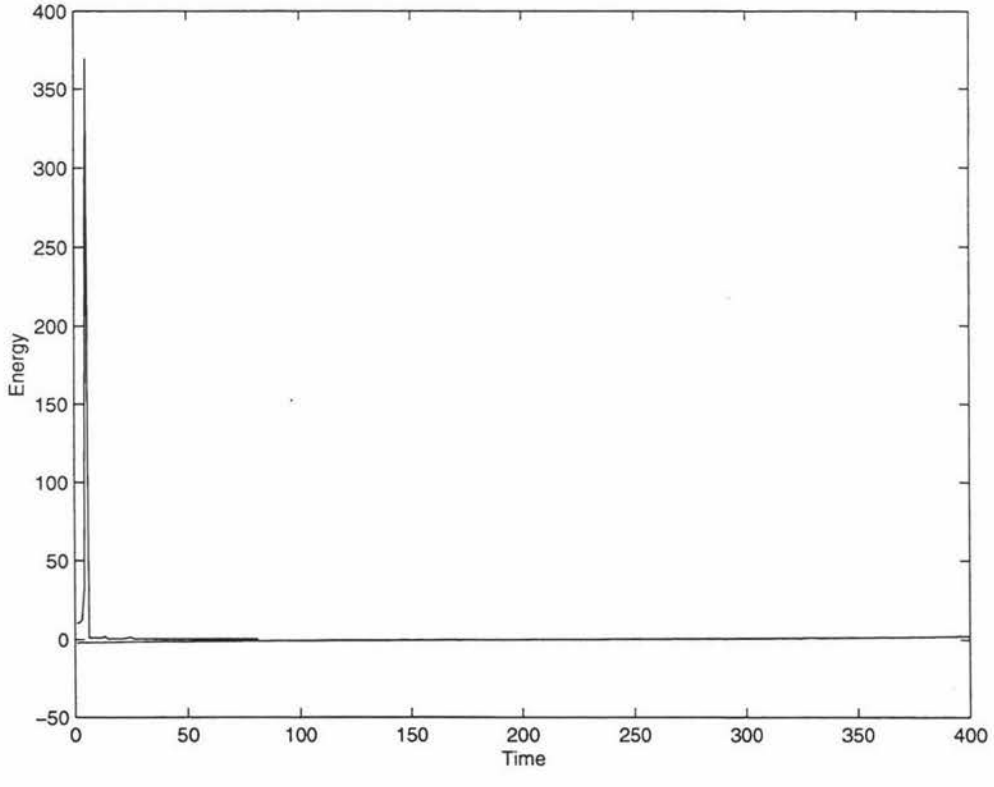


Figure 5.7: Model B energy

The maximum absolute value of isolated fluctuations due to error increased in magnitude with time because where there is a record to be broken it will be.

# Chapter 6

# Conclusion

By permuting the cells of a regular square tessellation, two attempts were made to approximately represent the flow of a fluid in artificial circumstances. Regardless of whether these were successful there is the question of whether a third such permutation model could or should have been devised. A vorticity value initially assigned to represent each cell was made to remain with the cell, so reflecting its advection if the permutation was reliable. Any representative other than vorticity should match some aspect of the flow with at least the same precision as the vorticity function, but vorticities were advected to mirror a flow with exact equivalence so there was no better alternative variable and no need to build a Model C.

Unless neither Model A nor Model B worked, but fortunately both Models exhibited aspects of the anticipated flow behaviour. As with any experiment it could be said that this concurrence was coincidental, yet judgement of the success of a model depends on a preconceived notion of what constitutes an acceptably refined approximation. The limits of acceptance, either qualitative or quantitative, must be set before an opinion about performance is

reached. Criteria established beforehand for acceptance of Models A and B were that they should appear to conserve enstrophy, total vorticity and kinetic energy besides reflecting the invertibility and area preservation of the fluid flow map $\phi$. In fact, invertibility and area preservation followed from permuting equal-area cells.

Because Model A did not alter vorticity values, it definitely conserved enstrophy and total vorticity, and although its slowness discouraged lengthy tests, there was no conclusive evidence that Model A did not conserve energy. Less certain was the preservation of enstrophy and total vorticity by Model B, although results were suggestive of conservation. Only by restoring the original vorticity values would there be a possibility that Model B could consistently conserve energy.

Lastly, the positive results obtained with both Models encourage the view that this attempt to build conservative permutation models of idealised fluid flow has been a success.

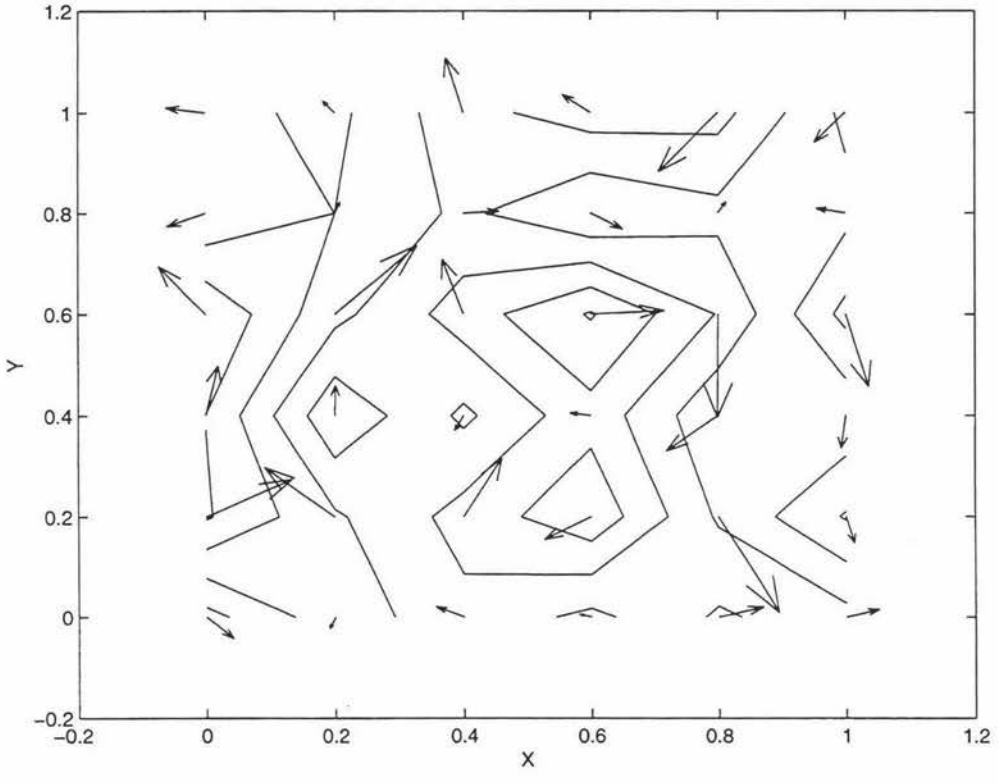


Figure 6.1: Flow through lattice sites indicated by velocity vectors and streamlines, step 6.

# Bibliography


Allen,M.P. and Tildesley,D.J.

[1] *Computer Simulation of Liquids.* New York: Oxford University Press, 1994.

Bagnoli,F.

[1] "General algorithm for two-dimensional totalistic cellular automata." *Jnl.Comp.Phys. 101* 1 (July 1992) 176–184.

Bussemaker,H.J. and Ernst,M.H.

[1] "Lattice gas automata with self-organization." *Physica A 194* (1993) 258–270.

Chorin,A.J. and Marsden,J.E.

[1] *A Mathematical Introduction to Fluid Mechanics.* 3d ed. New York: Springer-Verlag, 1993.

Codd, E.F.

[1] *Cellular Automata.* New York: Academic Press, 1968.

Coxeter,H.S.M.

[1] *Introduction to Geometry.* 2d ed. New York: John Wiley, 1969.

Feistauer,M.

[1] *Mathematical Methods in Fluid Dynamics.* New York: John Wiley, 1993.

Garzon, M.

[1] *Models of Massive Parallelism. Analysis of Cellular Automata and Neural Networks.* Berlin: Springer-Verlag, 1995.

Goles, E. and Martinez, S.

[1] *Neural and Automata Networks. Dynamical Behavio r and Applications.* Kluwer Academic Publishers, 1990.

Gutowitz, H.

[1] "A hierarchical classification of cellular automata." *Physica D 45* (1990) 136–156.

Haile, J.M.

[1] *Molecular Dynamics Simulation. Elementary Methods.* New York: John Wiley, 1992.

Hall, M.

[1] *Combinatorial Theory.* 2d ed. New York: John Wiley, 1986.

Kloeden, P.E. and Mustard, J.

[1] "Constructing permutations that approximate Lebesgue measure preserving dynamical systems under spatial discretization." *Int.Jnl.Bif.and Chaos* 7 2 (1997) 401–406.

Lax, P.D.

[1] "Approximation of measure preserving transformations." *Communications on Pure and Applied Mathematics XXIV* (1971) 133–135.

MacKinnon and others

[1] "CA models of solar flare occurrence." http://www.astro.gla.ac.uk/preprints/96-02html

McCrone,J.

[1] "Quantum states of mind." *New Scientist* (20 August 1994) 35–38.

McNamara,G.R.

[1] "Diffusion in a Lattice Gas Automaton." *Europhys.Lett. 12* 4 (1990) 329–334.

Markus,M. and Hess,B.

[1] "Isotropic cellular automata for modelling excitable media."

*Nature 347* (6 September 1990) 56–58.

Olver,P.J.

[1] *Applications of Lie Groups to Differential Equations.* 2d ed. New York: Springer-Verlag, 1993.

Packard,N.H. and Wolfram,S.

[1] "Two-dimensional cellular automata." *Jnl.Stat.Phys. 38* 5/6 (1985) 902.

Robinson,A.

[1] *Non-standard Analysis.* Amsterdam: North-Holland, 1966.

Rothman,D.H. and Zaleski,S.

[1] "Lattice-gas models of phase separation..." *Rev.Mod.Phys. 66* (4 October 1994) 1417–1479.

Stewart,I.

[1] "A new order." *New Scientist Supp.* (6 February 1993) 2–3.

Szepietowski,A.

[1] *Turing Machines with Sublogarithmic Space.* Berlin: Springer-Verlag, 1994.

Toffoli, T. and Margolus, N.

[1] *Cellular Automata Machines. A New Environment for Model ing.* Massachusetts Institute of Technology: MIT Press Series in Scientific Computation, 1987.

Toffoli and others.

[1] "Cellular-automata supercomputers for fluid-dynamics model ing." *Phys.Rev.Lett. 56* 16 (21 April 1986) 1694–1696.

Wolfram,S.

[1] "Statistical mechanics of cellular automata." *Rev.Mod.Phys. 55* 3 (July 1983) 601–644.

[2] "Cellular automata as models of complexity." *Nature 311* (4 October 1984) 419–424.