Cellular Automata based fluid simulation

Motivation: -

My personal motivation for undertaking this project is a deep appreciation of simple, minimal, and hopefully elegant, approaches to problem solving. This is tied to a curiosity about interesting computer architectures, again with a desire to avoid needless complexity or just to achieve high efficiency.

Cellular Automata are certainly a simple and minimal approach, arguably the simplest possible, so their appeal was natural. Why apply them to fluids? This is a more interesting question that will be discussed later.

What is a Cellular Automata (CA): -

There are very precise, strict, and sometimes even contentious, definitions of the requirements a model must satisfy to be considered a CA, but they can easily be defined in a more general way.

A CA consists of a lattice of 'cells', each of which is in one of a finite set of discrete states. Sets of rules are then applied the cells to evolve the system, the rules are applied to a cell and can have the local neighbouring cells states as inputs. These then output a state from the original finite set as a result, which becomes the evolved state of the cell being evaluated. The rules are applied to each cell in the lattice in turn, to create a new total system state.

Why apply a CA to a fluid?

Given the definition above, the notion of applying a CA to fluid simulation probably seems counter intuitive. In particular the phrase 'finite set of discrete states' seems an unnatural fit to the problem; a fluid is a continuum system. This only serves to make the approach more interesting, as it offers an alternative approach to the problem, which potentially may offer new insights to the system.

The common numerical approaches to computational fluids are very intensive, and subject to stability issues; potentially a CA could remove this complexity. Also on specialised hardware CA's can be run very efficiently.

How to represent a continuum with a CA: -

The most fundamental problem to be overcome is how to represent a continuum system with a set of finite states; the solution is actually quite simple. By averaging over a block of cells, the number of possible states increases – the larger the window that is averaged, the more possible states there are (See Fig.1). Therefore there is a trade off of spatial resolution against state resolution. In the limit of an infinite lattice, with infinite averages of infinite size, the CA would approach a true continuum system.

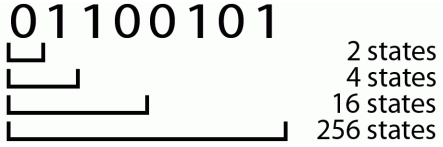


Fig.1 – Averaging the lattice

This method of generating continuum states creates an interesting situation, for a given averaged window value, there are many possible ways the finite states in that window could be arranged to generate the same result. It's possible that purely statistically some average values are more likely than others. This is seemingly analogous to the fundamental concepts of thermodynamics and statistical mechanics - partition functions and ensembles. Of course these principles have been applied to information theory also, so there seems to be a circular relationship here. This in itself is a potentially interesting topic to explore, how different finite representations of the same approximate continuum state behave.

Modelling fluids: -

While there is a vast array of literature on CA fluid models, after reviewing some, it was decided to attempt to work from first principles and derive a model from scratch in hope to gain more insight to working with CA's. The approach taken was to try and mimic a known simulation method, namely the numeric solutions to the Navier-Stokes equations.

$$(1)\quad \frac{\partial \vec{u}}{\partial t} = -(\vec{u}\cdot\nabla)\vec{u} - \frac{1}{\rho}\nabla p + \nu\nabla^2\vec{u} + \vec{F}$$

$$(2)\quad \nabla\cdot\vec{u} = 0$$

Equations 1 and 2 – Naiver-Stokes equations

Equations 1 and 2 show the Navier-Stokes equations for and incompressible, homogeneous fluid. The 4 terms on the right hand side of Equ.1 are all accelerations; the two critical terms are the first and third, self-advection and viscous diffusion. The self-advection term is particularly problematic as it is a non-linear phenomenon.

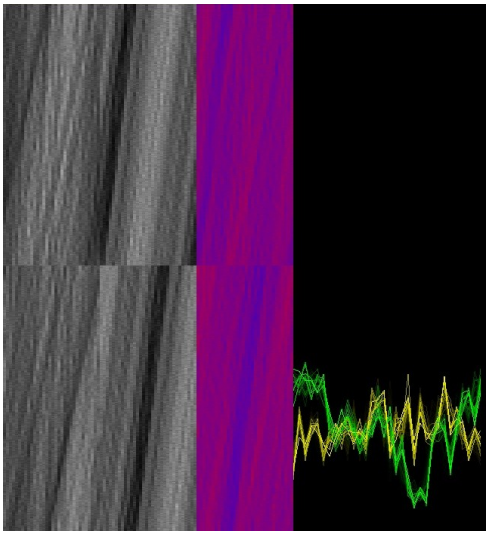
A common finite difference based approach to evolving this system simply repeatedly advects and diffuses the velocity in turn, so it seems CA models for advection and diffusion need to be found. For simplicity the 1 dimensional case was considered first before moving to 2 dimensions.

One Dimensional case: -

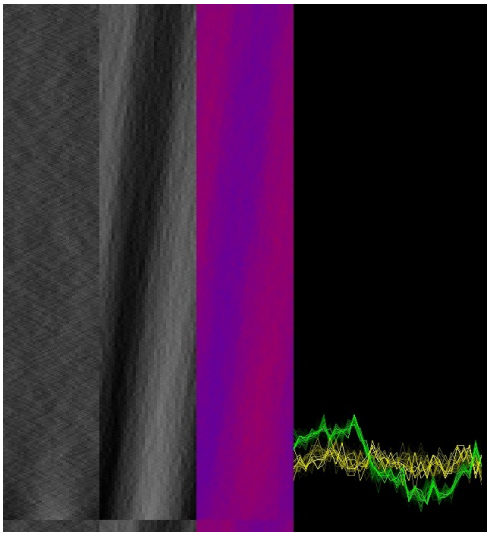
Advection in 1 dimension is simple, simply set each cell to the state of its neighbour, so the first real task was to model diffusion with a CA; two standard models where found. The first was a deterministic 3 state rule, easily implemented as a look up table, which considered two adjacent cells and permutated them based on their state. The rule was applied alternatively to even and odd pairings of cells; this even odd approach proved to be useful for all ca rules that act on non-overlapping blocks.

This model did successfully provide diffusion behaviour, but it didn't seem ideal. There always seemed to be some memory of the original distribution present

(see Fig.2) and under some conditions it resulted in oscillatory behaviour, also it seemed quite slow in terms of the rate of diffusion, requiring many time steps to be evaluated. Regardless, it remains interesting that such a simple deterministic system can approximate random behaviour.



Deterministic
Note there is still a clear trough and peak structure



Stochastic
Differing probabilities resulting in different diffusion rates

Fig.2 – Comparing Diffusion Rules

Because of these issues the rule was modified, making a simple intuitive change to the look up table – diffusion should always act from high to low gradient, this fundamental notion seemed to be violated by the original rule. While this did seem to improve the model, it still wasn't satisfactory. It's likely that this rule provided a statistical model, not a qualitative one, which is required for a visually acceptable fluid model.

The second rule was based from the very common reaction diffusion model, this is an incredibly simple stochastic CA – a stochastic CA is one where the rule can use probabilities to determine the new state. The rule is very simple - consider two adjacent cells, and swap them if a random number is less than some given value. This rule proved to be very good, giving smooth stable diffusion, the rate of which could be controlled with the probability parameter rather than just adjusting the time scale; for the step into 2d this was clearly the preferred approach.

A simple tool was made to test and compare these rules, see figure 2 for sample output. It displayed time series plots for two different CA's and then a difference of these two time series, also the current state of the system at each time step could be seen as a line plot. This tool is useful for general investigation of 1d CA's and more could have been done with it, for example comparing the different representations of the same averaged initial conditions.

Two Dimensions: -

Things become more complex when a 2 dimensional system is considered, here advection becomes the problematic term. Using a square lattice, diffusion is simple; the rule used for one dimension is easily extended into two. A 2x2 block is updated analogously to the 2 neighbouring cells in the 1d case, but the swap is

replaced by a 90° rotation to the left or right, each with a given probability. This rule gave a satisfying smooth and stable diffusion much like in one dimension

Advection is much more complex in 2d as direction need be considered; it is made worse by the fact that it is self-advection. Using the same square lattice as the diffusion each cell was given one of 5 states representing the 4 lattice directions, North, South, East, West, and a 0 velocity. The basic advection update is still relatively simple; the problem is that now two cells can 'move' into the same place. How to handle these collisions is still an open question, and it seems fundamental to the simulation, deterministic rules where tried but again stochastic behaviour seemed a better fit. The current rule simply picks a random perpendicular direction; unfortunately this doesn't seem to be so good. It seems that even with this fundamentally different approach it's the non-linear advection term that is the main source of trouble. Though the generated velocity field seems physically plausible, in that it is free from discontinuities, one has to conclude that in its current state the model just doesn't strongly resemble the desired system.

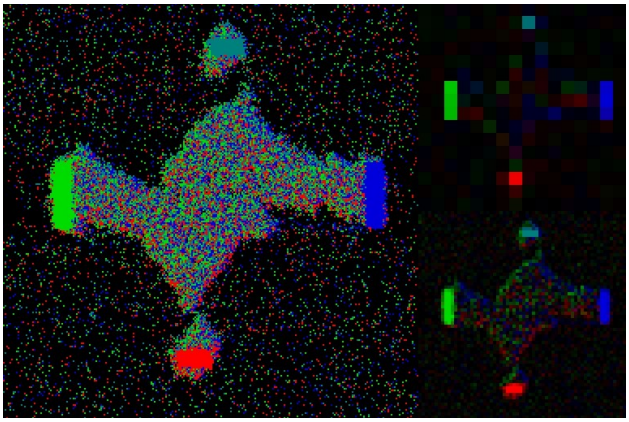


Fig 3 – Sample output from the 2d simulation, Left the discrete 512x512 CA lattice, Bottom right 62x62 average, Top right 16x16 average.

The program made to run display the simulation raised some interesting issues, particularly how to visualise a continuously changing velocity field? The approach take was to colour code the directions. Each average cell is effectively a 2d vector, the angle of this vector from the y axis was taken and used to pick a hue, a colour system that maps pure shades to $[0, 2\pi]$ range, this was then modulated by the magnitude of the vector. All of this, including the averaging down of the original state was performed on the GPU, allowing the CPU to concentrate on the quite intensive task of updating the CA.

While this successfully represents all available information in a visual way, it isn't so intuitive to analyse. One reason for this is that we aren't familiar with looking at velocity field directly, in our daily lives we are presented with the effects of velocity fields, for example an ink drop spreading in water. Perhaps it would be helpful to simulate the velocity field acting on something to get a better understanding. Other possibilities would be to try and visualise key features such as streamlines.

So why didn't it work?

There are many possible reasons the model isn't successful, certainly the seemingly inadequate advection rule has a large part to play. Also the fact that there is only one state in each direction may be a factor, as this means it is impossible to get any differential velocity in the basis directions; this just seems to be intuitively unnatural. Research on accepted CA fluid models all suggest that using a regular square lattice is not an ideal arrangement, hexagonal, triangle or even more complex lattices seem to offer more dynamic behaviour.

Likely the most significant floor can be found by going back to the original Naiver-Stokes equations, it is highly unlikely that continuity equation (2) is satisfied by this model. Commonly in finite difference approaches, in which this model has been derived to mimic, the continuity equation isn't satisfied, but pressure term in Equ.1 is often used as a free parameter to force it to be satisfied after the diffusion and advection terms. This is not considered in this model, and quite how to integrate it in the CA system remains unclear.

Conclusion: -

While the model may not of been successful, many things where learnt and avenues for further investigation have been opened. An obvious future direction would be to try and make a CA model that satisfies the continuity equation to see if it improves the results. Also trying a different lattice structure, or multiple parallel velocity states could improve the situation.

As has been previously mentioned, it would be very interesting to look at the effects of different representations of the same average initial conditions, and a very useful tool for doing this has all ready been developed with the one dimension case.