The Application of Cellular Automata to Weather Radar

Alistair Fitt* David Panton† Yanbo Wang‡

Abstract

A possible cellular automaton approach to weather (and in particular rainfall) modelling is considered. After posing a paradigm problem in a manner reminiscent of a numerical PDE solver and showing that the general approach appears to be valid, we consider some more detailed modelling and comment on how this could be used to construct a genuine finite-state cellular automaton.

1 Background Information

Numerical modelling of geographically large scale weather systems has been very successful in extending weather forecasts to periods of time of up to 5 days or more (The ill-posed nature of the governing Navier-Stokes equations mitigates against reliable forecasts for longer periods than this). For geographically small scale systems, such as localised thunderstorms, whose lifetime is on a much shorter timescale however, numerical modelling is not so effective since it appears that the dynamics of such systems are a great deal more chaotic. For such localised phenomena, weather forecasters rely heavily on radar observations to monitor the storm development. To do this, weather radar is normally used to measure the spatial distribution of raindrops suspended in air by detecting the returned signals of the transmitted electromagnetic waves. The returned signals are called weather echoes. By examining a sequence of radar images, it is possible to see how the echoes evolve and interact with one another. Depending on the weather system under consideration, the evolutionary sequence could be simple or extremely complicated. It is not uncommon to encounter cases where the echoes sequence begins with a sporadic distribution that appears to have no meaningful organisation; as time evolves, however, such echoes may merge with each other and develop into larger weather clusters that have organised structure. Under some circumstances, these clusters may remain intense and last for long periods of time. In other cases, the clusters may disperse gradually until all echoes have died away.

Many previous attempts have been made to simulate such evolution sequences. The success of artificial intelligence techniques is hard to measure and the physical foundations of such schemes are often opaque. The alternative approach of using the full Navier-Stokes equations is the standard *modus operandi* for many weather prediction programmes. Though the physical basis for such models is undeniably correct, many complications are present (see, for example [2]). Not only is it necessary to carry out computationally expensive three-dimensional unsteady calculations, but the nonlinearity of the equations means that computations with ostensibly similar initial data may give widely differing results. Many runs are thus required before a consensus can be reached.

Evidently, an approach that avoids the complexities of solving the full governing partial differential equations but still allows local predictions to be made is of great interest to weather forecasters.

* Faculty of Mathematical Studies, University of Southampton, Highfield, Southampton SO17 1BJ, UK

† Centre for Industrial and Applied Mathematics, University of South Australia, Mawson Lakes 5095 South Australia

‡ Department of Mathematics, Fudan University, Shanghai 200433, China

1.1 Project Description

The project description as defined by the Hong Kong Observatory may be stated as follows:

(i) is it possible to simulate radar image sequences using cellular automata?

(ii) if (i) is possible, how should the required cellular automaton rules be determined?

We see immediately that (i) and (ii) are may be thought of as a forward and an inverse problem, and may be posed as such:

1. Forward problem - is it possible to use a cellular automaton approach to simulate the evolution of rainstorms as shown on radar images?

2. Inverse problem - if it is, how can we deduce the underlying rules from the evolution for specific sets of rainfall data and radar images?

2 Cellular Automata and Conway's Game of Life

It does not seem impossible that rainstorm evolutionary sequences might be fairly similar to the *Game of Life*, which is played using a simple cellular automaton.

We shall first briefly review the important characteristics of a *cellular automaton* (CA). Essentially, a CA allows the evolution of simple computer graphical objects from simple distributions to possibly complex patterns, by applying a predefined set of simple rules. A genuine CA is normally considered (a) to have a structure that permits each cell to be in only a finite number of states and (b) to have a set of rules wherein the contents of each cell are changed only by the properties of the neighbouring (or nearby) cells.

In Conway's *Game of Life* (regarded by many as the "original" CA) a regular grid of cells, each of which may be considered to be in one of two states (namely either "occupied" or "unoccupied") is updated according to the following set of rules:

- if a cell has 0,1,4,5,6,7 or 8 occupied neighbours, then whatever its current state is "dies" (i.e. becomes unoccupied) at the next generation;
- if an occupied cell has exactly 2 or 3 neighbours it survives, in its occupied state, to the next generation;
- if an unoccupied cell has exactly 3 occupied neighbours it becomes occupied at the next generation.

By starting with some randomly chosen pattern of occupied cells one can play out the game of life over many generations or cycles of evolution. It can be plausibly argued that many of the structures that emerge are similar to the characteristics of tribal evolution and population movement. As well as oscillating structures of various different periods, it is possible to create objects that move with a constant speed across the computational domain ("gliders") and even objects that not only move at a constant speed, but also eject "particles" in a given direction (a "glider gun").

The ability of the *Game of Life* to simulate complex evolutionary patterns from a simple initial state using simple rules was undoubtedly a catalyst in the development of cellular automaton theory. Recently, it was claimed by Stephen Wolfram [1], that it is possible to simulate the evolution of the universe by using cellular automata with as few as 256 rules. If this somewhat dubious statement is true, then there seems to be no reason why it might not, in principle, be possible to use a CA to simulate the evolution of a sequence of radar images.

A simple observation regarding the evolution of cells and its relation to thunderstorm development is illustrated in figure 1. Figure 1(a) shows 2 separate cell systems (thunderstorms) that are bridged in the next generation as shown in figure 1(b). Such bridging, which is routinely observed in small scale weather systems can presumably be created using an appropriate set of rules such as the ones defined above.

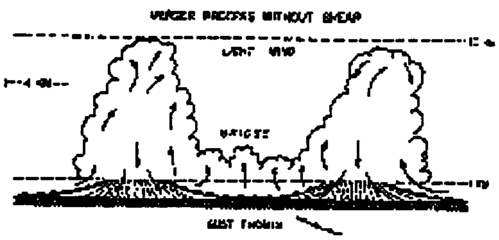

Figure 1(a)

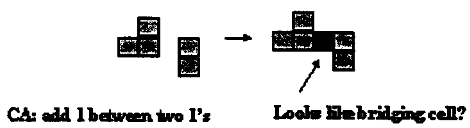

Figure 1(b)

Figure 1 shows a single aspect of the forward problem. Even more interesting is the inverse problem, i.e. given a sequence of radar images, how might it be possible to determine the rules behind the process? If this could be done, then it would be possible to develop some automatic procedures to determine the rules and predict, at least at short-range, the future development of any rainstorm event.

It is also worth noting that it does not appear as if cellular automata have previously been considered for weather modelling. Possibly the most closely-related CA application involved the prediction of the spread of forest fires ([6]). Cellular automata have also been used to model a wide range of phenomena, including cement hydration ([4]), rainforest dynamics ([3]), climate prediction ([7]) and the segregation of coal piles ([5]).

3 A Simple Model

We begin by considering a simple paradigm model for rainstorm evolution. At some stage a decision will have to be made concerning which physical properties of a rainstorm will be modelled, but the cell properties might, for example, include

- Wind velocity
- Temperature
- Pressure
- Moisture level
- Topography

After each evolutionary cycle these data must be updated to properly reflect the state of each cell. It was decided to test the first requirement with a very simple rule, namely to conserve "moisture" within each cell. (At present, we simply regard "moisture" as H_20 and do not distinguish between water vapour and raindrops.) We begin with some known initial moisture levels and wind velocity components for each cell, and trace the evolution from one cycle to the next using the conservation of moisture equation

$$\frac{\partial m}{\partial t} + (q.\nabla)m = -kH(m - m_c).$$

Here m denotes moisture fraction (i.e. the percentage of a given volume that is composed of water), $q = (u, v)$ is the wind velocity, k (dimensions s^{-1}) is a measure of moisture lost through rain and $H(.)$ is the Heaviside step function which ensures that moisture will not be lost as rain until a threshold moisture level m_c is reached.

We discretise this equation using central differences to take account of moisture in surrounding cells. Both upwind and downwind differences could also be employed, but either of these choices would only convect data in a single direction. Each cell is identified by its i, j position during generation cycle n. This yields a set of equations for each cell in the form

$$m_{ij}^{(n+1)} = \frac{1}{4}(m_{i-1,j}^{(n)} + m_{i+1,j}^{(n)} + m_{i,j-1}^{(n)} + m_{i,j+1}^{(n)}) -$$

$$\left(\frac{dt}{2}\right)[\left(\frac{u}{dx}\right)(m_{i-1,j}^{(n)} - m_{i+1,j}^{(n)}) + \left(\frac{v}{dy}\right)(m_{i,j-1}^{(n)} - m_{i,j+1}^{(n)})] - H(m_{i,j}^{(n)} - m_c)kdt.$$

Appropriate boundary conditions must be chosen which are consistent with the use of central differences. For illustrative purposes, we used the simple rule that if a cell is on the boundary then adjacent cells outside the region have the same cell value as that cell. Many other possibilities could also be considered, but probably in general some sort of "absorbing" boundary conditions should be incorporated so that weather features may pass out of the computational region without giving rise to spurious reflections.

4 Simulation Experiments

Code was written in Visual Basic to provide visual demonstrations of this simple model. In each case a grid of 100x150 was used in which it was assumed that each cell was of size 1kmx1km, ie $dx = dy = 1$. A time step of $dt = 1/3$ was used after some experimentation to ensure stability of solutions according to the standard CFL (Courant-Friedrichs-Levy) condition. A threshold moisture level of $m_c = 50$ was used with moisture loss controlled by $k = 0.05$.

Three separate scenarios were constructed, namely

1. Simple convection and moisture dissipation via rain. Up to 10 storm centres are randomly positioned on the grid each with a circular profile randomly chosen and limited to a maximum diameter of 50 cells. A wind velocity profile consistent with a south westerly was imposed on the grid.

2. As for the first scenario except for random pockets of moisture via new storm centres being created after 20 cycles, These new storm centres are randomly placed on the grid, and the range of their effect is random as in the first case.

3. A frontal line aligned from northwest to southeast is embedded in this example. The frontal line defines the boundary of a single storm system which is moving across the region from the southwest. Above this frontal line other storm systems are randomly generated in the same way described for the first two cases.

Each model was run for several evolutionary cycles until the simulated storm system event dissipated. Figure 2 shows the progression of scenario three in a sequence of four time stages. Movement of the storm front from a south westerly direction across the region can be clearly seen, along with emerging and waning localised storm systems. Although this scenario does not simulated the 'reinforcement' of storm systems, this phenomena could readily be included.

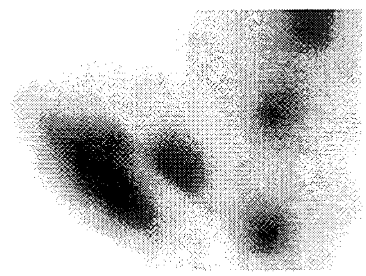

Figure 2(a)

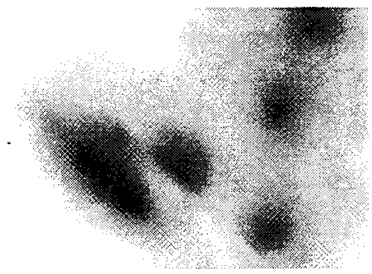

Figure 2(b)

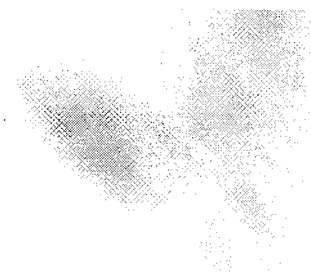

Figure 2(c)

Figure 2(d)

5 Possible Model Refinements

Thus far our approach to this problem has been virtually identical to a standard numerical PDE solving approach. It should be recalled, however, that the whole point of the specific approach that we were asked to take by the Hong Kong Observatory was to see if it was possible to employ a cellular automaton model to predict rainfall.

As noted above, the major difference between a simple numerical PDE solver and a cellular automaton is that each cell in a true cellular automaton may have only a finite number of states. Of course, one could argue that by the very nature of computers, every computer simulation actually has only a finite number of states. This is hardly the point, however, and it is generally accepted that in an 'elementary cellular automaton' each cell has only two states ('on' or 'off'), while in a more complicated cellular automaton each cell has perhaps a maximum of 10 states. (It is also generally agreed that the state of each cell may depend only on its local neighbours, but the simulations presented above evidently satisfy this anyway.)

Our approach is therefore to now propose some more involved and realistic update rules than those previously considered, which may be suited to a true cellular automaton approach. Though we shall propose rules in continuum form for simplicity, we assume that it is understood that each variable will eventually be interpreted from time step to time step as being in one of only a small number of possible states.

5.1 Model Equations

In order to propose a more detailed set of update rules, we assume again that the wind velocity $\boldsymbol{q} = (u, v)$ is given. Of course, it would be possible to further complicate the model by including equations for u and v. It is hard to see how this could be accomplished in a simple manner, however. We also assume that minimum number of variables that should be used in order to produce an accurate picture of the rainfall is three, namely

r: volume fraction of rain (water drops) in the air;

m: concentration of water vapour in clouds in the air;

T: temperature.

We now consider simple conservation rules for r, m and T. (Naturally, it may later emerge that it is necessary to keep track of more than simply these three quantities: we postpone such complications for the present, however.) We shall assume that

$$\frac{\partial r}{\partial t} + (q.\nabla)r = R_{cond} - R_{rain} \tag{1}$$
$$\frac{\partial m}{\partial t} + (q.\nabla)m = -R_{cond} - \alpha m V \tag{2}$$
$$\frac{\partial T}{\partial t} + (q.\nabla)T = -Q(R_{cond}) - \beta V. \tag{3}$$

In these equations α and β are constants, R_{cond} is a source term that arises from vapour condensing into drops (and thus $-R_{cond}$ represents drops evaporating), R_{rain} is a sink term arising from the loss of drops that actually fall as rain, V is moisture drawn up into the clouds by updraughts (and thus reflects the local topography), and Q denotes the temperature change arising from condensation.

Submodels are required for each of these quantities. Probably the simplest such models are to assume that

$$R_{cond} = \delta m H(T_0(m) - T) - \gamma r H(T - T_0(m))$$

where δ and γ are constants and H is the standard Heaviside function and $T_0(m)$ is the dew point,

$$Q(R_{cond}) = L R_{cond}$$

where L is the latent heat of the moisture,

$$R_{rain} = A(r - r_0)H(r - r_0)$$

where A is a constant and r_0 is the critical raindrop volume fraction that has to be reached for rain to start to fall,

$$V = B(T - T_v)H(T - T_v)$$

where B is a constant and T_v is the critical temperature for an updraught (thermal) to occur and finally

$$T_v = T_v(h)$$

where h reflects the local topography. The steady states of this model may easily be analysed. In particular, a steady state with $r = r_s < r_0$, $m = m_s$, $T = T_0(ms) < T_v$ (here there is no rain, drops and moisture are present in the clouds, and the temperature is below the critical updraught temperature), and also one with $r = r_s > r_0$, $m = m_s$, and $T > T_0$, $T > T_v$ (thermals are present, the temperature is relatively high, and it is raining).

6 Conclusions

In the discussion above we have only been able to give a brief idea of how a cellular automaton could be used for rainfall prediction. We have indicated how a finite-state CA could be constructed and commented on some of the decision that would have to be made during this process.

It is clear that the stakes are high in this venture: an accurate CA model for localised rainfall would be of great value in forecasting, and would render a large amount of computationally expensive three-dimensional runs unnecessary. By the same token, any simple set of rules that is able to accurately mimic the complex behaviour of rainstorms would have to encapsulate some complicated dynamics in a relatively simple manner. Clearly the details of the CA rules would have to be determined in conjunction with an intimate understanding of how storm and rainfall systems work.

One further item that must be discussed is the length scales of the approximations that are used. If a two-dimensional CA approach is to be used, then it is essential that typical vertical distances are much smaller than the size of grid used. Essentially, the CA approach requires an average to be taken in the z-direction. If, for example, one wished to model a large storm of active height 2km, then it would be very hard to pose any meaningful CA rules for cells of lateral dimensions 1km$\times$ 1km.

7 Acknowledgements

The authors of this report gratefully acknowledge the help of the many other attendees at the 1st Hong Kong Study Group with Industry who worked on this problem, and in particular the assistance provided by Malwina Luczak and John Ockendon.

References

[1] Wolfram S. Advances in Applied Mathematics, 7 (June 1986) 123-169.

[2] Bader, M.J., G.S. Forbes, J.R. Grant, R.B.E. Lilly and A.J. Waters, Images in Weather Forecasting, Cambridge U. Press, 1995

[3] Alonso D. & Sole, R.V., The DivGame Simulator: a stochastic cellular automata model of rainforest dynamics. Ecological Modelling, **133**, 131–141, (2000).

[4] Bentz, D.P., Coveney, P.V., Garboczi, E.J., Kleyn, M.F. & Stutzman, P.E., Cellular automaton simulations of cement hydration and microstructure development. Modelling and Simulation in Materials Science and Engineering, **2**, 783-808, (1994).

[5] Fitt, A.D. & Wilmott, P., Cellular-automaton model for the segregation of a two-species granular flow. Phys. Rev. A, **45**, 2383–2388, (1992).

[6] Karafyllidis, I. & Thanailakis, A., A model for predicting forest fire spreading using cellular automata. Ecological Modelling, **99**, 87–97, (1997).

[7] Palmer, T.N., A nonlinear dynamical perspective on model error: A proposal for non-local stochastic-dynamic parametrisation in weather and climate prediction models. Quart. J. Royal Met. Soc. **127**, 279–304, (2001).