

# MATH2501 Project

## Discrete Models of Competition

Felix Andrews

November 2002

### Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Cellular Automata</b>	<b>2</b>
<b>3</b>	<b>Logistic Growth</b>	<b>3</b>
<b>4</b>	<b>Intra-specific Competition</b>	<b>7</b>
<b>5</b>	<b>Conclusions</b>	<b>15</b>
<b>6</b>	<b>Extension: Inter-specific Competition</b>	<b>15</b>
<b>7</b>	<b>Bibliography</b>	<b>16</b>
<b>A</b>	<b>Appendix: Scilab code</b>	<b>16</b>

### List of Figures

1	Cellular Automata scheme for one species. . . . .	3
2	Logistic growth, $R = 0.8$ . . . . .	4
3	Logistic growth, $R = 2.2$ . . . . .	5
4	Effects of initial conditions and neighbourhood size . . . . .	6
5	Demonstration of initial condition patterns . . . . .	6
6	Intra-specific competition, $r = 1.5$ . . . . .	8
7	Convergence of CAs to the mean as grid size increases . . . . .	9
8	Intra-specific competition, $r = 2.2$ . . . . .	10
9	Intra-specific competition, $r = 3.0$ . . . . .	11
10	Emergence of stable oscillations in the global CA around $r = 2.7$ . . . . .	11
11	Oscillatory regions in the local CA . . . . .	12
12	Intra-specific competition, $r = 3.8$ . . . . .	13
13	Instability of local (radius 2) CA with $r = 5$ . . . . .	14
14	Cellular Automata scheme for two species. . . . .	16

## 1 Introduction

Simple mathematical models are often used to understand (and in some cases to predict) ecological processes such as predation, the spread of disease, and competition. The simplest models are most useful in qualitative analysis, especially stability analysis.

As is the nature of modelling, many simplifying assumptions are made. More complex and realistic models incorporate such factors as density dependence, time lag, refuge effects (non-uniformity of the environment), environmental and demographic stochasticity, interactions with other species, etc.

The models investigated here have various kinds of delay, ranging from a constant global delay to an implicit delay resulting from localised interactions. This property is achieved with the use of Cellular Automata.

## 2 Cellular Automata

A Cellular Automata (CA) is a grid of cells, each of which is in one of a finite set of states. A *transition function*, usually simple and often deterministic, defines the next state for each cell. On each iteration, or time step, every cell is updated. Ladd (1995, p.49) lists the three core properties of a CA...

**Parallelism:** On each iteration (time step) all cells are updated according to the conditions of the previous step only. In effect they are updated simultaneously.

**Locality:** The input to the transition function is the state of the cell and its local neighbours in the grid.

**Homogeneity:** The transition function uses the same algorithm for all cells in the same state.

Boundary conditions present a problem, as even edge cells need input from neighbours. Often the most precise strategy is to use a *torus* topology, where the edges wrap around transparently. However it is simpler to implement—as we will here—an *island* topology, where the edge cells remain in the empty state.

Cellular Automata can be used to model dynamic processes such as bushfires or urban sprawl, but usually only in a very general, qualitative way. Often only simple rules are needed for complex phenomena to emerge.

They may have some specific advantages for modelling simple population dynamics. The first is the fact that they are an integer-valued function: no fractional individuals can exist, as in reality. This is sometimes known as *demographic stochasticity*; we will come back to it later. The other main point is the locality of interaction on each time step, which is certainly realistic for most systems.

One problem with CA models compared with, for example, a difference equation approximating some dependent variable of interest, is the amount of computation required. On each iteration each cell must be processed, which usually involves the states of at least the 8 adjacent cells. As the grid is made finer (or bigger, depending on your interpretation), the running time scales as  $\mathcal{O}(tn^2)$  if it runs for  $t$  steps on an  $n \times n$  grid.

Using *Scilab* some significant efficiency improvements were possible. The transition functions used here operate on density. Rather than using a loop to count the neighbouring cells, summing the local submatrix proved to be 3 or 4 times faster. With this technique using a larger neighbourhood radius has a negligible effect on running time. Using a global neighbourhood is a special case which could be done with operations on the entire grid (a matrix) at once, running about 20 times faster. See the code in Appendix A.

### 3 Logistic Growth

The standard differential equation describing logistic growth is

$$\frac{dN}{dt} = RN(K - N) \quad (1)$$

where  $N$  is the population at time  $t$  and  $K$  is the “carrying capacity” of the environment.  $R$  is the *intrinsic rate of increase*, which is the difference of birth and death rates in absence of competition (Begon *et. al.*, 1996, p.247). The population grows almost exponentially when small, slowing as it converges to  $K$ . For simplicity we take  $x = N/K$ , so  $x$  is an index of the density. Then:

$$\frac{dx}{dt} = Rx(1 - x) \quad (2)$$

We can construct a homologous difference equation by discretising in time:

$$x_{t+\tau} - x_t = \tau Rx_t(1 - x_t)$$

Let  $\tau = 1$ :

$$x_{t+1} = x_t + Rx_t(1 - x_t) \quad (3)$$

From this we would like to construct a scheme which is spatially explicit and discrete—a cellular automata model where each cell represents an individual. The population dynamics of such a model should be an approximation of the “lumped” model (3). We can think of it in terms of state transitions.

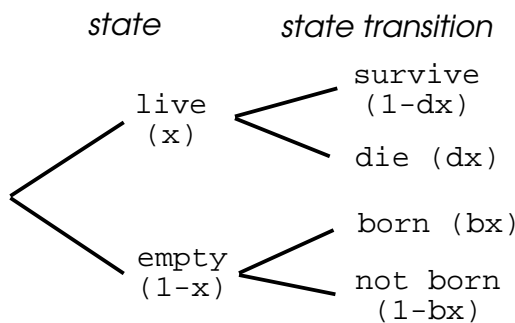


Figure 1: Cellular Automata scheme for one species.

The leftmost split in Figure 1 shows the two states in the system and their densities—this also corresponds to the probability of a cell being in that state.

For each state, the transition function determines the cell's next state according to the density  $x$ . Note that in general this is not the overall density (population) of the system, but the *local* density, within some neighbourhood of each cell. The birth and death rate factors  $b$  and  $d$  are to be determined as we will see shortly.

From this scheme we can work out the *average* density at the next time step, given the density at the current time step. The change in density is given by BIRTH - DEATH; only empty cells can experience birth, so BIRTH is  $bx$  proportion of the empty cells; only live cells can die, so DEATH is  $dx$  proportion of the live cells:

$$x_{t+1} \approx x_t + bx_t(1 - x_t) - dx_t^2 \quad (4)$$

By comparing coefficients between (4) and (3) we see that the best approximation is achieved by choosing  $b = R$  and  $d = 0$ . The logistic growth is enforced purely by the limited space of the lattice on which the cells exist. This model must be monotonically increasing (or at least non-decreasing) since there is no possibility of death occurring.

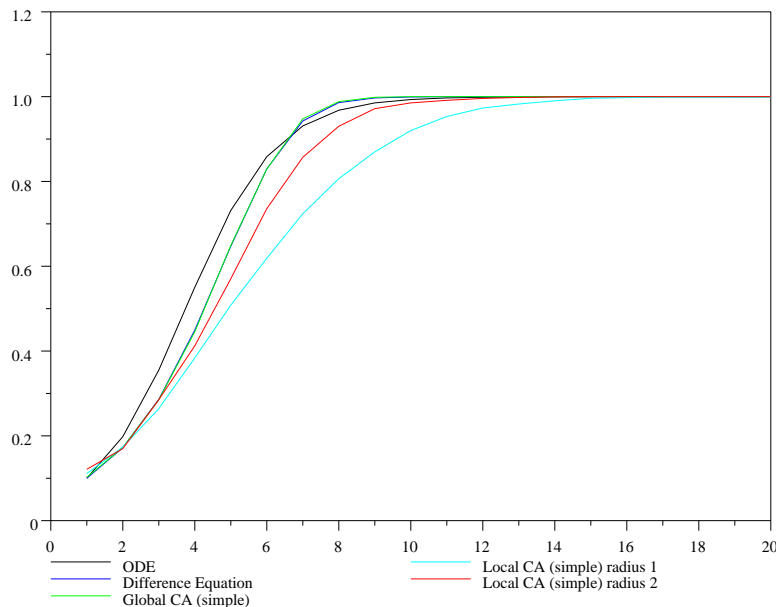


Figure 2: Simulations of logistic growth based on models (2), (3) and (4). Here  $R = 0.8$  and  $x_0 = 0.1$ . The CA grid was  $50 \times 50$  with a random initial distribution. Density is on the vertical axis and time step on the horizontal. A time step of 1 was used, even for the ODE.

This model is a reasonably good approximation of the continuous differential equation (2) (at least as good as the homologous difference equation, to graphical accuracy). It is a very good approximation of the difference equation (3) for small  $R$ . The ‘‘Global CA’’ considers the entire population when calculating

density; this represents a “well mixed system” (as in Kerr *et. al.*, 1997, p.173) where interaction and dispersal occur equally over the grid. In this scheme the spatial structure is irrelevant and it is essentially equivalent to a statistical model with demographic stochasticity. The resulting fluctuations about the mean scale with  $\mathcal{O}(N^{-1/2})$  (May, 1974, p.32), and as we have seen it should have the same mean as (3). Thus in the limit of a large population (fine grid) the density dynamics become equivalent to the deterministic difference equation (3).

The “Local CA” models show a slower convergence to the carrying capacity, which is to be expected since it takes time for the growth to propagate across the whole grid. The severity of this lag depends both on the neighbourhood size (from the most lag with radius 1 to zero lag with a global neighbourhood) and on the initial spatial distribution (the time taken to achieve equilibrium increases with a higher degree of initial clustering). This can be seen in Figure 4. The process can be understood by looking at the grids themselves—see Figure 5. Grid size is also a factor for clumped distributions: larger cell-areas of empty space take longer to propagate through. We usually use a random initial distribution partly because it is scale-invariant.

This first model is unrealistic since cells are immortal. Logistic growth is supposed to model the process of population regulation, which is a balance of

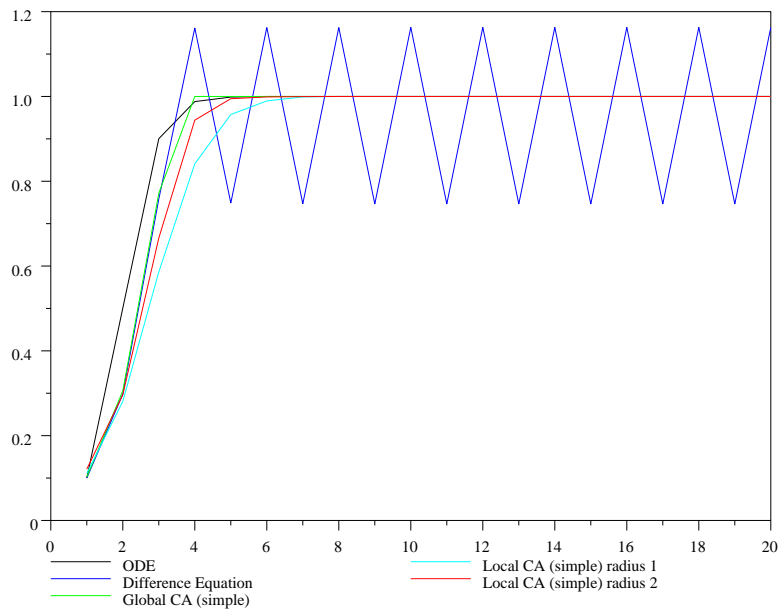


Figure 3: Simulations of logistic growth based on models (2), (3) and (4). Here  $R = 2.2$  and  $x_0 = 0.1$ . The CA grid was  $50 \times 50$  with a random initial distribution. Density is on the vertical axis and time step on the horizontal. A time step of 1 was used, even for the ODE.

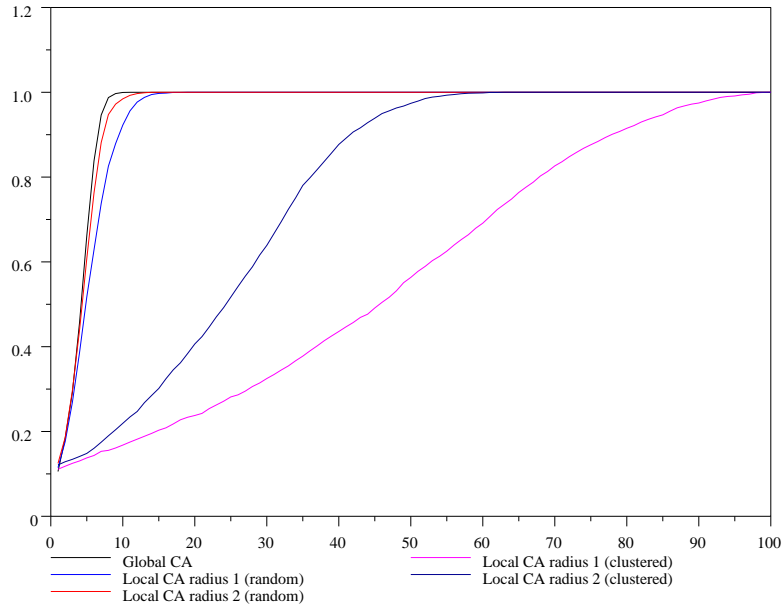


Figure 4: Simulations of logistic growth based on model (6). Here  $R = 0.8$  and  $x_0 = 0.1$ , although the initial density was distributed differently in the two cases, clustered and random (see Figure 5). The CA grid was  $50 \times 50$ . Density is on the vertical axis and time step on the horizontal.

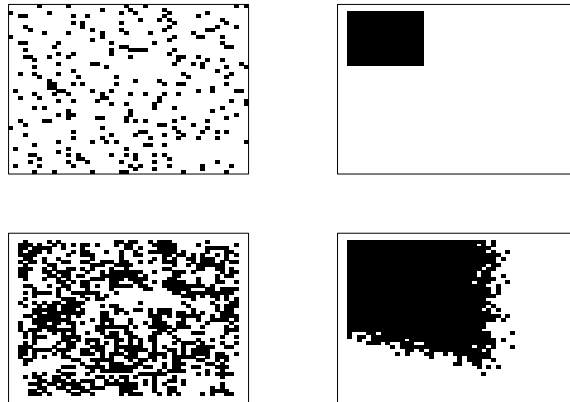


Figure 5: The effect of initial conditions on time taken to achieve equilibrium. The top grids show random and clustered initial spatial distributions, both have approximate density 0.1. At bottom-left is the random grid after just 3 iterations of the simple scheme (4) with neighbourhood radius 2. At bottom-right is the clustered grid after 20 iterations of the same scheme. This is the same situation as in Figure 4.

death and birth<sup>1</sup>. In addition we can see from Figure 3 that the difference equation (3) has qualitatively different dynamics. Since it operates on discrete time steps it is actually more analogous to a delay differential equation (Murray, 1993, p.38). Begon *et. al.* (1996, p.247) noted that the logistic differential equation (2) was “doomed to be a model of perfectly compensating density dependence.” The difference equation—with a high enough growth rate as in Figure 3—exhibits overcompensating density-dependence. Of course, they model different processes: the differential equation is for iteroparous populations (continuously breeding or overlapping generations) whereas the difference equation is for semelparous populations (non-overlapping generations) (Murray, 1993, p.36). We shall now attempt a better CA model of these dynamics.

## 4 Intra-specific Competition

A rescaled formulation of the logistic difference equation, which is easier to work with, is

$$x_{t+1} = rx_t(1 - x_t) \quad (5)$$

Note that the parameter  $r$ —controlling the rate of increase—is not the same as the previous  $R$ . Working with the CA density probability function (4):

$$\begin{aligned} x_{t+1} &\approx x_t(1 + b(1 - x_t) - dx_t) \\ &= x_t(b + 1 - x_t(b + d)) \end{aligned}$$

For  $d = 1$ :

$$x_{t+1} \approx (b + 1)x_t(1 - x_t) \quad (6)$$

This should approximate (5) for  $b = r - 1$ .

There are clearly trivial fixed points of (5) at  $x = 0$  and  $r = 0$ . The non-trivial fixed point  $x^*$  satisfies

$$\begin{aligned} x^* &= rx^*(1 - x^*) \\ \Rightarrow \frac{1}{r} &= 1 - x^* \\ \Rightarrow x^* &= 1 - \frac{1}{r} \end{aligned} \quad (7)$$

This shows that the population will become extinct for  $r \leq 1$ , and  $\lim_{r \rightarrow \infty} x^* = 1$ . This contrasts with the previous model (3) where  $x^* = 1$  for all  $R > 0$ .

Now let us examine the stability of this equilibrium. If we define  $F(x) = x_{t+1}$ , the eigenvalue of the system at this fixed point is

$$\lambda(x^*) = \left[ \frac{dF}{dx} \right]_{x=x^*} = r(1 - 2x^*)$$

---

<sup>1</sup>Migration is another fundamental process involved, but in terms of density flux it is equivalent to birth and/or death.

$$\begin{aligned}
&= r\left(1 - 2\left(1 - \frac{1}{r}\right)\right) \\
&= r - 2r + 2 \\
&= 2 - r
\end{aligned} \tag{8}$$

For stability, we require  $-1 < \lambda(x^*) < 1$ , *i.e.*  $1 < r < 3$ . If the eigenvalue is positive ( $1 < r < 2$ ), the convergence will be monotonic; if it is negative ( $2 < r < 3$ ),  $x_{i+1}$  will switch sign and we will get damped oscillations. May (1976, p.462) has shown that the system becomes unstable at  $r = 3$ , then there are stable limit cycles with period  $n$  doubling according to  $2^n \rightarrow \infty$  as  $r \rightarrow 3.5700$ , where true chaos begins. The chaos becomes more intense, with every integer period present by  $r = 3.8284$ , until  $r > 4$  when the population is driven to extinction.

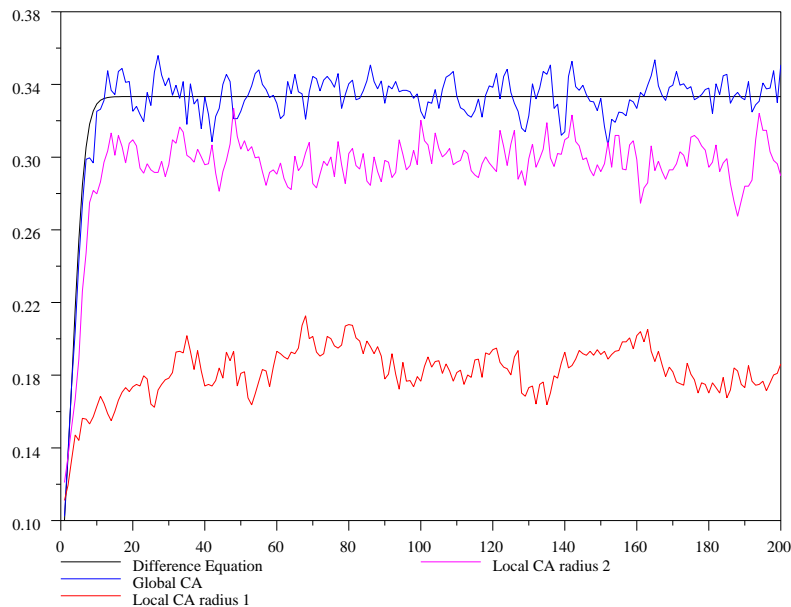


Figure 6: Simulations of single-species (intra-specific) competition based on models (5) and (6). Here  $r = 1.5$  and  $x_0 = 0.1$  on a random  $50 \times 50$  grid.

It is apparent from Figures 6 and 8 that the density predicted by the global CA is indeed centered around that of the difference equation, as we expect given the comments on demographic stochasticity above. The local CAs achieve a lower equilibrium density than the value expected for the difference equation, with the deficit greater for smaller neighbourhoods. Higher  $r$  values increase the equilibrium density, according to relation (7), but more so for the local CAs so that the effect of locality is reduced.

Figure 7 is the same situation as Figure 6 but also shows runs of each CA model with higher grid sizes. It is clear that the mean remains the same, but the variance around it decreases, as expected. The greater variance of the 1-radius-neighbourhood CA is probably due to a sort of demographic stochasticity

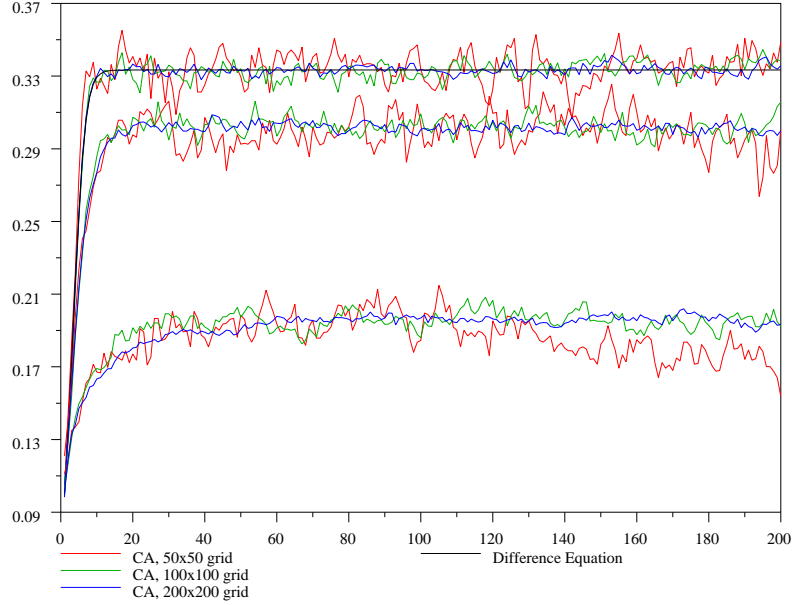


Figure 7: Convergence of CAs to the mean as grid size increases. This is the same situation as Figure 6: the highest group is the global CA, the middle group is the 2-radius CA and the lowest group is the 1-radius CA.

	Global CA	Local CA radius 2	Local CA radius 1
25x25 grid	$2.69 \times 10^{-4}$	$4.08 \times 10^{-4}$	$9.01 \times 10^{-4}$
50x50 grid	$1.45 \times 10^{-4}$	$1.69 \times 10^{-4}$	$3.43 \times 10^{-4}$
100x100 grid	$7.09 \times 10^{-5}$	$8.57 \times 10^{-5}$	$1.63 \times 10^{-4}$
200x200 grid	$3.67 \times 10^{-5}$	—	—

Table 1: Relative error per iteration of the global CA (6) compared with the difference equation (5). For the local CAs, approximate relative error was calculated in comparison to the 200x200 grid run. The time series used are the same as in Figure 7:  $r = 1.5$  and  $x_0 = 0.1$  on random grids of various sizes. The errors are averaged over 200 time steps.

effect, which would be severe since its neighbourhood sample is only 9 cells. This suggests that small populations may be more vulnerable to random fluctuations and genetic drift if their interaction is local in nature.

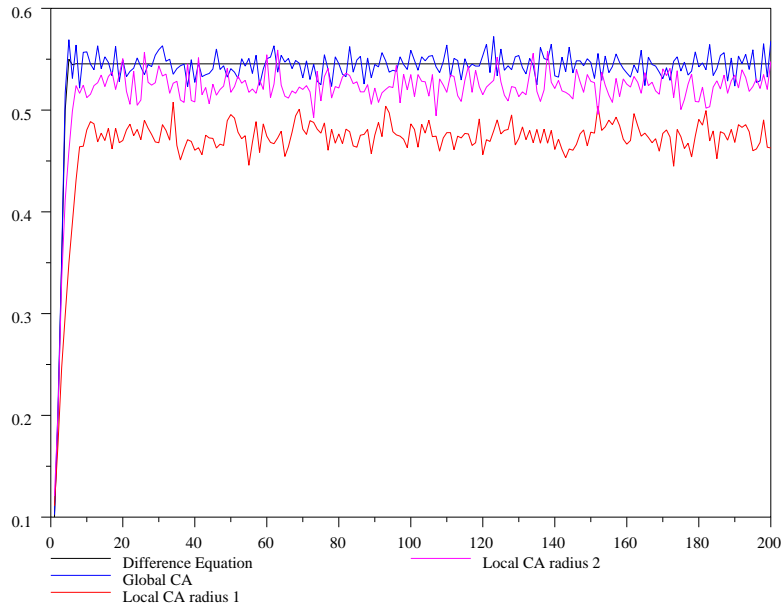


Figure 8: Simulations of single-species (intra-specific) competition based on models (5) and (6). Here  $r = 2.2$  and  $x_0 = 0.1$  on a random  $50 \times 50$  grid.

Table 1 gives the error in the global CA model as an approximation of the difference equation (5). Note that as the grid size is increased by a factor of 4, the error is reduced by only a factor of 2; that is, the error is indeed inversely proportional to the square root of population as May (1974, p.32) showed for demographic stochastic processes. The same basic pattern can be seen in the approximate errors of the local CAs, although they are centered around a lower mean.

Figure 9 shows the situation when  $r = 3$ , which is when the fixed point first becomes unstable (by equation 8). The difference equation (5) damps down initial oscillations to a lower-amplitude stable limit cycle. The Global CA corresponding to (6) does indeed show stable oscillations, of a greater amplitude. In fact, exploration revealed that stable oscillations emerged in the Global CA as early as  $r = 2.7$  (independent of grid size). Figure 10 shows the emergence of instability in the GCA. The intermediate stage has pulsations which then merge together into roughly stable limit cycles. It seems that the whole grid falls into a resonant frequency.

At this stage the local-neighbourhood CAs show no oscillations distinguishable from their usual demographic stochasticity. The general conclusion that localised population processes are more stable (over a large range of parameter values) than “well mixed” population processes is consistent with previous studies, for example Kerr *et al.* (2002). Figure 11 helps to explain why the

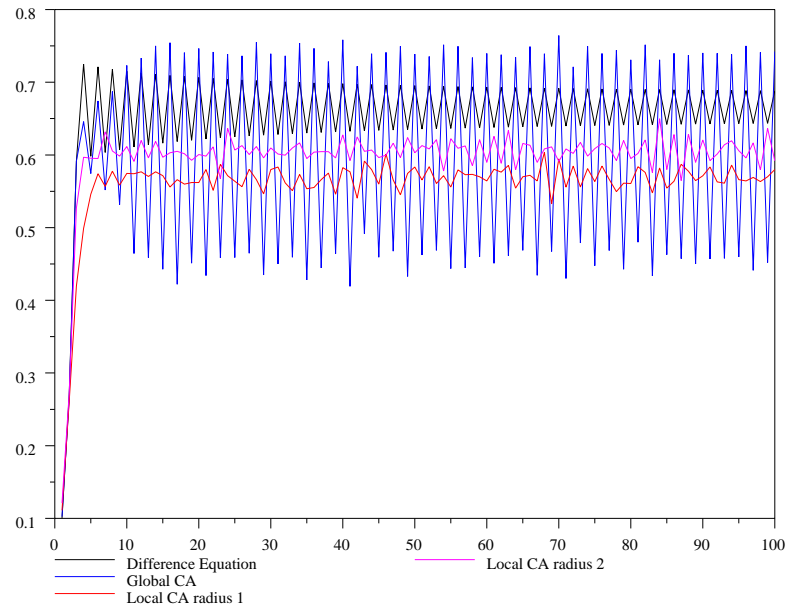


Figure 9: Simulations of single-species (intra-specific) competition based on models (5) and (6). Here  $r = 3.0$  and  $x_0 = 0.1$  on a random  $50 \times 50$  grid.

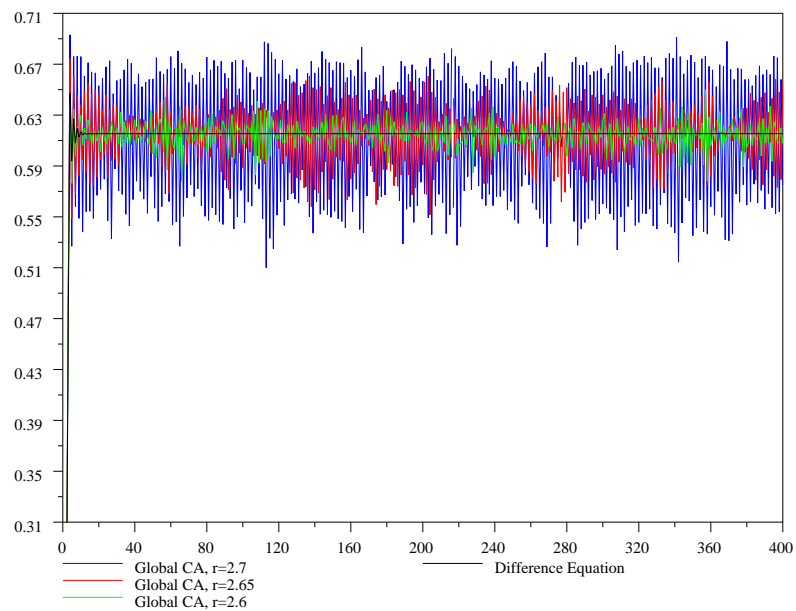


Figure 10: Emergence of stable oscillations in the global CA around  $r = 2.7$ .

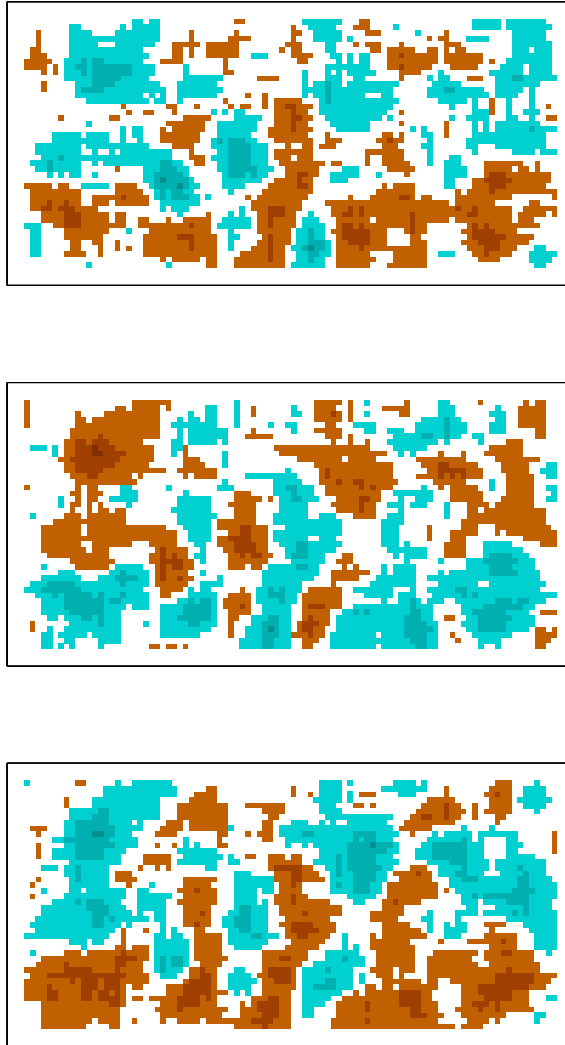


Figure 11: Oscillatory regions in the local CA. Each cell is coloured such that if its neighbourhood experienced growth in the last time step it is blue, death is brown, and no change in the neighbourhood density is white. To make the regions obvious, a radius of 3 was used. Here  $r = 3.0$  and  $x_0 = 0.3$  on a random  $50 \times 100$  grid. Time steps 2, 3 and 4 are shown in sequence.

local CAs remain stable longer than the global CA: there are in fact strong oscillations at the same  $r$  values, but they have are *localised*, with the scale of these regions related to the neighbourhood size. Since there are many regions in opposite phases of the growth/death cycle, the overall density is stable.

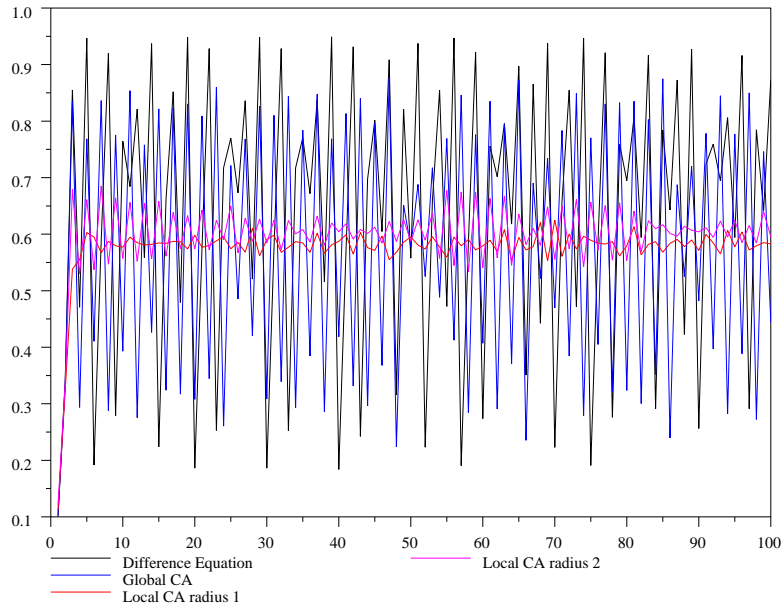


Figure 12: Simulations of single-species (intra-specific) competition based on models (5) and (6). Here  $r = 3.8$  and  $x_0 = 0.1$  on a random  $50 \times 50$  grid.

Figure 12 shows the situation when  $r = 3.8$ , which is in the region of chaotic dynamics. Corresponding to the difference equation, the Global CA also appears to be chaotic, although they do not match up on specific time steps due to the nondeterminism of the CA.

If  $r > 4$  the difference equation (5) approaches  $-\infty$  (the population goes extinct). However, in the global CA for  $r > 4$  windows of calm appear amid the chaotic oscillations.

In Figure 12 there seems to be some irregular pulsation phenomena in the 2-radius-neighbourhood CA—we might speculate that this is on a threshold where resonance can develop but dies away again. In fact for the 2-radius CA large damped oscillations appear around  $r = 4$ . By  $r = 5$  and higher there are strong oscillatory periods, but never entirely stable cycles; they collapse and reappear in an apparently random fashion. For example see Figure 13. By around  $r = 10$  it is basically stable in the sense of always having significant oscillations, although they vary in amplitude.

The 1-radius CA never goes oscillatory. With high  $r$  values it occasionally has brief pulses, seldom lasting for more than 10 iterations. Increasing the parameter beyond  $r = 9$  has no effect since at that level the birth process becomes deterministic.

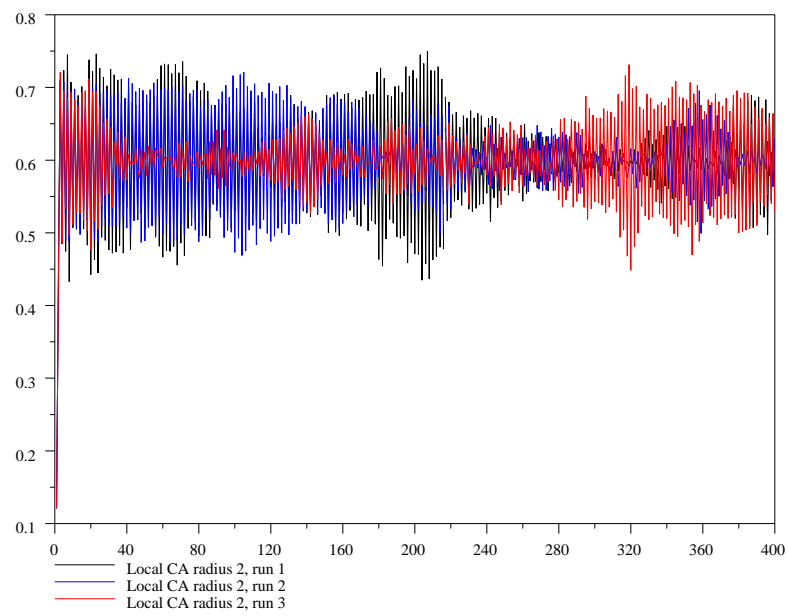


Figure 13: Instability of local (radius 2) CA. Here  $r = 5.0$  and  $x_0 = 0.1$  on a random  $50 \times 50$  grid. Differences between the runs are due to the stochastic nature of the model only, showing that at this stage the model is very unstable.

## 5 Conclusions

Discrete spatial models of intra-specific competition exhibit much more stable population dynamics than the non-spatial (lumped) discrete time model, if interaction and dispersal processes are local. On the other hand, if the system is well-mixed (a global neighbourhood for spatial processes) it is actually less stable than the difference equation.

Unanswered questions include:

- Why do the localised models reach a lower equilibrium than the global CA?
- Why does the global CA becomes unstable at a smaller  $r$  value than the difference equation it approximates?
- Do the CA models really exhibit chaotic dynamics, or just stochasticity?

## 6 Extension: Inter-specific Competition

Competition between several species is more interesting because questions of coexistence arise. The simplest model of continuous inter-specific competition is the 2-species Lotka-Volterra competition equations:

$$\begin{aligned}\frac{dx_1}{dt} &= r_1 x_1 (1 - x_1 - \alpha_{12} x_2) \\ \frac{dx_2}{dt} &= r_2 x_2 (1 - x_2 - \alpha_{21} x_1)\end{aligned}\quad (9)$$

Each species has a rate of increase  $r_i$  and a competition coefficient  $\alpha_{ij}$  which is the impact of species  $i$  on species  $j$  (measured in niche overlap). For example, if  $\alpha_{12} = 1$ , species 2 is competitively equal with species 1 over the whole niche of species 1 (*e.g.*, eats all the same food as species 1). This does not necessarily mean that  $\alpha_{21} = 1$ , as species 2 may have a wider niche. Again we can construct a homologous system of difference equations:

$$\begin{aligned}x_{1,t+1} &= x_{1,t} + r_1 x_{1,t} (1 - x_{1,t} - \alpha_{12} x_{2,t}) \\ x_{2,t+1} &= x_{2,t} + r_2 x_{2,t} (1 - x_{2,t} - \alpha_{21} x_{1,t})\end{aligned}\quad (10)$$

A general cellular automata scheme to encompass this is as follows. To simplify the diagram, only state transitions to a different state are shown.

As in the case of one species (4) we can determine the average density. Since the competition equations are symmetric, we will only consider that for species 1.

$$\begin{aligned}x_{1,t+1} &\approx x_1 + (1 - x_1 - x_2) b_1 x_1 - x_1 d_1 (x_1 + \alpha_{12} x_2) \\ &= x_1 + b_1 x_1 \left[ (1 - x_1 - x_2) - \frac{d}{b} (x_1 + \alpha_{12} x_2) \right] \\ &= x_1 + b_1 x_1 \left[ 1 - x_1 \left(1 + \frac{d}{b}\right) - x_2 \left(1 + \frac{d}{b} \alpha_{12}\right) \right]\end{aligned}\quad (11)$$

The  $x_1$  crowding term demands that  $d = 0$  for correspondence to the difference equation. However, this would eliminate the  $\alpha_{12}$  factor. This seems to be a limitation of a grid with only 3 states. We can use 2 parallel grids, one for each species (or, equivalently, have an extra state for double-occupation), which means for each species the probability of emptiness (actually *availability*) depends only on its own density. With  $d = b$  this gives

$$x_{1,t+1} \approx x_1 + b_1 x_1 (1 - 2x_1 - \alpha_{12} x_2) \quad (12)$$

We now only need to eliminate an  $x_1$  from the scheme. The approximation to (10) is achieved if each species' own density is not a factor in the state transition to death. In other words they don't die from crowding by their own kind; the effect is already incorporated by limited space.

The same approach should work for 3 or more species.

## 7 Bibliography

**Begon, M., Harper, J. L. & Townsend, C. R.** (1996) Ecology, third edition. Blackwell.

**Kerr, B., Riley, M. A., Feldman, M. W. & Bohanna, B. J. M.** (2002) Local dispersal promotes biodiversity in a real-life game of rock-paper-scissors. *Nature* Vol 418, July 2002, pp.171–174.

**Ladd, S. R.** (1995) C++ Simulations and Cellular Automata. M&T Books.

**May, R. M.** (1974) Stability and Complexity in Model Ecosystems, second edition. Princeton University Press.

**May, R. M.** (1976) Simple mathematical models with very complicated dynamics. *Nature* Vol 261, June 1976, pp.459–267.

**Murray, J. D.** (1993) Mathematical Biology, second edition. Springer.

## A Appendix: Scilab code

```
//--logistic growth--//
```

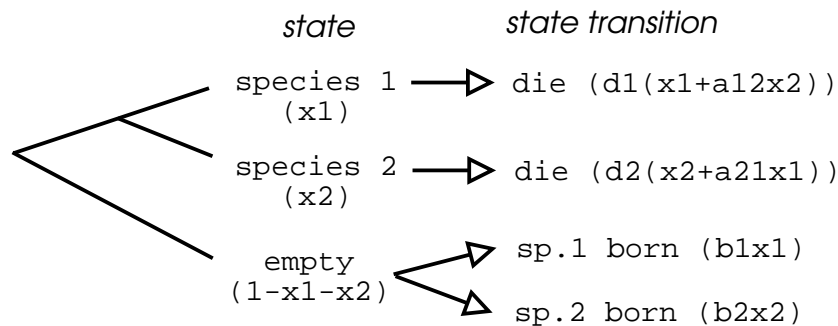


Figure 14: Cellular Automata scheme for two species.

```
function [dxdt] = onecomp_ode(t, x, R)
dxdt=R*x*(1-x);
endfunction
```

```
function [y] = onecomp_diff_simple(t, x0, R)
x=x0;
for tt=t
    y(tt)=x;
    x=x+ R*x*(1-x);
end
endfunction
```

```
function [y] = onecomp_simple(t, A, R)
cellarea = prod(size(A));
for tt=t
    y(tt)=max(size(find(A==1))) ./ cellarea;
    A=onecomp_simple_it(A, R);
end
endfunction
```

```
function [A_new] = onecomp_simple_it(A, R)
b=R;
[m, n] = size(A);
count = sum(A);
x=count / (m*n);
L=bool2s(A==1); // 1 for each live cell; 0 otherwise
E=bool2s(A==0); // 1 for each empty cell; 0 otherwise
E=E.*bool2s(rand(m,n)<(b*x)); // proportion of empty cells born
A_new=L+E;
endfunction
```

```
function [y] = onecomp_local_simple(t, A, R, radius)
cellarea = prod(size(A) - 2*radius);
for tt=t
    y(tt)=max(size(find(A==1))) ./ cellarea;
    A=onecomp_local_simple_it(A, R, radius);
end
endfunction
```

```
function [A_new] = onecomp_local_simple_it(A, R, radius)
b=R;
[m, n] = size(A);
A_new=zeros(A);
for i=(1+radius):(m-radius)
    for j=(1+radius):(n-radius)
count = sum(A(i-radius:i+radius, j-radius:j+radius));
x = count / (radius * 2 + 1)^2;
if (A(i,j)==1) then
    // live cell
```

```

    A_new(i,j)=1;
else
    // empty cell
    if (rand() < b*x) then
// born
A_new(i,j)=1;
    end
end
end
endfunction

```

```

/--intra-specific competition--//

```

```

//>plot2d(t,[x_diff,x_gca_4,x_lca1_4,x_lca2_4],leg="Difference Equation@Global CA@Local CA ra

```

```

function [y] = onecomp_diff(t, x0, r)
x=x0;
for tt=t
    y(tt)=x;
    x=r*x*(1-x);
end
endfunction

```

```

function [y] = onecomp(t, A, r)
cellarea = prod(size(A));
for tt=t
    y(tt)=max(size(find(A==1))) ./ cellarea;
    A=onecomp_it(A, r);
end
endfunction

```

```

function [A_new] = onecomp_it(A, r)
b=r-1;
[m, n] = size(A);
count = sum(A);
x=count / (m*n);
L=bool2s(A==1); // 1 for each live cell; 0 otherwise
E=bool2s(A==0); // 1 for each empty cell; 0 otherwise
L=L.*bool2s(rand(m,n)<(1-x)); // proportion of live cells die
E=E.*bool2s(rand(m,n)<(b*x)); // proportion of empty cells born
A_new=L+E;
endfunction

```

```

function [y] = onecomp_local(t, A, r, radius)
cellarea = prod(size(A) - 2*radius);
for tt=t
    y(tt)=max(size(find(A==1))) ./ cellarea;
    A=onecomp_local_it(A, r, radius);

```

```

end
endfunction

function [A_new] = onecomp_local_it(A, r, radius)
b=r-1;
[m, n] = size(A);
A_new=zeros(A);
for i=(1+radius):(m-radius)
  for j=(1+radius):(n-radius)
    count = sum(A(i-radius:i+radius, j-radius:j+radius));
    x = count / (radius * 2 + 1)^2;
    if (A(i,j)==1) then
      // live cell
      if (rand() < (1-x)) then
        // survives
        A_new(i,j)=1;
      end
    else
      // empty cell
      if (rand() < b*x) then
        // born
        A_new(i,j)=1;
      end
    end
  end
end
endfunction

//--two species competition--//
// (experimental)

// y_ode = ode(y0,t0,t,list(twocomp_ode,r,alpha));
function [dDdt] = twocomp_ode(t, D, r, alpha)
dDdt(1)=r(1)*D(1)*(1-D(1)-alpha(1,2)*D(2));
dDdt(2)=r(2)*D(2)*(1-D(2)-alpha(2,1)*D(1));
endfunction

function [y1,y2] = twocomp_diff(t, D0, r, alpha)
D1=D0(1);
D2=D0(2);
for tt=t
  D1_new=r(1)*D1*(1-D1-alpha(1,2)*D2);
  D2_new=r(2)*D2*(1-D2-alpha(2,1)*D1);
  D1=D1_new;
  D2=D2_new;
  y1(tt)=D1;
  y2(tt)=D2;
end
endfunction

```

```

function [y1,y2] = twocomp(t, A, r, alpha)
cellarea = size(A, '*');
for tt=t
    A=twocomp_it(A, r, alpha);
    y1(tt)=max(size(find(A==1))) ./ cellarea;
    y2(tt)=max(size(find(A==2))) ./ cellarea;
end
endfunction

```

// this doesn't use the new formulation yet!

```

function [A_new] = twocomp_it(A, r, alpha)
b=r-1;
[m, n] = size(A);
count1 = max(size(find(A==1)));
count2 = max(size(find(A==2)));
D1=count1 / (m*n);
D2=count2 / (m*n);

L1=bool2s(A==1); // 1 for each live sp1 cell; 0 otherwise
L2=bool2s(A==2); // 1 for each live sp2 cell; 0 otherwise
L1=L1.*bool2s(rand(m,n)<(D1 + alpha(1,2)*D2)); // proportion of live sp1 die
L2=L2.*bool2s(rand(m,n)<(D2 + alpha(2,1)*D1)); // proportion of live sp2 die
E1=bool2s(A==0); // 1 for each empty cell; 0 otherwise
E2=E1; // copy for sp2
E1=E1.*bool2s(rand(m,n)<(b(1)*D1)); // proportion of empty cells born into sp1
E2=E2.*bool2s(rand(m,n)<(b(2)*D2)); // proportion of empty cells born into sp2
X=find(E1==1 & E2==1); // get indices of any conflicting births
E=E1+(E2*2); // merge birth matrices
E(X)=round(rand(X))+1; // randomly resolve conflicts
A_new=E+L1+(L2*2);
endfunction

```