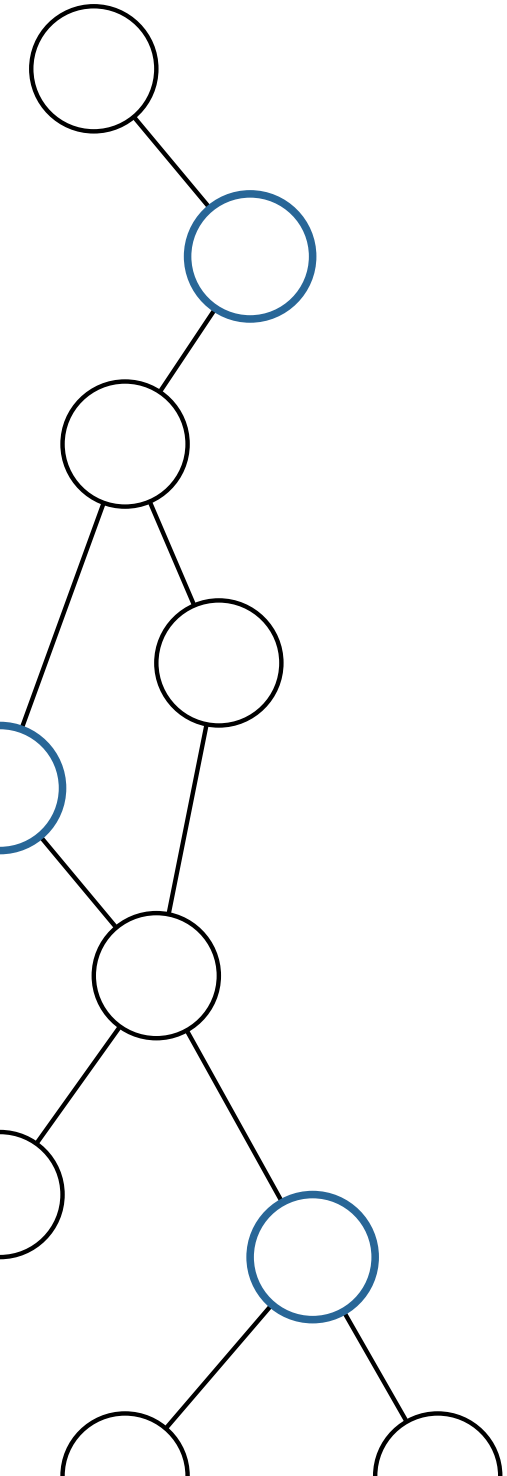


UQAR – Chaire de Recherche EEC

Spatial models in ecology

Analytical and numerical methods

Dominique Gravel

http://www.chaire-eec.uqar.ca/

August 23, 2013

UQAR

Outline

Program a random walk model. Consider a single individual whose spatial coordinates at time t_0 are $(0, 0)$. Now consider the following simple rule: each time step, pick up randomly (X, Y) coordinates among the eight immediate neighbours with a single unit movement in direction X and Y with probability 0.5.

- Plot the spreading of the individual over time
- Run the model 100 times and record the distance of the individual at time t = 10 and t = 100

Definition: random moving of particles suspended in a fluid (a liquid or a gas) resulting from their bombardment by the fast-moving atoms or molecules in the gas or liquid.

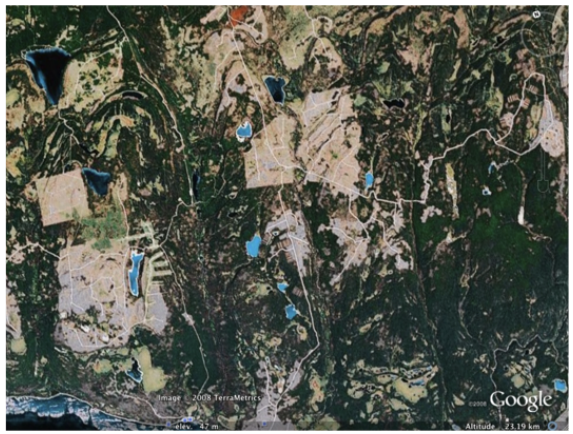
Could be use to detect non random features of random number generators.

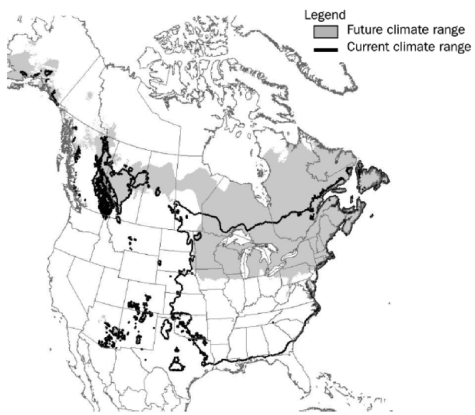
An old and rich litterature on diffusion equations and its connection to random walk process. Connecting diffusion and Brownan motion was one of Einstein's three accomplishments that earned hum his Nobel prize!

In ecology, it forms the basis of a class of spatial models. More generally it is fundamental to understand spatial spreading, of organisms and disturbances. Random walk is the driving force beind neutral models. Also used to study the structure of phylogenetic trees.

Important spatial processes in ecology

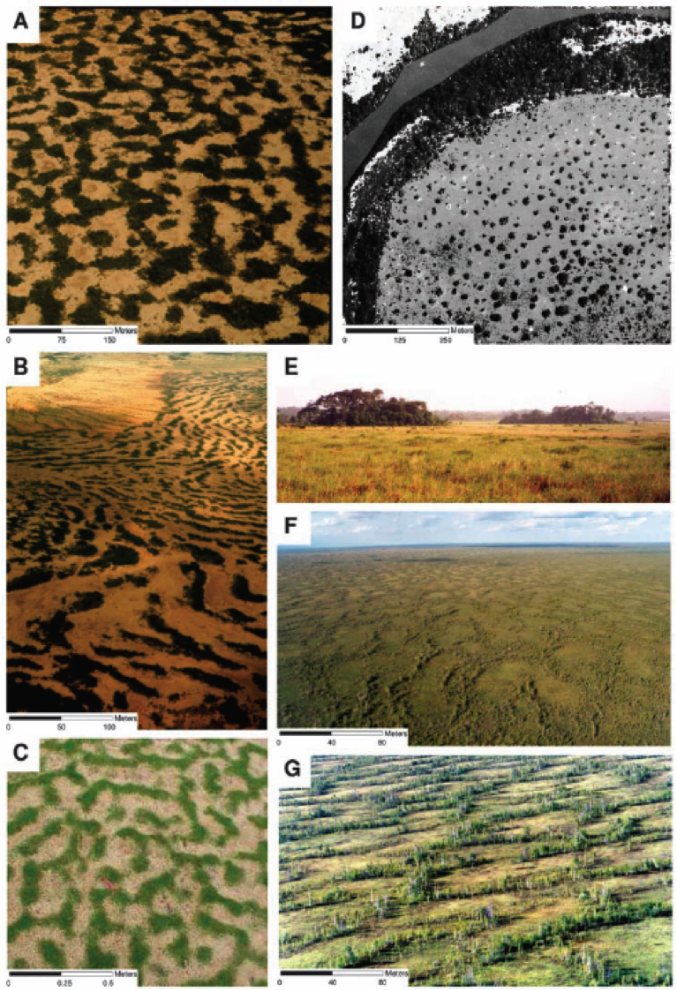
- Dispersal
- Natural disturbances
- Synchrony, Moran effect
- Spreading of invasive species
- Metapopulation dynamics, fragmentation, habitat loss
- Species distribution
- Estimating species diversity





The importance of spatial structure

- Why?
- What characteristics? (e.g. stationarity, dynamics, scale etc...)
- Impacts?



Distinguishing types of spatial models

- Space
- Time
- State variable

Distinguishing types of spatial models

Discrete space

- Patch models (2, 3,...,n)
- Landscape models
- Cellular automaton
- Lattice models

Continuous space

- Microscopic models
- Macroscopic models

Conceptual objectives

- Diffusion process
- Cellular automaton
- Metapopulation dynamics & spatially explicit dispersal

Technical objectives

- Partial differential equations
- Cellular automaton
- Analysis of spatial structure (patch size distribution)
- Introduction to root solving
- Program spatially explicit dispersal

In this section, we will derive a model expressing the change in density (dynamics) continuously over space, time and the state variable. First, consider a dispersal kernel $K(y, x)$ giving the probability that an individidual leaving position y ends up at position x.

Now suppose that individuals leaves location x at rate φ. This movement means that the number lof loss individuals from any particular location x is $\varphi n(x, t)$.

And similarly, the number of individuals going from location y to location x is $\varphi n(y, t)K(y, x)$. Note that because all individuals must land somewhere, the integral $\int K(y, x)dy$ must be equal to 1. The movement for a one dimensional system can therefore be described as follow:

$$\frac{\partial n(x,t)}{\partial t} = \varphi \int K(y, x)n(y, t)dy - \varphi n(x, t)$$

The left hand side of the equation represents the immigration, while the right hand side represents emigration. This equation is named *integrodifferential* and it has been shown that we could recover the diffusion equation from it.

We begin with a Taylor series expansion of $n(y, t)$ about position x:
$n(y, t) = n(x, t) + \frac{\partial n(x,t)}{\partial x}(y - x) + \frac{1}{2}\frac{\partial^2 n(x,t)}{\partial x^2}(y - x)^2 + ...$

And throw in equation (9.1) to get the ugliest equation ever:
$\frac{\partial n(x,t)}{\partial t} = \varphi \int K(y, x)[n(x, t) + \frac{\partial n(x,t)}{\partial x}(y - x) + \frac{1}{2}\frac{\partial^2 n(x,t)}{\partial x^2}(y - x)^2]dy - \varphi n(x, t)$

I will skip the simplifications, but just keep in mind that if movement is symetric (there is no bias toward one direction) and we respect the condition of the dispersal kernel having an integral equal to 1, then by some magical trick we obtain the nice equation (trust me, or at least Wilson 2000!):
$\frac{\partial n(x,t)}{\partial t} = D\frac{\partial^2 n}{\partial x^2}$

Isn't it convenient? Parameter $D = \varphi R^2/2$ is the diffusion coefficient, and R is the mean square displacement distance.

Diffusion tends to homogenize the distribution of organisms and therefore one solution to the diffusion equation is a constant uniform distribution, $n(x, t) = n^*$. This result is somewhat trivial, and therefore a more interesting question is *What are the temporal dynamics of a small spatial perturbation about this equilibrium solution?*

Formalizing the problem of spatial stability, we assume a spatiotemporal perturbation described by the function
$n(x, t) = n^* + n_0 e^{-t/T} sin \frac{2\pi x}{L}$

where n_0 represents the initial perturbation at time $t = 0$. The first factor $e^{-t/T}$ is a time-dependent decay function, which multiplies the amplitiude by $1/e$ when $t = T$. The factor $sin \frac{2\pi x}{L}$ represents the spatial variation of the intitial perturbation. Parameterl L is the perturbation's wavelength and represent peak to peak distance.

We now conduct a stability analyslis by substituing the latest equation for the perturbation in the diffusion model:

$$\frac{\partial}{\partial t}(n^* + n_0 e^{-t/T} sin\, \frac{2\pi x}{L}) = D\, \frac{\partial^2}{\partial x^2}(n^* + n_0 e^{-t/T} sin\, \frac{2\pi x}{L})$$

We are again stuck with an unreadable equation (unfortunately this is systematically the case with spatial models....). But after some algebra found in a good text book, we obtain the simplified solution:

$$T = \frac{L^2}{4\pi^2 D}$$

which is not much meaningful in first sight. It basically tells us that the time for a perturbation of a given structure L to dissipate is inversely proportional to diffusion rate D.

A more interesting case arises when we add *reaction*, i.-e. a local interaction among organisms (in other words, when we add ecology!).

The simplest diffusion-reaction model was originally studied by Fisher and later by Skellam. In one dimension, the density changes according to the following equation:

$$\frac{\partial N}{\partial t} = D\frac{\partial^2 N}{\partial x^2} + rN$$

with boundary conditions:

$N(x = 0, t = 0) = N_0$

$N(x = \pm\infty, t)$ is finite

Skipping the tedious integration of the model, we get the solution:

$$N(x, t) = \frac{N_0}{2\sqrt{\pi Dt}} e^{kt - \frac{x^2}{4Dt}}$$

Ploting the result

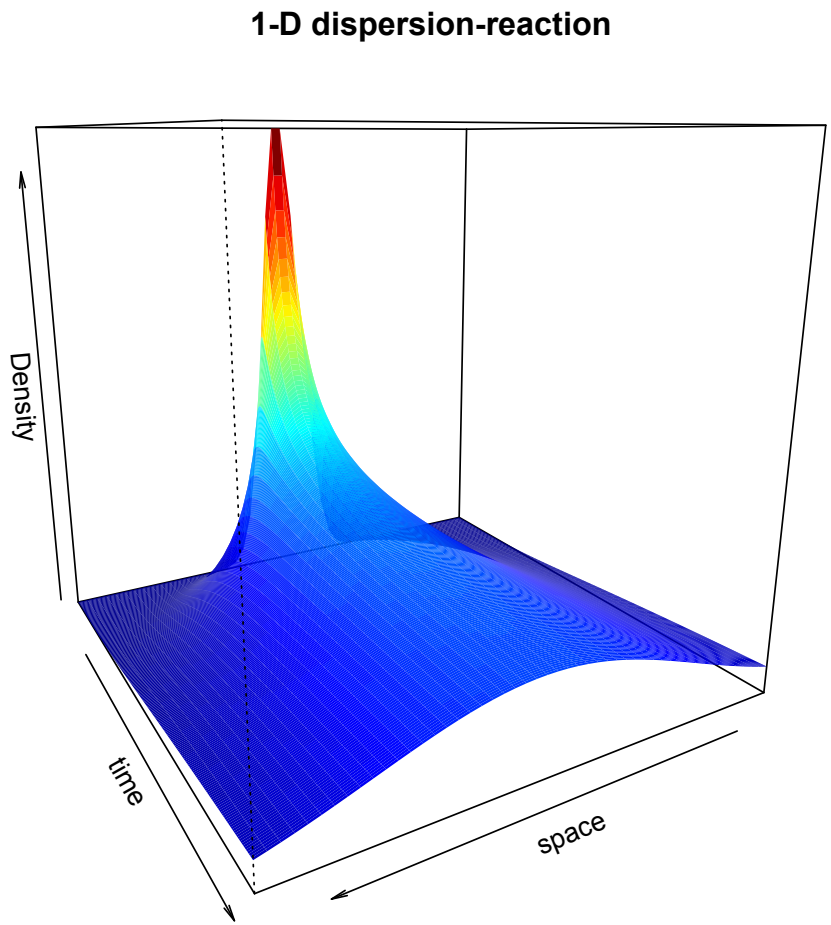


Fig. 5.4

Soetaert and Herman - chapter 5

The most meaningful result comes from the study of spreading rate of invading organisms. Following Kellam's (1953) derivations of the Fisher's diffusion model with logistic growth, we obtain the following spreading rate:
$S = 2\sqrt{rD}$

which holds with standard diffusion or an exponential dispersal kernel. This equation basically tells us three things:

- Spreading increases with the square root of the diffusion coefficient (which happens to be the square of the mean dispersal distance - trivial result)
- It also increases with the growth rate in the invading area
- Carrying capacity in the resident area does not impact migration rate (note that it might under other model formulations)

Numerical integration in time: jumping from one point in time to the next.

Numerical integration in space: computation of the fvalue of the function in a finite number of discriete spatial locations.

Consider the Kolmogorov diffusion equation on a one dimension landscape:
$\frac{\partial N}{\partial t} = -\frac{\partial Flux}{\partial x} + rN(1 - \frac{N}{K})$

where
$\frac{\partial Flux}{\partial x} = D \frac{\partial N}{\partial x}$

We numerically approximate the rate of change in one cell i as (be aware of the switch to ODEs):

$$\frac{dN_i}{dt} = -\frac{\Delta_i Flux}{\Delta x_i} + +rN_i(1 - \frac{N_i}{K})$$
$$\frac{dN_i}{dt} = -\frac{Flux_{i,i+1}}{Flux_{i-1,i}} + +rN_i(1 - \frac{N_i}{K})$$

where Δx_i is the length of the cell and Δ_i denotes that the flux gradient is to be taken around the cell i. In other words, as the difference of the flues at the interfacte with the next cell (i, $i + 1$) and the previous one, ($i - 1$, i).

The flux is simply given by:

$$Flux_{i-1,i} = -D_{i-1,i}\frac{N_i - N_{i-1}}{\Delta x_{i-1,i}}$$

where $\Delta x\, i - 1, i$ is the dispersion distance, ie the distance between adjacent cells.

```r
# Diffusion rate (m2/time)
D = 0.3

# Intrinsic growth rate
r = 0.05

# Carrying capacity
K = 1

# Cell size
delx = 1

# Number of cells
numboxes = 60

# Sequence
Distance = seq(from=0.5,by=delx,length.out=numboxes)
```

We use here the diffusion-reaction equation discretized per cell to calculate the rate of change in each cell. There are therefore *numboxes* differential equations. The flux between cells is calculated using the function *diff*, which takes the gradient between adjacent locations.

```
1 model <-function(t,N,parameters)
2 {
3 deltax     <- c (0.5,rep(1,numboxes-1),0.5)
4 Flux       <- -D*diff(c(0,N,0))/deltax
5 dN    <- -diff(Flux)/delx  + r*N*(1-N/K)
6
7 # the output: the rate of change in each cell
8 list(dN)
9 }  # end of model
```

Densities are initiated with 0 in all cells, except for the two central ones, which are at carrying capacity.

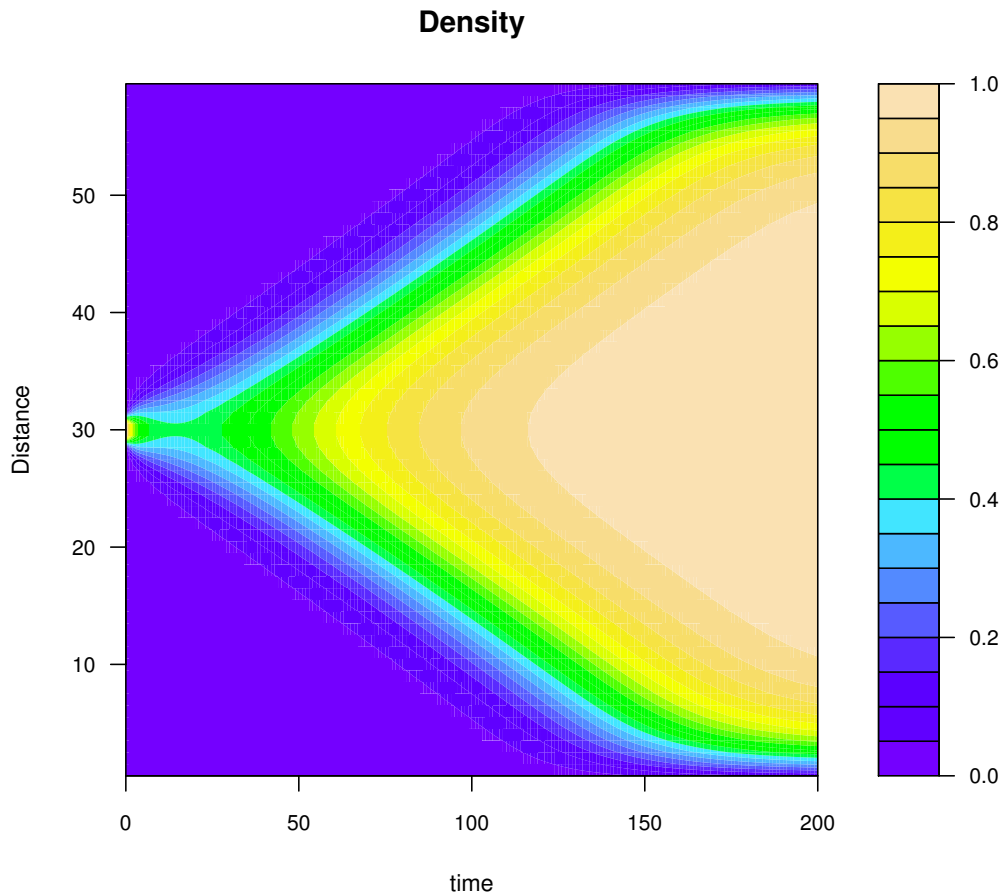
```
N = rep(0,times=numboxes)       # individuals/surface

N[30:31] = K

state = c(N=N)             # the state variables are initialised
```

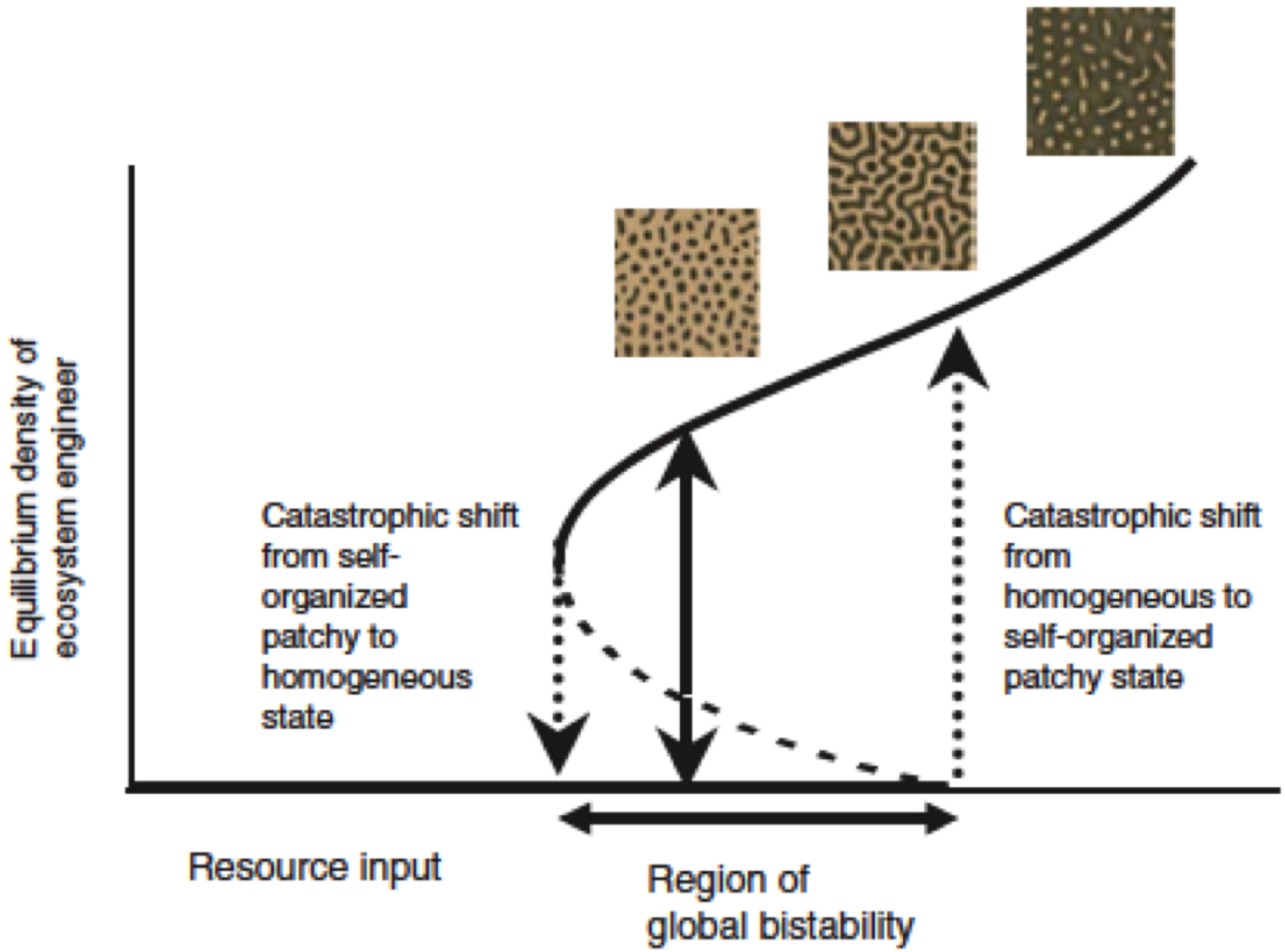
The model is simply run using the function *ode* of the package *deSolve*.

```
1
2 times = seq(0,200,by=1)    # output wanted at these time intervals
3
4 out = ode(state,times,model,parms=0)  # ode is integration routine
5
6 DENSITY   <- out[,2:(numboxes  +1)]
```

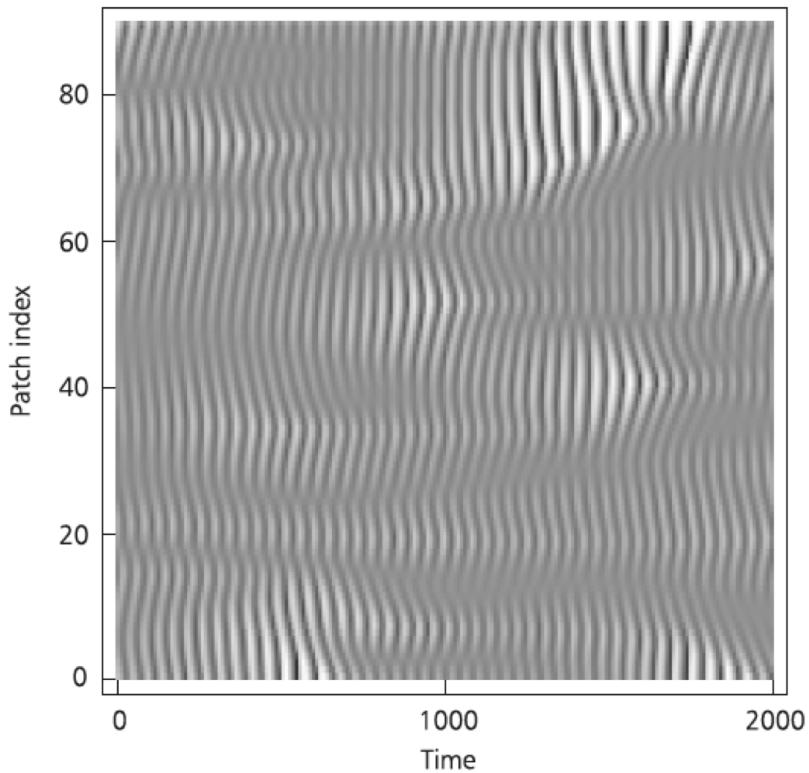
Here I use directly the code provided in Soetaert's book (as a did for a big chunk of the slides...).

```
1
2  # set margins
3  par(mfrow=c(1,1))
4  par(oma=c(0,0,3,0))
5
6  # set colors
7  color= topo.colors
8
9  # make the plot
10 filled.contour(x=times,y=Distance,DENSITY,color= color,
11 xlab="time", ylab= "Distance",main="Density")
```


Density
Distance
time
0
50
100
150
200
10
20
30
40
50
0.0
0.2
0.4
0.6
0.8
1.0


Equilibrium density of ecosystem engineer
Catastrophic shift from self-organized patchy to homogeneous state
Catastrophic shift from homogeneous to self-organized patchy state
Resource input
Region of global bistability

Now program a diffusion version of the Rosensweig-MacArthur model of predator-prey interactions. You should eventually get to a result similar to what Jansen (2000) found:

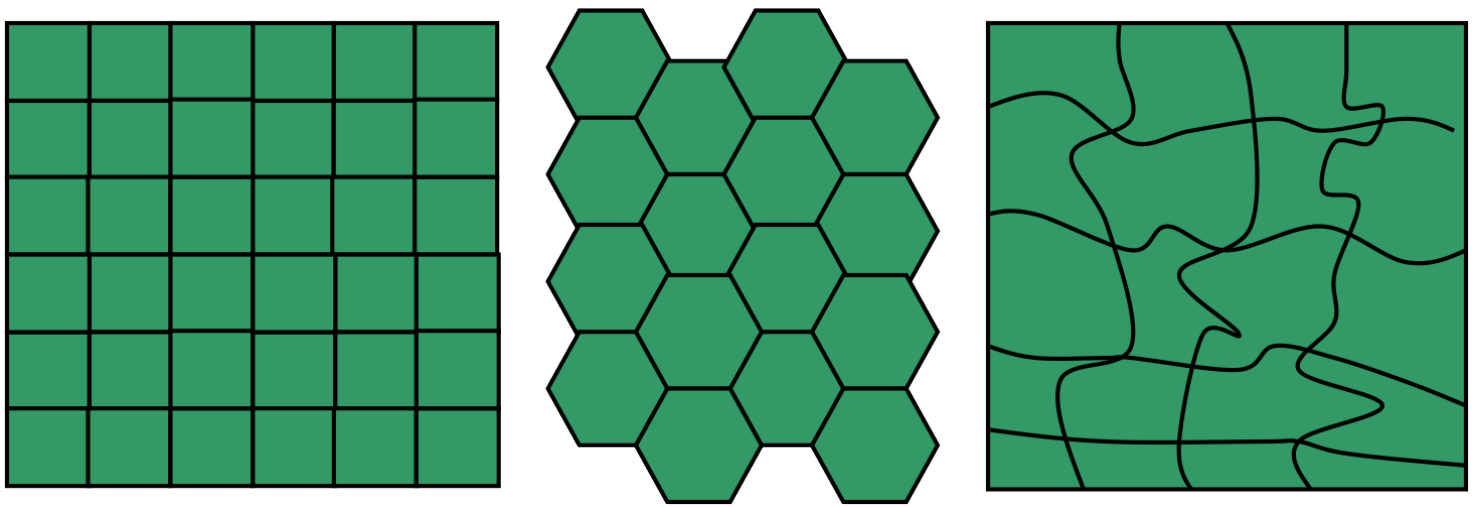


- Discrete space. Divided in cells, culd be 1D, 2D or 3D
- Cells could be in different *states*
- Cell state changes in discrete time
- All cells could be updated at once, or subsequently
- Changes obey rules, which could be deterministic or stochastic
- Transition rules are function of the state of the focal cell and the neighbours

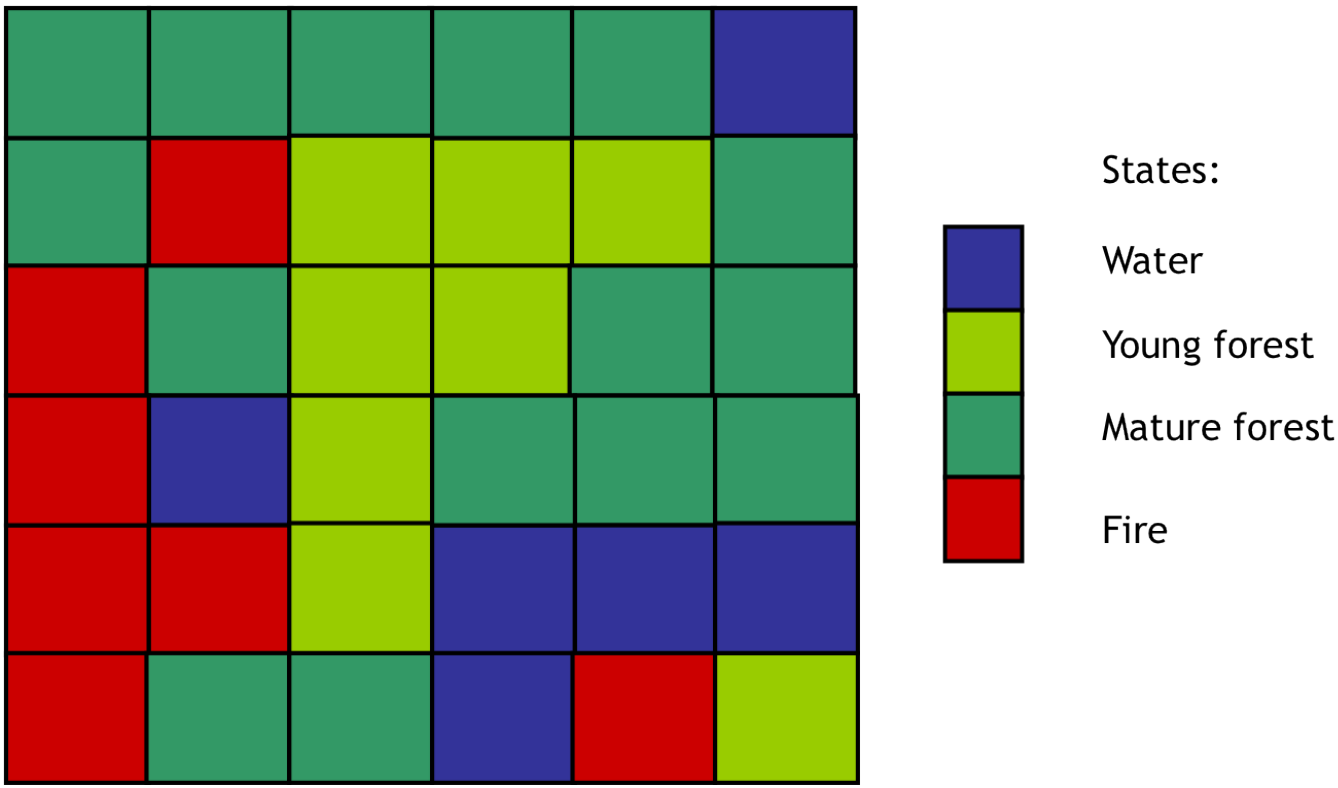
Fundamental question: is it possible to generate complex and non random structures from simple rules?

- Computer science, maths & physics
- Von Nuemann (1940)
- Conway (1970)
- Wolfram (1980)

Landscape structure

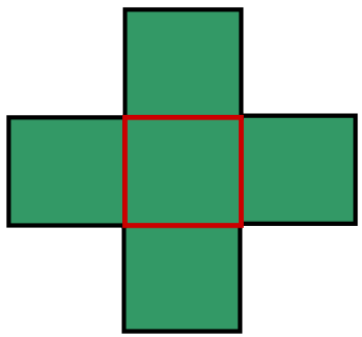


States

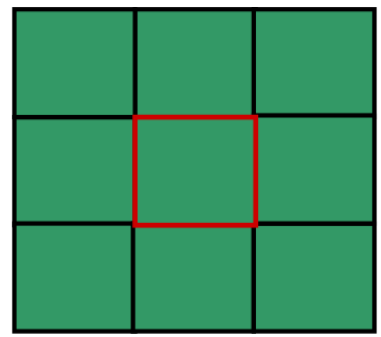


Neighborhood types

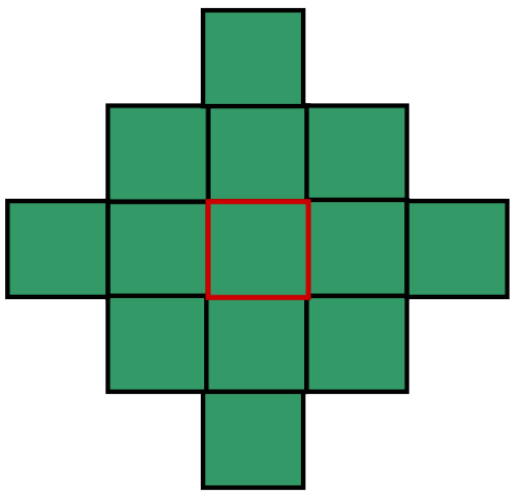
Von Neumann neighbourhood

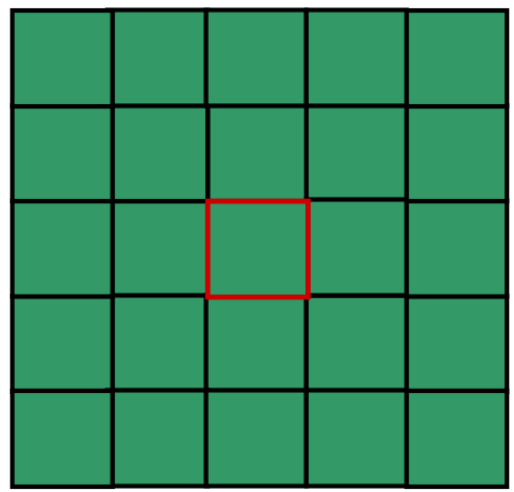


Moore neighbourhood

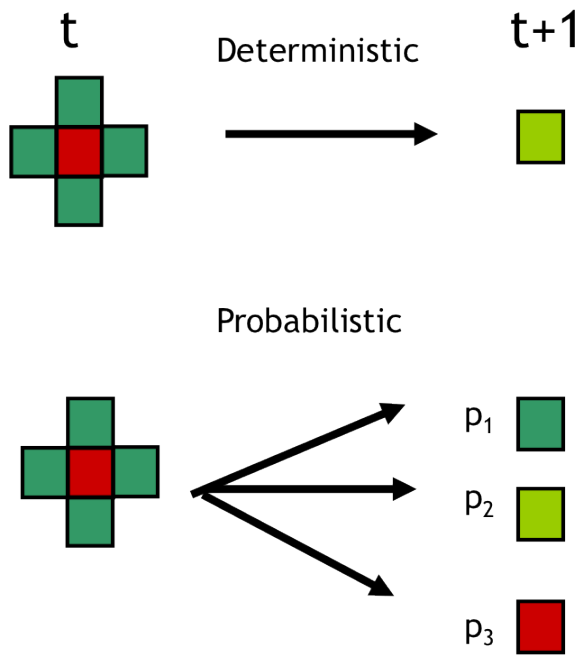


Neighborhood types





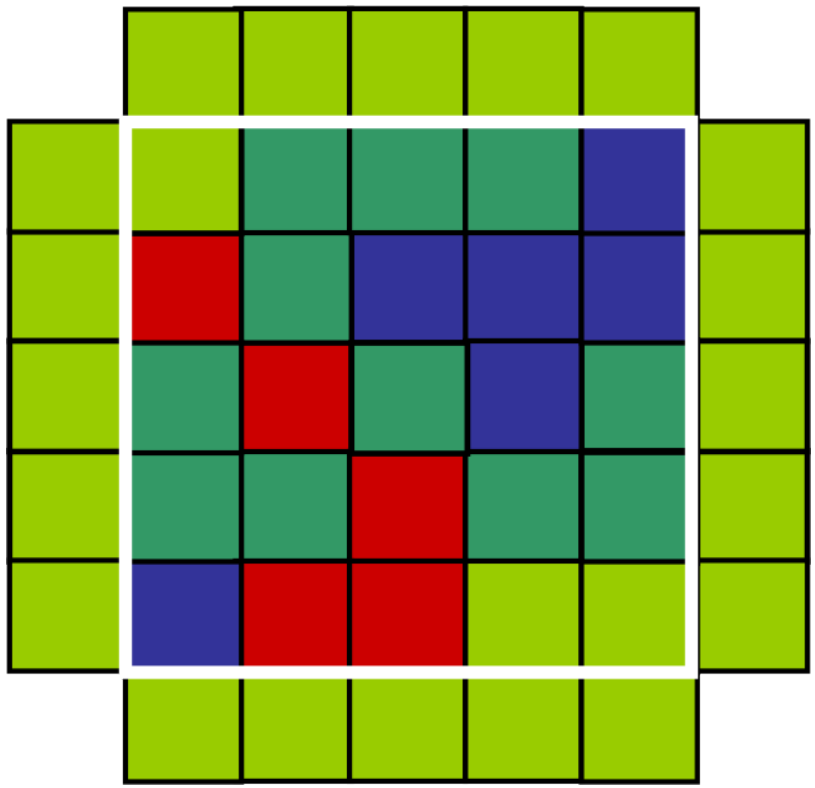
Transitions



Where $p_1 + p_2 + p_3 = 1$

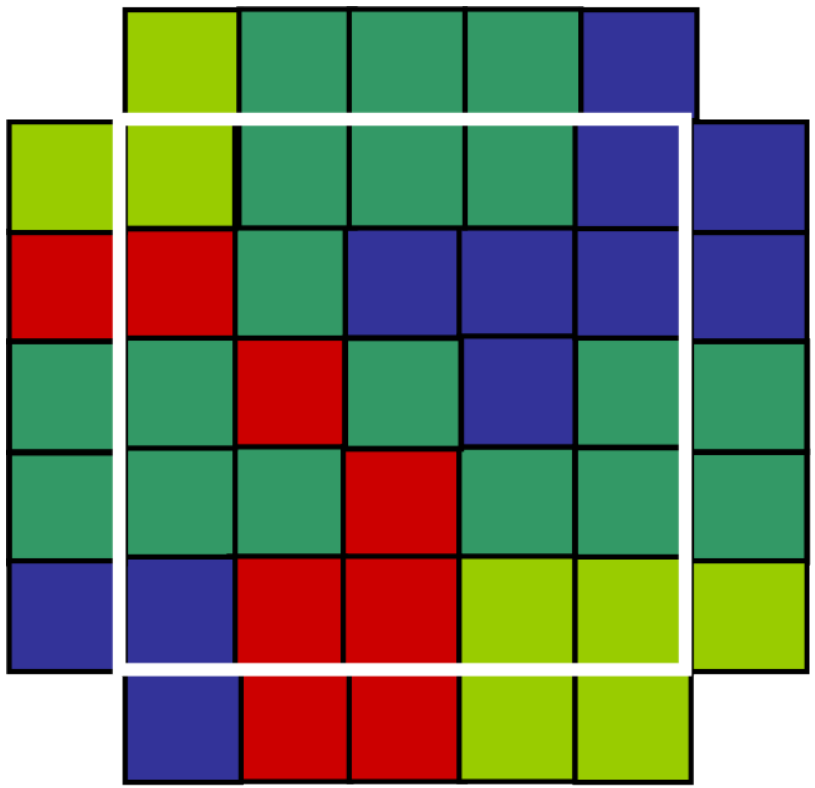
Boundary conditions

Buffer zone



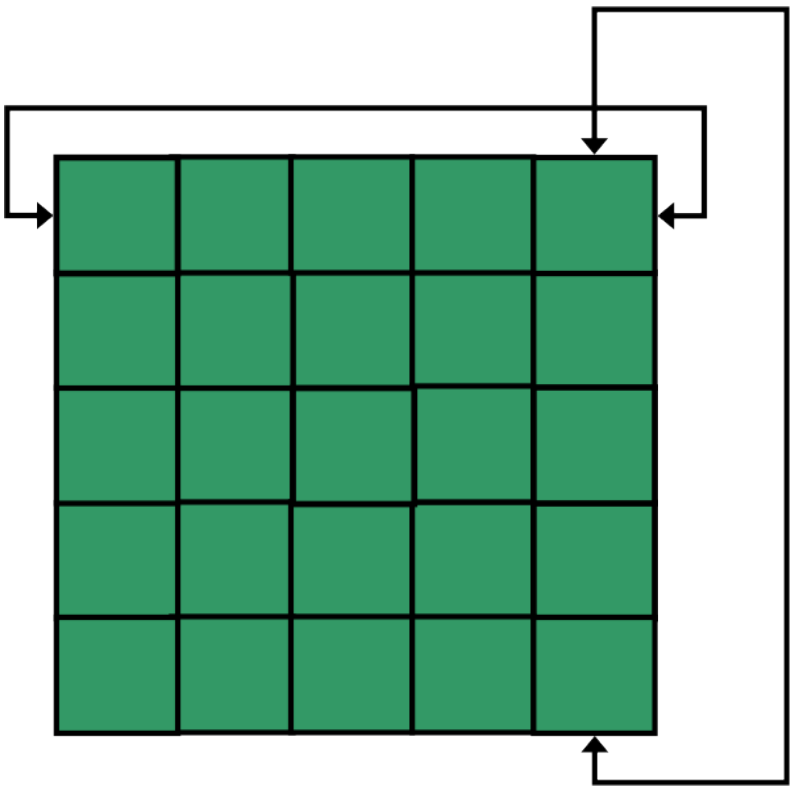
Boundary conditions

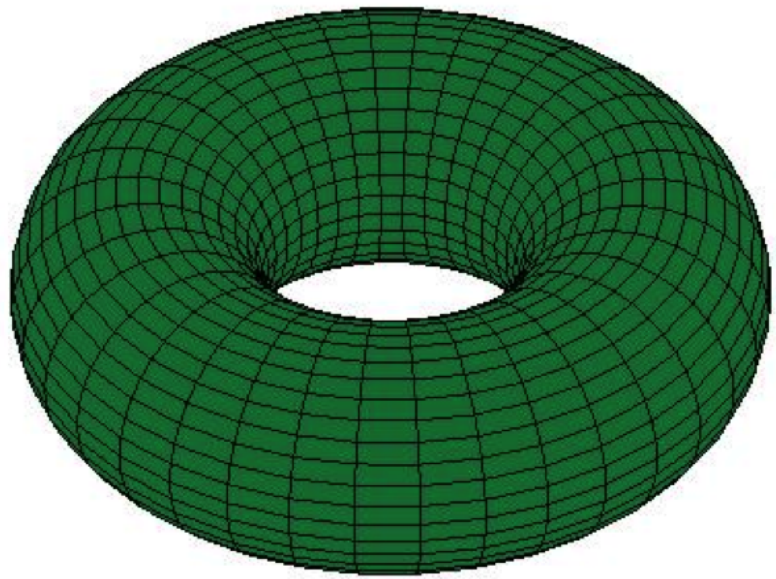
Mirror

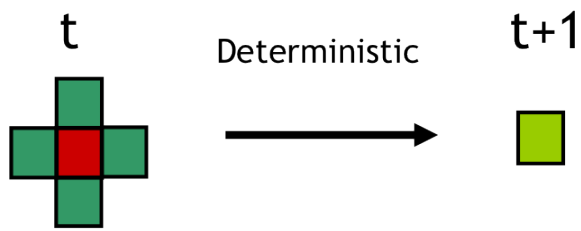


Boundary conditions

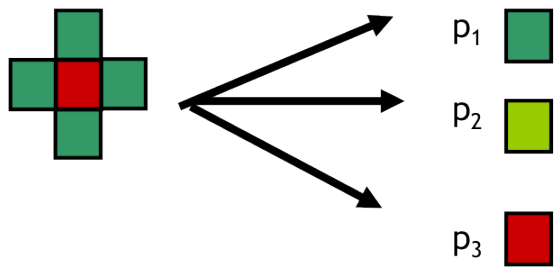
Torus







Probabilistic



Where $p_1 + p_2 + p_3 = 1$

Simple deterministic rules (possibility of 256) to determine the transition between white (0) and black (1) cells.

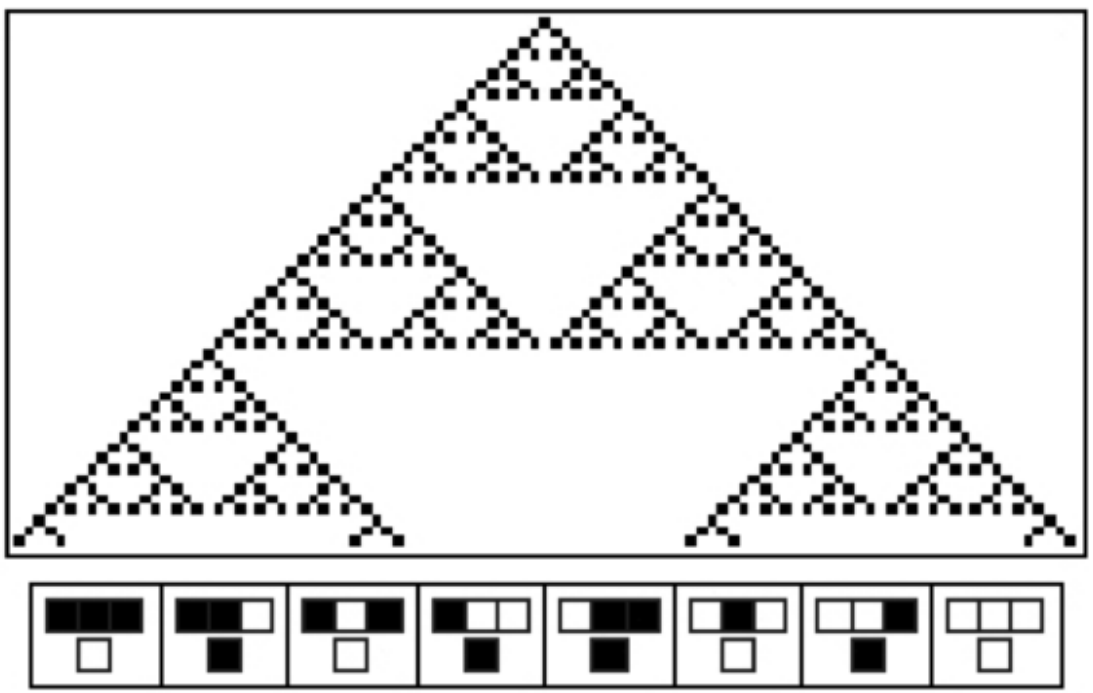
A classic example: rule 90

- Adding two white cells gives a white cell
- Adding to black cells gives a white cell
- Adding one white and one black gives a black

Example

package CellularAutomaton

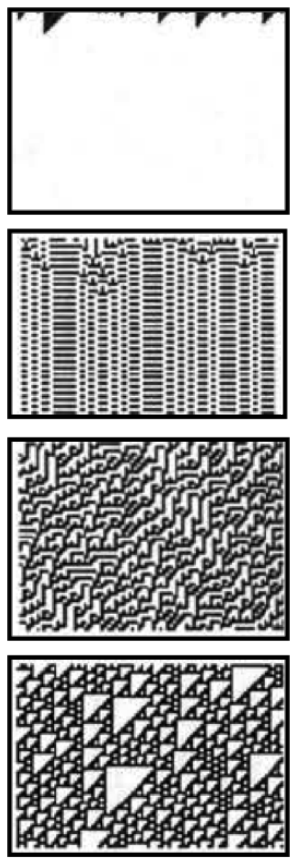
```
1 library(CellularAutomaton)
2
3 # Example with rule 90 and run for 100 time steps
4 res = CellularAutomaton(n = 90, t = 100)
5 res$plot()
```



Rule 90 in nature!

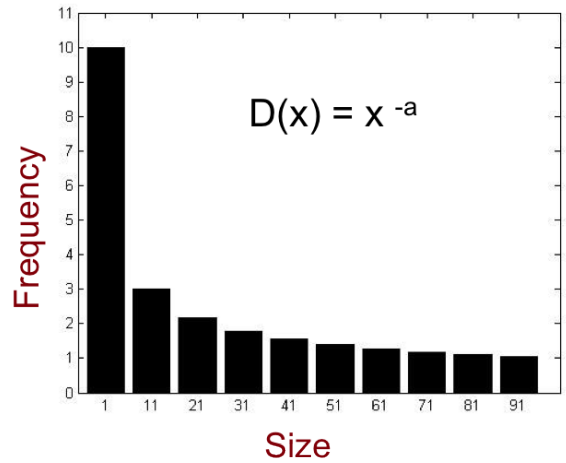
Wolfram's classification

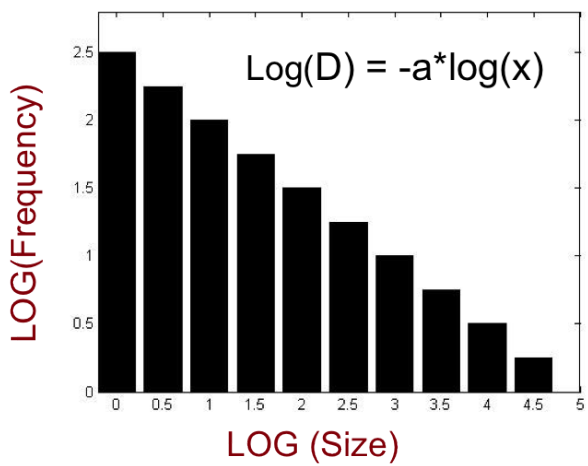
- Stable & homogeneous
- Periodic
- Chaotic or aperiodic
- Complex



Forest fire model

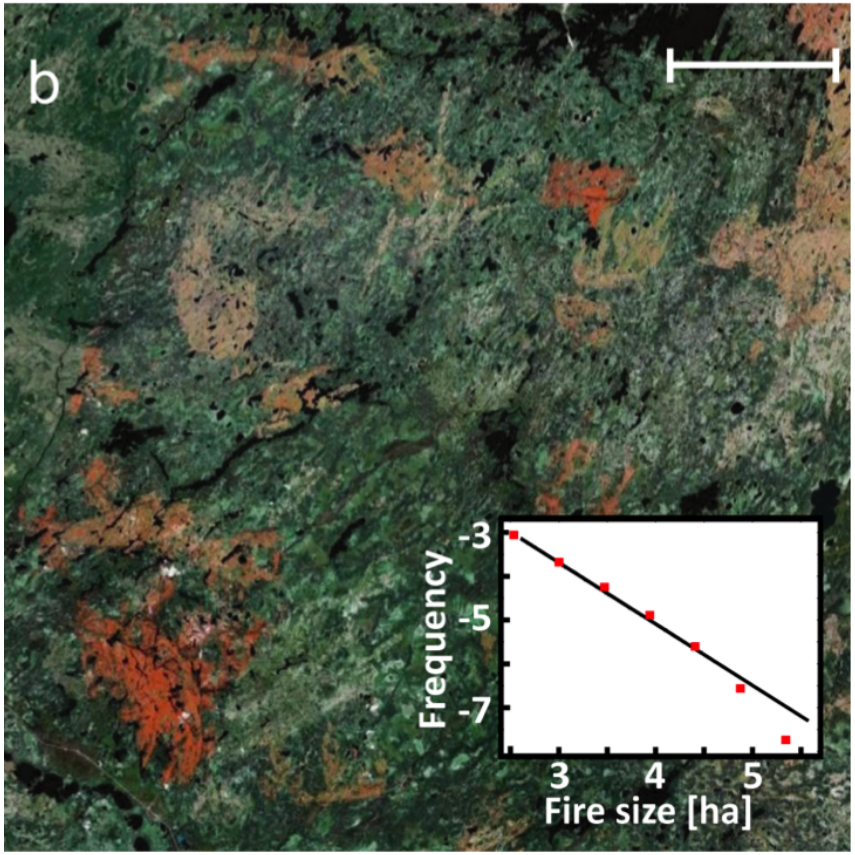
Power law distribution of forest fire size





Forest fire model

- CFS database
- All fires $>$ 200ha between 1960 and 1999
- Data for the Canadian Shield



What is the simplest model that could reproduce this structure?

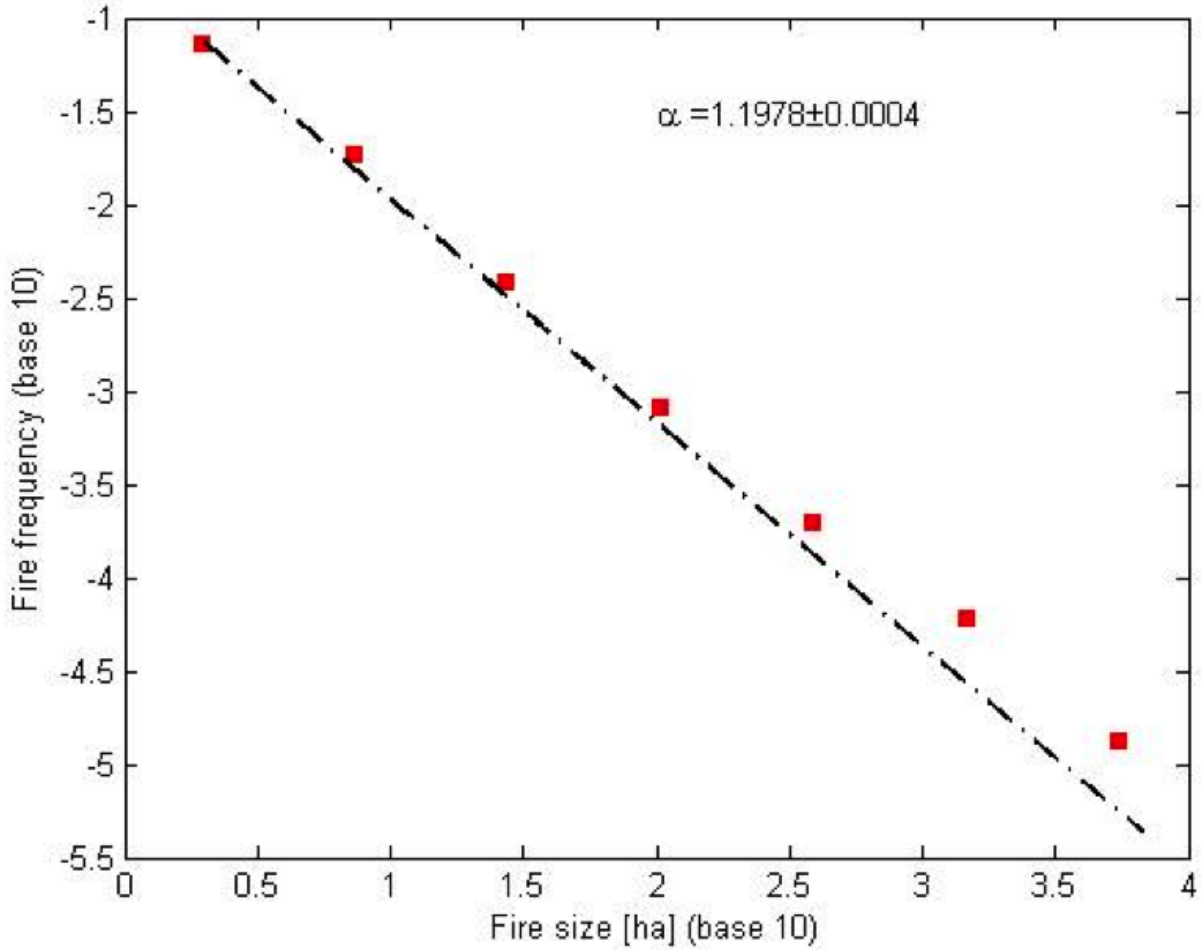
First model was published by the physicists Drossel & Schabl (1992).
Malamud et al. (1998) later published a simplification.
Three possible states:

- Empty
- Occupied by a three
- In fire

- A cell in fire turns into empty state
- An empty cell is colonized by a tree at probability p
- A cell occupied by a tree burns if one of the neighbours is in fire or burn at probability f if no neighbour is in fire

Forest fire model

Power law



Consider an ensemble of patches that could be either empty or occupied. Define p as the fraction of the landscape that is occupied. The following model describes temporal dynamics of occupancy:

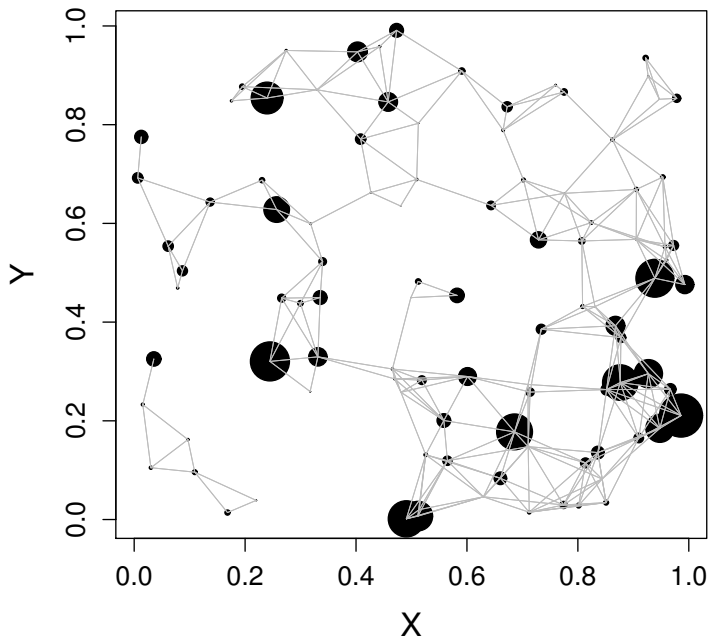
$$\frac{dp}{dt} = cp(1 - p) - ep$$

where c is the colonization rate and e is the extinction rate. Solving the model at equilibrium yields:

$$p^* = 1 - \frac{e}{c}$$

The fundamental result of metapopulation theory is that the species will persist provided that $c > e$.

The equilibrium occupancy will depend on the spatial distribution of the different patches.



Probability of an offspring leaving patch ym reaching patch x and establishing a population is:
$c_y p_y k(y, x) A_y^b$

where c_y is the number of offspring produced per unit of surface at patch y, (y, x) is the dispersal kernel yielding the probability a single offspring leaving y and getting to x, p_y is the probability that patch y is occupied, A_y is the area of the patch y and b is a parameter relating area to the number of offspring produced.

Flipped around, the probability of that offspring not reaching that patch:
$1 - c_y p_y k(y, x) A_y^b$

Considering all patches, we get the probability C_x of at least one offspring reaching patch x and establishing a population :
$1 - \prod(1 - c_y p_y k(y, x) A_y^b)$

Looking at the other side of the equation, we have the extinction function:
$E_i = \frac{e_x}{A_x^c} p_x$

We now get the model:
$\frac{dp_x}{dt} = C_x(1 - p_x) - E_x$

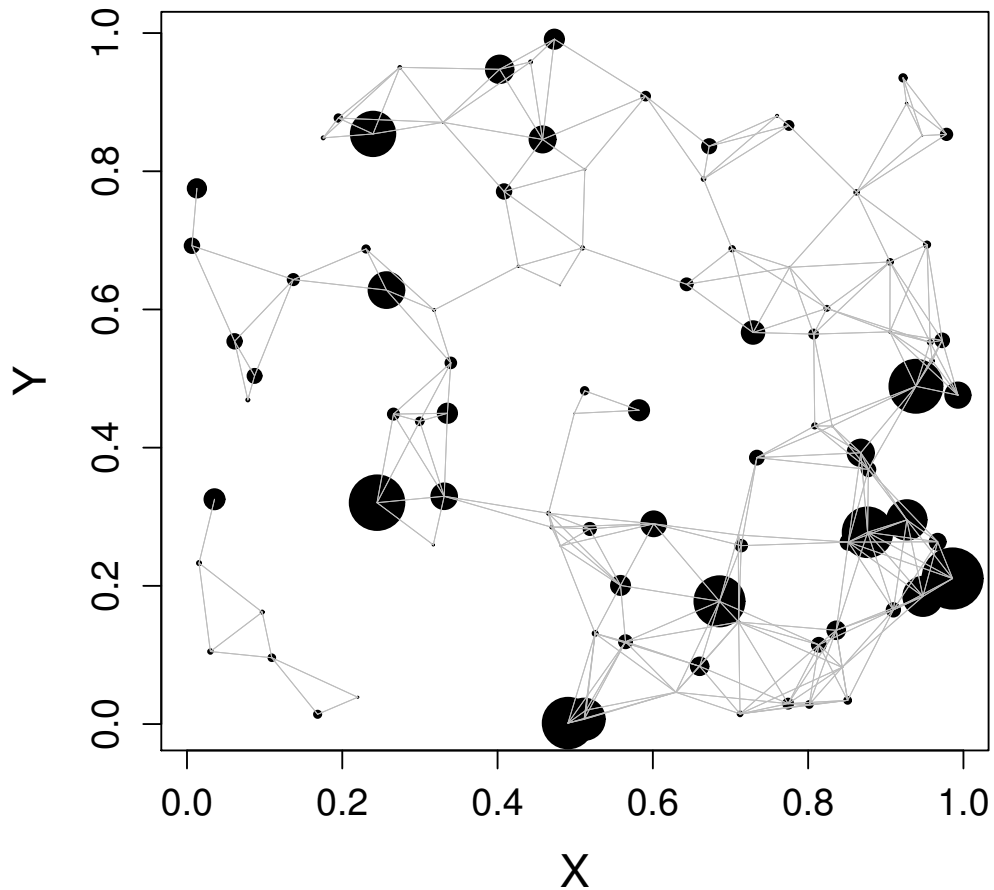
And the equilibrium probability of occurrence at patch x:
$p^{x*} = \frac{C_x}{C_x + E_x}$

This solution is a likelihood function that could be fitted to empirical data and used to evaluate the different parameters. We take the sum ($\sum p_x$) over all patches of the landscape to have the number of patches that are occupied:

Solving the model on R

```
rm(list = ls())
library(rootSolve)

# Create a random metapop structure
n = 100
r = 0.15
XY = cbind(runif(n), runif(n))
A = rexp(n = n, rate = 1)

distMat = as.matrix(dist(XY, method = 'euclidean', upper = T, diag = T))
adjMat = matrix(0, nr = n, nc = n)
adjMat[distMat < r] = 1
diag(adjMat) = 0
```

```
# Plot the random landscape
x11(height = 5.5, width = 6)
par(mar=c(5,6,2,1))
plot(XY[,1],XY[,2],xlab = "X", ylab = "Y",cex.lab = 1.5, cex.axis = 1.25,cex = A, pch = 19)
adjVec = stack(as.data.frame(adjMat))[,1]
XX = expand.grid(XY[,1],XY[,1])
YY = expand.grid(XY[,2],XY[,2])
XX = subset(XX,adjVec==1)
YY = subset(YY,adjVec==1)
arrows(x0 = XX[,1],x1=XX[,2],y0 = YY[,1], y1 = YY[,2], length = 0,lwd = 0.1, col = "grey")
```

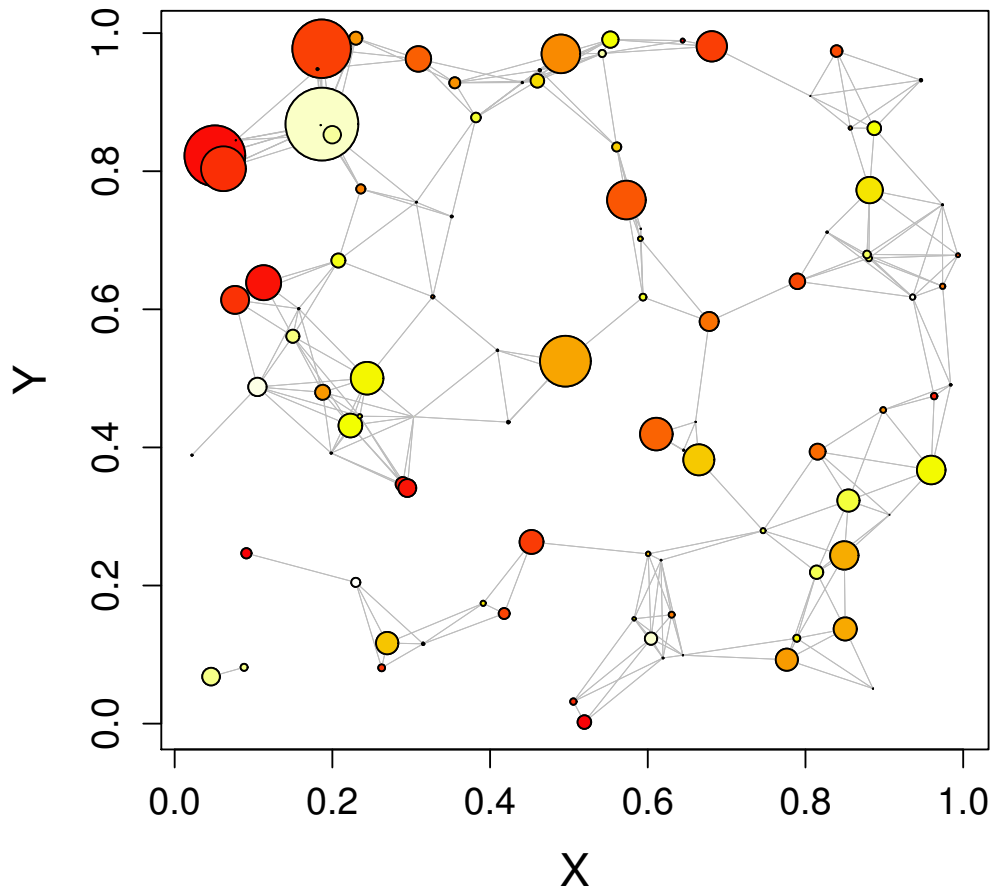

Y
X
0.0
0.2
0.4
0.6
0.8
1.0

```
# MAIN FUNCTION:
model1 = function(p,adjMat,A,b,c,e,f) {
# Number of links per node
degree = apply(adjMat,2,sum)

# Weighted connectivity matrix
adjMatw = adjMat*matrix(degree^-1,nr = n,nc=n, byrow=F)
adjMatw[adjMatw=="NaN"] = 0

# Outgoing propagules
emigr = f*A^b*p

# Pairwise colonization probability
ColProbMat = adjMatw*matrix(emigr,nr = n,nc=n, byrow=F)

# Pairwise probability of no colonization even
noColProbMat = 1 - ColProbMat

# Cumulative probability of an immigration event taking place
ColProbVec = 1 - apply(noColProbMat,2,prod)

# Expected occupancy
px = ColProbVec/(ColProbVec + e/A^c) - p
return(px)
}
```

Solving the model on R

```
1  # OPTIMIZATION
2  get_px = function(adjMat,A,b,c,e,f) {
3  degree = apply(adjMat,2,sum)
4  model2 = function(p) model1(p,adjMat,A,b,c,e,f)
5
6  # Start with the solution of a mean-field metapop model
7  Start = numeric(n) + 1 - e/(mean(degree)*mean(A)^(b+c)*f)
8
9  # Perform the root solving
10 multiroot(model2,start = Start,positive=TRUE)[[1]]
11 }
```

```
# PLOTTING THE RESULTS
# Solve the model
eq_px = get_px(adjMat,A,b=0,c=0,e=0.1,f=0.5)

# Rank the patches according to equilibrium occurrence probability
RK = rank(eq_px)
col = heat.colors(n = n)
vec.col = numeric(n)
for(i in 1:n) vec.col[i] = col[RK[i]]

# Add points to the figure
points(XY[,1],XY[,2],pch=21,bg=vec.col,cex = A)
```


Y
X
0.0
0.2
0.4
0.6
0.8
1.0

See the following files:

- patch_model
- lottery_model
- examples
- spatial_graphs
- simulations