

Partial differential equations: Self-organizing systems

4.1 Introduction	2
4.2 Functions of time and space	3
4.3 The continuity equation	3
4.4 Models for flux: advection and diffusion.....	5
4.5 Discretization	7
4.6 Exercise: Self-organization of vegetation in arid ecosystems	8

4.1 Introduction

Self-organization is the spontaneous emergence of spatially ordered patterns without any external agent imposing it. The system arranges itself from disorder or randomness into a more ordered pattern. To say that a system is self-organized is to say that it is not governed by top-down rules, instead, local interactions generate patterned, ordered structures at higher levels with recognizable dynamics. Since the origins of order in self-organizing systems are the subtle differences among components and the interactions among them, system dynamics cannot be understood by decomposing the system into its constituent parts. Thus, the study of self-organizing systems is *synthetic* rather than *analytic*.

In this chapter we will study spatial self-organization using partial differential equations. In self-organizing systems, the relation between cause and effect is typically non-linear: small causes can have large effects and large causes can have small effects. This non-linearity can be understood from *feedbacks* between the system's components. Each component affects the other components, but these components in turn affect the first component. Thus, the cause-effect relation is circular. Any change in the first component is fed back via its effects on the other components to the first component itself. Feedbacks can have two basic values: *positive* or *negative*. Feedback is said to be positive if the recurrent influence reinforces or amplifies the initial change. In other words, if a change takes place in a particular direction, the reaction being fed back takes place in that same direction. Feedback is negative if the reaction is opposite to the initial action, that is, if change is suppressed or counteracted, rather than reinforced. Negative feedback stabilizes the system by bringing deviations back to their original state. Positive feedback, on the other hand, makes deviations grow in a runaway, explosive manner.

A process of self-organization typically starts with a positive feedback phase, where an initial fluctuation is amplified, spreading even more quickly, until it affects the complete system. Once all components have aligned themselves with the configuration created by the initial fluctuation, the configuration stops growing; it has "exhausted" the available resources. Now the system has reached an equilibrium. Since further growth is no longer possible, the only possible changes are those that reduce the dominant configuration. However, as soon as some components deviate from this configuration, the same forces that reinforced that configuration will suppress the deviation, bringing the system back to its stable configuration. This is the phase of negative feedback. In a more complex self-organizing system, there will be several interlocking positive and negative feedback loops, so that changes in some directions are amplified while changes in other directions are suppressed. This can lead to very complicated, difficult to predict behaviour.

In ecology, a self-organizing-system perspective might reveal how spatial patterns such as patches and boundaries might arise in a complex, heterogeneous community. In the exercise here below, you will learn that different ordered spatial vegetation patterns in arid ecosystems might be the result of spatial self-organization as a consequence of feedbacks among different components of this system (vegetation, surface water and soil water) and the spatial interactions that take place. Since the magic of self-organization lies in the connections and interactions among the parts of the system, we first need to define some fundamental rules that govern the local interactions among the constituent parts.

4.2 Functions of time and space

Here we look at changes of processes in space and time. You can think, for instance, of transport of matter (molecules, cell, organisms, solutes) in combination with biological or (bio-)chemical reactions. We will derive equations for the transport of matter in terms of changes in concentrations c of that matter. In the present approach, state variables are continuous and have measurable units. Also time and space are measured in real units. We assume one-directional movement in the x direction; you can think of transport through a pipe (see Figure 4.1); x represents the location in that pipe and t represents time. The concentration can thus be represented as $c(x, t)$.

4.3 The continuity equation

The continuity equation is the fundamental rule underlying all changes in spatial distribution (see also the previous heat and groundwater flow exercises). It can be very simply explained verbally in the following way:

- Consider a small area;
- Calculate the change in quantity of matter in that small area as the quantity of matter that flows into that area minus the quantity that flows out of that area plus the quantity of matter produced minus the quantity of matter lost within that area.

We will now formulate this simple rule mathematically.

Consider the concentration $c(x, t)$ (quantity of particles per volume) in a straight pipe with constant surface A (Figure 4.1); x represents location in the pipe and t represents time.

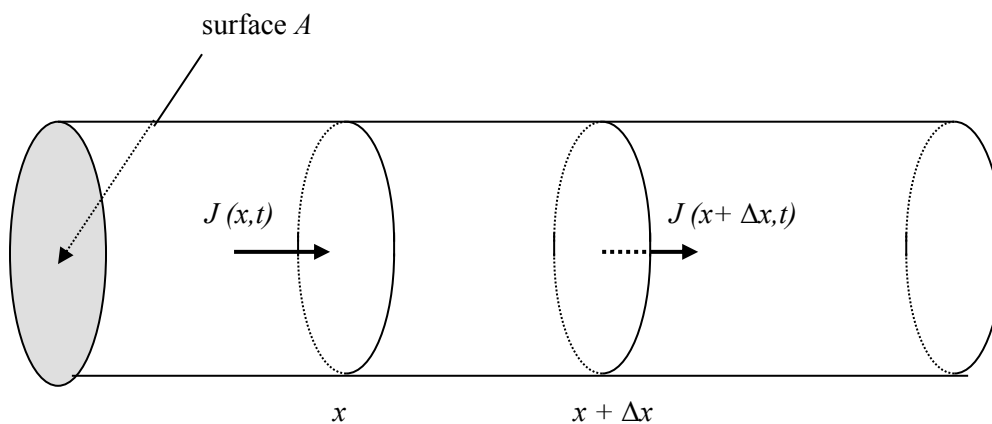


Figure 4.1: Solute transport through a pipe.

Let us now focus on the small piece of pipe between x and $x + \Delta x$. We look at the changes in the concentration c by taking two different effects into account:

- 1) Flow of particles in and out of this small piece of pipe through its “ends” x and $x + \Delta x$;
- 2) Processes (which could be biological, or (bio)chemical reactions) that can lead to production or losses from that piece of pipe.

We will now derive the continuity equation of that small piece of pipe in quantity of particles; we can write:

$$\left(\begin{array}{c} \text{Increase in} \\ \text{quantity of} \\ \text{particles} \\ \text{between } t \\ \text{and } t + \Delta t \end{array} \right) = \left(\begin{array}{c} \text{Flow in} \\ \text{between} \\ t \text{ and} \\ t + \Delta t \end{array} \right) - \left(\begin{array}{c} \text{Flow out} \\ \text{between} \\ t \text{ and} \\ t + \Delta t \end{array} \right) + \left(\begin{array}{c} \text{Quantity of} \\ \text{particles that is} \\ \text{produced} \\ \text{between } t \text{ and} \\ t + \Delta t \end{array} \right)$$

The quantity of particles within that small piece of pipe is $c(x, t)A\Delta x$ (concentration times volume). Therefore, we can write for the term at the left hand side:

$$\left(\begin{array}{c} \text{Increase in} \\ \text{quantity of} \\ \text{particles} \\ \text{between } t \\ \text{and } t + \Delta t \end{array} \right) = c(x, t + \Delta t)A\Delta x - c(x, t)A\Delta x$$

The flow of particles through the ends at x and $x + \Delta x$ are determined by the flux $J(x, t)$, that is, the quantity of particles per unit surface area per time unit at location x and time t flowing through the pipe (in positive direction). For the first two terms at the right hand side we can thus write:

$$\left(\begin{array}{c} \text{Flow in} \\ \text{between} \\ t \text{ and} \\ t + \Delta t \end{array} \right) - \left(\begin{array}{c} \text{Flow out} \\ \text{between} \\ t \text{ and} \\ t + \Delta t \end{array} \right) = [J(x, t)A - J(x + \Delta x, t)A]\Delta t$$

Note that a flux in positive x -direction contributes positively to the quantity of particles in the small piece of pipe at location x , but negatively at location $x + \Delta x$; this explains the signs in this equation. For the last term at the right hand side we write:

$$\left(\begin{array}{c} \text{Quantity of} \\ \text{particles that is} \\ \text{produced} \\ \text{between } t \text{ and} \\ t + \Delta t \end{array} \right) = [\sigma(x, t)A\Delta x]\Delta t$$

in which $\sigma(x, t)$ is the quantity of particles per time unit per unit of volume on location x on time t produced ($\sigma > 0$) or lost ($\sigma < 0$) through biological or (bio-)chemical processes. We can now write the continuity equation as follows:

$$c(x, t + \Delta t)A\Delta x - c(x, t)A\Delta x = [J(x, t)A - J(x + \Delta x, t)A]\Delta t + [\sigma(x, t)A\Delta x]\Delta t$$

Note that all terms have the same physical dimension: quantity of particles. First we divide by A and by Δt :

$$\frac{c(x, t + \Delta t)\Delta x - c(x, t)\Delta x}{\Delta t} = J(x, t) - J(x + \Delta x, t) + \sigma(x, t)\Delta x$$

Now we divide by Δx :

$$\frac{c(x, t + \Delta t) - c(x, t)}{\Delta t} = \frac{J(x, t) - J(x + \Delta x, t)}{\Delta x} + \sigma(x, t)$$

Finally we take limit of Δx to 0 and Δt to 0, so we get:

$$\frac{\partial c(x, t)}{\partial t} = -\frac{\partial}{\partial x} J(x, t) + \sigma(x, t)$$

This is the one-dimensional continuity equation. Note the “−” before the derivation towards x .

4.4 Models for flux: advection and diffusion

The continuity equation that we found does not yet contain all the information needed to determine concentration $c(x, t)$: we therefore need a relation between flux $J(x, t)$ and concentration $c(x, t)$. This relation is dependent on the nature of the physical, biological or (bio)chemical process that we study. We consider two important special cases: advection and diffusion.

4.4.1 Advection

Consider a flowing liquid (like water) containing the particles that we study; these particles adopt the velocity of the liquid. We call this process advection. Let $v(x, t)$ be the velocity of the liquid, then we can write:

$$J(x, t) = v(x, t)c(x, t)$$

(check dimensions!), and substituting this in the continuity equation gives:

$$\frac{\partial c(x, t)}{\partial t} = -\frac{\partial}{\partial x} [v(x, t)c(x, t)] + \sigma(x, t)$$

This is the one-dimensional equation of advection. If we consider the velocity of the liquid independent of location x and time t , this equation reduces to:

$$\frac{\partial c(x, t)}{\partial t} = -v \frac{\partial}{\partial x} c(x, t) + \sigma(x, t)$$

4.4.2 Diffusion and the law of Fick

The particles move randomly (sometimes referred to as random walk). The movement of all the particles together is called diffusion. Consider that there are 100 particles present at x and 60 particles at $x + \Delta x$. On average 50 move from x to $x - \Delta x$ and 50 from x to $x + \Delta x$. Likewise, 30 particles move from $x + \Delta x$ to x and 30 move from $x + \Delta x$ to $x + 2\Delta x$. If we consider only the transport between x and $x + \Delta x$, then we see a net movement of $50 - 30 = 20$ particles from x (where there were 100 particles initially) to $x + \Delta x$ (where there were 60 particles initially). So, there is a net flow of particles from areas with high densities to areas with low densities, having a levelling effect. Further analysis shows that this flow is proportional to the differences in concentration: this relationship is called the law of Fick:

$$J(x, t) = -D \frac{\partial}{\partial x} c(x, t)$$

in which D is the diffusion coefficient. Substituting this in the continuity equation gives:

$$\frac{\partial c(x, t)}{\partial t} = D \frac{\partial^2}{\partial x^2} c(x, t) + \sigma(x, t)$$

This is the one-dimension equation for diffusion.

4.5 Discretization

As you would have guessed already, we can model these processes in MATLAB. As we often discretize in rectangular grids we now consider grid cells.

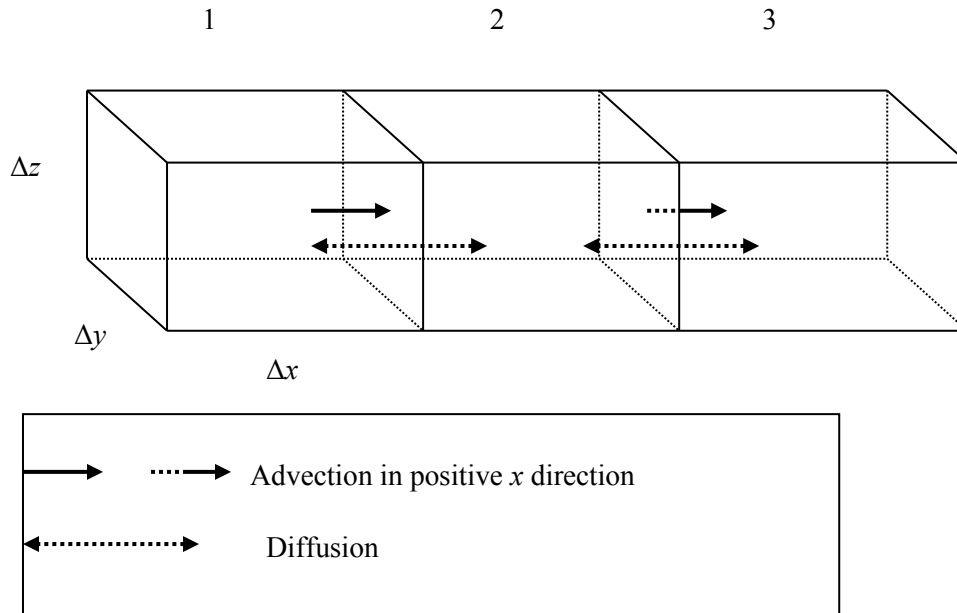


Figure 4.2: Advection and diffusion through grid cells.

4.5.1 Advection

We first consider advection, whereby the liquid flows with a constant velocity in positive x -direction. We are interested in quantities of particles that flow in and out of the grid cell per unit of time and how that affects the concentration of those particles. We call the advection from cell 1 to cell 2 $Advx1$ and the advection from cell 2 to cell 3 $Advx2$. We can write:

$$Advx1 = v c(1) \Delta y \Delta z$$

$$Advx2 = v c(2) \Delta y \Delta z$$

Note that because we opted to calculate the advection in quantities per unit of time we have to multiply the flux of particles (given by $v c(1)$ and $v c(2)$ respectively) by the flow surface area ($\Delta y \Delta z$).

The net advection in the x direction ($NetAdvx$) is now given by:

$$NetAdvx = Advx1 - Advx2$$

The concentration in cell 2 is now given by:

$$c(2) = c(2) + (NetAdvx / \Delta x \Delta y \Delta z) \Delta t$$

4.5.2 Diffusion

We now consider diffusion. We again calculate in quantities of particles flowing in and out of cell 2 per unit of time. We can write:

$$Diffx1 = -D [c(2) - c(1)] \Delta y \Delta z / \Delta x$$

$$Diffx2 = -D [c(3) - c(2)] \Delta y \Delta z / \Delta x$$

The net diffusion in the x direction ($NetDiffx$) is now given by:

$$NetDiffx = Diffx1 - Diffx2$$

The concentration in cell 2 is now given by:

$$c(2) = c(2) + (NetDiffx / \Delta x \Delta y \Delta z) \Delta t$$

4.6 Exercise: Self-organization of vegetation in arid ecosystems

Exercise 4.6.1 Read the article *Self-organization of vegetation in arid ecosystems* enclosed in this reader. Study the equations 1a-1c.

- Identify the term that describes water infiltration into the soil.
- Give the expression for water infiltration into bare soil (where there are no plants).
- Give the expression for water infiltration if plant density goes to infinity.
- Draw the relationship between plant density and water infiltration.
- Explain in your own words how this feeds back to plant growth in the model.

Exercise 4.6.2 Download the file `exerciseflat.m` and study the script (see end of this chapter). This is a discretization of the reaction-diffusion model; the reaction, diffusion, net flow and update calculations have been left out.

- Include these in the script.
- Give a dimension check for flow and update. (NB: there is a typo in Rietkerk et al. 2002; the unit of g_{\max} should be $\text{mm g}^{-1} \text{m}^2 \text{d}^{-1}$)
- Run the model with MATLAB with the given parameter values until stability is reached and show the output.
- Explain in your own words the most important assumptions of the model and the dynamics you observe.

Exercise 4.6.3 We will now try to analyze why plants grow in spatial patterns.

- Change the parameter W_0 from 0.15 to 0.5. Give an interpretation of the meaning of W_0 and the parameter change.
- Run the model. What do you observe and why?
- What causes the plants to grow in spatial patterns in the model? This is now a hypothesis based on our model that can be tested in the field.
- How would you verify this in the field?

Exercise 4.6.4 Desertification can occur as a result of decrease in rainfall, an increase in grazing or a sudden disturbance of the vegetation by humans. First we will analyze what will happen with decreasing rainfall.

- Change the parameter R from 1.3 to 1.5 and 1.1 mm d^{-1} . Show the model output. What is your interpretation of the observed dynamics?
- Do a similar exercise by increasing grazing in the model and report your findings.
- Now, restore original parameter values, let R be 1.1 mm day^{-1} and reduce the initial plant biomass very drastically. What do you observe and why? What is your interpretation of these results?

Exercise 4.6.5 One of the main underlying assumptions of the model `exerciseflat.m` is that the ground is completely flat. Make a file `exerciseslope.m` in which the flow of surface water is in one direction only from upper row downwards, reflecting a hillside, by introducing an advection term for surface water flow instead of a diffusion term. Choose a constant downhill water flow velocity v of 30 m day^{-1} . Note that the shape of the model now has to be a complete circuit. This is called “periodic boundary conditions” (see below and see in the Appendix C3 of the reader):

```
pop(1,:)=pop(NY-1,:);
pop(NY,:)=pop(2,:);
pop(:,1)=pop(:,NX-1);
pop(:,NX)=pop(:,2);
```

(for P , W and O respectively).

- Why do you need periodic boundary conditions?
- Run the model. What do you observe? What is your interpretation of these results?
- How can you change the direction of the surface water flow (from lower row upwards)? Explain this.

```
% MODEL EXERCISEFLAT
% Spatial redistribution of surface water.
% Stefan Dekker, Willem Bouten, Maarten Boerlijst en Max Rietkerk.
% Rietkerk et al. 2002. Self-organization of vegetation in arid ecosystems.
% The American Naturalist 160(4): 524-530.
% DeltaX, DeltaY, DifP, DifW and DifO have unrealistic values that differ from original
% publication to increase computation speed for educational purposes.

clear all

% System discretisation (see remark above)
DeltaX=101; % (m)
DeltaY=101; % (m)

% Diffusion constants for plants, soil water and surface water (see remark above)
DifP=100; % (m2.d-1)
DifW=1; % (m2.d-1)
DifO=1000; % (m2.d-1)

% Initial fraction of grid cells with bare ground
frac=0.90; % (-)

% Parameter values
R = 1.3; % Rainfall (mm.d-1)
alpha = 0.1; % Proportion of surface water available for infiltration (d-1)
W0 = 0.15; % Bare soil infiltration (-)
rw = 0.1; % Soil water loss rate due to seepage and evaporation (d-1)
c = 10; % Plant uptake constant (g.mm-1.m-2)
gmax = 0.05; % Plant growth constant (mm.g-1.m-2.d-1)
d = 0.4; % Plant senescence rate and grazing rate (d-1)
k1 = 3; % Half saturation constant for plant uptake and growth (mm)
k2 = 5; % Half saturation constant for water infiltration (g.m-2)

% Number of grid cells
m=100
NX=m;
NY=m;

% Timesteps
dT=1; % timestep
Time=1; % begin time
EndTime=5000 % end time
PlotStep=10; % (d)
PlotTime=PlotStep; % (d)

% Initialisation
popP = zeros(m,m);
```

```

% Initial state
for i=1:m,
    for j=1:m,
        if (rand > frac)
            popO(i,j)=R/(alpha*W0); % Homogeneous equilibrium surface water in absence of plants
            popW(i,j)=R/rw;          % Homogeneous equilibrium soil water in absence of plants
            popP(i,j)=90;             % Initial plant biomass
        else
            popO(i,j)=R/(alpha*W0); % Homogeneous equilibrium surface water in absence of plants
            popW(i,j)=R/rw;          % Homogeneous equilibrium soil water in absence of plants
            popP(i,j)=0;             % Initial plant biomass
        end;
    end;
end;

% Timesteps
while Time<=EndTime,

% Reaction = ???
    dO;
    dW;
    dP;

% Diffusion: calculate Flow in x-direction : Flow = -D * dpopP/dx; ???
    FXP;
    FXW;
    FXO;

% Diffusion: calculate Flow in y-direction: Flow = -D * dpopP/dy; ???
    FYP;
    FYW;
    FYO;

% calculate netflow ???
    NetP;
    NetW;
    NetO;

% Update ???
    popW;
    popO;
    popP;

    Time=Time+dT;

% PLOTTEN

    PlotTime=PlotTime-dT;
    if PlotTime<=0,
        imagesc (popP); TITLE 'vegetation'
        colorbar
        drawnow;
        PlotTime=PlotStep;
    end
end

```