Assignment 6: Cellular Automata

Luca Bertoni
Alice Hartog

Environmental Systems Analysis

Faculty of Geoscience
Utrecht University
The Netherlands
March 22, 2021

Exercise 1

The flammability f can be modelled as follows:

$$f = f_0 + c * exp(a) \tag{1}$$

where $a \in [2, 4]$ is the age of the vegetation.

The term f_0 reflects the fact that, regardless of its age, biologically speaking any vegetation has a probability greater than 0 to take fire. The coefficient c indicates that for some types of vegetation the age has a significant influence on its flammability, whereas for other types it is lower. A great c implies that the age of the vegetation notably affect the vegetation flammability; on the other hand, a low c means that the age has negligible effects on the flammability of the vegetation.

Being f a probability and given Equation 1, the following range constraints apply:

- $f_0 + c * exp(2) \leq f \leq 1$
- $f_0 \in [0, 1]$
- $0 \leq c \leq \frac{1-f_0}{exp(4)}$

Exercise 2

Assuming time steps of 1 year, n strikes per time step and a flammability f, the number of fires per year per ha is

$$\# \ fires = \frac{n * f}{m * m} \tag{2}$$

A realistic value for wildfires in the Mediterrain is around 20,000 a year [1].

Exercise 3

Probability f_0, the coefficient c and the initial vegetation proportion pv have a high incidence in fire propagation. When one of these three parameters increase, there is always more probability to see fire in the area. This also depends on the number of yearly lightning strikes of course. The model takes all these parameters into consideration together with the number of cells already burnt and, depending on their value, determines if the cells burn or not. If a cell burns, then neighbour cells will also burn, with a probability that depends on their state.

Updating the lattice with a synchronous method the model is faster in calculating the fire propagation. Figure 1 shows the result of the asynchronous method and Figure 2 the synchronous one.

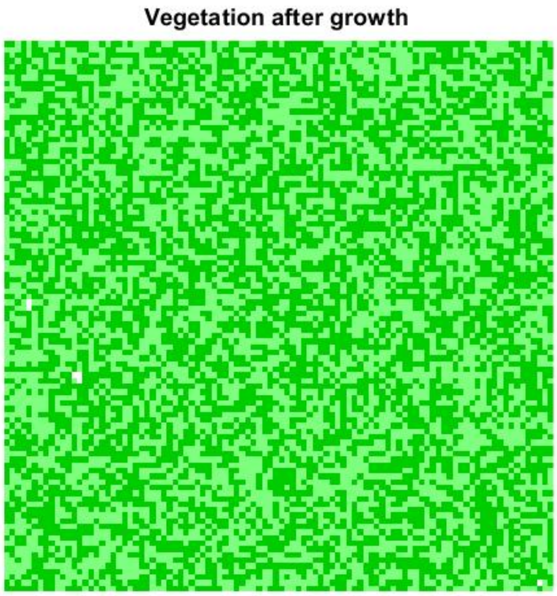

Figure 1: *Vegetation after growth for the asynchronous lattice update with pv=0.5 and* f_0*=0.5.*

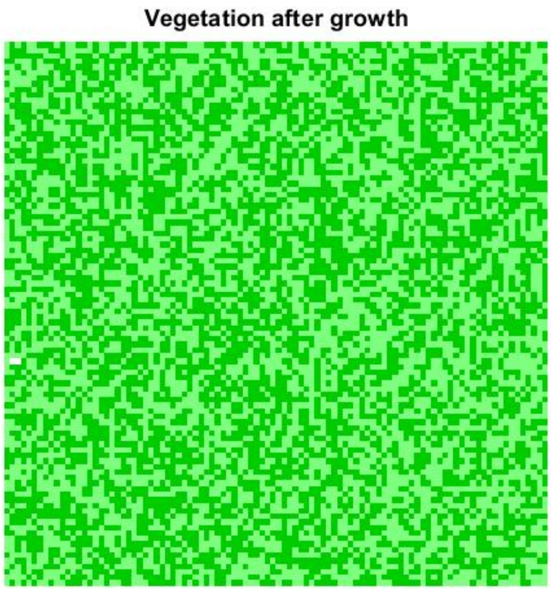

Figure 2: *Vegetation after growth for the synchronous lattice update with pv=0.5 and* f_0*=0.5.*

Exercise 4

A variable that counts the number of the burned cells, i.e. the fire size, is calculated adding the following code:

```
Burning = sum(sum(M==5));
```

right before the end of the `while` in the 'Fire Propagation' section.

Exercise 5

In order to calculate the probability of fire and the average size of fire when it starts the following code has been added:
- Initialised arrays `Firestarted = zeros(1,TotSim)` and `BurnedArea = zeros(1,TotSim)`.
- In a `for` loop of `TotSim`=50 simulations all is calculated, while in the 'Initial lightning strikes' part is taken into account the times a fire starts in `Firestarted(r)=1` array. In the part 'Fire propagation' the array `BurnedArea(r) = Burning` takes into account the burned area in each simulation.
- At the end of the `for` loop, the variables
`SimWithFire = sum(Firestarted)`
and
`FinalBurnedArea = sum(BurnedArea)` are calculated.
- In the end, the probability of fire and the average size of fire are calculated as:

$$ProbaF = \frac{SimWithFire}{TotSim} \tag{3}$$

and

$$AvSizeF = \frac{FinalBurnedArea}{SimWithFire} \tag{4}$$

Exercise 6

Now the model will be run also for different values of f_0 and n.
In order to do it for f_0. 20 values are used from 0 to 0.45. The number 0.45 is chosen so that f never exceeds 1 (being $c = 0.01$, we have $f_0^{max} = 1 - c * exp(4) = 0.45$) a new `for`. The result is shown in Figure 3.

NOTE: In this assignment we used the provided value of $f_0 = 0.5$ with $c = 0.01$ even though it leads to a probability $f > 1$ for old vegetation.

Part of the code used is the following:

```
f0 = linspace(0,0.45,20);
for Loopf = 1:20
(...)
ProbaF(Loopf) = SimWithFire/TotSim;
AvSizeF(Loopf) = FinalBurnedArea/SimWithFire;
end
```

The same loop is done with n changing from 1 to 20, obtaining the plot shown in Figure 4. As it can be noticed, the probability of fire is highly dependent on the number of lightening. In particular, this is true when the parameter n is between 0 and $\sim$ 10. After this threshold, the system stabilizes at a probability of fire close to 1.

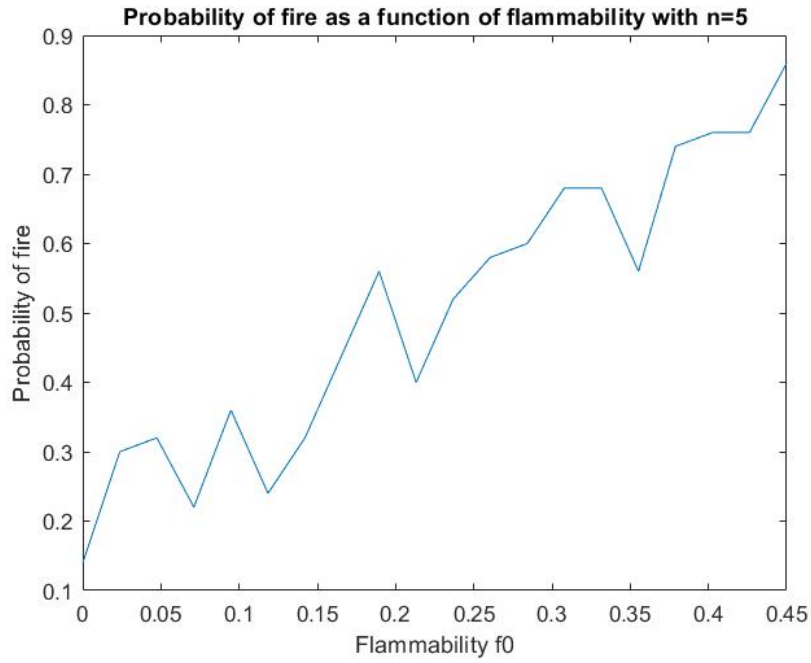

Figure 3: *Probability of fire for different values of flammability and fixed n=5.*

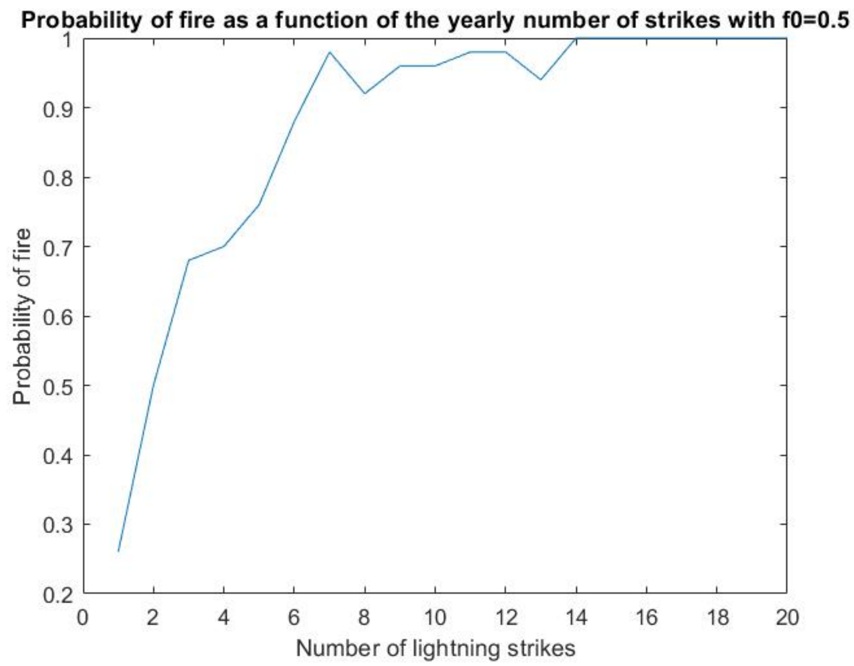

Figure 4: *Probability of fire for different values of yearly number of lightning strikes and* $f0$*=0.5.*

Exercise 7

Figures from 5 to 9 show the annual evolution of the vegetation and the fires. As it is possible to notice, given the evenly spread vegetation the size of the fire increases significantly between year 1 and 2. After year 2, the portions of vegetation that took fire in year 2 are empty, while the portions that were populated in year 2 are either burning or older vegetation.

In year 4, the region with old vegetation took densely fire. Fire is now evenly spread in the entire forest, except for the portion that was burning in year 3. In fact, having no vegetation the fire couldn't spread.

After the big fires of year 4, in year 5 the vegetation is recovering but a significant amount of ground is empty. As a consequence, no fire is present.

Overall, the size and the location of the fire are determined by stochastic processes. However, being the flammability a function of the vegetation (presence or absence, age etc.), some patterns of the fire are more probable than others and they can be recognized looking at the vegetation status.

The assumptions of the model responsible for this behaviour are the flammability that increases with the age of vegetation and the fact that empty cells don't take fire.

Figure 5: *Vegetation and fire at year 1 with c=0.01, f0=0.5 and n=5.*

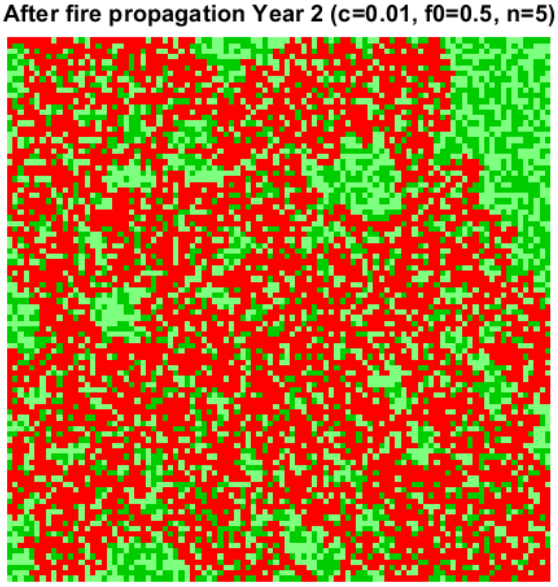

Figure 6: *Vegetation and fire at year 2 with c=0.01, f0=0.5 and n=5.*

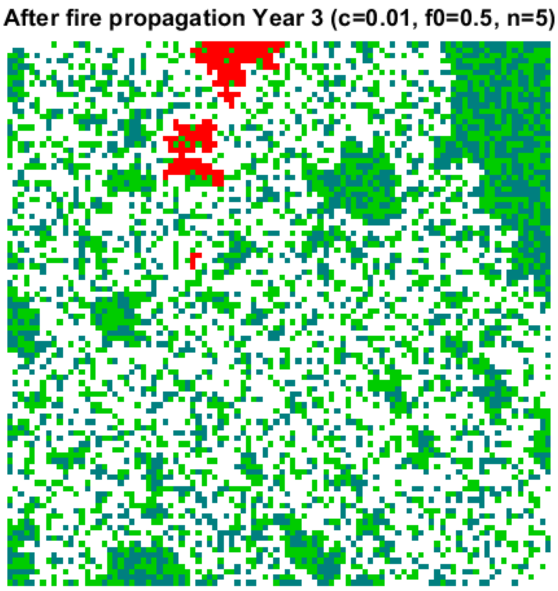

Figure 7: *Vegetation and fire at year 3 with c=0.01, f0=0.5 and n=5.*

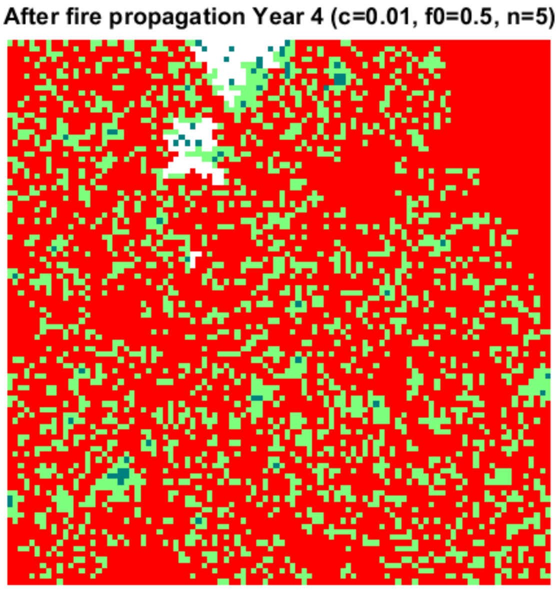

Figure 8: *Vegetation and fire at year 4 with c=0.01, f0=0.5 and n=5.*

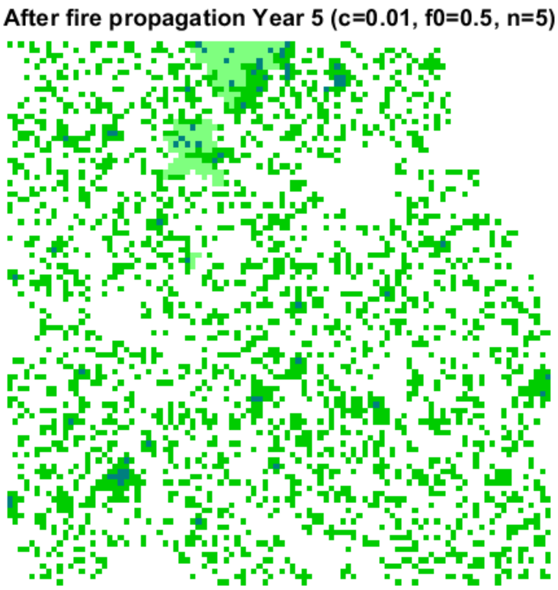

Figure 9: *Vegetation and fire at year 5 with c=0.01, f0=0.5 and n=5.*

Exercise 8

Now the mortality of vegetation is taken into account. Old vegetation can, at any step, either burn or die. In order to consider this differentiation, the variable `mort` represents the mortality probability, here set to 0.2, and the code is:

```
for i = 2 : m-1
    for j = 2 : m-1
        if M(i,j)==4
            if rand<mort
                    M(i,j)=1;
            end
        end
    end
end
```

at the end of the 'Vegetation growth' part. Figures 10 to 14 show the annuel evolution of vegetation and fire.

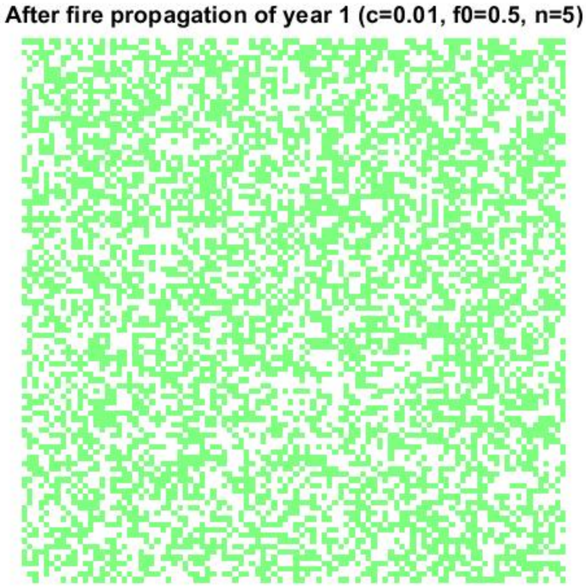

Figure 10: *Vegetation and fire at year 1 with probability of mortality=0.2, c=0.01, $f0$=0.5 and n=5.*

As can be seen, in year 2 all the cells that were empty in year 1 have become young vegetation and some of them took fire together with other mature and old cells. All these burnt cells are then empty in year 3, in which there is only one burnt cell because of the lack of vegetation. All the empty cells become young vegetation thus in year 4 the majority of cells are vegetated and the fire area is huge. Finally, in year 5 happened a situation very similar to year 3.

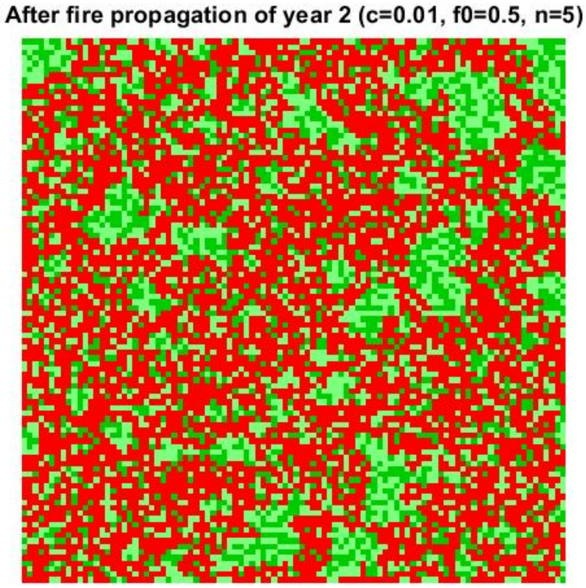

Figure 11: *Vegetation and fire at year 2 with probability of mortality=0.2, c=0.01,* $f0$*=0.5 and n=5.*

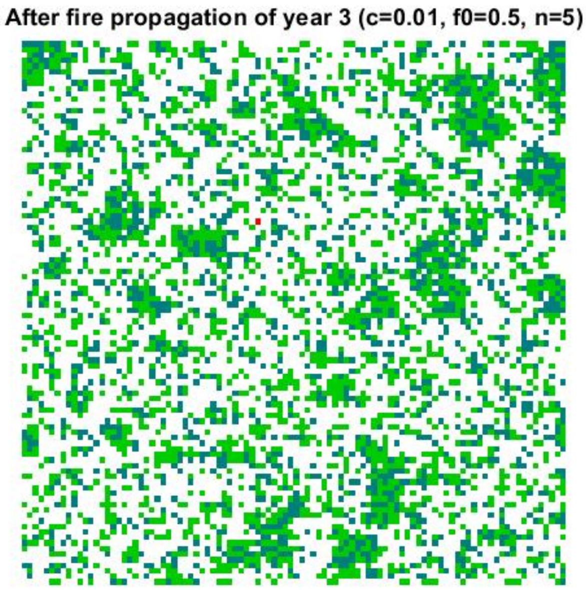

Figure 12: *Vegetation and fire at year 3 with probability of mortality=0.2, c=0.01,* $f0$*=0.5 and n=5.*

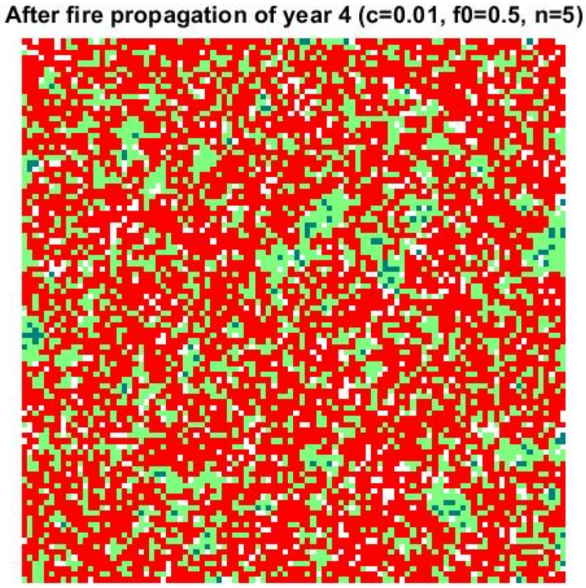

Figure 13: *Vegetation and fire at year 4 with probability of mortality=0.2, c=0.01,* $f0$*=0.5 and n=5.*

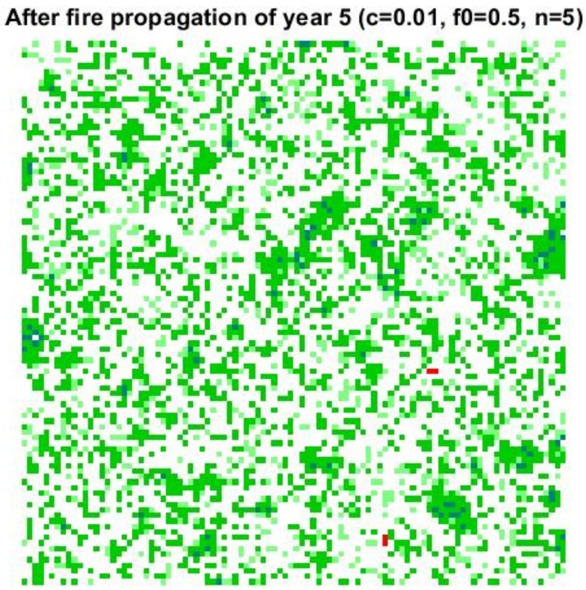

Figure 14: *Vegetation and fire at year 5 with probability of mortality=0.2, c=0.01,* $f0$*=0.5 and n=5.*

Exercise 9

Now, in addition to vegetation mortality, there is a revegetation probability: an empty cell has a certain probability to revegetate or not and this depends on its neighbours. The more vegetated neighbours a cell has, the higher the probability to revegetate. In order to implement this, a new function has been created (RevegetatingProbability) and added the code

```
if M(i,j)==1
    if rand<RevegetatingProbability(i,j)
        M(i,j)=2;
    end
end
```

added before the cells update in part 'Vegetation growth'.

The revegetating probability is strictly dependent on the neighbour cells, as just said. In fact, this probability is defined as

$$N = \frac{N}{4} \tag{5}$$

meaning that if it has 4 vegetated neighbours the probability of revegetating is 1 and if the neighbours are empty cells the probability is 0 (other intermediate possibilities lead to a probability of 3/4 or 1/2).
Figures 15 to 19 show the annual vegetation and fire.

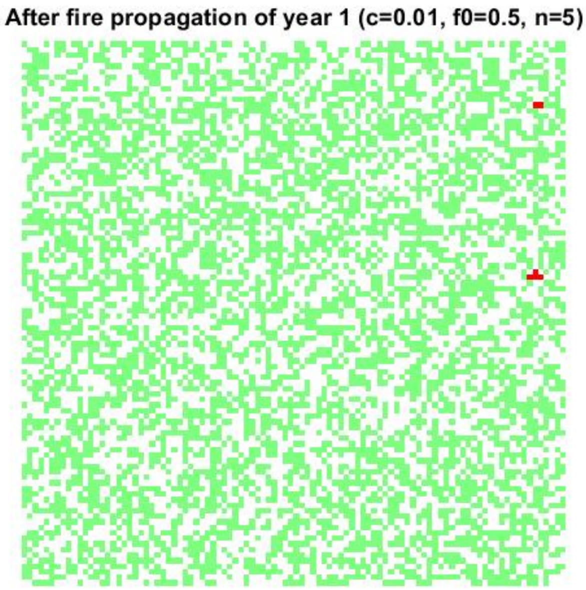

Figure 15: *Vegetation and fire at year 1 with a certain revegetating probability, probability of mortality=0.2, c=0.01, f0=0.5 and n=5.*

In year 2 many cells that were empty in year 1 revegetate and a fire starts. In year 3 many mature cells have a fire and the others just become old vegetation. In year 4 most of the previously fired cells are empty and, since the burnt area was quite big, many cells

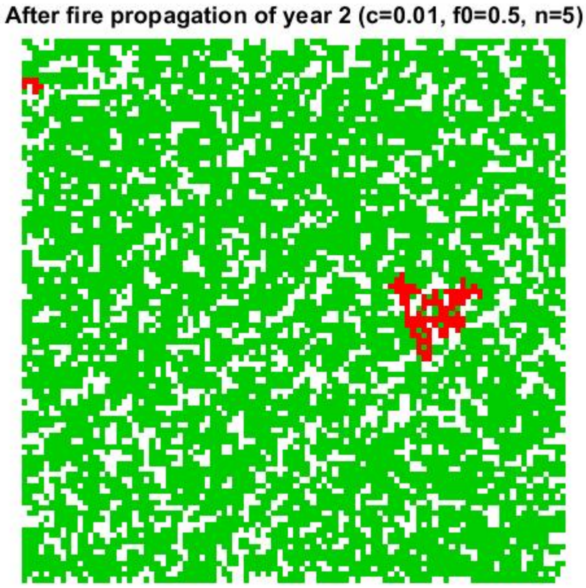

Figure 16: *Vegetation and fire at year 2 with a certain revegetating probability, probability of mortality=0.2, c=0.01, f0=0.5 and n=5.*

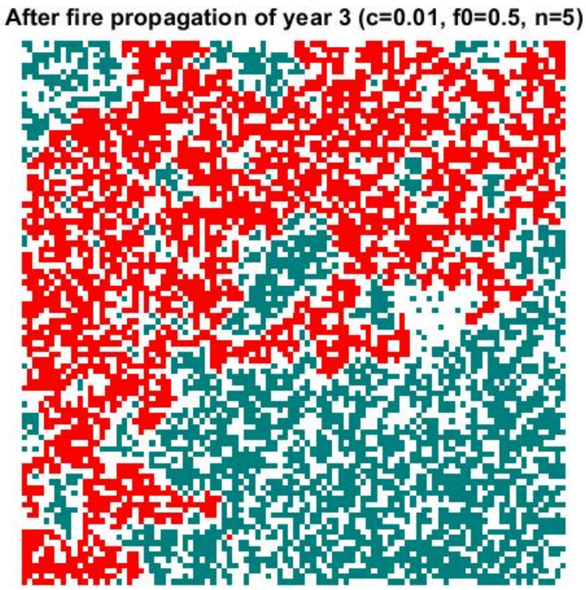

Figure 17: *Vegetation and fire at year 3 with a certain revegetating probability, probability of mortality=0.2, c=0.01, f0=0.5 and n=5.*

don't have vegetated neighbours thus not revegetate and remain empty. In year 5 the same as year 4 happens and many cells become just empty.

The main event here is that if an empty cell has a certain probability to revegetate depending on its neighbours, with big fires the area that remains empty is important.

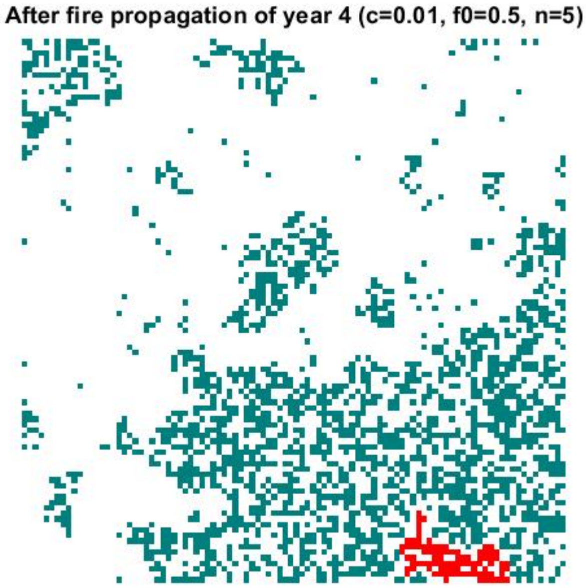

Figure 18: *Vegetation and fire at year 4 with a certain revegetating probability, probability of mortality=0.2, c=0.01, f0=0.5 and n=5.*

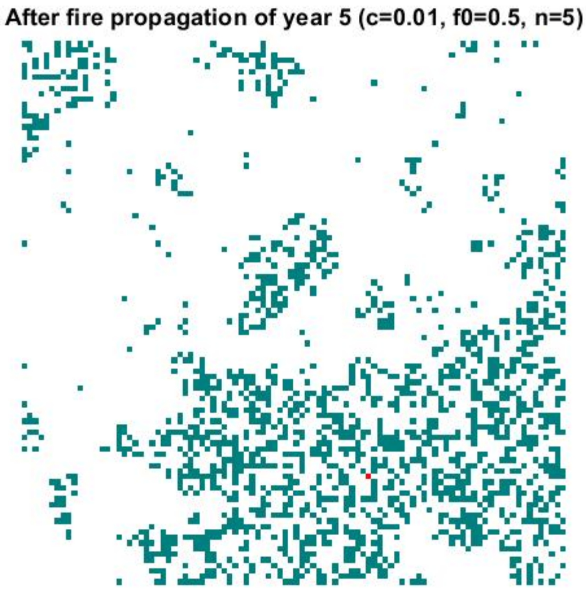

Figure 19: *Vegetation and fire at year 5 with a certain revegetating probability, probability of mortality=0.2, c=0.01, f0=0.5 and n=5.*

Code until Exercise 6

```
%% 1. Setup
```

```
clear all;

% Define model parameters
m  = 100;  % Size of the CA: total number of cells = m*m
pv = 0.5;  % Initial proportion of vegetated cells.
f0 = 0.5;  % Parameter for flammability model
c  = 0.01; % idem
n  = 5;    % Number of lightning strikes per year

% Define an appropriate color map for visualisation
white      = [1 1 1];       % State 1: Empty
lightgreen = [0.5 1 0.5];   % State 2: Young vegetation
green      = [0 0.8 0];     % State 3: Mature vegetation
darkgreen  = [0 0.5 0.5];   % State 4: Old vegetation
red        = [1 0 0];       % State 5: Fire
mycolors   = [white; lightgreen; green; darkgreen; red];

% Loop to calculate probability for 50 repetitions
TotSim = 50;

% Loop over different f0
f0 = linspace(0,0.45,20);
for Loopf = 1:20

% Loop over different n
% n = linspace(1,20,20);
% for Loopn = 1:20

    Firestarted = zeros(1,TotSim);
    BurnedArea = zeros(1,TotSim);

    for r = 1:TotSim


        %% 2. Create state matrix

        M = zeros(m,m);

        % The first and last rows and columns are empty (1)
        M(:,1) = 1;
        M(1,:) = 1;
        M(:,m) = 1;
        M(m,:) = 1;


        % Otther cells are (randomly) empty, or occupied by young vegetation
        R = rand(m,m);
        for i=2:m-1
            for j = 2:m-1
```

```
                if R(i,j) < pv
                    M(i,j) = 2 ; % Young vegetation
                else
                    M(i,j) = 1 ; % Empty
                end
            end
        end

        % Count number of vegetated cells
        Vegetation = sum(sum(M==2 | M==3 | M==4));


        %% 3. Initial lightning strikes
        for k = 1 : n  %for loop for the f0 loop
%          for k = 1 : n %for loop for the n loop
            % Randomly select a grid cell (but exclude the first/last
            % rows and columns)

            i = randi(m-2)+1; % random row in range 2..nrow-1
            j = randi(m-2)+1; % Random column in range 2..ncol-1

            % If there is vegetation on that cell, compute flammability
            % and test
            % if it is unlucky enough to catch fire.
            if M(i,j) == 2 || M(i,j) == 3 || M(i,j) == 4
                f = f0(Loopf) + c*exp(M(i,j)); %for the f0 loop
% Probability of burning of the cell
                f = f0 + c*exp(M(i,j)); %for the n loop
                if rand < f % compare random number with flamability
                    M(i,j) = 5 ; % Unlucky, the cells starts to burns
                end
            end
        end

        % Count number of initially burning cells
        BurningIni = sum(sum(M==5));

        if(BurningIni>0)
        Firestarted(r) = 1;
        end

        %% 4. Visualisation of initial state (inc. lighning)

        % figure(1); clf;
        % colormap(mycolors);
        %
        % image(M);
        % axis equal; axis tight; axis off; % Tight & square
        % title('Just before fire propagation')
```

```
    % drawnow;

    %% 5. Fire propagation

%           figure(2); clf;
%           colormap(mycolors);

    % Matrix V keeps track of the number of burning neigbours of
    % each grid cell.
    V = BurningNeighbors(M);

    % Compute the spreading of fires from the lighning strike sites.
    % The propagation goes step by step. We keep on trying to expand
    % the fires, until it stopped growing.
    %
    % Catching fire is a stochastic process. We draw random numbers
    % ONCE to guarantee that individual grid cells have a fixed chance
    % of getting burned.

    R = rand(m,m);

    FireGrowing = BurningIni>0; % We went from 0 to >0 fires...
    while FireGrowing

        % Then try to expand the current fire
        [M,V,NewBurned] = FireProgressionSync(M,V,R, f0,c); % spread fire

        % Draw map of updated states
    %       image(M);
    %       axis equal; axis tight; axis off; % Tight & square
    %       title('After fire propagation')
    %       drawnow;

        % Do we need to go for another round of fire expansion?
        FireGrowing = NewBurned > 0;


    end
     Burning = sum(sum(M==5));

     BurnedArea(r) = Burning;
    %% 6. Vegetation growth

    for i = 2 : m-1
        for j = 2 : m-1
            if     M(i,j)==1, M(i,j)=2; % Empty -> Young
            elseif M(i,j)==2, M(i,j)=3; % Young -> Mature
            elseif M(i,j)==3, M(i,j)=4; % Mature -> Old
            elseif M(i,j)==5, M(i,j)=1; % Burning -> Empty
```

```
                end
            end
        end
        Vegetation = sum(sum(M==2 | M==3 | M==4)); % Count

        % figure(3); clf;
        % colormap(mycolors);
        %
        % image(M);
        % axis equal; axis tight; axis off; % Tight & square
        % title('Vegetation after growth')
        % drawnow;



    end
SimWithFire = sum(Firestarted);
FinalBurnedArea = sum(BurnedArea);

ProbaF(Loopf) = SimWithFire/TotSim;
AvSizeF(Loopf) = FinalBurnedArea/SimWithFire;

% ProbaF(Loopn) = SimWithFire/TotSim;
% AvSizeF(Loopn) = FinalBurnedArea/SimWithFire;
end

figure(1);
plot(f0, ProbaF);
xlabel('Flammability f0');
ylabel('Probability of fire');
title('Probability of fire as a function of flammability with n=5');

% figure(1);
% plot(n, ProbaF);
% xlabel('Number of lightning strikes');
% ylabel('Probability of fire');
% title('Probability of fire as a function of the yearly number
% of strikes with f0=0.5');

%% 7. End of program
```

Code for Exercise 7, 8 and 9

Code for the new function *RevegetatingProbability*

```
function N = RevegetatingProbability(M)

[nrow, ncol] = size(M);
```

```
% Initialize V as empty matrix (0 burning nbrs)
N = zeros(nrow,ncol);

% For every grid cell, count the number of burning nbrs
for i = 2 : nrow-1
    for j = 2 : ncol-1
        % lower nbr burning?
        if M(i-1,j) == 2 || M(i-1,j) == 3 || M(i-1,j) == 4
            N(i,j) = N(i,j)+1 ;
        end
        % Upper nbr burning?
        if M(i+1,j) == 2 || M(i+1,j) == 3 || M(i+1,j) == 4
            N(i,j) = N(i,j)+1 ;
        end
        % Left nbr burning?
        if M(i,j-1) == 2 || M(i,j-1) == 3 || M(i,j-1) == 4
            N(i,j) = N(i,j)+1 ;
        end
        % Right nbr burning?
        if M(i,j+1) == 2 || M(i,j+1) == 3 || M(i,j+1) == 4
            N(i,j) = N(i,j)+1 ;
        end
    end
end


N = N/4; % Probability of 1 for each neighbour vegetated
```

Code for the exercises

```
%% 1. Setup

clear all;

% Define model parameters
m  = 100;  % Size of the CA: total number of cells = m*m
pv = 0.5;  % Initial proportion of vegetated cells.
f0 = 0.5;  % Parameter for flammability model
c  = 0.01; % idem
n  = 5;    % Number of lightning strikes per year
mort = 0.2;   %Probability for the vegetation to die

% Define an appropriate color map for visualisation
white      = [1 1 1];       % State 1: Empty
lightgreen = [0.5 1 0.5];   % State 2: Young vegetation
green      = [0 0.8 0];     % State 3: Mature vegetation
darkgreen  = [0 0.5 0.5];   % State 4: Old vegetation
red        = [1 0 0];       % State 5: Fire
mycolors   = [white; lightgreen; green; darkgreen; red];
```

```
%% 2. Create state matrix

M = zeros(m,m);

% The first and last rows and columns are empty (1)
M(:,1) = 1;
M(1,:) = 1;
M(:,m) = 1;
M(m,:) = 1;

% Otther cells are (randomly) empty, or occupied by young vegetation
R = rand(m,m);
for i=2:m-1
    for j = 2:m-1
        if R(i,j) < pv
            M(i,j) = 2 ; % Young vegetation
        else
            M(i,j) = 1 ; % Empty
        end
    end
end

% Count number of vegetated cells
Vegetation = sum(sum(M==2 | M==3 | M==4));

for t = 1:5
    %% 3. Initial lightning strikes
    for k = 1 : n
        % Randomly select a grid cell (but exclude the first/last
        % rows and columns)
        i = randi(m-2)+1; % random row in range 2..nrow-1
        j = randi(m-2)+1; % Random column in range 2..ncol-1

        % If there is vegetation on that cell, compute flammability
        % and test
        % if it is unlucky enough to catch fire.
        if M(i,j) == 2 || M(i,j) == 3 || M(i,j) == 4
            f = f0 + c*exp(M(i,j)); % Probability of burning of the cell
            if rand < f % compare random number with flamability
                M(i,j) = 5 ; % Unlucky, the cells starts to burns
            end
        end
    end

    % Count number of initially burning cells
    BurningIni = sum(sum(M==5));

    %% 4. Visualisation of initial state (inc. lighning)
```

```
%       figure(1); clf;
%       colormap(mycolors);
%
%       image(M);
%       axis equal; axis tight; axis off; % Tight & square
%       title('Just before fire propagation')
%       drawnow;

    %% 5. Fire propagation

%       figure(2); clf;
%       colormap(mycolors);

    % Matrix V keeps track of the number of burning neigbours of
    % each grid cell.
    V = BurningNeighbors(M);

    % Compute the spreading of fires from the lighning strike sites.
    % The propagation goes step by step. We keep on trying to expand the
    % fires, until it stopped growing.
    %
    % Catching fire is a stochastic process. We draw random numbers ONCE to
    % guarantee that individual grid cells have a fixed chance of getting
    % burned.
    R = rand(m,m);

    FireGrowing = BurningIni>0; % We went from 0 to >0 fires...
    while FireGrowing

        % Then try to expand the current fire
        [M,V,NewBurned] = FireProgression(M,V,R, f0,c); % spread fire

        % Draw map of updated states
%           image(M);
%           axis equal; axis tight; axis off; % Tight & square
%           title('After fire propagation')
%           drawnow;

        % Do we need to go for another round of fire expansion?
        FireGrowing = NewBurned > 0;
    end

        figure(); clf;
        colormap(mycolors);
        image(M);
        axis equal; axis tight; axis off; % Tight & square
        title(sprintf('After fire propagation of year %d (c=0.01,
        f0=0.5, n=5)', t));
```

```
        drawnow;
    %% 6. Vegetation growth
RevegetatingProbability = RevegetatingProbability(M);
    for i = 2 : m-1
        for j = 2 : m-1
            if M(i,j)==1    % For Exercise 9
                if rand<RevegetatingProbability(i,j)
                    M(i,j)=2;
                end
            end

            if M(i,j)==2, M(i,j)=3; % Young -> Mature
          % elseif M(i,j)==1, M(i,j)=2; % Empty -> Young % Only when
          % not doing Exercise 9
            elseif M(i,j)==3, M(i,j)=4; % Mature -> Old
            elseif M(i,j)==5, M(i,j)=1; % Burning or dead -> Empty
            end
        end
    end
    for i = 2 : m-1
        for j = 2 : m-1
            if M(i,j)==4
                if rand<mort
                     M(i,j)=1;
                end
            end
        end
    end



    Vegetation = sum(sum(M==2 | M==3 | M==4)); % Count

%       figure(3); clf;
%       colormap(mycolors);
%
%       image(M);
%       axis equal; axis tight; axis off; % Tight & square
%       title('Vegetation after growth')
%       drawnow;
end
%% 7. End of program
```

References

[1] https://effis-gwis-cms.s3-eu-west-1.amazonaws.com/effis/reports-and-publications/annual-fire-reports/2019$_{Fire}$$_{Report}$$_{High}$$_{Res}$$_{final}$$_{PT}$$_{correction}$/$Annual_{Report}$$_2$019$_{final}$$_{to}$