# Programmeren (FEB22012)

## 5. Exercise

Deadline for submission: 2011-09-18 23:59 CET

## Simulating Forest Fires

Australia's government is worried about the path of destruction forest fires are carving into the Outback of Australia. Global warming appears to be taking its toll as long dry periods render the forests of Australia extremely vulnerable to spontaneous combustion and to random people who just like to torch a beautiful piece of nature. In order to be able to effectively fight the fires, the government is obliged by law to declare a state of emergency when 6% of a natural reserve has been burned. The Australian government would like you to give them a reliable estimation of the number of days they typically have until they have to declare a state of emergency.

In order to tackle this problem, we first simplify the problem by making several assumptions. First, we assume the natural reserve to be rectangular. Second, we model the outbreaks of forest fires as a stochastic process and assume the time between two outbreaks to be exponentially distributed. The location of an outbreak is drawn from a uniform distribution. If the location of a new fire turns out to have already been burned, the outbreak will simply be ignored in our model. Third, we assume that due to a lack of wind, forest fires grow in a circular way with a steady pace.

As our assumptions render the number of days until declaration of a state of emergency a stochastic variable, simulation of the outbreaks and growth of forest fires over time is a useful technique for addressing the request made by the Australian government. As every simulation may yield different outcomes due to the randomness in our model, multiple simulations can provide a more reliable approximation of the number of days the Australian government typically has until they have to declare a state of emergency.

In this exercise, you will be simulating forest fires in MATLAB in accordance with the model and assumptions presented above. The main algorithm for the simulation is detailed in pseudocode in Algorithm 1.

**Algorithm 1**: Pseudocode for the forest fire simulation.

```
initialize day
initialize burned area
determine starting day for next fire
while burned area $\leq$ threshold do
   spread existing fires
   if today is the starting day for next fire then
      determine coordinates of center
      if coordinates not burned yet then
         start fire
      end
      determine starting day for next fire
   end
   update burned area
   visualize forest
   proceed to next day
end
```

# Exercise

1. Write a function that simulates the outbreak and growth of a series of forest fires over time until a state of emergency is declared.

   (a) The inputs of this function should be:

   - The width in kilometers of the considered area.
   - The length in kilometers of the considered area.
   - The percentage of burned forest area needed in order to declare a state of emergency.
   - The speed in kilometers per day with which the radius of a forest fire increases.
   - The mean number of days until the outbreak of a new forest fire (exponentially distributed).

   (b) This function should produce as output the number of days after which a state of emergency is declared.

   (c) The contents of this function should reflect the pseudocode presented above. The function should make use of two other functions, i.e., one function for checking whether pairs of coordinates have been burned and another function for visualizing the simulation. We will implement these functions in the next steps.

   (d) ***Hint:*** You may want to consider keeping track of all forest fires by storing them in a matrix with, e.g., columns representing the two coordinates of their centers, as well as the current radius of each existing fire.

2. Create a function that checks whether sets of coordinates have been burned by any of the current fires.

   (a) Depending on how the forest fires are stored in your application, this function should at least take as input:

      - A matrix with all sets of coordinates to be checked.
      - The matrix with information on all current fires, including the coordinates of their centers.

   (b) The output of this function should be a vector, signalling for each set of coordinates whether this point has been burned (1) or not (0).

   (c) This function should first compute the Euclidean distances between the requested locations and all current forest fires. Then, if the distance of a location to any fire is smaller than or equal to the fire's radius, the location can be considered to have been burned.

   (d) ***Hint:*** given an $n \times p$ matrix $\mathbf{A}$ with $n$ coordinates in $p$-dimensional space and an $m \times p$ matrix $\mathbf{B}$ with $m$ $p$-dimensional coordinates, the $n \times m$ matrix of their Euclidean distances $\mathbf{D}$ can be computed as

   $$\mathbf{D} = \sqrt{\mathbf{x}\mathbf{1}_m + \mathbf{1}_n\mathbf{y}' - 2\mathbf{A}\mathbf{B}'}, \tag{1}$$

   with $\mathbf{x}$ an $n \times 1$ vector where the elements represent the sum of squared coordinates of $\mathbf{A}$, $\mathbf{y}$ an $m \times 1$ vector with the sum of squared coordinates of $\mathbf{B}$, $\mathbf{1}_n$ an $n \times 1$ vector of ones, and $\mathbf{1}_m$ a $1 \times m$ vector of ones.

   (e) ***Hint:*** you may find the MATLAB function `any` useful.

   (f) Use this function in the main simulation in order to check whether a new location of a forest fire has already been burned as well as to estimate the burned percentage of the complete considered area.

   (g) ***Hint:*** only check for every squared kilometer whether it has been burned or not. You may want to use the MATLAB function `meshgrid` to generate all coordinate pairs at this level of granularity.

3. Create a function that visualizes the forest fires over time.

   (a) The visualization may look like the snapshot presented in Figure 1.

   (b) ***Hint:*** you can plot circles in MATLAB by using the `rectangle` function (as logical as having to press Start to shut down Windows).

   (c) ***Hint:*** prevent your animation from slowing down by clearing the figure before plotting an iteration. You can use the MATLAB command `cla` for this.

   (d) ***Hint:*** in order to be able to actually see your animation, you may need to briefly pause the animation after plotting an iteration.

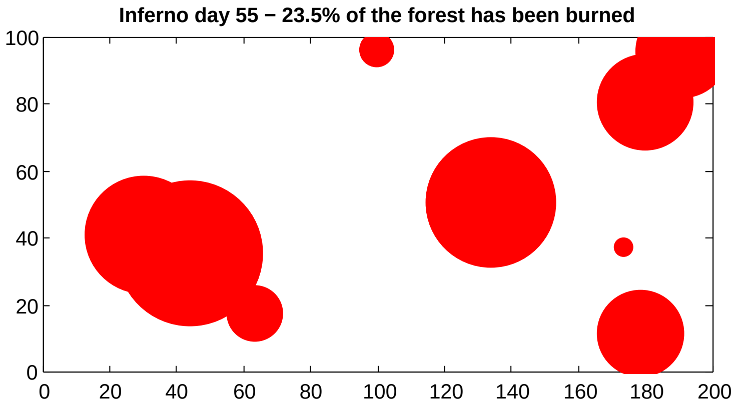

Figure 1: Example of a visualization of the forest fire simulation.

4. Write a script that runs 1,000 simulations of forest fires in an area of 200 by 100 kilometers. Suppose that the time in between the outbreak of two subsequent fires is on average 7 days (exponentially distributed) and that fires typically spread with a speed of 0.5 kilometers per day. How many days does it take on average until the government has to declare a state of emergency after the critical amount of 6% of the forest has been reduced to nothing but ash? Plot a histogram with the results of your simulation. You may want to change your simulation function by adding a parameter with which you can toggle the visualization on and off.