

Physics 218 – Spring, '07 - Assignment #4 – A Forest Fire Simulation

Overview: We wish to look at an example of a “percolation” problem. Percolation problems are characterized by having randomly occurring “bonds” which connect the nodes of a regular grid. The extent to which the system has large regions of connected nodes is often a key element of a percolation problem. In some sense, the properties of the systems are set once the bonds have been determined, but it often takes a fair computational effort to analyze the resulting system.

Forest fires may be viewed as a percolation problem. Connectedness is critical to the spread of a forest fire. Testifying to this is the firefighter’s use of “fire breaks” (creating a swath of forest free of trees) as an important fire-fighting tool.

Our model of a forest will be built upon a $N \times N$, doubly subscripted array, $F(i,j)$. A value of 0 at a location (i,j) indicates that there is no tree at that location. A value of 1 indicates a living tree at that location. A value of 2 indicates a tree is burning at that location. A value of 3 indicates a dead tree at that location. To prepare the “forest” select a number at random from 0 to 1 and create a tree at location (i,j) if that random number is greater than p for some fixed (for each forest) value of p between 0 and 1. So a forest with $p = 0.01$ would be a very sparse forest indeed, with, on average, only $0.01 * N * N$ trees in the entire forest. But a forest with $p = 0.5$ would have, on average, $N * N / 2$ trees in the forest.

Once the forest is prepared, at time $t = 0$, convert every **live** tree on the bottom edge of the forest ($i = 0$, any j) into a burning tree. For subsequent time steps, convert every live tree that is next to a burning tree (burning in this time step) into a burning tree for the next time step and convert every tree that is burning in this time step into a dead tree for the next time step. By “next to” I mean the tree immediately up or down in row number or left or right in column number (**not** those connected by diagonal lines). This is called the von Neumann neighborhood of a cell. Continue to run the simulation until there are no more burning trees.

Product: Now you want to collect some statistics. We will look at two things. We want to produce a graph of the fraction, f , of the forest that was burned as a function of p . So, $f = (\text{number of dead trees at end of simulation}) / (\text{number of live trees at the beginning of the simulation})$. The other thing is to produce a graph of the time (*i.e.*, the number of time steps) it takes for the fire to burn itself out, T , as a function of p . Make at least one version of your program that allows you to see the forest burning.

Now, it is not sufficient just to do one run at each value of p . You will not get the same answer every time. Because of the random nature of the forest, f and T will vary from run to run. So, you will need to average the results of potentially many runs to get good statistics. Do enough runs so that the standard deviation of the mean of f is less than 0.1 for any given value of p . Choose a value of N that gives a reasonable run time. I haven’t explored this in detail with these laptops, so use your judgment. When programmed in *C* or *Java* on slower machines, we regularly did 200×200 forests. You may want to debug

your algorithms with $N=10$ or so, but for the graphs, try to push N as much as you can. Note that the graphs change shape as a function of N so you have to pick a value of N and stick with it in order to gather data for your graphs.

A key to this assignment is to use the functions available to you in IDL and avoiding (where possible) doubly subscripted FOR loops to cycle over the locations in the forest. Here are some potentially useful functions: WHERE, TOTAL, RANDOMU, REFORM, LOADCT, TVSCL, STDDEV. Bowman's book has a chapter on color and will be very handy if you want to add color to your forest images. (Note: You must program in IDL for this assignment.)

Project ideas:

How do your results change if you use a Moore neighborhood instead of a von Neumann neighborhood? If there is some probability less than one that a tree next to a burning tree will be ignited? If the fire is started at one point in the center rather than along an entire edge? If you use periodic boundary conditions? You could also explore how the shape of the graphs change with N . Could you imagine some evolutionary scheme involving more than one type of plant and see if a stable population develops over time and after multiple fires?