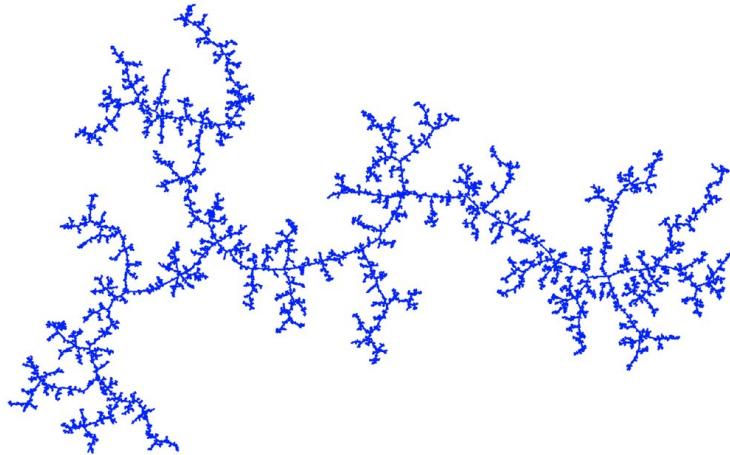


Fractals and Percolation

Luis David Ruiz Ortiz Dominik Pastuszka Malek

July 29, 2024



The geometry of a fractal tree. (Credit: Tom Hutchcroft [1])

Abstract

The main objective of percolation theory is to study the behaviour showcased by systems of nodes distributed over a lattice, and the properties displayed by the clusters of nodes that form when we consider them as part of an interconnected network. In this project, we will review some basic properties of percolative and fractal systems as well as obtain useful results for different types of aggregates via numerical simulations. We will also apply this to solve the real-world problem of forest fires and provide further applications of percolation.

Keywords: *Fractals, percolation, critical probability, fractal dimension, scale invariance, phase transition.*

Table of Contents

1 Brief theoretical introduction	2	4.2 Triangular lattice	9
2 Brief description of the code structure	3	4.2.1 Cluster analysis and critical probability	10
3 Hoshen-Kopelman algorithm	4	4.2.2 Average cluster size	10
4 Static site percolation in a regular lattice	5	4.2.3 Cluster size distribution	11
4.1 Square lattice	5	4.2.4 Behaviour near the critical point. Fractal dimension	11
4.1.1 Cluster analysis. Critical probability	5	5 Fire in the woods	12
4.1.2 Average cluster size	6	6 Dynamic and three-dimensional lattice percolation	14
4.1.3 Cluster size distribution	7	7 Conclusions	15
4.1.4 Behaviour near the critical point. Fractal dimension	8	8 References	15

1. Brief theoretical introduction

Percolation theory was first brought forward by S. R. Broadbent in 1954 in a symposium about Monte Carlo methods in an attempt to find the probability of finding a continuous path that connects the opposite ends of a lattice whose connections have been severed with a certain probability. This problem was addressed shortly thereafter by Broadbent and J. M. Hammersley [2], and a seemingly simple question turned out to be the source of a profound theory of mathematical phase transitions that led to the development of multiple interesting results.

Although it can easily be generalized to a wide variety of different cases, for the sake of simplicity let's first consider a square lattice, where nodes are places in an equally spaced grid.

Between each node and its nearest neighbors, there is a connection that joins them together. We can now begin to cut the connections randomly with a certain probability. Initially, we have a system in which the ends of the lattice are connected, even if we begin to remove some small number of connections. However, as the number of severed connections increases, the probability of finding a path that joins the farthest ends decreases. This is the basis of the so-called *bond percolation*.

Alternatively, one can also consider a model in which all nodes are always connected with their nearest neighbors, but it is now the nodes that we randomly remove from the lattice, leaving unoccupied states that are left disconnected from the rest of the nodes. This is called *node percolation*.

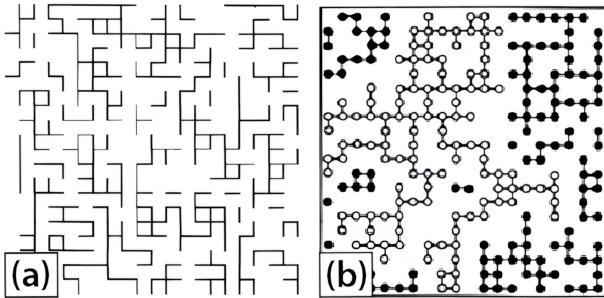


Figure 1: Schematic representation of bond percolation (a) versus site percolation (b). (Credit: A. Bunde et al. [3]).

In both models, the probability parameter p controls the creation/removal of the randomly-chosen connections or nodes. To fix the notation from now onwards, this p will denote the probability of a node/bond being active. The probability of removing them will therefore be $(1 - p)$.

We define a *cluster* as a group of nodes directly connected among each other, such that there is a connected path between any two nodes of the cluster. For a very high value of p , the majority of the nodes of the network will be in a single cluster; maybe with some small isolated clusters inside. For a low value of p , a majority of the connections will be lost and there will be a large amount of disconnected small clusters.

Between those cases there appears to be a *critical probability* p_c above which the first percolating cluster can be found [4]. A cluster is said to be percolating if it contains a connected path from one extreme of the network to the other. Such chain of nodes is called a *percolating path*. If such a cluster can be found, we say the system is *percolating* (or supercritical), while for the opposite case, we say it is *non-percolating* (or subcritical) [5].

The fact that by varying the “creation/removal probability” p the system turns from a percolating to a non-percolating state can be understood as a *phase transition* [6], that occurs at the critical point $p = p_c$ (also known as percolation threshold). As we increase the probability p in the subcritical phase, the “length” of the clusters is expected to increase until $p = p_c$, when a cluster that connects extremal points of the lattice is formed and coexists with a number of medium and small size ones. If we continue increasing the probability $p > p_c$, one should find that the percolating cluster contains more and more points until it spans the whole lattice.

It can be proven [7] that right at the critical probability $p = p_c$, the system of clusters behaves as a fractal, a geometrical object whose “mass” (in this case, the number of nodes) is proportional to the linear size L of the system via a *power law*:

$$M \propto L^{D_f} \quad (1)$$

where D_f is the *fractal dimension*.

This proportionality relation implies that the system is *scale invariant* (or self-similar). This is due to the fact that, as we change the scale at which we observe the fractal, the mass will change accordingly leaving the same pattern at all scales.

As the system gets closer to the critical point, a variety of magnitudes start being described by power laws, allowing the determination of several critical exponents from a numerical simulation.

To mention a couple of examples of the parameters that the theory predicts will be governed by a power law (though not all will be covered in this project), we have already established that the mass of the clusters will depend on their scale via a power law whose exponent will represent the fractal dimension (d_f); the size distribution of clusters is expected to be characterized by the Fisher critical exponent (τ) and close to p_c the correlation length will diverge as the power law $\xi \sim |p - p_c|^{-\nu}$.

This exponents are expected to be independent of the lattice microscopic structure or size (as long as it is sufficiently large to avoid finite size effects). This characteristic is what is known as the *universality* of critical exponents and show that some aspects of this kind of critical phenomena are governed by the same laws.

2. Brief description of the code structure

The code used in this project can be found uploaded at <https://github.com/domipm/Complex-Systems-Percolation/tree/main>. In this section, we will briefly describe the functionality and purpose of all the written files and how we simulated percolation processes with them.

Most of the program is written in C++, with certain parts such as the graphing script and some additional examples written in Python.

In the main folder, we can find the header file *percolation.hpp* which contains all the necessary class definitions and functions used for the *Node*, *Lattice*, and *Cluster* objects.

We begin the process by generating a file via the *lattice_generator.cpp* file. In this program, we can choose between a square, triangular, and randomly distributed lattice, given the number of nodes, the length of the square in which the nodes are generated (bounds for the positions), and the probability for each node being active or not. This then outputs the lattice into the file *lattice.txt*.

Afterwards, *hoshen-kopelman.cpp* reads this lattice file and groups the nodes which are at a distance equal or less-than the threshold distance (using the algorithm described in section 3), and generates another file called *lattice_sorted.txt*, which contains the same information as before but now each node has a cluster index parameter.

Optionally, we can plot the result of this file via the script *graph_sorting.py* to visually see the different clusters formed.

Finally, we run the program *cluster_stats.cpp* to find how many different clusters we have and all the necessary parameters to perform the analysis. For each cluster, it computes the number of nodes it contains and how much does it extend in each direction. Finally it checks if the system contains a percolating cluster and determines consequently if the file *lattice_sorted.txt* corresponds to a percolating or non percolating scenario.

In the repository, we can also find some folders, into which are grouped the remaining code files used for all additional parts of the project, with most being just tools to study the behaviour of the system while looping over some parameter.

The *forest fire* folder contains all the code necessary for section 5, in which we aim to provide a solution to the spreading of fires in a some more realistic forest using percolation in non-regular lattices.

Lastly, the folders *galaxy percolation* and *three-dimensional percolation* contain some basic codes that were used to illustrate possible further applications of percolation theory in section 6.

3. Hoshen-Kopelman algorithm

Since we are interested in studying the different properties of the clusters of nodes formed in percolating systems, we must first decide how we will be generating or finding these clusters. To do this, there are mainly two routes we can take: the first one is to simply generate the clusters while we perform the percolation algorithm, activating or not nodes neighboring an initial node and adding them to the cluster. This is the algorithm presented in [7], and while it is simpler to implement, we believe it to be less generalizable.

Therefore, in this project, we have decided to implement a generalized Hoshen-Kopelman algorithm, which aims to find clusters by labelling the already-generated nodes using the distance between them. This lets us find clusters in percolating systems in non-lattice environments, for example, when generating nodes at random positions, while still being applicable to lattice environments such as the square or triangular lattices. The algorithm we have implemented follows the one presented by Ahmed Al-Futaisi and Tadeusz W. Patzek in [8], and its main steps are as follows:

1. Generate or read the positions, indices, cluster indices, activation status, etc. of all nodes.
2. For all active nodes, find their active neighbors (nodes within a certain threshold distance).
3. Scan over all active nodes. If a given node has no labelled neighbors, start a new cluster and add the node to it. If the node has at least one labeled neighboring node, set the cluster index of the node to the minimum between its own value and the minimum value of the neighbors.
4. Perform previous step iteratively until the cluster indices of all nodes converge to a solution that labels each node and all of its neighbors with the minimum cluster index.

With this algorithm, we can now find all the different clusters in any kind of lattice, and it can be easily generalized to three dimensions. As an example, figure (2) showcases how, for various initial distributions of nodes, the algorithm successfully manages to distinguish and label all the different clusters, making it easy to then study the properties of each cluster.

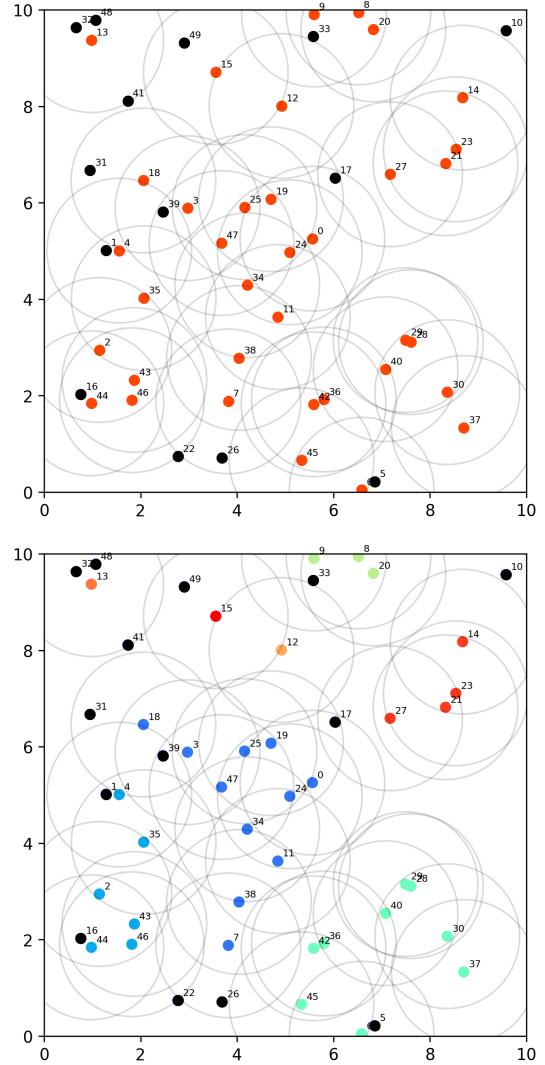


Figure 2: Extended Hoshen-Kopelman (HK) algorithm applied to find clusters of active nodes in a randomly distributed lattice with $N = 50$ nodes, activated with probability $P = 0.65$, and with a neighbor-finding threshold distance of $D = 1.5$. The image on top shows the initial, unlabeled nodes, while the second image shows the output after grouping the nodes into clusters via the HK algorithm. The different color points represent nodes in clusters, while black-colored points represent inactive nodes.

4. Static site percolation in a regular lattice

4.1. Square lattice

We begin our analysis by considering a square lattice, generating a grid of equally spaced $L \times L$ nodes and a probability of node activation p . To determine if a given node is active or not we assign it a random number p_i between 0 and 1. If this number is smaller than p the node will be active. Considering a distance threshold of $D = 1.25$, we ensure that all nodes will have at most 4 neighbors, corresponding to nodes situated at the left, right, top and bottom (that is, we will not consider the nodes situated in the diagonal directions as neighbors). Figure (3) shows an example of a square lattice at the percolation probability, where the fractal properties of the system are showcased.

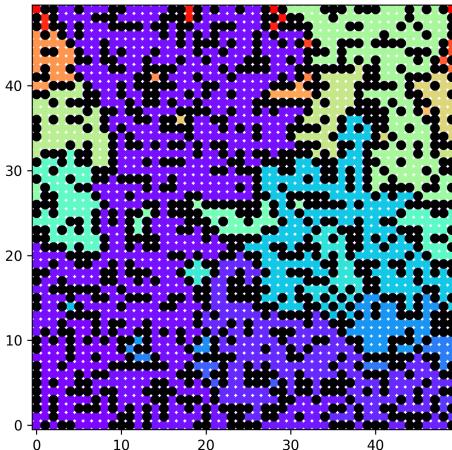


Figure 3: Example of a square lattice (of side length 50 nodes) at the critical probability ($p \sim 0.58$) showcasing self-similarity.

4.1.1. Cluster analysis. Critical probability

In this section we are going to study the critical probability and how it depends on the size of the lattice. It is important to establish the criteria that will be used to determine if a system percolates. The concrete values obtained in low scale lattices will be dependent on the specific criteria, even if the behaviour in the $L \rightarrow \infty$ limit is expected to be the same regardless.

We will say that the system percolates if it contains at least one cluster with scale equal to the side length of the lattice. We define the *scale*

of a cluster to be the maximum between its lineal extension in the horizontal and vertical direction. Being more precise, if the x component of all the nodes in the cluster are within the interval $[x_{min}, x_{max}]$ and the y components are contained in $[y_{min}, y_{max}]$, the scale of the cluster will be $s = \max\{(x_{max} - x_{min}), (y_{max} - y_{min})\}$.

To minimize statistical fluctuations in the determination of the percolation threshold, the steps followed to calculate p_c for a given lattice size are:

1. Perform one simulation for each probability from 0 to 1 in steps of 0.1. In each simulation, check whether the system percolates or not. There will be a probability for which the transition between percolating to non percolating behaviour occurs. In an interval of 0.2 around that transition probability a finer study is performed to get a more accurate value for p_c .
2. Within that range, 4 simulations are made for each probability in intervals of 0.02. We will register the fraction of the times that the system percolates for each probability. The critical probability will be fixed to the one that gives rise to percolation 50% of the times.
3. In case of non conclusive results, we will increase the resolution to 0.01 and perform the same analysis.

We have studied how the critical probability evolves as the side length of the lattice ranges from 20 to 500. The results are shown in the figure 4.

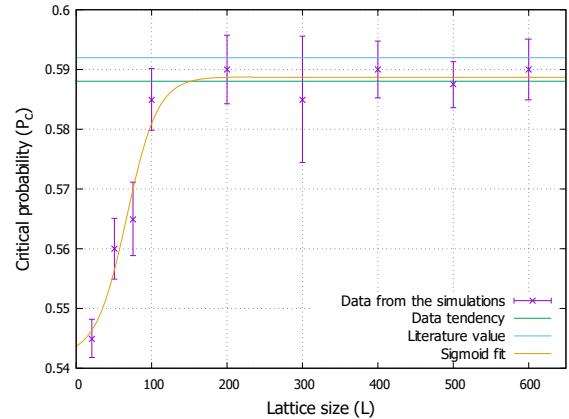


Figure 4: Critical probability as a function of lattice size for a square lattice.

The errors have been computed to take into account both the statistical dispersion and the finite discretization (precision) of the probability. The expression used is:

$$\Delta p = \sqrt{\left(\frac{\delta p}{\sqrt{12}}\right)^2 + \left(\frac{\sigma}{\sqrt{N}}\right)^2} \quad (2)$$

Which is the standard way of combining errors due to precision and statistical ones. In that expression, δp represents the spacing between probabilities, N is the number of simulations performed for the fine tuning and σ^2 is an effective variance. Since there is no parameter that is being measured various times, we have chosen to work with an effective standard deviation obtained from the probability of percolation distribution. The standard deviation σ will be the distance from the percolation probability to the closest probability that percolates either 0% or 100% of the time 5.

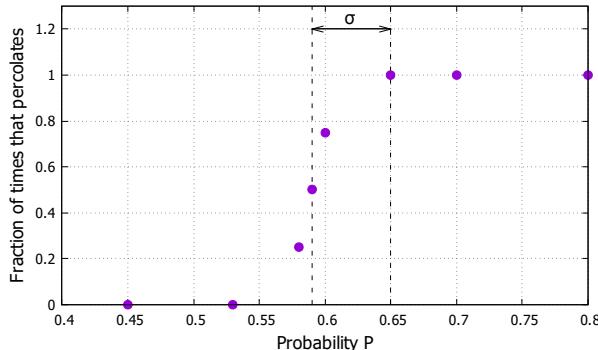


Figure 5: Effective standard deviation.

Regarding the critical probability and its evolution with L there are some aspects worth commenting. The first is that, at small lattice sizes, there are significant variations. A part of that can be traced back to low statistical significance (a denser sample of sizes may make the tendency clearer) but it is clear that there is still an intrinsic dependency of p_c with the largeness of the lattice. Those finite size effects were to be expected since the exact calculations for percolation probabilities are always made accounting for an infinite lattice.

The behaviour of the critical probability as the lattice gets larger quickly stabilizes, converging towards a value of $\bar{p}_c = (0.588 \pm 0.003)$. This value is computed as the average of the last 6 points. Since there is no analytical solution to the

site percolation threshold problem in a 2D square lattice, we can only compare it to other simulations.

Though our approximate result is slightly lower than the ones found in the literature [9], the relative error is in the order of 0,7%; so we can consider it a reasonably good agreement within the margins of uncertainty. To go one step further, we wanted to fit our data to a function that could describe the behaviour of the system in the big size limit. Since we know the data is supposed to converge in the limit $L \rightarrow \infty$, we choose a bounded function with a limiting value we can use to infer the value of p_c in the thermodynamical limit. We choose a sigmoid with 4 free parameters for the fitting:

$$p_c(L) = \frac{a}{1 + \exp(-bL + c)} + d \quad (3)$$

With this expression, we can calculate the critical probability in the thermodynamical limit, which is given by $p_c(\infty) = a + d$. So by knowing the parameters of the fit we can infer a reasonable estimation of the limiting p_c that our model predicts beyond the points we have simulated. By this method, the critical probability obtained for the infinite lattice limit is $p_c(\infty) = 0.589 \pm 0.013$, which is slightly more in line with the literature. Combining both of our results (average of simulated points and limit of the fitting curve), we get the result we are going to take as representative of the critical probability for the square lattice $p_c = 0.588 \pm 0.007$.

4.1.2. Average cluster size

In this section, we study the average cluster size \bar{s} of the square lattice, which is calculated as the average number of nodes contained in all finite clusters. Figure (6) shows the obtained value of the average cluster size as a function of the probability p for various lattice sizes. These values have been obtained by averaging the results of five simulations, and the sampled probabilities have been chosen in such a way that more points are studied around the critical probability and less points are simulated away from it, as to maximize the precision.

We can see that the graph has three distinct parts: an initial slow rise of the average cluster size as we increase the probability, a maximum value which is reached right below the critical probability,

around $p = 0.58464$, and a rapid fall-off right after as the average cluster size tends to one.

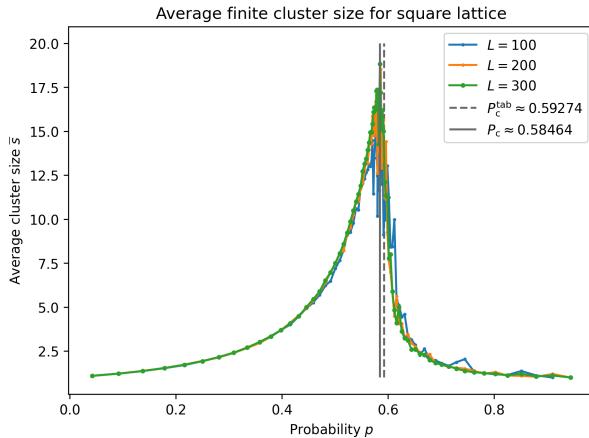


Figure 6: Average cluster size as function of probability for various square lattice sizes.

This behaviour is qualitatively the one expected given the characteristics of the system. In the subcritical phase, the system consists on a series of small disconnected clusters. As more nodes get activated, they will either become a new 1-noded cluster or spawn connected to an existing cluster. The former possibility (that will not increase the average cluster size) is only relevant at very low values of p ; whereas the latter (that does increase \bar{s}) becomes the most likely option as the probability gets closer to p_c . This is why, in this phase, the slope of the rise gets steeper as p grows.

The later decrease is explained by the percolating cluster progressively incorporating more of the medium size clusters. The fact that the descent in average size starts immediately after p_c is an indication that the phase transition between percolating and non-percolating system occurs abruptly.

4.1.3. Cluster size distribution

In figure (7) we can see represented the number of clusters with a given size, n_s , for different probabilities with respect to the critical probability. All of them were obtained using a square lattice of side length 300 nodes, which we estimate is enough to obtain statistically significant results within a reasonable computation time.

In the first figure (7a), we can see the cluster size distribution for a probability below the critical, in particular we have chosen $p = 0.25$. In this case, the distribution clearly follows an exponential decrease (the vertical axis is in logarithmic scale), meaning most clusters contain one or at most a couple nodes, while bigger clusters are very unlikely to be found.

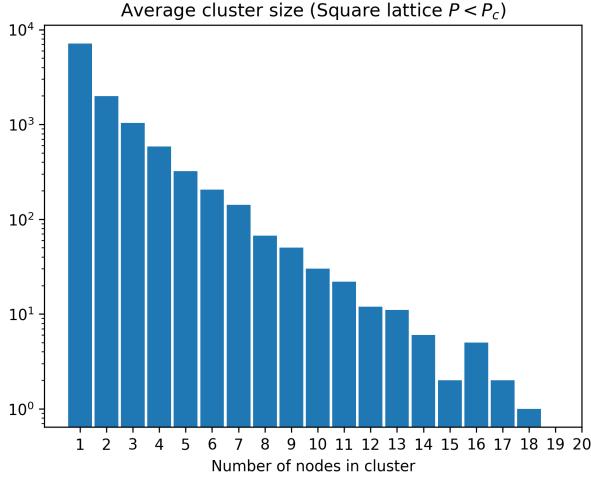
As for the second figure (7b), it showcases the size distribution just below the percolation probability, for a value of $p = 0.58464$. With the help of the embedded log-log plot, we can see that we obtain a distribution that follows a power law. This is expected, since at the critical probability we obtain an infinite cluster and the system presents self-similarity. To further study this, a power law fit was performed with a fitting function:

$$f(n_s) = a n_s^{-\tau} \quad (4)$$

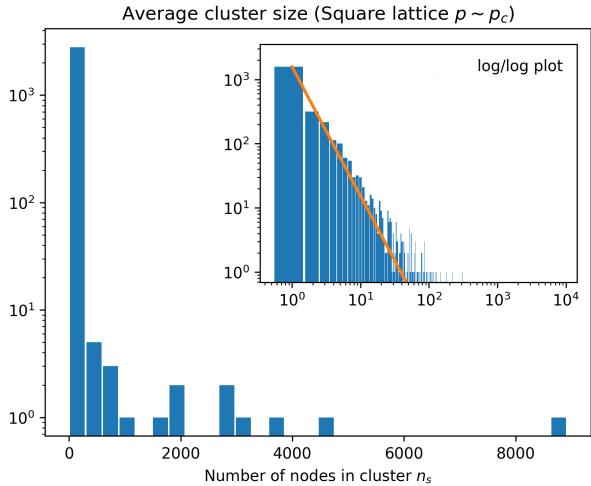
To compute this critical exponent the average of 10 simulations was considered, obtaining a value of $\tau = 2.015 \pm 0.008$, which is in good agreement with the Fisher exponent whose value in the literature is $\tau = 187/91 \approx 2.0549$ [10].

There is still room for improvement in this point though. Obtaining sufficient statistical significance for the very large cluster sizes is computationally demanding but more simulations will reduce the statistical noise that affects the region around 10^2 in the log/log plot. This may help our obtained value of τ to get closer to the generally accepted one.

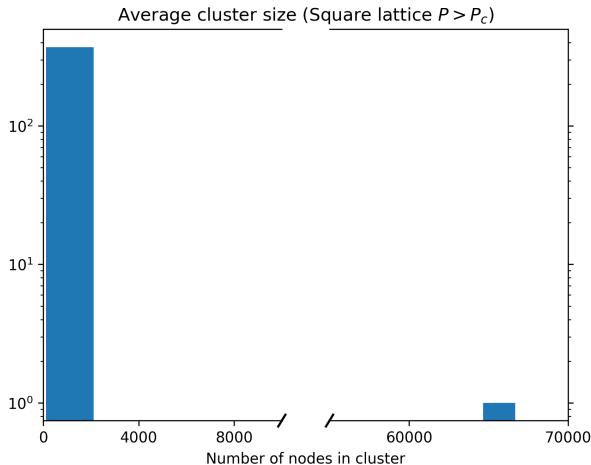
Finally, figure (7c) shows the same distribution for a probability $p = 0.75$, well above the critical. The result is quite trivial, showcasing how in this case we have one cluster containing most of the nodes (the percolating or infinite cluster) and a lot of small isolated clusters containing each a smaller number of nodes.



(a) Below critical probability ($p = 0.25$).



(b) At critical probability ($p = 0.58464$). Orange line represents power law fit.



(c) Above critical probability ($p = 0.75$).

Figure 7: Frequency of cluster with given number of nodes, representing the node distribution at different probabilities for the square lattice (of side length $L = 300$ nodes).

4.1.4. Behaviour near the critical point. Fractal dimension

The power law obtained for the cluster sizes near the critical probability was a first indication that the system has some form of scale invariance at the point where the phase transition occurs. However, we have decided to perform a deeper analysis of the fractal nature of the cluster distribution. The most common way of characterising these kind of systems is to obtain their *fractal dimension* (d_f). For a given figure, this parameter is calculated in terms of how its mass depends on the scaling factor according to:

$$M(s) = C s^{d_f} \quad (5)$$

With C being some proportionality constant. It is important to note that this definition refers to a single fixed shape that gets rescaled according to s . We are going to perform the study taking one mass and one scale for each cluster in a configuration very close to the critical point. For the expression (5) to be applicable we are making the assumption that the scaling properties of all clusters is going to be the same and, therefore, we can compute a single fractal dimension from data of multiple clusters. This is a well founded assumption given the fact that the percolation process is not biased in principle towards the generation of any particular spatial pattern.

In our case, the mass of a cluster is computed considering each node as contributing with a unit of mass, so the total will be the number of nodes the cluster contains. The scaling factor will be computed as established in section 4.1.1.

Having this, we can compute the fractal dimension just below the critical probability for each lattice size studied. We will make a linear fit of the logarithm of the mass of the clusters with respect to the logarithm of the scale and the slope of that fit will be the fractal dimension.

For each lattice size, this computation will be performed for three node configurations with the same p and we will assign the value of the fractal dimension to the average of the results found. Figure (8) shows the fractal dimension computed in this way:

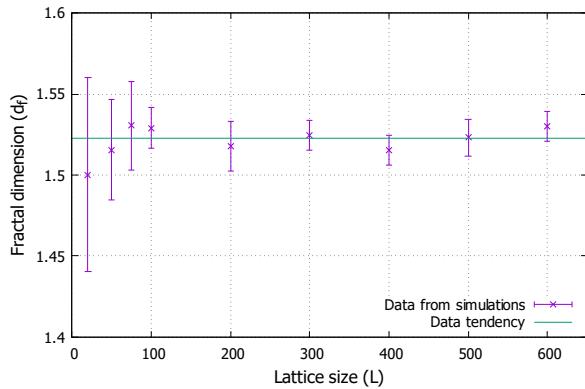


Figure 8: Fractal dimension as a function of the lattice size for square lattice.

The uncertainties in the fractal dimension parameter have been calculated according to the expression:

$$\Delta d_f = \sqrt{u_B^2 + \left(\frac{\sigma(d_f)}{\sqrt{N}} \right)^2} \quad (6)$$

to take into account the intrinsic error margin that each parameter from the fit has and its statistical variability. In this context, and being d_f a derived magnitude, u_B will depend on the individual errors associated with each fit parameter (Δd_f^i) as follows:

$$u_B(d_f) = \sqrt{\sum_{i=1}^3 \left| \frac{\partial d_f}{\partial d_f^i} \right|^2 (\Delta d_f^i)^2} = \frac{1}{3} \sqrt{\sum_{i=1}^3 (\Delta d_f^i)^2} \quad (7)$$

We can see that the value of the fractal dimension does not have much variation with the size of the lattice. However, the error bars associated to d_f are significant for small size lattices. This is due to the fact that, for small lattices, there are fewer clusters. This increases the uncertainty in the slope of the fit and that uncertainty propagates to the average fractal dimension.

The average fractal dimension found from our simulations is $d_f = 1.521 \pm 0.009$. We expected to obtain a value between one and two since the clusters fill the 2-dimensional space less efficiently than a solid object ($d_f = 2$) because of their rough edges and holes. It is, however, smaller than the value found in the literature [11] of $d_f = 1.72 \pm 0.04$.

We attribute this to a systematic error driven by the fact clusters are not evenly distributed in scales. The region of small scales is much more populated but also has a larger dispersion (9).

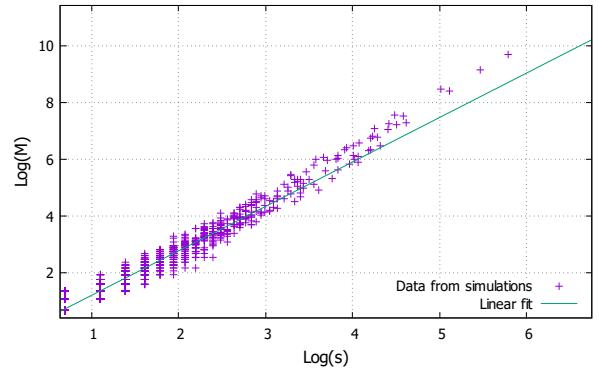


Figure 9: Logarithm of the mass of the clusters as a function of the logarithm of the scale. This example was calculated for a lattice of $L = 400$.

This asymmetry makes the low scale clusters have a greater weight in the least square calculation and reduces slightly the slope of the fit. A further methodological analysis may be needed to study whether this is a systematic error that needs to be taken into consideration.

The discrepancy in d_f , in any case, is not dramatic ($\sim 10\%$ of relative difference) and the value calculated here still gives a clear insight about the scaling properties of the cluster in this critical point.

4.2. Triangular lattice

Following a similar procedure as for the square lattice in the previous section, the triangular lattice can be seen as a square lattice whose every second row is displaced by half a distance unit. We again consider L as the size of the lattice, and it corresponds to the number of nodes in each row. The distance threshold is kept as $D = 1.25$ so as to ensure each node has at most 6 neighbors. Figure (10) showcases the self-similarity of a triangular lattice at the critical percolation probability.

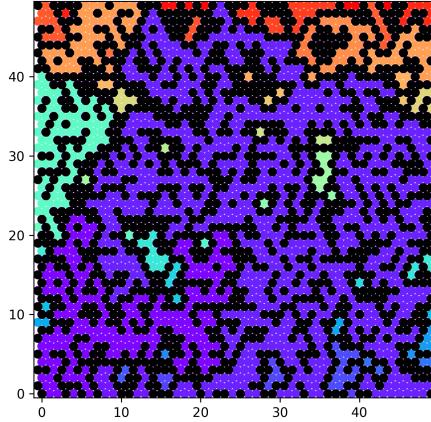


Figure 10: Example of a triangular lattice ($L = 50$ nodes) at the critical probability ($p = 0.5$).

4.2.1. Cluster analysis and critical probability

As it has been done for the square lattice, the goal here is to find the critical probability for a range of lattice side lengths.

The results are shown in (11) and we can see a similar pattern of convergence for large L with the difference that in this time all values (even for small sizes) are closer to that limiting value. This can be attributed to the fact that, for a triangular lattice, the site percolation threshold can be shown to be exactly $1/2$.

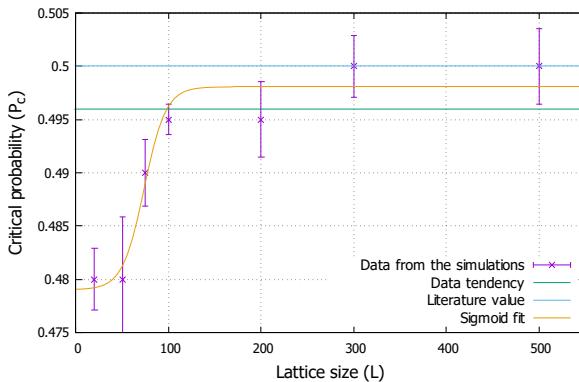


Figure 11: Critical probability as a function of lattice size.

We will perform an analogous analysis as the one for the square lattice in order to obtain the value of this critical probability from our simulations. The first approach, averaging the probability obtained from the higher lattice size

points, gives a result of $\bar{p}_c = 0.496 \pm 0.004$. As we can see, the obtained value is in accordance with the exact result within the error margins, though slightly lower since the tendency for small lattices is to have a smaller p_c .

Performing the same logistic fit as for the square lattice we can obtain an estimate result for the critical probability in the thermodynamical limit in terms of the parameters of the fit. In this case, the result obtained is $p_c(\infty) = 0.50 \pm 0.12$. Though the errors carried from the uncertainty in the fit parameters are significant, the result matches immaculately with the expected result. Combining both results, we get a final percolation threshold of $p_c = 0.50 \pm 0.06$.

4.2.2. Average cluster size

In figure (12) we can see the average cluster size \bar{s} represented as a function of probability p for various sizes of a triangular lattice. We can see that the maximum occurs almost precisely at the critical probability, approximately at $p = 0.50143$.

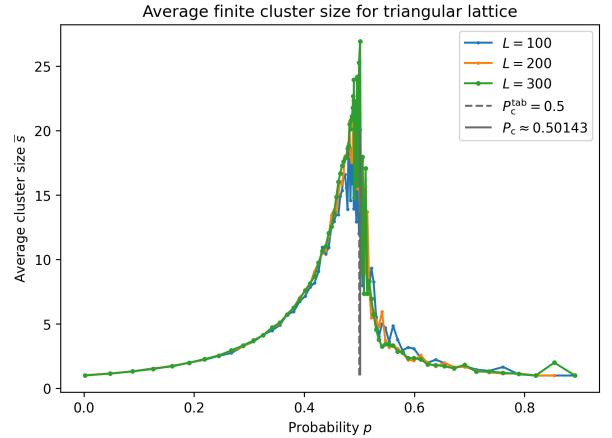


Figure 12: Average cluster size as function of probability for various triangular lattice sizes.

The discussion is identical to the one for the square lattice. Even if the geometry of the lattice influences the details and the numerical result for the critical probability, the general aspects and the behaviour of the system in both phases and in the transition point are the same for both kinds of lattices.

4.2.3. Cluster size distribution

Similarly to the square lattice case, we have also studied the cluster size distribution for the triangular lattice, taking again a side length of $L = 300$ nodes. This distribution for various values of the probability can be found in figure (13), where the transition between the system being in two distinct states can be seen.

In general, for the triangular case we have found a very similar behaviour as for the square lattice. In the first figure (13a) we have the cluster size distribution for a probability $p = 0.25$ below the critical probability. Again, an exponential decrease is found, however now the graph with the vertical axis in logarithmic scale looks less like a straight line (such as that for the square case) and more like a power law. This is because the critical probability for the triangular lattice is lower than for the square lattice, and therefore the chosen probability is closer to the critical than before.

In the second figure (13b), the distribution is shown for the critical probability $p = 0.5$, and the behaviour found is again a power law, signaling the scale invariance of the system. Performing a similar power law fit as in the square lattice (4), we have found the critical exponent to be: $\tau = 1.484 \pm 0.003$. This result differs from the one obtained for the square lattice, which seems to be in disagreement with the universality of the critical exponents. Incrementing the lattice size does not affect this result significantly, therefore we exclude the possibility of it being a statistical anomaly and instead being fundamentally dependent on the structure of the lattice.

Lastly, figure (13c) shows the cluster size distribution above the critical probability, $p = 0.75$, where we again find the percolating or infinite cluster containing most of the nodes in the system and a lot of smaller clusters containing fewer nodes.

4.2.4. Behaviour near the critical point. Fractal dimension

Lastly, we have computed the fractal dimension for the triangular lattice just below the critical probability. The methodology has been the same as the one described in 4.1.4 and very similar results were found (14).

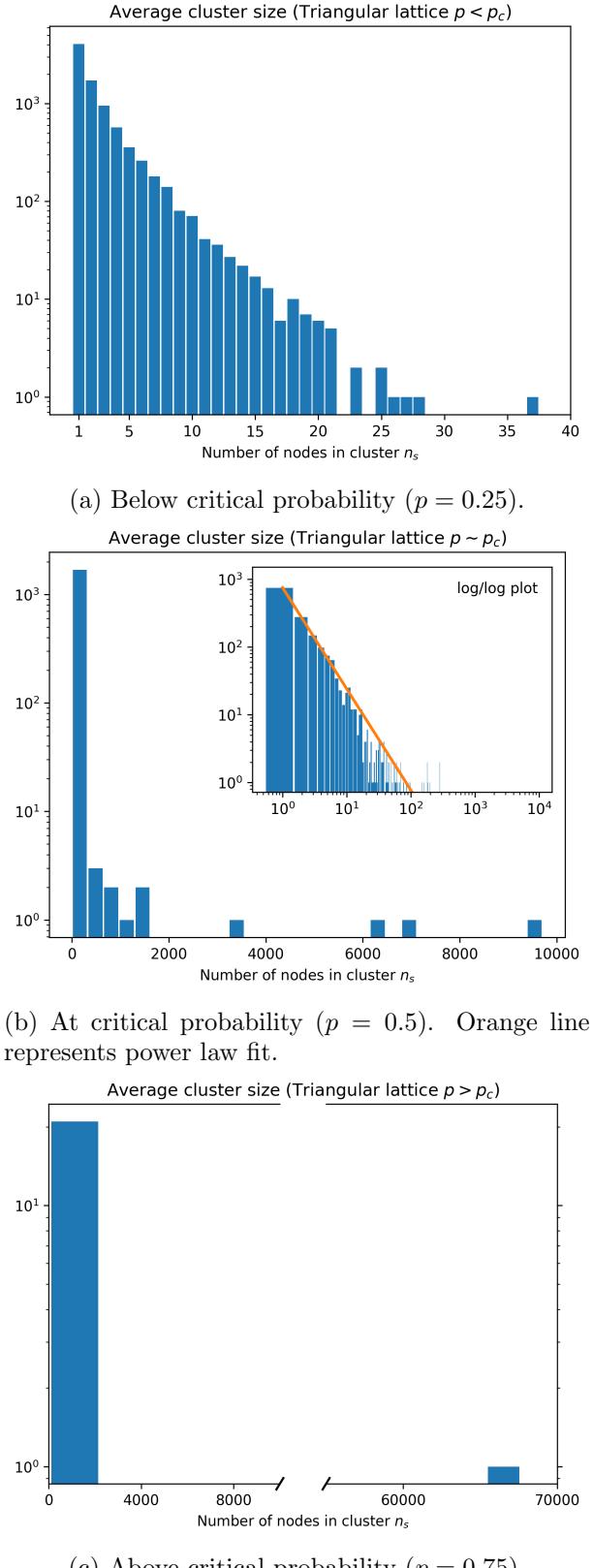


Figure 13: Frequency of cluster with given number of nodes, representing the node distribution at different probabilities for the triangular lattice (of side length $L = 300$ nodes).

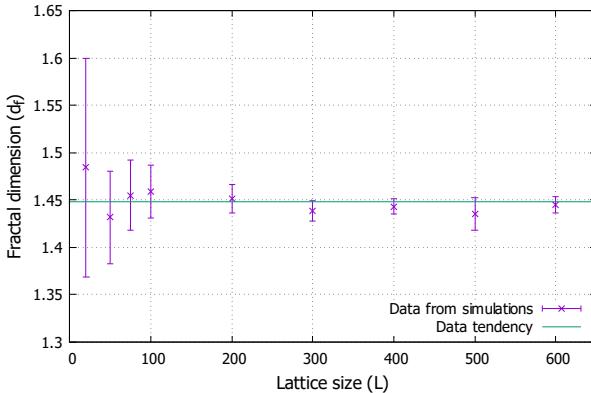


Figure 14: Fractal dimension as a function of lattice size for a triangular lattice.

The pattern is similar to the one found for the square lattice, increasing the convergence of the values while reducing the error bars as we increase the lattice size. The average fractal dimension found in this case is $d_f = 1.45 \pm 0.02$, which deviates significantly both from the result for the square lattice (5%) and from the value from the literature (15%).

As discussed in section 4.1.4, we attribute it to an underestimation of the slope of the fit caused by the dispersion of the data in the low mass end of the range. To solve this, a fit can be done only for large values of cluster mass and scale but the criteria for determining which points to include is not clear.

5. Fire in the woods

In this section, our aim is to give an answer to the main proposed problem in this project: given a forest where one tree catches on fire, how many trees do we have to cut down in order to stop the fire from spreading over all the forest?

In order to do so, we have decided to slightly modify the system to make the forest more realistic. Considering each distance unit equals one meter, we will generate a randomly distributed lattice with two different types of trees generated with a density similar to that of Yosemite's National Park in California, USA ([12]), where big trees are very common and a lot of data is available. In particular, we have chosen a tree density of 0.1475 trees per square meter.

Incorporating the different types of trees is done by them having a different neighbor distance threshold, so the way they propagate the fire differs (those with a higher distance threshold will spread the fire more than the others). We can think of it as them being two different tree species (one which catches fire more easily than the other) or trees of two different sizes (big trees versus smaller shrubs). In particular, we have chosen a distance threshold of $D_b = 5$ m for bigger trees and $D_s = 3$ m for smaller ones. Figure (15) shows the distribution of trees in this simulated forest.

Our objective in this section will be to study how the ratio between big trees and smaller shrubs, which has been shown to affect the severeness of forest fires [13], affects the critical percolation probability. To do so, we will focus on studying the average cluster size for various probabilities, having fixed the ratio to big-to-small trees. We will only sample enough points as to obtain an estimate value of the critical probability and see how the ratio of different trees affects it in a general sense, since the exact probability will depend too much on variables such as the density and size of the forest, as well as random fluctuations.

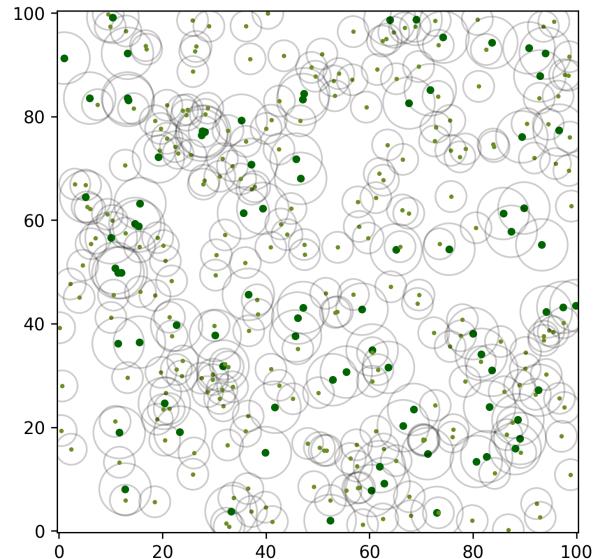


Figure 15: Example of (a small patch of a lower density of) our simulated forest, where the big dark points represent bigger trees, while smaller light green points represent shrubs or smaller trees. The black transparent circles around each point represents the neighbor distance threshold of each tree.

Due to this dependence, throughout this section we will fix the size and density of the forest, as well as generate the forest with always the same seed for the random number generator (this way, we always generate the same random lattice).

Figure (16) shows the average cluster size obtained as a function of probability for various ratios of big-to-small trees. We know that the percolation occurs at the probability for which the average cluster sizes reaches its maximum and then falls off, and even though near these maxima we don't have too much precision, we can obtain an estimate of this critical probability and study its general behaviour after varying the tree ratio.

Comparing with the average cluster size graphs of the regular lattices from the previous section, we can see that this graph is more erratic: apart from the maximum which tells us that the system has percolated, we find other maxima at higher probabilities which we believe are completely due to the random distribution of our forest, and the fact that other big clusters appear apart from the percolating cluster.

The approximate critical probability for each ratio value is represented in table (1) and we can see that, in general, as we decrease the number of big trees and increase the number of small ones, the critical probability shifts to higher values. This indicates that, the higher the proportion of trees with smaller area of effect, the less trees will have to be cut in order to stop fire from percolating, something that is completely expected. If the goal is to increase the overall tree density, this small/less fire-propagating trees are a way of increasing the number of trees while keeping the system in the non-percolating phase. Another thing to notice is the fact that the initial exponential increase slows down as we decrease the ratio (the percolation process "slows down" in terms of probability). If we only generate small trees ($R = 0$), the system does not percolate. This corresponds to a forest where the only tree species will be the one with small range of fire propagation. This $D_s = 3\text{ m}$ is small enough in comparison with the average distance between trees in our model so that there are always disconnected tree clusters.

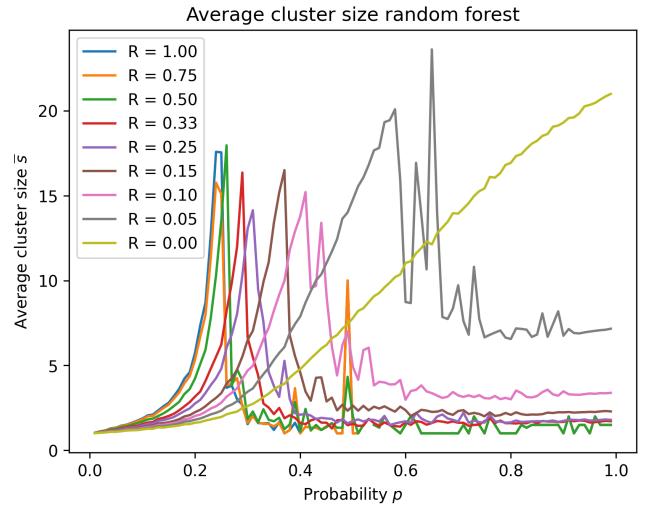


Figure 16: Average cluster size as function of probability for various tree ratios R , for a fixed random lattice of side length $L = 500\text{ m}$ and density 0.1475 trees/m^2 . Big trees have a neighbor distance threshold of $D_b = 5\text{ m}$ while small ones $D_s = 3\text{ m}$.

Tree Ratio R	Critical Probability p_c
1.00	0.2401
0.75	0.2442
0.55	0.2600
0.33	0.2905
0.25	0.3104
0.15	0.3696
0.10	0.4097
0.05	0.5780
0.00	∞

Table 1: Approximate critical probability of system for various tree ratios.

All these numerical results depend entirely on the type of forest we generate and the concrete tree distribution that the seed gets rise to. However, we expect the general behaviour found here to be extrapolable to other random lattices (similar results were found for other randomly generated lattices).

There is still plenty of possible extensions and further work that can be done in this model of fire propagation. For once, a more exhaustive analysis of the critical probability, gathering results for a statistically significant number of starting tree distributions, may provide insight about how the concrete shape of the random forest affects the percolation and whether there are patterns in the tree distribution that can contribute to increasing or decreasing the critical probability.

One of the main advantages of our object-based percolation model is that it has a great potential for extensions. We have only differentiated trees by their neighbour distance threshold but new parameters can be included for all nodes of the lattice to implement more complex characteristics in relation to fire propagation. This could be a simple basic model to study how biodiversity affects the propagation of fire through the forest.

Outside of this particular model, a more realistic study of fire propagation through percolation theory will have to take into account the dynamics of the fire and not only the distribution of tree clusters. This can be done with dynamical percolation theory, which we will briefly address in the next section.

6. Dynamic and three-dimensional lattice percolation

Although percolation theory was at first developed to solve a very specific problem, the fact that generalizations of networks of interconnected nodes appear frequently in nature leads to the fact that we can apply it to multiple and diverse systems. This in turn allows us to model the system and even predict its behaviour, obtaining useful results along the way even without having to use any theoretical results.

An interesting and unexpected application of percolation theory arises in the field of astrophysics and cosmology. It has been found that percolation theory can be useful to analyze the geometrical and structural properties of the cosmic web [14]. Percolation techniques have been also used to study clustering of galaxies [15].

Staying in the astrophysical realm, during this project we have managed to replicate, at least to some extent, the results presented by L. S. Schulman and P. E. Seiden in [16], which models star formation in spiral arms of galaxies following the theory of propagating star formation. We have done this by applying the two-dimensional random lattice algorithm with some slight modifications as to account for the differential rotation of the galaxy and the radially distributed nodes. Figure 17 showcases a dynamic percolation simulation where the obtained star forming regions reproduce a spiral arm structure similar to the ones of galaxies.

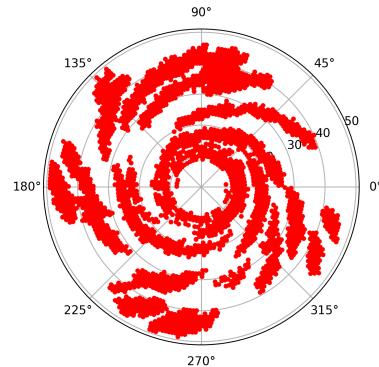


Figure 17: Result of applying a dynamic percolation algorithm on a lattice of spherically distributed nodes with differential rotation.

The algorithm used in this project is also easily generalizable to three-dimensional lattices, which can be useful to study, for example, the flow of liquids through porous materials [8]. In figure 18 is illustrated a simple three-dimensional square lattice after applying the clustering algorithm.

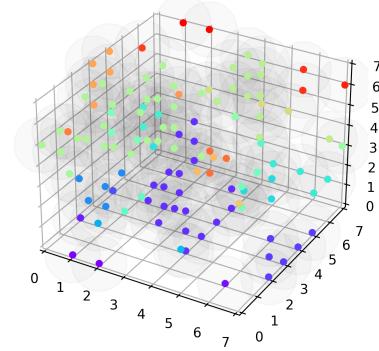


Figure 18: Percolation clusters formed in a three-dimensional square lattice.

7. Conclusions

In this project we have developed a set of codes for studying site percolation in networks. The main conclusions reached are:

- For both the square lattice and the triangular lattice, we have studied the percolation threshold and how it varies with the lattice size. In both cases we have found p_c to converge in the $L \rightarrow \infty$ limit. The limiting values we have calculated are $p_c = 0.588 \pm 0.007$ for the square and $p_c = 0.50 \pm 0.06$ for the triangular.
- We have determined how the average cluster size evolves as the probability changes. We have used this result to corroborate the phase transition that the system experiences at the percolation probability, manifested in a drastic fall of the average cluster size immediately after the critical probability.
- We have shown that the cluster size distribution follows a power law at $p = p_c$ and we have calculated the fractal dimension of the system by relating the mass of the clusters with their scaling factor; obtaining $d_f(\text{sqr}) = 1.521 \pm 0.009$ and $d_f(\text{tri}) = 1.45 \pm 0.02$.
- For the randomly distributed “forest” with two distinct distance thresholds we obtained a high dependence of the behaviour of the system with the particular arrangement of nodes. Fixing the seed allows us to analyze the effect that the tree ratio has on the percolation probability by studying the average cluster size plot for various small/big tree ratios.
- We give an overview of other applications of percolation theory, such as dynamical percolation to study astrophysical phenomena. We have also shown an example of the generalizations for our model with 3D percolation.

8. References

- [1] Tom Hutchcroft. Mathematically percolating. <https://www.caltech.edu/about/news/mathematically-percolating>, Aug 2022.
- [2] Simon R Broadbent and John M Hammersley. Percolation processes: I. crystals and mazes. In *Mathematical proceedings of the Cambridge philosophical society*, volume 53, pages 629–641. Cambridge University Press, 1957.
- [3] Armin Bunde, Paul Heitjans, Sylvio Indris, Jan Kantelhardt, and Markus Ulrich. Anomalous transport and diffusion in percolation systems. 06 2007.
- [4] Dietrich Stauffer and Amnon Aharony. *Introduction to Percolation Theory*. Taylor and Francis, 2003.
- [5] Vincent Beffara and Vladas Sidoravicius. Percolation theory. 2005.
- [6] Harry Kesten. What is... percolation? *Notices of the American Mathematical Society*, 53(5), 2006.
- [7] Dpto. Electromagnetismo y Física de la Matemática Universidad de Granada. Fisica computacional / lecciones. https://ergodic.ugr.es/cphys/index.php?id=lec_percolacion.
- [8] Ahmed Al-Futaisi and Tadeusz W Patzek. Extension of hoshen-kopelman algorithm to non-lattice environments. *Physica A: Statistical Mechanics and its Applications*, 321(3):665–678, 2003.
- [9] Jesper Lykke Jacobsen. High-precision percolation thresholds and potts-model critical manifolds from graph polynomials. *Journal of Physics A*, 43, 1 2014.
- [10] Naeem Jan. Large lattice random site percolation. *Physica A: Statistical Mechanics and its Applications*, 266(1):72–75, 1999.
- [11] Richard F Voss. The fractal dimension of percolation cluster hulls. *J. Phys. A: Math. Gen*, 17:373–377, 1984.
- [12] J.A. Lutz, J.W. van Wagtendonk, and J.F. Franklin. Twentieth-century decline of large-diameter trees in yosemite national park, california, usa. *Forest Ecology and Management*, 257(11):2296–2307, 2009.
- [13] James Lutz, Tucker Furniss, Sara Germain, Kendall Becker, Erika Blomdahl, Sean Jeronimo, C. Cansler, James Freund, Mark Swanson, and Andrew Larson. Shrub communities, spatial patterns, and shrub-mediated tree mortality following reintroduced fire in Yosemite National Park, California, USA. 13, 01 2017.
- [14] J. Einasto, I. Suhhonenko, L. J. Liivamägi, and M. Einasto. Extended percolation analysis of the cosmic web. *Astronomy and Astrophysics*, 616, 8 2018.
- [15] Anatoly Klypin and Sergei F. Shandarin. Percolation technique for galaxy clustering. *apj*, 413:48, August 1993.
- [16] Lawrence S Schulman and Philip E Seiden. Percolation and galaxies.