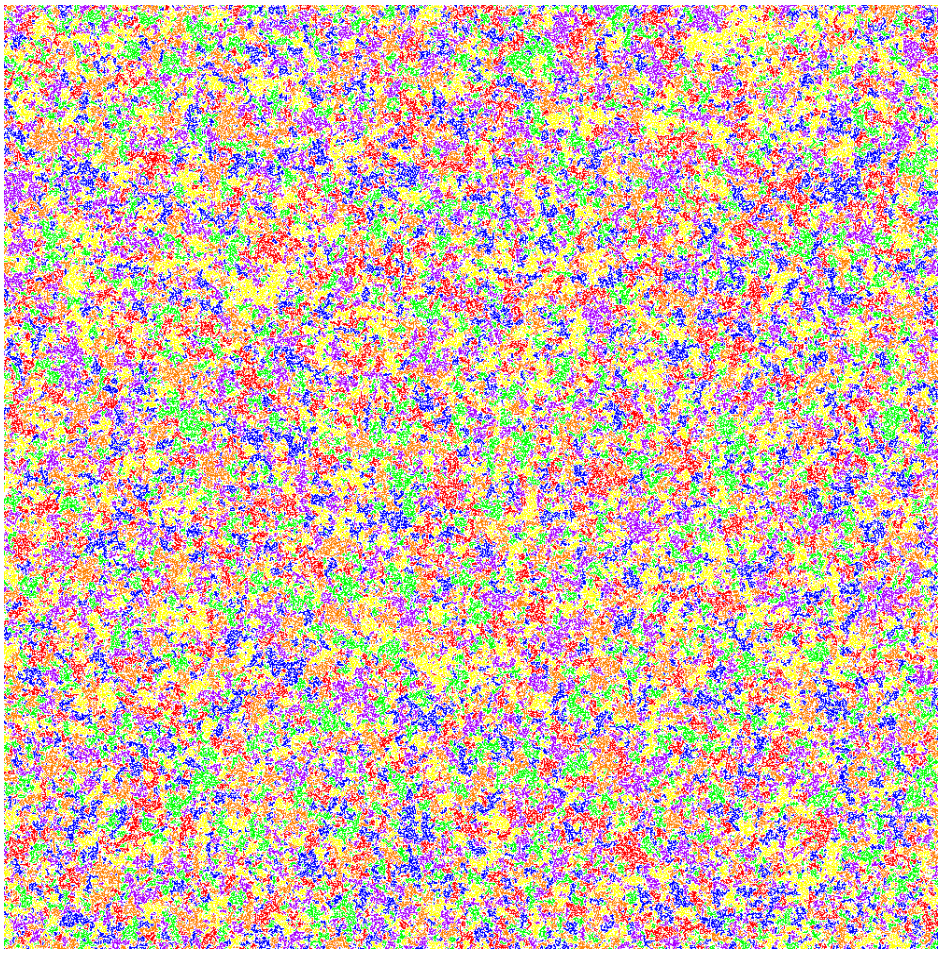
Newcastle University

Percolation Theory: The Influence of Obstacles

School of Mathematics & Statistics

MAS8091

2014/15



Author:
Emma Jones

Supervisor:
Dr. Nick Parker

Abstract

Percolation theory provides the foundation for further understanding of random networks and complex systems, which are the basis of many applications. In this report, the concept of percolation theory is introduced, exploring the fundamental ideas that are needed to understand and create the system. The computational cost of such methods will be discussed, highlighting the reason for the chosen characteristics used within the simulations. The direction is then shifted to explore the conventional obstacle-free 2D systems, verifying the results against known results for site percolation. Finally, obstacles will be introduced into the system, outlining their applicability to real-life applications and the implications observed on percolation behaviour. To our knowledge, these quantitative results have not been considered within a uniform 2D system before. It is shown that the presence of obstacles suppresses network formation and significantly modifies the percolation threshold across the system.

Acknowledgements

I would like to thank my supervisor Dr Nick Parker for introducing me to the topic of percolation theory and for providing encouragement and support throughout the project. I would also like to thank my friends and family for their help and support throughout the year, particularly when questioning computational and LaTeX errors. Finally, I would like to thank Dr Michael Beaty for granting me access to the Banach server, along with spending some time explaining how to access it to me. Without this, the computation time for some of my results would have been much longer.

Contents

1 Introduction **2**

2 What is Percolation Theory? **4**
2.1 Lattices and Occupation 4
2.2 Connectedness and Clusters 4
2.3 Site, Bond and Site-Bond Percolation 5
2.4 An Example Percolating System 5
2.5 Visualising Individual Clusters 6
2.6 Visualising the Biggest Cluster 7
2.7 The Spanning Cluster 7
2.8 Percolation Threshold 9

3 Method for Generating the System **11**
3.1 Numerical Method 11
3.2 Labelling the Clusters 11
3.2.1 The *bwconncomp* Function 11
3.2.2 The Computational Cost of Increasing the System Size 14

4 Percolation in the Absence of Obstacles **17**
4.1 Percolation Threshold 17
4.1.1 Determining p_c 18
4.1.2 Effect of Changing the System Size on p_c 20
4.2 Universal Scaling Properties 21
4.2.1 Average Cluster Size 21
4.2.2 Fraction of Sites in the Spanning Cluster 24
4.2.3 Correlation Length and Radius of Gyration 26
4.2.4 Fractal Dimension 29

5 Percolation in the Presence of Obstacles **33**
5.1 Introducing Obstacles into the System 33
5.2 The Influence of Obstacles on p_c 35

6 Conclusion **39**

Bibliography **40**

Chapter 1

Introduction

Within mathematics, random networks are described as random graphs which are generated by a probability distribution or random process. Such random networks are used all over the world in a variety of applications some of which include airline routes, the Internet and power grids. These examples, along with many other applications, fundamentally rely on the relationships of the connections between each of the components of the system. Some exemplars of random networks are displayed in Figure 1.1. By deleting a single component from the system, it can be detrimental to it's functionality. This can be exhibited, in connection to the earlier examples, through Internet routers or power line failures. These random networks can be modelling using *percolation theory* [1].

Percolation theory is commonly used to model disordered systems to provide a clearer representation of the processes that are occurring. Historically, the first mention of percolation theory dates back to World War II when Flory and Stockmayer (see [2] and [3] for review) utilised it in order to describe how macromolecules are formed through chemical bonds forming between smaller molecules. This process provides the foundation to gelation which leads to chemical bonds forming that span across the whole system. A common example of this is jelly, in which the gelatin protein fibres intertwine and coil together causing solidification to occur. Despite the earlier mention, the first application of mathematics to percolation theory is associated with the 1957 publication, '*Percolation processes I. Crystals and mazes*', written by Broadbent and Hammersley [4]. Broadbent and Hammersley developed this model for a specific example, namely liquid flowing through porous rock with the percolating network corresponding to the open pores in the

(a) (b) (c)

Figure 1.1: Examples of applications of percolation theory. Figure (a) shows the network of international airline routes [5], (b) shows the result of the solidification of jelly [6] and (c) shows a porous rock in which non-porous strata may be present [7].

rock. Through this, the name percolation became attached to the theory. Nowadays, percolation theory has proven to be theoretically important across areas such as physics, biology and geophysics through the use and development of fractal and scaling properties.

This a popular area of research in which many results are still unknown [8].

In Chapter 2, the concept of percolation theory will be introduced, exploring the fundamental ideas that are needed to understand and create the system. The discussion will then move on to access the computational cost of creating the systems and how the chosen parameters used in calculating the main results were decided.

This will then advance to explore some of the main results of percolation theory, including the percolation threshold and the universal scaling properties. These will help to develop an understanding of the behaviour of the system and confirm that the results produced from this method of simulation are in-line with known results.

The standard results of percolation theory are developed on uniform and homogeneous grids. As such, these networks do not provide a realistic insight into naturally occurring networks which are irregular and inhomogeneous. With this in mind, in Chapter 5, obstacles (that is, localised regions which do not contribute to the network of interest) will be introduced into the uniform 2D percolation system. To our knowledge, this scenario has not been previously considered. Such obstacles may represent, for example, pieces of fruit in jelly, oceans in land-based communication networks or non-porous strata enclosed in otherwise porous rock. The effect these obstacles have on the percolation properties, mainly the percolation threshold, will then be discussed. This is done with a view to develop the understanding of general defects in networks and provide a more realistic approach to percolation theory within applications.

Chapter 2

What is Percolation Theory?

In the introduction, the origins and ideas of percolation theory were discussed. With a variety of physical and theoretical applications, percolation theory provides an important and practical model of random networks. Considering the practical side of these models, it is essential to address the questions what is percolation theory and how does it work? In order to answer these questions, how the percolating system is set-up and its key mathematical properties will be introduced.

2.1 Lattices and Occupation

To begin with, consider an empty lattice of grid points. This lattice can be of any shape or size. For example, standard lattice shapes used within the field include triangular, honeycomb, square (2D) and cubic (3D) lattices, all of which come in a variety of discretised sizes from two points in each dimension to an effectively infinite extent. It is a notable feature of percolation theory that the qualitative behaviour exhibited by differently shaped and sized systems are the same, with only the quantitative results changing. Once the shape and size of the lattice has been chosen, the next step is to randomly assign a number to each of the sites, ensuring that each site is treated independently. After each site has been assigned it's own random number, an occupational (threshold) probability p, with $0 \leq p \leq 1$, is set to identify which sites are occupied within the system. Through this, it can be said that each site is occupied with probability p and unoccupied with probability $1 - p$. Therefore, the level of site occupancy depends on the value of p, with more sites being occupied for increasing p and less for decreasing p. By introducing this occupational probability, the system obtains a binary representation, where 1 represents an occupied site and 0 represents an unoccupied site. This forms the basis of the system from which various properties can be analysed.

2.2 Connectedness and Clusters

The first and foundational analysis that can be made is observing the connectivity between the occupied sites. A group of two or more connected sites is called a *cluster*. By identifying the clusters within the system, a foundation is formed for an array of analysis and properties which will be explored in Chapter 4. The idea of connectivity, as described in the definition of a cluster, can vary from system to system. For 2D, which is the main focus of this report, there are two different types of connectivity, namely 4-connectedness and 8-connectedness. The former, otherwise known as nearest neighbour, implies that it is only the adjacent sites that are connected to each other, whereas the latter, otherwise known as next-nearest neighbour, includes diagonal sites in the analysis and implies that

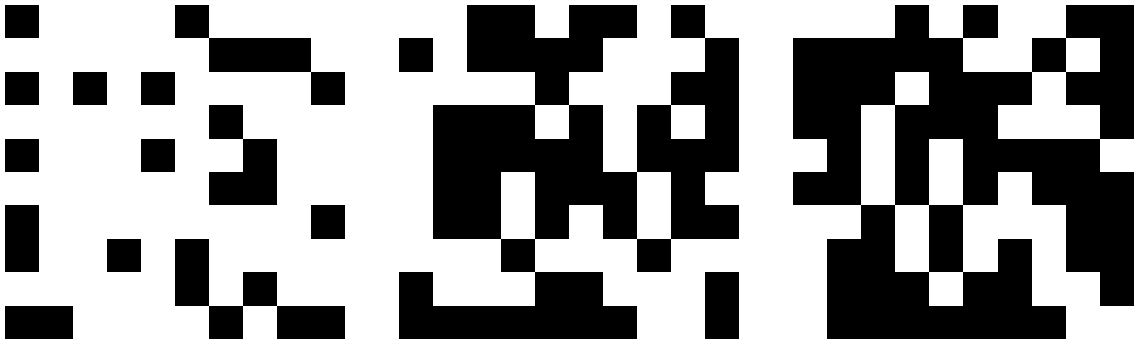
all adjacent and diagonal sites are connected to each other. The two types of connectivity generate different quantitative results but not different qualitative results. For this purpose, it is key to highlight that 4-connectedness will be used throughout this report.

2.3 Site, Bond and Site-Bond Percolation

Another factor that should be taken into consideration when creating or modelling the system is the type of percolation used. Within percolation theory, there are three varieties of percolation that are commonly studied. These are site percolation, bond percolation and site-bond percolation. Site percolation describes when sites are occupied with probability p, and is the type of percolation used in the above discussion. Bond percolation describes the formation of a bond between two sites with probability p. Once again, the states of occupancy are occupied and unoccupied, where occupied signifies a bond between two sites and unoccupied signifies no bond has formed. By using the definitions of site and bond percolation, a relationship can be formed to create site-bond percolation. Site-bond percolation relies on two occupational probabilities, p and q, where p is the probability that a site is occupied and q is the probability that a bond has formed between two sites. It is said that two occupied sites inhabit the same cluster if there is a continuous path of connected occupied sites that also have occupied bonds between each of them. Generally, the state of the system is described using both of the parameters however, there are some exceptions to this. When $p = 1$, site-bond percolation collapses down to bond percolation and similarly, when $q = 1$, site-bond percolation collapses down to site percolation [9]. Once again, like connectivity, the various different types of percolation produce different quantitative result but don't affect the qualitative results. Due to this, it is important to highlight that site percolation will be used throughout.

2.4 An Example Percolating System

Now the foundations of the system have been established, a simple example will be considered to help visualise the process. Following the method, we start by taking a 10×10 square lattice. Assigning each site independently with a random number and setting an occupational (threshold) probability, p, a binary representation of the system can be formed. This can then be plotted using a black and white colour map to characterise the state of occupancy, with black representing an occupied site and white representing an unoccupied site. As mentioned earlier, changing the value of p impacts the number of sites that are occupied and, hence, the connectivity between the sites. In order to help visualise this, Figure 2.1 has been created and shows three different values of p, namely $p = 0.3, 0.5$ and 0.7. Considering $p = 0.3$ first, Figure 2.1(a) shows that there are a small number of sites occupied which typically exhibit two types of behaviour. The occupied sites either stand alone, isolated by the surrounding unoccupied sites or they form small clusters consisting of a small number of sites therefore highlighting that connectivity between sites is low at low values of p. Now as the probability starts to increase, more and more sites become occupied with the number of isolated sites decreasing and bigger clusters forming. This is demonstrated in Figure 2.1(b) which is a system at $p = 0.5$. Increasing the probability further to $p = 0.7$, Figure 2.1(c) shows that the system is mainly dominated by one big cluster, that percolates across the system, with smaller clusters surrounding it.

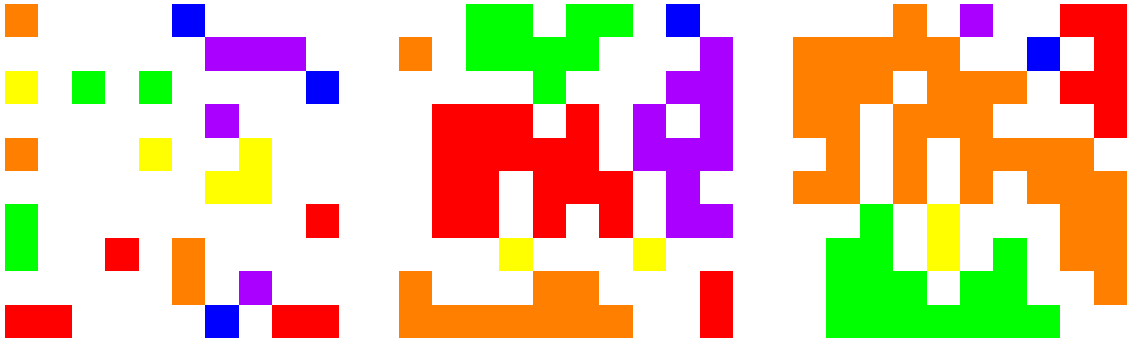


(a) (b) (c)

Figure 2.1: Black and white representation of a 2D percolating system for increasing probabilities, $p = 0.3$ (a), 0.5 (b) and 0.7 (c), where black represents the occupied sites and white represents the unoccupied site.

2.5 Visualising Individual Clusters

Considering a small system, like the 10×10 system shown in Figure 2.1, identifying the different clusters from these black and white images isn't too hard of a task. However as the system size starts to increase, for example to 100×100, $1,000 \times 1,000$ or $10,000 \times 10,000$, the ability to distinguish between the different clusters by eye diminishes, particularly for larger values of p. In an effort to make these clearer, the method of plotting was adapted to ensure that each of the clusters were plotted in a different colour. This not only aids the eye when identifying the clusters but accentuates the transformation of the clusters. Taking the example systems shown in Figure 2.1 and plotting them with the adapted method, the images shown in Figure 2.2 were produced.



(a) (b) (c)

Figure 2.2: Colour representation of the percolating system shown in Figure 2.1 for increasing probabilities, $p = 0.3$ (a), $p = 0.5$ (b) and $p = 0.7$ (c), where each of the clusters are represented by a different colour.

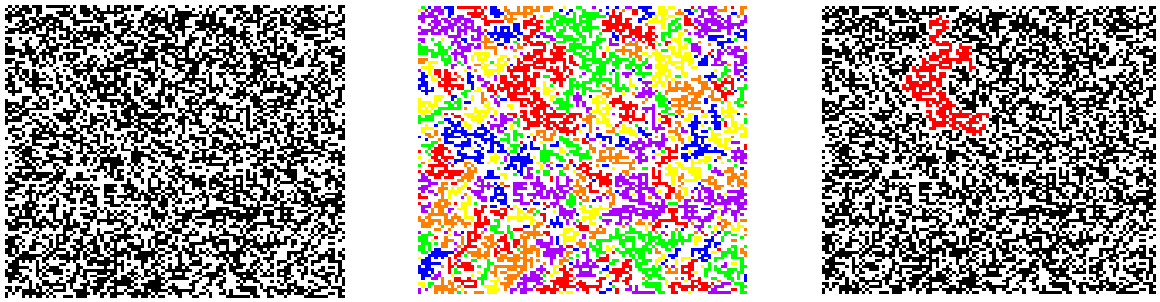
Through the ability to distinguish between the different clusters, this further supports the characteristics of the clusters that were discussed earlier. Looking at the system for $p = 0.5$ and $p = 0.7$, see Figure 2.2(b) and Figure 2.2(c) respectively, there is a drastic change occurring from which some of the larger clusters, present at $p = 0.5$, connect together and begin to percolate across the whole of the system. This is shown by the

orange cluster in Figure 2.2(c) and is a key characteristic that will be explored in more detail later.

These examples serve to demonstrate the role of the clusters in the system as p is changed. This is the foundation of percolation theory. Formally, percolation theory describes the numbers and properties of these clusters. These properties provide an insight into a variety of applications including the prevention of forest fires and oil extraction. There are a number of key properties that are studied in relation to percolation theory. These include the percolation threshold, the average cluster size and correlation length, all of which, along with others, will be addressed in more detail in Chapter 4. The concepts will be introduced and the effect of changing p on the properties will be discussed.

2.6 Visualising the Biggest Cluster

In many instances, percolation theory is concerned with the biggest cluster as the cluster tends to dominate network and exposes the largest connected network in the system. A particularly important feature within percolation theory is when the biggest cluster spans the system, which will be discussed shortly. With this in mind, it is insightful to identify and highlight the biggest cluster. An example of this is shown in Figure 2.3(c) for a 100×100 system at $p = 0.5$. The same network is presented in three different visualisation methods, namely black and white (a), all the clusters coloured (b) and only the biggest cluster coloured (c). Here, it can immediately be seen that, for $p = 0.5$, the biggest cluster is relatively small in comparison to the system.



(a) (b) (c)

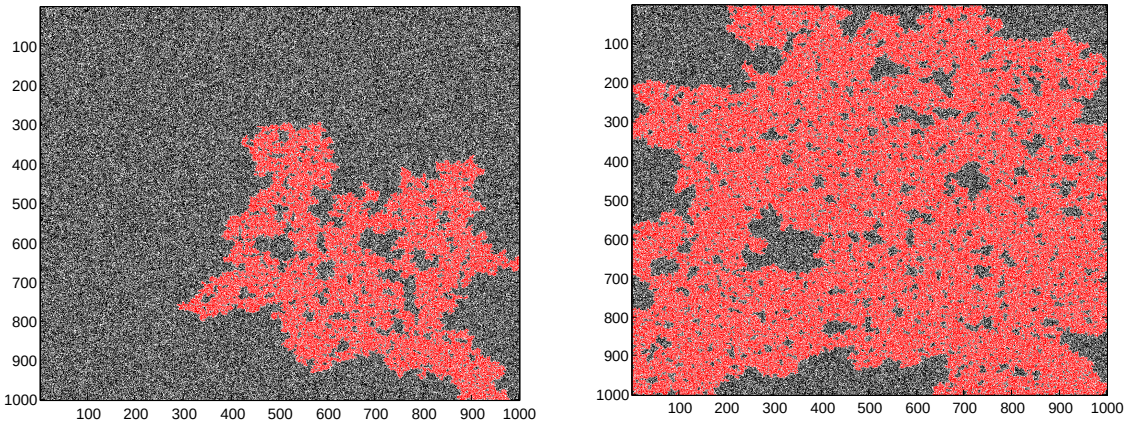
Figure 2.3: Progression through the different visualisation methods for a 100×100 system at $p = 0.5$. Figure (a) shows the occupied sites in black and unoccupied sites in white, (b) represents each of the clusters with a different colour and (c) highlights the all the sites in the biggest cluster in red, the remaining occupied sites in black and the unoccupied sites in white.

2.7 The Spanning Cluster

As described earlier, for small values of p the clusters are small and too isolated to percolate across the system. It is also true that for large values of p, the biggest cluster spans across the whole of system. Therefore, for some intermediate value of p, a transformation between isolated clusters to a cluster which spans the system is expected. Figure 2.3

shows that $p = 0.5$ is too small a value for the biggest cluster to span the whole system. Increasing p from this point, the biggest cluster grows in size.

For $p = 0.59$ (Figure 2.4(a)), we can see that the biggest cluster is beginning to stretch across the system adopting a sparse structure. Comparing this with $p = 0.6$ (Figure 2.4(b)), the biggest cluster has undergone a dramatic change, becoming much more prevalent and spanning across the whole of the system with a tenuous structure. From this point onwards as p increases, the density of the biggest cluster will increase with the remaining occupied sites becoming part of it. Consequently the size of the remaining finite clusters decreases [9]. This characteristic biggest cluster is called a *spanning cluster*, which can formally be defined as a cluster that spans from one side of the system to the other. Taking a 2D system as an example, a spanning cluster occurs when the cluster spans from left to right, top to bottom or both. The spanning cluster, otherwise known as the *infinite cluster*, is also said to be infinite due to it spanning the whole of the system. This is particularly demonstrated when the size of the system tends towards infinity.



(a) $p = 0.59$ (b) $p = 0.60$

Figure 2.4: The transition of the spanning cluster at the percolation threshold, p_c. Figure (a) shows the biggest cluster close to, but before, p_c and (b) shows the transformation of the biggest cluster into a spanning cluster just after p_c has been reached.

Besides the extraction of information about the system, the spanning cluster has a widely relevant importance throughout the applications of percolation theory. In physical terms, the spanning cluster represents a phase transition, which are frequently observed throughout nature. An example of this is shown in the solidification of jelly. The spanning cluster would be the first point at which the jelly would start to show any solid-like properties i.e. it is the first point at which the gelatin protein fibres begin to interlace and recoil causing solidification to occur. Some other examples include the transition between insulators and conductors, along with modelling the spread of forest fires to produce prevention methods. The importance of the spanning cluster is further supported by the fractal properties it exhibits close to it's formation. The presence of 'cracks' and 'holes' within in the cluster demonstrates the characteristics of fractal behaviour, however it is necessary to show the existence of these on all length scales. This will be explored, along with the fractal dimension of the spanning cluster, in more detail in Section 4.2.4.

2.8 Percolation Threshold

Having established the importance of the spanning cluster, we now want to find when it first occurs. For each type of system, there exists a critical occupation (threshold) probability at which this transition takes place. The spanning cluster is said to first appear at this critical occupation probability, which is more commonly known as the *percolation threshold* and is denoted by p_c. A formal definition of the percolation threshold is as follows: the percolation threshold, p_c, is the occupational probability at which an infinite cluster (i.e. spanning cluster) first appears in an infinite system. It is important to note here that this describes a system of infinite size. Realistically, it is not possible to simulate an infinite system however the same behaviour is still applicable with the additional coupling of statistical uncertainty on each iteration.

The quantitative value of p_c depends on two main factors, the shape and the dimension of the system. Recall that the underlying lattice can arise in variety of shapes, including square (2D), cubic (3D), honeycomb and triangular lattices. The shape of the system has a significant effect on the percolation threshold, as connectivity of occupied sites heavily depends on the occupation probability p and the number of connections available. Through altering the shape of the lattice, and consequently the system, the connectivity changes. For example, using a 2D square lattice, there are 4 neighbours available to make connections with. However, when you consider a triangular lattice, there are 6 neighbours available to make connections with. By increasing the number of connections that can be made, the connectivity has been altered, hence effecting the percolation threshold. The other contributing factor is the dimension of the system. Altering the dimension of the system, once again, has an effect on the connectivity between sites, mainly through the number of sites available to form a cluster with. The weighting of the influence these two factors have on p_c is much higher for the dimension than the lattice type. In order to demonstrate this, we will consider each in more detail. First, let the dimension, d, be fixed. Considering the number of connections available, z, in each of the lattices alone, the results for p_c can be observed in Table 2.1. As z increases, the value of p_c decreases. Comparing this with a fixed z and varying dimension, d, we can see that increasing the dimension of the lattice causes a significant decrease in p_c, despite having the same level of connectivity available. The decrease observed here is more significant than any seen when changing the value of z. This supports the heavier dependence of p_c on the dimension. It is necessary to note that p_c also depends on the type of percolation being used (site, bond or site-bond). However, due to site percolation solely being used throughout this report, this dependence will not be explored in detail here.

Type of Lattice	z	d	p_c
Honeycomb	3	2	0.6962
Square	4	2	0.592746
Triangular	6	2	0.500000
Cubic	6	3	0.3116

Table 2.1: The value of the percolation threshold, p_c, for various different lattices, highlighting their connectivity, z, and dimensions, d [8].

As demonstrated by the results shown in the above table, there are a variety of known results for p_c. These, however, are not all exact results. It took approximately two decades for the first numerical estimation of square bond percolation to be found in 1960.

Since then, exact known results have been found for triangular site, triangular bond and honeycomb bond percolation. This implies that the known results for honeycomb and square site percolation, referred to in Figure 2.1, are not exact. However, they are correct to a certain degree, mainly showing fluctuations in the final decimal places shown above. This problem is also extended to percolation among higher dimension systems. With this in mind, for the systems that we have been working with, namely finite 2D square lattices, the reference value of p_c that will be used is the close approximation of $p_c = 0.592746$ [8].

The occurrence of the critical occupation probability, p_c, in percolation theory lends it to be a mathematically challenging and extensive subject. As p_c is approached from both sides, the system exhibits inherently different behaviour. Through this, it provides a basis for complex physical systems that undergo a phase transition through the alteration of a particular parameter. An example of this is the critical concentration of gelatin required for jelly to set (at a fixed temperature). Percolation allows a simple model to be produced, making it more mathematically accessible whilst preserving features of the more complex systems [10]. By modelling systems through the use of percolation theory, we will go on to explore a variety of properties that are characteristic to any system in Chapter 4. The foundational knowledge will then be used to evaluate the effect of introducing obstacles into the system in Chapter 5.

Chapter 3

Method for Generating the System

Chapter 2 introduced the concept of percolation theory and how the systems are created. In this chapter, the generation of such systems will be explored, focusing on the method, functions and algorithms used along with the computational cost of altering the system size.

3.1 Numerical Method

Our main focus is site percolation on a two-dimensional square lattice. The percolating systems were generated using Matlab and were based on the following protocol:

1. A matrix, representing the square lattice, is populated with random numbers. Generally, the systems used throughout this report are based on $1,000 \times 1,000$ lattices, however systems as small as 10×10 and as large as $10,000 \times 10,000$ have also been used.

2. An occupation threshold probability p is defined in the range $0 \leq p \leq 1$.

3. Using p, the state of occupancy of the sites can be determined. Sites with values up to p are said to be occupied and are re-assigned with a value of 1; otherwise the sites are said to be unoccupied and are reassigned with a value of 0.

4. Based on 4-connectedness, the clusters are identified. This is the most challenging and time-consuming step of the procedure, and is performed using the in-built Matlab function *bwconncomp*, which will be discussed below.

5. Various quantities of interest are extracted from the cluster data to gain an insight into the properties of the system, e.g. size distribution and correlation length.

6. Steps 1 - 5 are repeated for different random realisations, as required, in order to gain statistical information. Typically, throughout this report, 1,000 iterations have been performed.

3.2 Labelling the Clusters

3.2.1 The *bwconncomp* Function

As discussed earlier, identifying the different clusters is a key part of the analysing the systems' behaviour, allowing further analysis of a variety of properties to take place. In order to distinguish between different clusters, a cluster multiple labelling algorithm

can be used. Within Matlab this has been encapsulated by the in-built *bwconncomp* function. The functionality of *bwconncomp* requires two main pieces of information, the object/image to be analysed and the type of connectivity used, which in this case is 4-connectedness (discussed further in [11]). The object/image to be analysed needs to be in the form of a binary matrix. This requirement is immediately fulfilled here through the implementation of the occupation probability, p, which identifies which sites are occupied and unoccupied. However this is not always the case. Through this, a binary matrix is produced with the occupied sites being represented by a 1 and unoccupied represented by a 0. Having established this, the cluster multiple labelling algorithm can be performed.

The *bwconncomp* function performs the following steps to find the connected sites within the binary matrix:

1. Search for the next unlabelled site, i.
2. Perform a flood-fill algortihm to label all the sites connected to site i.
3. Repeat steps 1 and 2 until all sites are labelled.

The algorithm then produces a structure which contains details on the type of connectivity used, the size of the image that has been analysed, the number of clusters found within the object (i.e. system) and the size of each cluster. Using this information, a new matrix can be assigned to be the matrix containing labelled clusters and a new variable can detail the cluster sizes. The output of this is a symbolic image showing each cluster labelled with a unique integer number. Through this, all of the sites contained within this cluster have the same unique integer label. By utilising this method, various measurements can be performed on the labelled cluster information [11].

The algorithm is performed column by column and starts the sweep from the top left-most column. There are two prelimanary variables that are created to store the number of clusters in the system and the size of each cluster found. Both of these are displayed in the structure that is outputted by *bwconncomp*. These variables are two one-dimensional arrays, denoted here by $\mathbf{C}_s$ and $\mathbf{C}_l$ which represent the cluster sizes and the cluster numbers respectively. Using these two variables and the occupancy of sites, the methodology of the algorithm is described in the flowchart shown in Figure 3.1.

$\mathbf{C}_s$ and $\mathbf{C}_l$ are an integral part of the labelling process but are used in fundamentally different ways. $\mathbf{C}_l$ is used to label each occupied site using the following method:

- If both sites to the north and west of the occupied site i are unoccupied, i.e. it has an entry of 0, then site i is given the current index of $\mathbf{C}_l$. This index is set to 1 at the beginning of the algorithm and is increased upon finding a new cluster.
- If the site to the north OR the west of the occupied site i are occupied, i.e. it doesn't have an entry of 0, the site i is assigned the label of the occupied site.
- If both sites to the north and west of the occupied site i are occupied, then site i and the site to north of site i are both assigned the label of the site to the west of site i.

Through the use of $\mathbf{C}_l$, the label j ($j = 1, 2, 3, \dots$) of each site can be found through satisfying $\mathbf{C}_l(j) = j$. The label for each site i that will be stored in $\mathbf{C}_l$ is identified using the following argument. First, let the original entry of site $(i-1)$ be j_1 and the label assigned to site $(i-1)$ be j_2. Then for site i, the following values can be assigned:

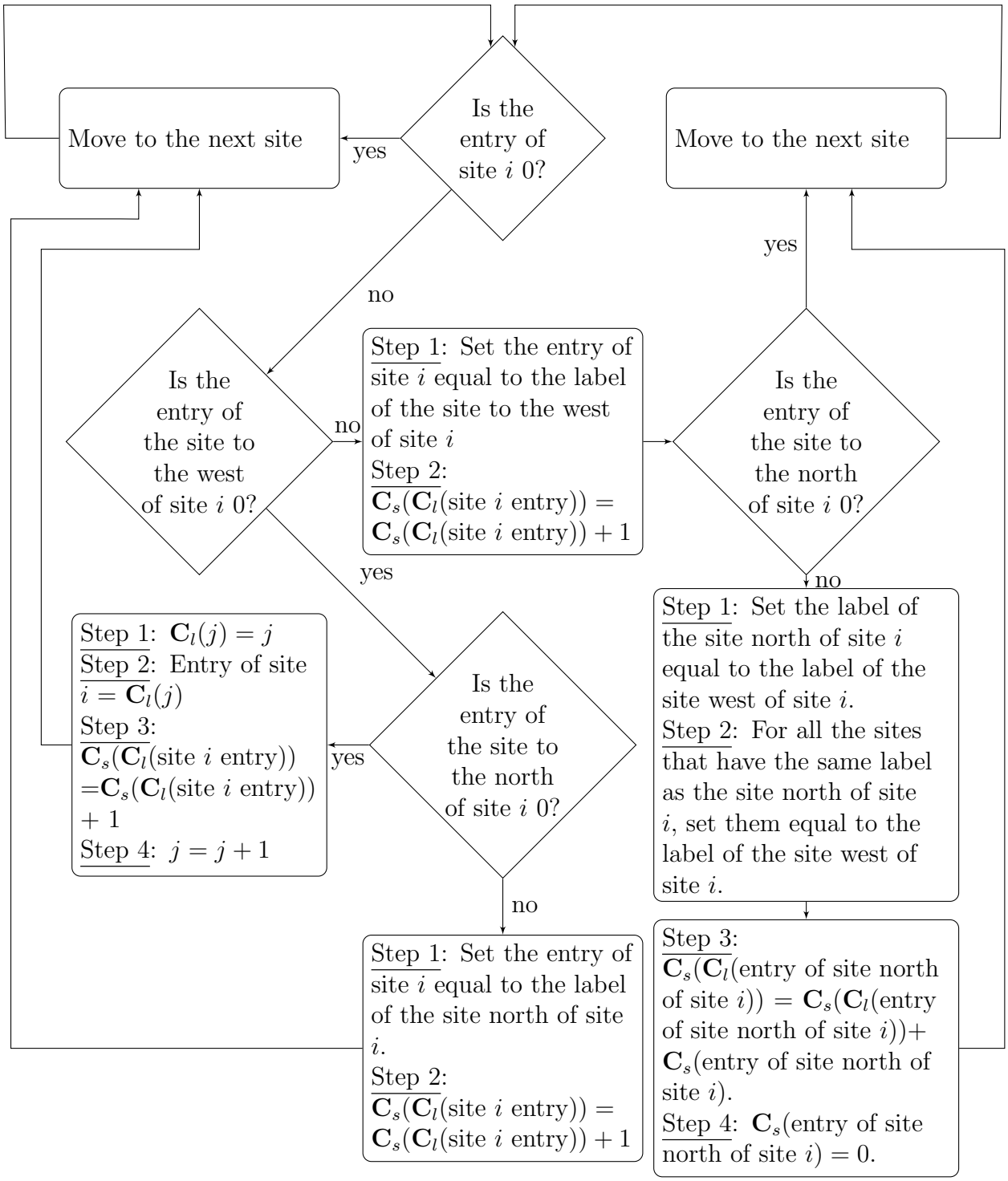


Figure 3.1: Flowchart describing the cluster labelling algorithm

$\mathbf{C}_l(j_1) = j_2$ and $\mathbf{C}_l(j_2) = j_2$. Hence, resulting in j_2 being stored as the label for site i. Having established the uses of $\mathbf{C}_l$, the uses of $\mathbf{C}_s$ will now be explored. $\mathbf{C}_s$ is used to store the different cluster sizes. In order to do this, whenever an occupied site is assigned a label, the corresponding entry of $\mathbf{C}_s$ is increased by one. In the case where both the sites to the north and west of site i are occupied but have different labels, it is necessary to merge the two clusters together and ensure all sites have identical labels. To do this, the size of the cluster that has the same label as the entry of the site to the north of i, is

added to the size of the cluster that has the same label as the site to the west of i. The size of the cluster corresponding to the site north of i is then zeroed out. Repeating this method allows the algorithm to perform a complete sweep. This provides an accurate labelling of all occupied sites and the size of all the clusters too. Due to this, only one sweep of the algorithm is necessary [12].

0	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0
1	1	0	1	0	0	0	0
0	0	0	0	1	1	0	0
0	0	0	0	0	1	1	0
0	1	0	0	1	1	1	1
0	1	1	0	0	0	0	1
0	0	0	0	0	1	0	1

(a) Binary matrix

0	0	3	0	5	0	8	0
1	0	3	0	5	0	8	0
1	1	0	4	0	0	0	0
0	0	0	0	6	6	0	0
0	0	0	0	0	6	6	0
0	2	0	0	6	6	6	6
0	2	2	0	0	0	0	6
0	0	0	0	0	7	0	6

(b) Labelled cluster matrix

Figure 3.2: The input and output of the cluster multiple labelling algorithm. Figure (a) shows the binary matrix that is used for the input while (b) shows the output with all the clusters distinguishably labelled.

Using the *bwconncomp* function creates an efficient way of labelling multiple clusters relatively quickly. An example of the labelled cluster output for $p = 0.5$ is shown in Figure 3.2. The output of the algorithm produces easily distinguishable clusters whilst providing data that can be analysed further in a variety of ways. This demonstrates the importance of a cluster multiple labelling algorithm (or connected component labelling technique) within many applications, be this in percolation theory or other areas of research. One of the main complications of using this method is the computational cost that accompanies it. We will now go on to explore how this has affected choosing the size of the system to analyse.

3.2.2 The Computational Cost of Increasing the System Size

Cluster multiple labelling techniques are computationally expensive to operate. The completion time of this algorithm is commonly found to be critical in the feasibility of a given algorithm. The algorithm, discussed in more detail in Section 3.2.1, works by assigning a label to an occupied site and attempting to propagate the label to the site's east and south neighbours [11]. The time taken to perform a full sweep varies dependent on the size of the system, denoted by L, number of clusters and the size of clusters found in the system. Note that the latter two directly depend on the value of p. To demonstrate the effect of changing these two factors (i.e. size and p), the time taken for the foundation code to run has been measured and recorded in Table 3.1. These measurements were obtained using the same machine, as the results are deterministic of the machine used. This provides a realistic representation of how the time scales as the system size, L, is increased. The foundation code consists of the following steps: assigning a matrix size, generating the matrix, and applying the occupation probability p, running the *bwconncomp* function, creating the labelled cluster matrix and extracting the cluster size information from the output. This is performed using the following set of commands:

```
mat = 10;
p = 0.5;
g = double(rand(mat,mat)>1-p);
connected_comp = bwconncomp(g, 4);
labelled_clusters = labelmatrix(connected_comp);
clusters = double(labelled_clusters);
cluster_size = cellfun(@numel, connected_comp.PixelIdxList);
```

These seven lines of code are the basis of all calculations performed and allow the results discussed in Chapter 4 to be obtained. It runs on every iteration performed, therefore if the completion time is lengthy then the whole calculation is destined to be so as well. Evaluating the time taken for the foundation code to run, shown in Table 3.1, we can see that for fixed p, increasing the system size by a factor of 10×10, the time taken increases. This increase starts off quite small at first, demonstrated by the increase from 10×10 to 100×100, however grows substantially in size with a higher order increases. This is shown from 100×100 to $1,000 \times 1,000$ to $10,000 \times 10,000$. With such a significant increase from $1,000 \times 1,000$ to $10,000 \times 10,000$, the process becomes more computationally expensive to operate.

System Size	Time (s)		
	$p = 0.3$	$p = 0.5$	$p = 0.7$
10^2	0.001533	0.001632	0.001526
100^2	0.006344	0.004610	0.002494
$1,000^2$	0.769116	0.485236	0.146629
$10,000^2$	117.224844	105.540672	53.526654

Table 3.1: Time take for one iteration of the base code to completely run for a variety of probabilities, namely $p = 0.3$, $p = 0.5$ and $p = 0.7$.

As all of the known results for percolation theory, for example p_c, are characteristic of an infinite system, it follows that as L is increased, the results become more reliable and in-line with the known values. As a result of this, one iteration of a $10,000 \times 10,000$ system produces more reliable results than one iteration of a 100×100 system. It should be noted, though, that averaging data over multiple iterations for a smaller system does provide data that is comparable to one iteration of a $10,000 \times 10,000$ system. Realistically, simulating an infinite system is not possible therefore simulating a finitely large system is the next best option. Due to the computational cost linked with performing multiple iterations of a $10,000 \times 10,000$ system, it proves to be much more time effective to perform multiple iterations of a $1,000 \times 1,000$ system instead. Through the use of multiple iterations, the statistical uncertainty has been reduced leading to more reliable results that are comparable to a $10,000 \times 10,000$ system. This shows that changing L produces different quantitative results but not different qualitative results when several iterations are carried out. As a result of this, many of the results discussed in Chapter 4 are calculated using 1,000 iterations on a $1,000 \times 1,000$ system giving an insight into the general trend with low statistical uncertainty. However, when needed, iterations on a $10,000 \times 10,000$ system are performed to ensure the behaviour is as expected.

Looking back at Table 3.1, the effect of changing p on the time taken can also be observed. Fixing the size of the system, it can be seen that there is a general decreasing

trend in time as p is increased. Therefore, for lower to middle values of p, the *bwconncomp* function takes longer to run. The reason for this is the abundance of isolated clusters during the progression through this region of probabilities. Comparing these results with $p = 0.7$, there is a considerable decrease in the time taken. This can be explained by the increase of p past the value of p_c, which results in more and more sites being added to the biggest cluster hence leaving less clusters to be analysed. The observation of this highlights that the time taken for one complete iteration is not uniformly spread across all probabilities. This highlights that some take longer than others, all of which create another factors into computation time.

Finding an optimum set of conditions for numerical work is desirable but presents a variety of factors that need to be considered. The choices for the size of the system have been explored with comments supporting the main choices that were implemented. Alongside this an overall view of the effect that different factors have on the system has been provided.

Chapter 4

Percolation in the Absence of Obstacles

In Chapter 2 and 3, the concept of percolation theory and the methods of simulation were introduced. In this Chapter, the focus will shift towards some of the key properties exhibited by the convention case of percolation in a uniform, obstacle-free system. Due to percolation being a random process, each simulation of a system will produce a unique set of clusters. Therefore, to explore the average properties of the system, it is necessary to analyse the statistics of the clusters [8]. The properties will be explored with an emphasis on understanding the behaviour as p is varied. First of all, the main result of the percolation threshold will be explored, establishing a method of calculation and ensuring that the known results are reproduced. Once this has been confirmed, the idea of universal scaling properties will be introduced and several behavioural properties that exhibit this will be considered. These include the average cluster size and fractal dimension of the spanning cluster. It should be noted that all results discussed have been calculated using a $1,000 \times 1,000$ square lattice based system.

4.1 Percolation Threshold

As mentioned in Chapter 2, the spanning cluster first appears at a value of p called the percolation threshold, which is denoted by p_c. Therefore for all $p > p_c$, there is a spanning cluster present in the system which increases in size as p increases. The system exhibits a variety of singular behaviour at p_c which is characterised through the use of critical exponents. These critical exponents depend on the dimensionality of the system [16]. These singular properties will be discussed in more detail in Section 4.2. In order to explore these properties, it is necessary to ensure that the value of p_c produced from our simulations are accurate and in line with the known results. This can be done through the analysis of the first occurrence of the spanning cluster.

Considering the biggest cluster for a variety of system sizes across 3 different probabilities, Figure 4.1 displays the results. Focusing on when $p = 0.3$ briefly, it can be seen that as the system size, L, is increased, the ability to identify the biggest cluster, becomes much more of a difficult task. This is due to the increasing number of isolated clusters present within the system. As L is increased the number of isolated clusters that can form also increases. Considering the case where $p = 0.5$, a similar type of behaviour is exhibited with the biggest cluster being less prominent as L is increased. Once again this is due to the increasing abundance of sites and connections to be made. For $p = 0.5$, the images present that the biggest cluster has not yet transitioned into a spanning cluster. With this in mind, $p = 0.7$ is considered. Here, the images no longer express a reduction in the prominence of the biggest cluster as L is increased. The biggest cluster, which has

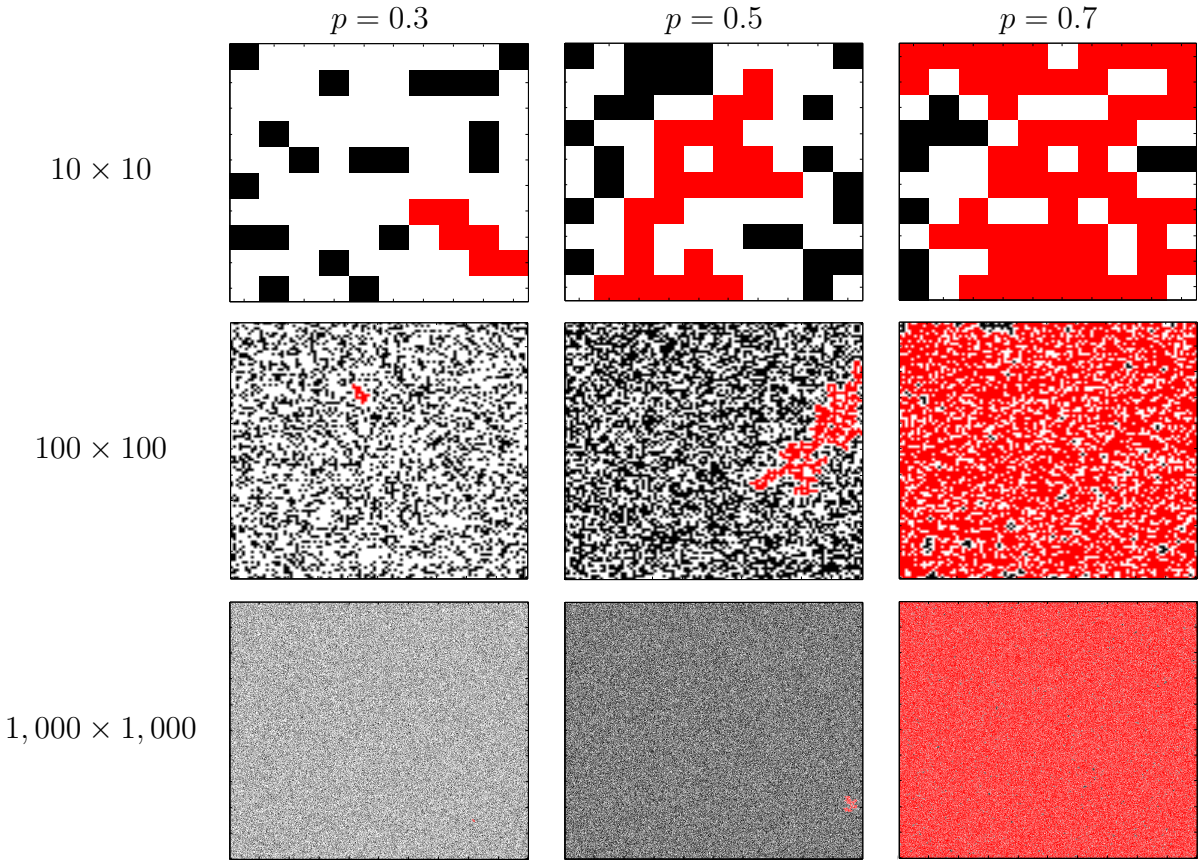


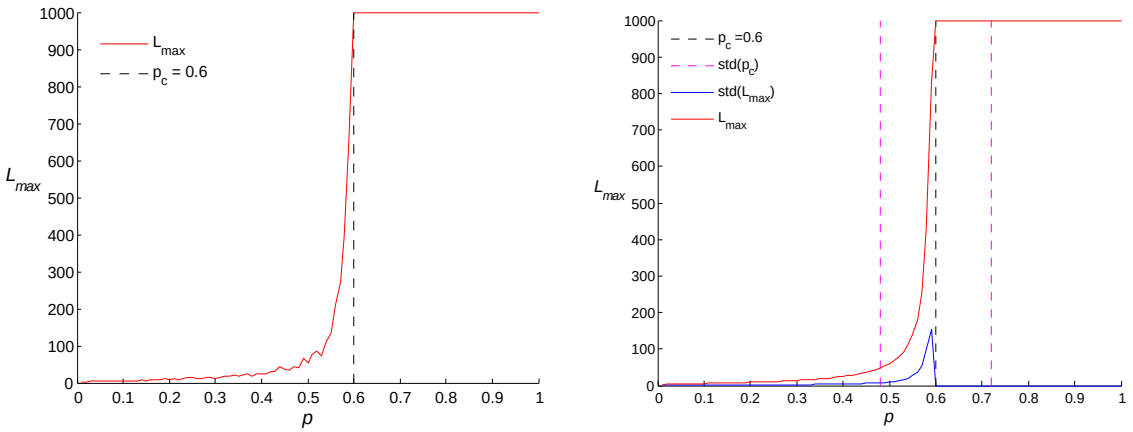
Figure 4.1: Comparison of the prominence of the spanning cluster for different system sizes at several values of p. The spanning cluster is highlighted in red, with the remaining occupied sites in black and unoccupied sites in white.

undergone a transition into a spanning cluster here, maintains the level of prominence throughout the alterations. This further supports the idea that the behaviour of the clusters is fundamentally different, dependent on which side of p_c the value of p sits on. Comparing the images for $p = 0.5$ and $p = 0.7$, the system undergoes a dramatic change within this region, resulting in a spanning cluster being formed. Examining this range of p in more detail, the value of p_c can be found.

4.1.1 Determining p_c

In order to determine p_c, it was found to be beneficial to introduce a new variable called L_{max}. This is defined as the maximum dimension (i.e. height or width) of the biggest cluster present in the system at the chosen probability. In order to understand how L_{max} changes, it was calculated at each increasing increment of p. The results were recorded for one iteration and for the averaged data of 1,000 iterations. The findings can be seen in Figure 4.2.

Looking at Figure 4.2(a), the data for one iteration displays a small amount of noise for $p < p_c$ with a general increasing trend. As p is increased, the gradual increasing trend evolves into a sharp increase close to the known value of p_c. It can then be seen that as soon as p_c is reached, L_{max} plateaus at the maximum dimension of the system (which in this case is 1,000). Plotting one iterations worth of data provides us with an insight into the characteristics of the variable, and hence the system. However only simulating



(a) (b)

Figure 4.2: L_{max} vs p for one iteration (a) and averaged over 1,000 iterations (b). Figure (a) shows L_{max} vs p (red line) highlighting the value of p_c (black dotted line). Figure (b) shows the averaged data for 100 iterations for L_{max} vs p (red line) along with the standard deviation of L_{max} (blue line), the value of p_c (black dotted line) and the standard deviation of p_c over all iterations (magenta dotted lines).

over one iteration gives way to statistical uncertainty. Therefore, to decrease this, 1,000 iterations were performed and an average of the data was produced. The results of this are displayed in Figure 4.2(b) and show a smoother curve that follows the same trend as displayed in Figure 4.2(a). To gain an insight into how much the behaviour varies from iteration to iteration, the standard deviation over the 1,000 iterations performed was calculated. This is represented by the blue line in Figure 4.2(b). For small values of p, far away from p_c, there is very little deviation in the data. However, as $p \to p_c^-$ (i.e. p approaches p_c from the left), there is a sudden increase in the deviation, reaching a maximum just before p_c is reached. This can be explained by the tenuous structure of the spanning cluster and the random generation of the system on each iteration. Both of these properties produce fluctuations in the data through changes in the number and size of the clusters produced. As discussed earlier in this section, the system exhibits fundamentally different characteristics on either side of p_c. Combining this with variations of when the spanning cluster will appear in each system, some deviation is inevitable. The deviation then diminishes quickly to zero as L_{max} plateaus. This is explained through the maximum dimension being reached with more and more sites becoming part of the spanning cluster from that point onwards. Figure 4.2 also displays the values of p_c produced from the simulations (see the black dotted line on both images). For both images, the value of $p_c = 0.6$, which is closest to the known value of p_c possible (with increasing increments of 0.01 for p). This indicates that the single iteration in Figure 4.2(a) had a good estimation of p_c, however the standard deviation of p_c (shown by the magenta dotted lines in Figure 4.2(b) demonstrates that this is not always the case. The calculated value of p_c varies by approximately 1.2 across all iterations. Despite these fluctuations, the average value is still consistent with the known result and therefore presents support towards a reliable method of simulation.

4.1.2 Effect of Changing the System Size on p_c

In Section 4.1.1, the value of p_c was determined for a $1,000 \times 1,000$ system through multiple simulations. After finding this is within a small range of the known value of p_c, it would be interesting to compare the estimations for different system sizes over 1,000 iterations. Simulating these different systems and averaging the data to obtain a value of p_c produces the results shown in Figure 4.3. This displays each of the found values for p_c along with error bars to indicate the standard deviation of the data.

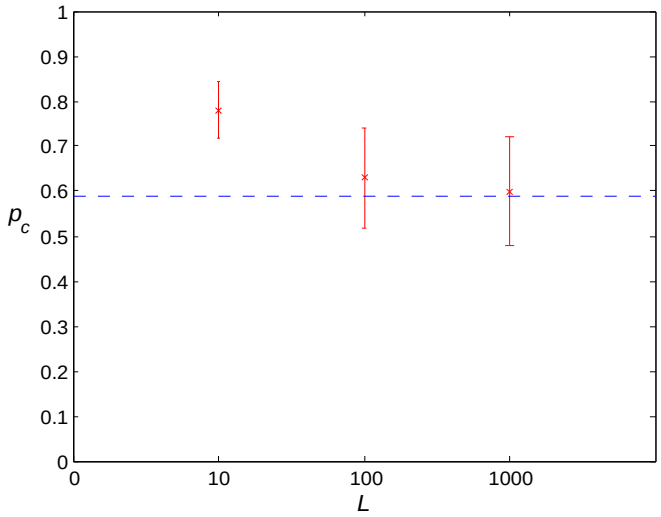


Figure 4.3: p_c vs different system sizes L. The results are shown with a red cross at the value and display the standard deviation in the form of error bars. The known value of $p_c = 0.592746$ is also shown (blue dotted line).

Comparing p_c for each system size, L, the approximation becomes closer to the known value of p_c as L is increased. This is in-line with the idea that p_c is, for an infinitely-sized system, producing more accurate results as L tends towards infinity. It may seem slightly contradictory to this though when observing the increase in standard deviation as L increases. This is due to the sparse nature of the spanning cluster and the fluctuations in behaviour from system to system (as it is a random process). With the tenuous and sparse structure of the spanning cluster, the value of p_c oscillates around the known value on each iteration. As there is more scope for this to occur in a larger system, the standard deviation of the data will increase, however the overall averaged data is more accurate. For a 10×10 system, the value of p_c is far away from the known value of p_c, however as L is increased, the results gain more accuracy tending towards $p_c = 0.592746$ quickly. This is shown through $L = 1,000$. Each of these results were produced using averaged data over 1,000 iterations. With this in mind, it is important to note that if the number of iterations were increased for the 10×10 system then the approximation for p_c would tend towards the known value for p_c. This further supports the ideas about changing the system size discussed in Section 3.2.2 and the justification of using a $1,000 \times 1,000$ system for the typical system.

4.2 Universal Scaling Properties

There are several properties that can be determined about the system, many of which exhibit singular behaviour at, or around, p_c. Many of these properties exhibit critical behaviour around p_c and are expressed using critical exponents. A critical exponent is expressed at a critical point or value, for which a quantity is found to be proportional to the power of another quantity that is characteristic of the system [13]. An example of a critical exponent is the power of the distance away from the percolation threshold, i.e. $|p - p_c|^\beta$, where β is the critical exponent. These critical exponents behave the same independent of which shape or size the system is. As a result of this, the critical exponents are said to be *universal* over the system shape and size. However their values alter if the Euclidean dimension, d, of the system is changed, e.g. shifting from 2D to 3D. From this universal behaviour, general scaling laws can be deduced. The aim of this section is to verify the universal behaviour for our model through looking at four key properties that are characteristic of system throughout percolation theory. These properties are the average cluster size, the fraction of sites in the spanning cluster, the correlation length and the fractal dimension of the clusters. The first property to be discussed is the average cluster size.

4.2.1 Average Cluster Size

Previously, the importance of the spanning cluster, and the information that can be extracted from it, have been discussed. Despite this still being the case, to gain an insight into the overall behaviour of the clusters, the average cluster size will be explored in this section. To begin this analysis, the following question is presented:

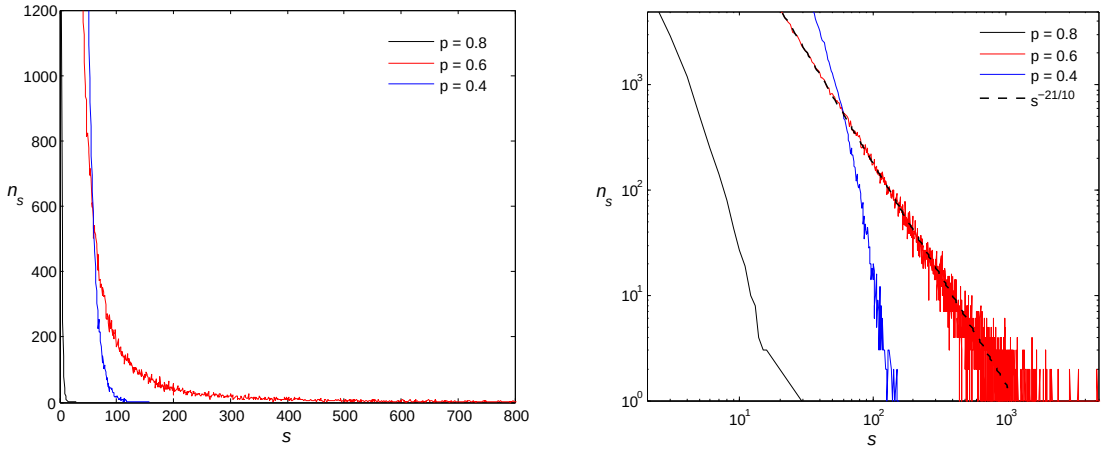
"*If a random occupied site in a system was chosen, on average, how big would the cluster that it is a part of be?*"

In order to answer this question, several variables and properties (or probabilities) need to be introduced [8].

To begin with, we consider the clusters of size s, otherwise known as *s-clusters*, and the number of s-clusters per site in the system, otherwise known as the *normalised cluster number*, denoted $n_s(p)$. Through defining the cluster number in this way, instead of considering the total number of s-clusters in the system, it provides an analysis that is independent of the system size L. Alternatively, this can be expressed through the probability that a given site in the system belongs to an s-cluster. This is given by $n_s s$. For the variable n_s, an exact form or equation has not been obtained for $d > 1$. This is due to the clusters being able to position themselves in a large variety of ways, unlike for $d = 1$ where the clusters are all formed along a straight line. This problem is apparent for all cluster sizes, with no alleviation for particularly small clusters [13]. To consider the behaviour of n_s, the results obtained for three probabilities, namely $p = 0.4, 0.6$ and 0.8, have been plotted in Figure 4.4(a). It is important to note here, that the biggest cluster has not been included in these calculations and the image is focused on a small section of the data to help gain understanding of the trend.

Looking at $p = 0.8$ (black line), there is a sharp and sudden decrease for small s with the data coming to an abrupt ending at $s \approx 30$. As $p = 0.8 > p_c$ and the biggest cluster having been excluded, i.e. the spanning cluster as $p > p_c$, the only remaining occupied sites create an abundant collection of small isolated clusters. Hence justifying the large

density of clusters at small s and presence of no large clusters. Considering $p = 0.4$ (blue line), there is less of a rapid decline over small values of s however similar behaviour to $p = 0.8$ is expressed. As $p = 0.4 < p_c$, there are many small clusters within the system, none of which dominate over the others in size. Although the clusters are slightly larger than for $p = 0.8$, the same behaviour is displayed. Finally considering $p = 0.6$, there is a much steadier decrease in n_s as s increases, with $n_s \to 0$ as $s \to \infty$. This is apparent due to the sparse nature of the spanning cluster and the prevalence of the remaining clusters around it. For large s at $p = 0.6$, there is a finite (but small) chance of finding a cluster.



(a) (b)

Figure 4.4: Linear plot (a) and loglog plot (b) of the number of s-clusters, n_s vs s for three probabilities, namely $p = 0.4$ (blue), 0.6 (red) and 0.8 (black). The loglog plot (b) displays the small scale fluctuations in behaviour and provides a power law behaviour comparison with $s^{-\frac{21}{10}}$ which has also been plotted (black dashed line).

Examining the decaying behaviour over the three probabilities, a loglog plot of the data was considered to identify if any power law behaviour was apparent. Figure 4.4(b) shows that the $p = 0.4$ (blue) and $p = 0.8$ (black) tail off quickly indicating that there is no significant change of finding a cluster at large s and hence no power law distribution. Focusing on $p = 0.6$, taking the noise (which is accentuated through using the logarithm function on the data) out of the consideration, the data looks to approach a straight line with a clear decreasing trend. Using this analysis, a power law distribution can be obtained. Fitting this to the graph, power law behaviour of $s^{-\frac{21}{10}}$ was found giving another insight into the behaviour of the clusters.

Having studied the characteristics of n_s for several probabilities, the variable can be used to introduce other variables. In furtherance to n_s, the probability that a particular site belongs to any finite cluster is given by the summation $\sum_s n_s s$, where it is a summation over all cluster sizes s. Combining the above findings together, the probability that a cluster of a given site in the system consists of exactly s sites is calculated by $w_s = \dfrac{n_s s}{\sum_s n_s s}$ i.e. the probability that a given site belongs to an s-cluster divided by the probability that a given site belongs to any finite cluster. Using these results, the average cluster size, denoted by S, can be measured using the following formula [8],

$$S = \sum w_s s = \sum \frac{n_s s^2}{\sum n_s s}.$$

Calculating the values of S for all p, allows the behaviour of S to be observed. The findings calculated for one iteration and averaged data over 1,000 iterations are shown in Figure 4.5(a) as the red dashed line and the black line respectively. From these values, it can be seen that the findings for one iteration are similar to the averaged data and gives a pretense that this may always be the case. However, this is not the case. The general behaviour shown in Figure 4.5(a), by the single iteration data, is consistent from iteration to iteration, however there are large fluctuations in S close to p_c. This implies that the smaller clusters will be averaged much more as a result of their frequent and consistent appearance. Evidence to support this further is demonstrated through the behaviour of n_s, which was discussed earlier. The dramatic fluctuations close to p_c are due to the sparse structure of the spanning cluster, which causes a greater effect on the averaging. To demonstrate the extremity in the fluctuations of S, the maximum value of S has been recorded over several iterations and the results are presented in Table 4.1. Using only five iterations, the variation can already be observed, showing the vast differences in values that can obtained and justifying the averaged maximum value of approximately 5.8 x 10^4.

Iteration number	Maximum value of S
1	56736
2	37539
3	77086
4	51616
5	94028

Table 4.1: The maximum value of S for five different iterations, to display the large fluctuations in the the data on each iteration

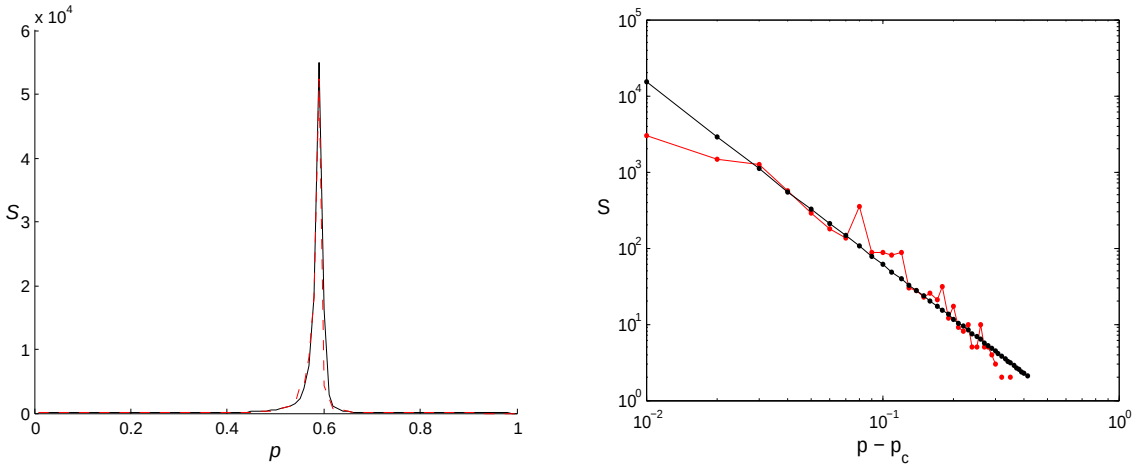
Focusing more on the behaviour of S, Figure 4.5(a) highlights how the average cluster size diverges as $p \to p_c$. This is reasonable behaviour which is supported by the presence of the infinite cluster, as p surpasses p_c, and large finite clusters, as $p \to p_c^-$. Therefore, an average over these two cases produces a very large finite cluster size just below p_c. The data shown in Figure 4.5(a) shows a distinguished sharp finite peak close to p_c. As the system size, L, is increased and tends towards infinity, this peak disappears, with the values of S tending towards infinity at p_c. Despite this, all other behaviour exhibited by the data shown, would be present as $L \to \infty$.

Having considered the characteristic behaviour of S over p, we want to determine the universal scaling properties for the divergence of S close to p_c. This can be characterised by the following proportionality [14],

$$S(p) \propto |p - p_c|^{-\gamma} \quad \text{for } p \to p_c, \tag{4.1}$$

where γ is the critical exponent and the known value for $\gamma = \frac{43}{18}$ for $d = 2$ [8].

In order to confirm that S exhibits this behaviour as $p \to p_c$, it is beneficial to use a loglog plot on the initial data. In accordance to Equation (4.1), the data corresponding to the absolute values of $p - p_c$ has been used. Figure 4.5(b) shows the applicable data from S vs p (red line) and the $|p-p_c|^{-\frac{43}{18}}$ (black line). It is useful to note that the first two data points for S (red line) can be disregarded due to their proximity to p_c. Due to the average that has been calculated when obtaining the data, and the fluctuations that occur around p_c, these are not true representations of the system's behaviour. Considering the data outside the 0.02 range of p_c, the critical exponent behaviour is apparent. The data



(a) (b)

Figure 4.5: Linear plot (a) and loglog plot (b) of the average cluster size S vs p. Figure (a) shows S vs p calculated for one iteration (red dashed line) and averaged data over 1,000 iterations (black solid line) and (b) displays the loglog plot of S vs p with the 1,000 iteration averaged average cluster size data (red line) and the critical exponent line $|p - p_c|^{-\frac{43}{18}}$ (black line) to show the similiarity.

S shows a close relationship with $|p - p_c|^{\frac{-43}{18}}$ from the outset with only small fluctuations, causing deviation away from this. These are made more apparent by the logarithmic function exaggerating the small scale noise. Through the relationship shown between S and $|p - p_c|^{-\frac{43}{18}}$, the proportionality shown in Equation (4.1) is supported for the method of simulation used.

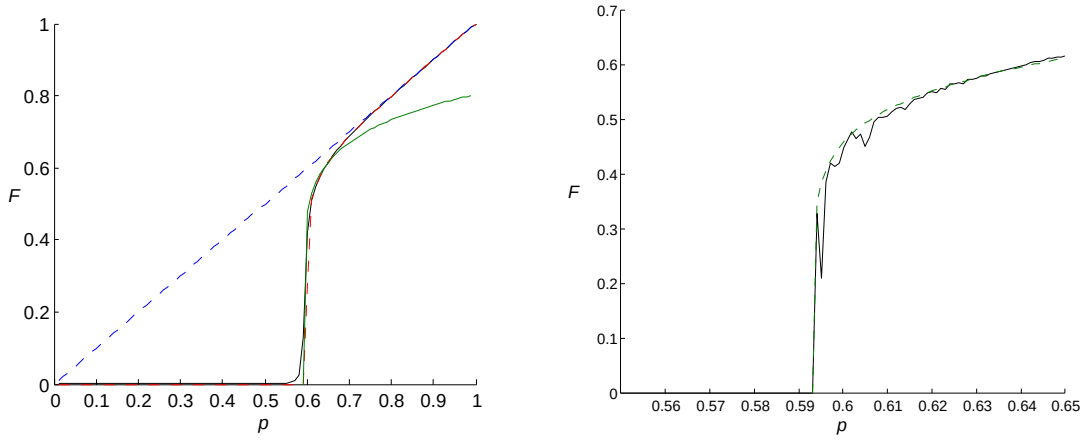
4.2.2 Fraction of Sites in the Spanning Cluster

In Section 4.2.1, the average cluster size was addressed. In this section, the focus will shift from all the clusters to the spanning cluster once again. The fraction of sites in the spanning cluster will be studied for increasing p, which will provide useful information about how the spanning cluster evolves after it first occurs. The fraction of sites in the spanning cluster, which is denoted by F, is defined by the following equation,

$$F(p) = \begin{cases} 0, & \text{for } p < p_c, \\ (p - p_c)^{\beta}, & \text{for } p \to p_c^+, \end{cases} \tag{4.2}$$

where β is the critical exponent and characterises the abrupt changes of the fraction of sites in the spanning cluster as p_c is approached. It is useful to note that the known value for β is $\beta = \frac{5}{36}$ for $d = 2$ [15].

By plotting the result for F vs p in Figure 4.6(a), for one iteration (red dashed line) and averaged over 1,000 iterations (black solid line), we can see that as $p \to p_c^-$, $F = 0$ until p_c is reached. As soon as $p = p_c$, F increases rapidly within 0.02 of p_c and then the rate of increase slows, evolving into a linear increase which then tends towards 1 as $p \to 1$. These characteristics are exhibited by both the averaged and single data which tells us that close to p_c, there can be a spanning cluster which contains a very small fraction of the total sites of the system. This implies that it has a tenuous structure and is incredibly sparse.



(a) (b)

Figure 4.6: Fraction of sites in the spanning cluster, F, vs p. Figure (a) shows the F for one iteration (red dashed line), F averaged over 1,000 iterations (black solid line), $F = p$ (blue dashed line) to highlight the linear behaviour as $p \to 1$ and $(p - p_c)^\beta$ (green solid line) to show the similiarity with the critical exponent behaviour. Figure (b) shows one iteration of F for a small region of p. This region is focused around p_c with increasing increments of 0.001 (black solid line). $(p - p_c)^\beta$ (green dashed line) is also plotted to gain more of an insight into the similiarity of F with the critical exponent behaviour.

In order to help visualise the linear relationship mentionned above, the line $F = p$ has been included and can be seen in Figure 4.6(a) (blue dashed line). By displaying this, it can be seen that after approximately $p = 0.8$, there is a purely linear increase in the fraction of sites in the spanning cluster. This is due to the remaining occupied sites slowly becoming part of the spanning cluster. After the dramatic transformation that the spanning cluster undergoes close to p_c, the remaining occupied sites use the spanning cluster's tenuous structure in their favour. This is done through maximising the exposed surface area and slowly becoming connected and part of the spanning cluster as p is increased. This linear behaviour is not an artifact created through the use of a finitely sized system. If an infinite system was used, the same linear behaviour would be observed due to the spanning cluster's characteristics.

Comparing the averaged data and the single iteration data in Figure 4.6(a), i.e. the black solid and red dashed lines respectively, the averaged data has a small but noticeable tail close to $p_c = 0.59$. The tail is due to the fluctuations in the value of p_c on each iteration, all of which arise because of the random nature of each system that is generated. Hence producing a purely statistical increase. When comparing this to the single iteration data, there is a much more sudden and rapid increase as the percolation threshold is reached. This behaviour is much more representative of the nature of the system. Considering the tail once more, it is important to note that its presence in the averaged data would slowly diminish if the system size was increased. This would culminate with it completely vanishing as the system size, $L \to \infty$. This would produce a much sharper increase at p_c. Despite the misrepresentation with the tail, it is necessary to compare both averaged and single sets of data to ensure that there are no artifacts effecting it. Through performing this check here, we can see that there are none present. This allows the use of a single iterations worth of data to access the critical exponent behaviour in more detail.

Figure 4.6(a) shows a plot of the data versus the power law fit. The critical exponent

behaviour is shown in Figure 4.6(a) as a comparison with the both sets of data for F. Looking at the critical exponent displayed in relation to the data, the relationship between the two is not distinctively strong, especially during the rapid change at p_c. In order to obtain a clearer view of the relationship between the simulated data and the critical exponent, smaller increments of 0.001 were used for increasing p. These were focusing on the region 0.55 - 0.65 to capture the behaviour at p_c in more depth. Through this method, a better estimation of p_c can also be obtained.

Bringing our attention to Figure 4.6(b), by reducing the size of the increments within this region, the behaviour becomes more precise. When comparing this to $(p - p_c)^{\frac{5}{36}}$ (green dashed line), the relationship is much stronger than previously seen in Figure 4.6(a). Highlighting the immediate area at p_c, the single iteration data (black solid line) agrees with the critical exponent data. Despite fluctuations, due to noise produced from one iteration, the single iteration data continually returns to the critical exponent line retaining the relationship. It is only between 0.64 and 0.65 that the behaviour is seen to tail off, which is similar to that displayed in Figure 4.6(a). The strength of this relationship justifies the use of $(p - p_c)^{\beta}$ for our method of simulation.

The fraction of sites in the spanning cluster exhibits a drastic change at p_c forming a sparse structure. At this point, the spanning cluster contains a quickly increasing fraction of the system's occupied sites [16]. As a result of this, the spanning cluster has fractal properties, which we will now go on to explore. Before the fractal properties can be explored in full, it is necessary to consider the correlation length and radius of gyration first.

4.2.3 Correlation Length and Radius of Gyration

In the previous sections, the main focus has been the distribution of clusters, their size and the spanning cluster. In this section, the geometry of the clusters will be discussed in more detail. New variables will be introduced which will then allow us to go on and explore the fractal geometry of the clusters in Section 4.2.4. This will then provide us with information about the clusters and their densities on all length scales.

In order to explore the geometry of the cluster, the cluster's linear size will be introduced through the use of the radius of the cluster. Through defining several variables based on radii, the correlation length, ξ, can be defined and explored. To begin, the position of the i^{th} occupied site in a s-cluster, denoted by $\mathbf{r}_i$, is noted. From this, the centre of mass of the cluster of size s, denoted by $\mathbf{r}_{cm}$, is calculated using the following equation,

$$\mathbf{r}_{cm} = \frac{1}{s}\sum_{i=1}^{s} \mathbf{r}_i. \tag{4.3}$$

This provides the position of the centre point of the s-cluster. Using Equation (4.3), the average of the squares radii can be calculated. This is denoted by R_s and can be found using the following equation,

$$R_s^2 = \left\langle \mid \mathbf{r}_i - \mathbf{r}_{cm} \mid^2 \right\rangle = \frac{1}{s}\sum_{i=1}^{s} |\mathbf{r}_i - \mathbf{r}_{cm}|^2. \tag{4.4}$$

By defining R_s^2, the average square distance to $\mathbf{r}_{cm}$ within an s-cluster can be found. Considering this in the two-dimensional plane being used, if the cluster was rotated around

the axis perpendicular to, and originating from, the $\mathbf{r}_{cm}$ of the cluster, this would result in the angular momentum and kinetic energy of rotation being the same, similar to if all the sites were situated on a ring with radius R about the axis. Through this, R_s^2 will be known here after as the *radius of gyration* [8].

The use of the radius of gyration is commonly used throughout many applications of percolation theory. This is due to it providing a more useful and easily obtainable quantity, in comparison to other methods. An example of such a method is to calculate the summation of the average number of sites an occupied site is connected to. This can be obtained using the summation of the average cluster size S, discussed in Section 4.2.1, for a particular site i [13]. Due to this being of less use in general, it will not be explored in more detail here.

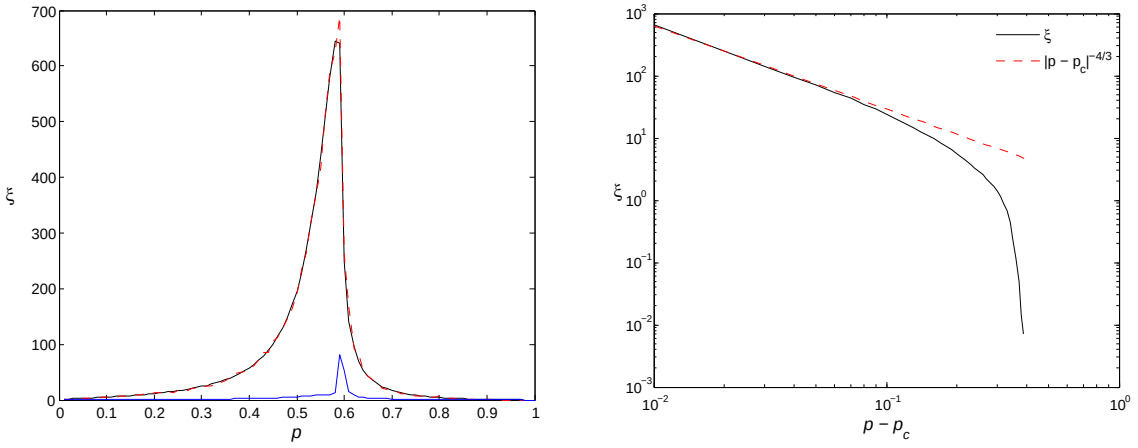
Using Equation (4.4), s (the size of a cluster) and n_s (the number of s-clusters), the *correlation length* can be defined. The correlation length, denoted ξ, is the average distance between two sites that are part of the same finite cluster, representing the characteristic length scale of the clusters. Therefore, the correlation length tells us the characteristic linear size of the finite clusters present in the system, for values of p above and below of p_c [9]. With this in mind, the correlation length is defined using the following equation,

$$\xi^2 = \frac{2\sum_s R_s^2 s^2 n_s}{\sum s^2 n_s}. \tag{4.5}$$

This equation follows from the fact that $2R_s^2$ is the average square distance between two different clusters and that a site belongs to an s-cluster with probability $n_s s$ within which it is then connected with s sites [8].

In order to observe how ξ is effected when the value of p is changed, the values have been calculated for increased p and are displayed in Figure 4.7(a). To ensure there are no artifacts arising in the data, a single iterations worth of data (red dashed line) and averaged data over 1,000 iterations (black solid line) have been plotted. The standard deviation of the data over these 1,000 iterations has also been plotted to help gain an insight into how much the data varies from run to run.

Comparing the single iteration data and the averaged data, the same general behaviour is displayed across the two. For small p, ξ is small due to the system consisting of only small isolated clusters. As p starts to increase, ξ increases displaying exponential growth as $p \to p_c^-$. This exponential growth implies that ξ should diverge at p_c. Here, we can see that the data does not diverge, but peaks a finite point. This is solely due to the finite nature of system. If the system size was increased and tended towards infinity, the value of ξ would diverge off to infinity as well. This is a direct consequence of the correlation length being highly influenced by the sparse structure of the spanning cluster at p_c and a higher probability of two occupied sites belonging to the same cluster. After p_c is reached, the spanning cluster is then no longer considered when calculating ξ due to its influence in the divergence of ξ. As a result of this, the correlation length rapidly decreases after p surpasses p_c, with a slowly decreasing trend towards 0 as p tends towards 1. Here ξ is also said to describe the typical radii of the largest hole in the spanning cluster, as these are the remaining occupied sites. Considering the standard deviation of the averaged data (blue solid line), there is little deviation for most p values. The deviation then suddenly peaks as p_c is reached, which is a result of the value of p_c fluctuating as each system is randomly generated. The standard deviation then diminishes once again as $\xi \to 0$. The peak in standard deviation supports the less severe and smoother peak of



(a) (b)

Figure 4.7: Linear plot (a) and loglog plot (b) of the correlation length, ξ vs p. Figure (a) shows ξ vs p for one iteration (red dashed line), averaged over 1,000 iterations (black solid line) and the standard deviation over the 1,000 iterations (blue solid line). Figure (b) shows the loglog plot of the averaged data of ξ corresponding to the values for $p - p_c$ shown (black solid line) and the critical exponent behaviour given by $|p - p_c|^{-\frac{4}{3}}$ (red dashed line).

the averaged data. This is caused by the fluctuations in cluster sizes, in comparison to the sharp peak of the single iteration.

Having explored the behaviour of ξ over varying values of p, we wanted to determine the universal scaling properties of ξ as $p \to p_c$. This can be characterised by the following proportionality,

$$\xi \propto |p - p_c|^{-\nu} \quad \text{for } p \to p_c, \tag{4.6}$$

where ν is the critical exponent and the known value for $\nu = \frac{4}{3}$ for $d = 2$. This behaviour indicates that there is exactly one length ξ which determines the critical behaviour exhibited in an infinite system [13].

To confirm that ξ exhibits this behaviour as $p \to p_c$, a loglog plot has been generated to compare the averaged data with the power law, characterised by the critical exponent ν for $p - p_c$. This is shown in Figure 4.7(b). Looking at the averaged data (black solid line), we can see there is a strong relationship with $|p - p_c|^{-\frac{4}{3}}$ (red dashed line). Even when the misrepresentations of data, that are due to the averaging of fluctuations close to p_c, are taken out of the consideration, the relationship is still strong. It is only when the averaged data starts to tail off (due to the original data tending towards zero at a slower rate than close to p_c), that the relationship breaks down between the two. The clear relationship displayed with the lower values of $p - p_c$, support the power law behaviour characterised by ν, and hence justifying the results obtained through the model.

Having established that $\xi = \infty$ for $p = p_c$, we are able to proceed and explore that fractal properties of the spanning cluster.

4.2.4 Fractal Dimension

In Section 4.2.3, the radius of gyration and correlation length were introduced. It is with the use of these characteristics that the fractal dimension of the spanning cluster can be addressed.

Aforementioned throughout the report, the spanning cluster has a very sparse and tenuous structure when it first appears. Recalling the random network description in the introduction, we know that the removal of a single site can be detrimental to it's connectivity. At this point, i.e. $p = p_c$, the spanning cluster is an example of a fractal and presents 'holes' and 'cracks' in its structure not only at first glance but on all length scales. To explore this in more detail, the idea of self-similarity will be discussed.

The initial spanning cluster displays an internal fractal geometry which is demonstrated through the density of the spanning cluster's dependence on the length scale [8]. In order to highlight this dependence, the mean mass of the cluster calculated within a circle of radius r is considered. This is denoted by $M(r)$. Through the use of $M(r)$, a formal definition of the fractal geometry of the clusters is given through the following equation,

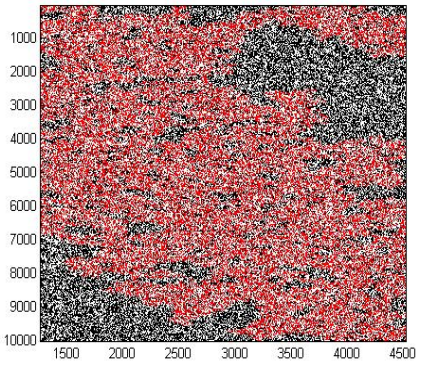
$$M(r) \sim \begin{cases} r^{d_f}, & \text{for } r \ll \xi, \\ r^{d}, & \text{for } r \gg \xi. \end{cases} \tag{4.7}$$

Here r is the radius of the circle that $M(r)$ is measured in, d_f is the fractal dimension and d is the Euclidean dimension. The known result for $d_f = \frac{91}{48}$ for $d = 2$ [13].

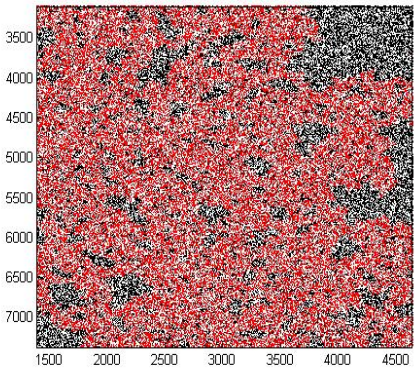
Equation (4.7) tells us that close to p_c for $r \ll \xi$, the finite clusters and spanning cluster are self-similar and are expressed by the power law behaviour of the fractal dimension r_{d_f}. Here ξ is the characteristic length that, when crossed, determines critical and non-critical behaviour [13]. This self-similarity can be observed in Figure 4.8.

Figure 4.8(a) displays the spanning cluster present in a $10,000 \times 10,000$ system at $p = 0.5932$ (close to p_c due to the random nature of the system). The spanning cluster at this length scale is seen to be full of 'holes' and 'cracks', which is a base characteristic of fractal behaviour. Taking a smaller section of this image and magnifying it to the original cluster size (shown in Figure 4.8(b)), the 'holes' and 'cracks' are still as prevalent across the image. Repeating this method twice more, shown in Figure 4.8(c) and 4.8(d), the characteristic is not lost and when comparing them with the original spanning cluster, especially if the scales were removed, it is difficult to distinguish between them. Here, exploring a range in which the images are still within an acceptable resolution, has resulted in the same characteristic behaviour being exhibited in each case. Unfortunately, it has not been possible to create further magnified images, due to loss of resolution. However as the size of the system is increased, particularly tending towards infinity, the number of images that are able to be produced increases, all of which would display the discussed behaviour.

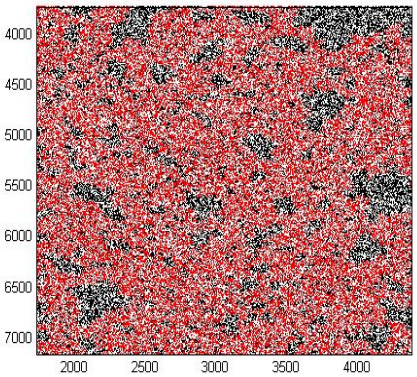
The self-similarity demonstrated in Figure 4.8, results in the clusters being characterised by the fractal dimension for $r \ll \xi$. This identifies that, particularly at $p = p_c$, nearly all the occupied sites belong to the spanning cluster's infinite network and nearly all of the unoccupied sites are part of an infinite network of 'connected' unoccupied sites [8]. This presents the distribution of a large cluster along with a large hole/s that are connected by the linear size of the finite clusters, with regards to the correlation length ξ. When $p \gg p_c$ the remaining small isolated clusters appear quite homogeneous as the 'holes' of the spanning cluster. Overall, this concludes that the clusters are typically



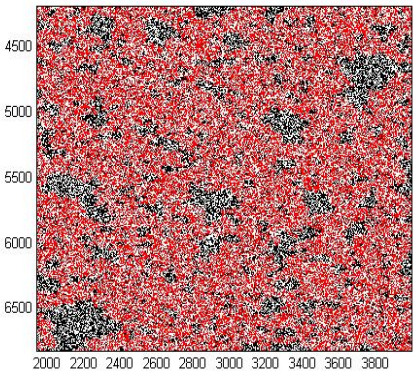
(a) Spanning cluster at $p = 0.5932$



(b) Rescaled image of a section of 4.8(a)



(c) Rescaled image of a section of 4.8(b)



(d) Rescaled image of a section of 4.8(c)

Figure 4.8: Original (a) and rescaled ((b), (c) and (d)) images of the spanning cluster in a $10,000 \times 10,000$ system to demonstrate the spanning cluster's fractal properties.

homogeneous on length scales $r \gg \xi$ and are proliferate with holes for length scales $r \ll \xi$.

After reviewing the behaviour of the fractal dimension of the system above, it is necessary to consider the fractal properties for our model. In order to do this, the variable described as the number of occupied sites within a circle of radius r, denoted by N_r, is introduced. Monitoring the effect of increasing the radius r for several probabilities is displayed in Figure 4.9.

Focusing attention on the five probabilities shown in Figure 4.9, namely p =0.2, 0.4, 0.6, 0.8 and 1, it can be see that N_r increases as p is increased. For low p values, displayed here by $p = 0.2$ and $p = 0.4$ (the green and maroon lines respectively), N_r is very small and never surpasses over 200 for any particular value of r. Comparing this to the higher values of p, i.e. $p \geq p_c$, N_r displays more parabolic behaviour, with $p = 1$ showing similarity to r^2 at first glance.

Examining the power law behaviour of the probabilities, a loglog plot of the data was considered. Figure 4.10(a) shows the loglog plot for all the probabilities. From this plot, the behaviour of $p = 0.2$ (green line) and $p = 0.4$ (purple line) can be determined, concluding that no power law behaviour is displayed. This is due to the large number of small isolated clusters within the system which causes no two occupied sites to ever be too far apart. Even for $p = 0.4$, where the average cluster sizes has increased, no characteristic power law behaviour is displayed. In contrast to this, there is $p = 0.6$ (blue

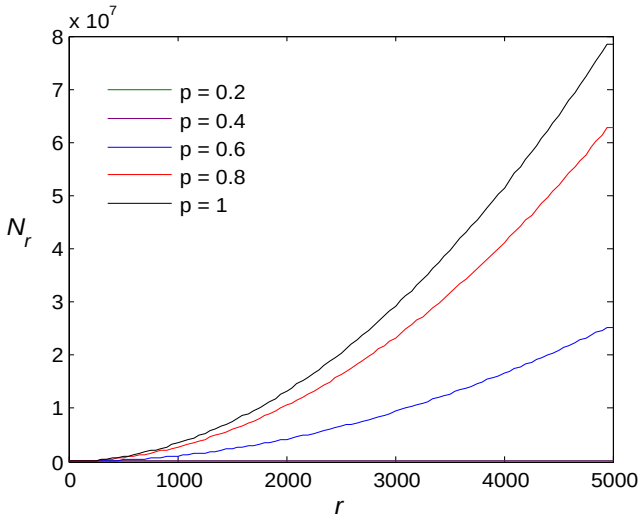


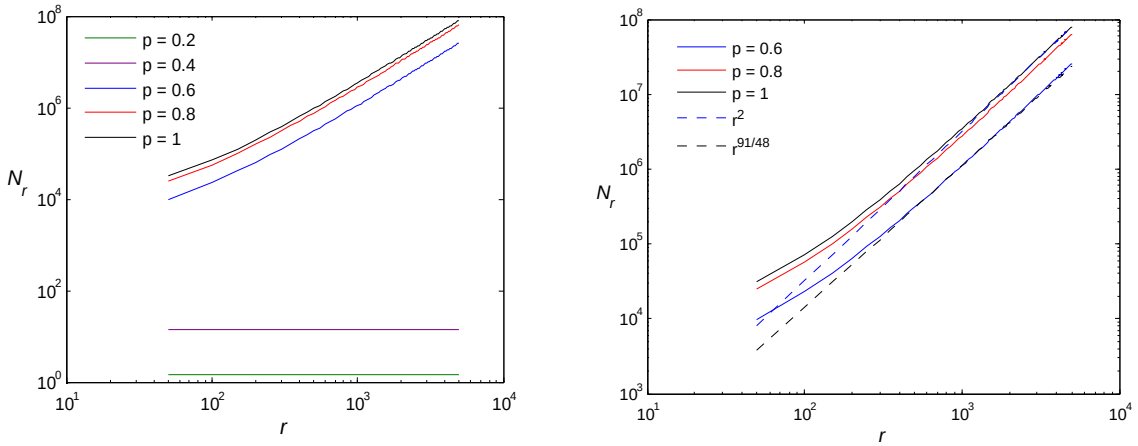
Figure 4.9: Number of occupied sites with radius r, N_r vs r for several probabilities, namely $p = 0.2, 0.4, 0.6, 0.8$ and 1, which are represented by the green, maroon, blue, red and black lines respectively.

line), $p = 0.8$ (red line) and $p = 1$ (black line). All three of these probabilities show linear increasing behaviour on the loglog plot which implies power law behaviour within the original data (exhibited in Figure 4.9). Isolating these three probabilities, allows further analysis of the behaviour to take place.

Observing the initial behaviour of all three probabilities, a tail in the data is noticeable. These uncharacteristic tails are due to the discretised nature of the system. The use of radii on a square grid produces oscillations in the number of sites included in each circle. This is further accentuated by the binning method used to quantify the data. The number of sites in the bins with small radii is much smaller than the catchment for larger radii, hence resulting in less data being averaged and causing more deviation at low r. The calculations for N_r were repeated for $10,000 \times 10,000$ to confirm the tail is not a lack of data or accuracy. The presence of the tail in a $10,000 \times 10,000$ system and at $p = 1$ confirms that it is not due to statistic uncertainty. For $p = 1$, all the sites in the system are occupied and belong to the spanning cluster, with no number of iterations changing this behaviour.

Taking the tail out of the consideration, the general linear increasing trend can be observed. Considering $p = 1$ (black solid line) first, we know that all the sites in the system are occupied at this point and hence are part of the spanning cluster. Examining the 2D system that we are in, taking circles of radii r produces an area of $2\pi r$. With all the sites occupied, this will result in a parabola curve which is displayed in Figure 4.9. Looking at the loglog plot for this, a gradient of 2 is expected. In order to deduce if this behaviour is present here, r^2 has been plotted in Figure 4.10(b) (blue dotted line). Comparing the behaviour of N_r vs r for $p = 1$ and r^2, it apparent that there is a close relationship between the two as r increases. Once again, due to the discretised nature of the system, the relationship is not extended across all the data, however it is still apparent. Having confirmed the expected behaviour for $p = 1$, demonstrated through the above reasoning and Equation (4.7) with $d = 2$, $p = 0.6$ will now be considered.

Looking at the loglog plot N_r vs r for $p = 0.6$, we can gain an insight into the behaviour of the occupied site at the closest value to p_c the increments used will allow. At



(a) (b)

Figure 4.10: Loglog plots of N_r vs r in order to deduce the fractal dimension. Figure (a) shows the loglog plots of N_r vs r for five probabilities, namely $p = 0.2, 0.4, 0.6, 0.8$ and 1, which are represented by the green, maroon, blue, red and black lines respectively. Figure (b) displays the loglog plots of N_r vs r for $p = 0.6, 0.8$ and 1 (once again represented by the blue, red and black lines respectively) along with r^2 (blue dotted line) to exhibit the expected behaviour at $p = 1$ and $r^{\frac{91}{48}}$ (black dotted line) to exhibit the fractal dimension present at $p = 0.6$.

this value of p, the spanning cluster will have a sparse structure implying it is most likely to exhibit fractal characteristics. Recalling the known value of ν, the fractal dimension is shown by $r^{\frac{91}{48}}$. Therefore, in addition to $p = 0.6$, $r^{\frac{91}{48}}$ has also been plotted in Figure 4.10(b) (black dotted line). Comparing the two lines, there is a strong relationship between them in an intermediary range between the end of the tail and the slight increase in the data at the end. The strong relationship within this range signifies the spanning cluster at $p = 0.6$, and more importantly at $p = p_c$, boasts a fractal dimension of $\frac{91}{48}$. Recalling the spanning cluster shown in Figure 4.8, this conclusion is demonstrated visually.

Having established the fractal dimension for model, the use of this may be being questioned. Why is the fractal dimension or having fractal properties useful? As mentionned in the previous sections, there are a variety of real-life applications for percolation theory, from networks such as airline routes to modelling the spread of forest fires. By quantifying the fractal dimension of the clusters on all length scales, described in Equation (4.7), the behaviour of the system can be further understood and applied. An example of this is used when calculating the average concentration of oil in rocks, which is frequently used in oil extraction. Through the use of the fractal dimension, the size and sparsity of the clusters (i.e. holes and pockets of oil in the rock) can be deduced and the concentration of oil can be estimated. This facilitates the decisions made on which route of extraction is to be used [8].

Chapter 5

Percolation in the Presence of Obstacles

In Chapters 2 and 4, systems of regular shapes and sizes were considered and results regarding them have been discussed. Using these types of systems is useful in gaining understanding into the critical behaviour displayed, however these types of systems are not typically found throughout nature.

Considering percolation theory throughout nature gives rise to continuum percolation which is said to be the most natural example of percolation theory. Unlike site, bond and site-bond percolation, all of which arise in systems formed from different types of lattices, continuum percolation describes the percolation of points that are randomly positioned throughout some continuous space [1]. An example of the use of continuum percolation is typically used in the description of porous materials.

In order to find a middle ground between the two considerably different types of systems, and make the systems more applicable to real life, obstacles will now be introduced into the system. Although this seems like an obscure idea, it provides a useful insight into modelling disrupted systems which widely occur throughout nature. Considering this in terms of an application of percolation theory, the spread of forest fires can be reduced through the strategic cutting down of trees within the forest. Through cutting down carefully selected trees, obstacles are being introduced into the system, i.e. the forest. This causes disruptions in its ability to percolate which could then reduce the fire's ability to spread.

In this Chapter, the focus will be shifted to evaluate how introducing these obstacles into the system affects the system's ability to percolate. Through this, any changes in the percolation threshold will be discussed. As before, a $1,000 \times 1,000$ square system will be used to simulate and generate results.

5.1 Introducing Obstacles into the System

Obstacles arise throughout nature in a large array of shapes, sizes and numbers. In the interest of generating clear results from the influence of the obstacles, identical square obstacles have been introduced at regular intervals throughout the system. This provides a basis of analysis upon which the idea can be built upon. For example, by randomly generating the position of the obstacles along with changing the shape and sizes of them. Although this does not provide a direct insight into obstacles that occur throughout nature, it builds a foundation of understanding.

Before the obstacles are introduced into the system, two variables need to be introduced so the characteristics of the obstacles can be determined. To characterise the obstacles in terms of a more generally applicable variable, the width of the square obsta-

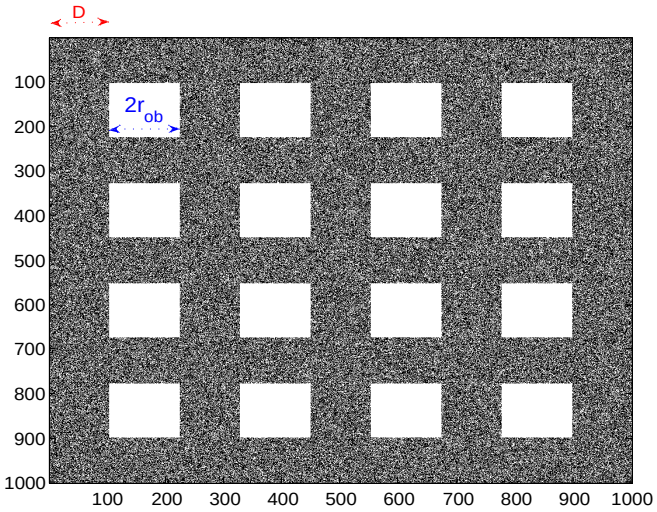
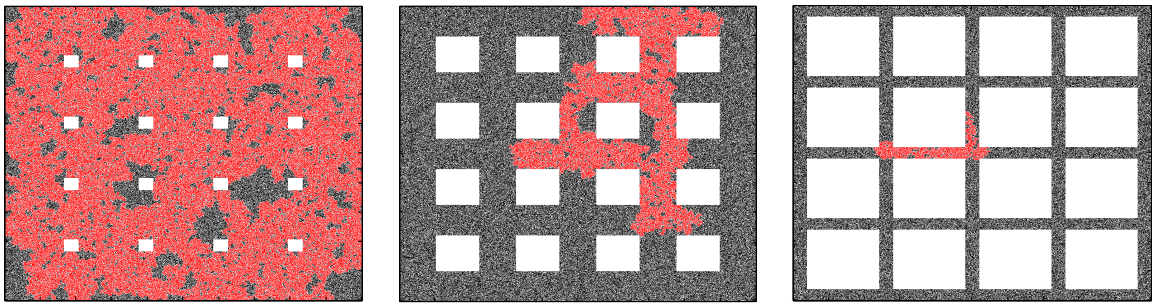


Figure 5.1: System with obstacles displaying the parameters that are used. Here $2r_{ob}$ is the width of the obstacle and D is the distance between each of the obstacles or the obstacle and the boundary.

cles is denoted by $2r_{ob}$ where r_{ob} denotes the *half-width of the obstacle*. In addition to this, the distance between the obstacles is denoted by D. Although this will not be used in much detail here, apart from to ensure that the obstacles are regularly positioned, D would play a larger role when the obstacles were randomly positioned. To demonstrate both of these variables visually, a system with obstacles included has been generated in Figure 5.2. Here, sixteen obstacles have been placed into the system. As a result of each of the obstacles characteristics (i.e. shape, size, position and number) being liable to vary in large amounts, the number, shape and positioning will be fixed from here on in. This is to allow the effect of changing the size of the obstacles to be observed without any other influence.

Having established the characteristics of the obstacles used in the model, the effect of changing r_{ob} can now be observed. Starting at $r_{ob} = 20$ and increasing the value in 3 increments of 40, the systems behaviour has been monitored for a given probability, namely $p = 0.6$. Highlighting the biggest cluster in red, the remaining occupied sites in black and the unoccupied sites in white (as introduced in Chapter 2), the effect of the



(a) (b) (c)

Figure 5.2: Introduction of obstacles of different radii into a $1,000 \times 1,000$ system. Example radii are $r_{ob} = 20$ (a), $r_{ob} = 60$ (b) and $r_{ob} = 100$ (c). All examples are shown for $p = 0.6$.

obstacles on the system can be observed. The results of this are shown in Figure 5.3. For $r_{ob} = 20$, the ability of the biggest cluster to percolate across the system does not seem to be hindered. However, as r_{ob} is increased to 60, the presence of the obstacles starts to take effect. By highlighting the biggest cluster in red, we can see that the biggest cluster is no longer a spanning cluster at $p = 0.6$, which would normally be expected. Increasing r_{ob} further, to $r_{ob} = 100$, results in the biggest cluster reducing even more in size. This concludes that a spanning cluster is much less probable at $p = 0.6$.

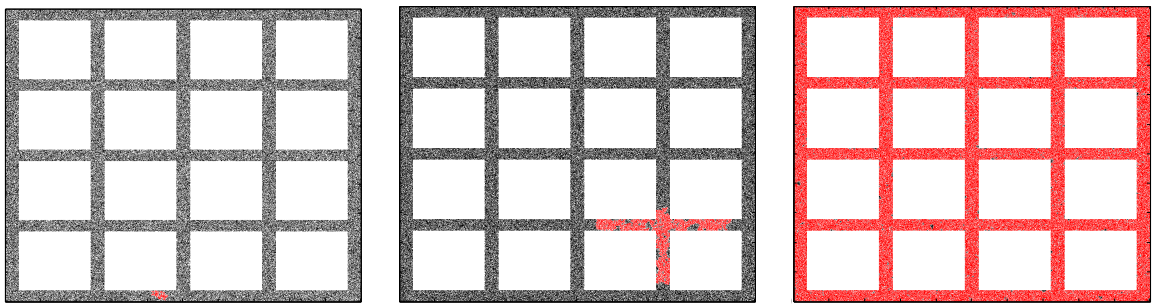
With an increase in r_{ob} causing a decrease in the size of the biggest cluster across a given p, it would now be beneficial to determine how changing p effects the behaviour of the system and the influence the obstacles have on the value of p_c.

5.2 The Influence of Obstacles on p_c

Before determining the effect of introducing obstacles on the value of p_c, we will consider how increasing p effects the size of the biggest cluster for a given obstacle size.

Using our knowledge of p_c (see Section 4.1), the probabilities used here are aimed to be more directly around the known value of p_c. Considering the effect of changing p on a system with larger sized obstacles to demonstrate a more extreme case, $r_{ob} = 100$, has been used. With this in mind, the system has been plotted for three probabilities, $p = 0.5$, $p = 0.6$ and $p = 0.7$, and is displayed in Figure 5.3.

Looking at $p = 0.5$ (Figure 5.3(a)), we can see that it displays similar behaviour to the standard system, although the biggest cluster is slightly smaller. Increasing the probability to $p = 0.6$ (Figure 5.3(b)), the biggest cluster has increased in size however does not span across the system. This is a contrast to the expected behaviour of the standard system at $p = 0.6$. Comparing this to $p = 0.7$ (Figure 5.3(c)), the biggest cluster has transformed into a spanning cluster, exhibiting the same behaviour as the standard system at this value of p. This highlights that the transition between finite clusters and the spanning cluster has shifted, with the percolation threshold now sitting somewhere between $p = 0.6$ and $p = 0.7$.

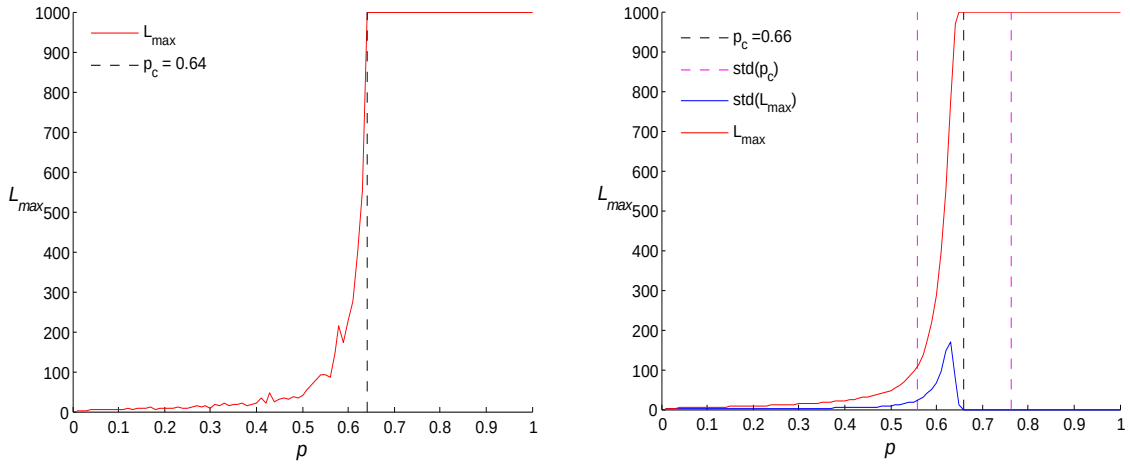


(a) (b) (c)

Figure 5.3: The effect of changing p on a $1,000 \times 1,000$ system containing obstacles with a radius $r_{ob} = 100$. Three probabilities are used to show the effect, namely $p = 0.5$ (a), $p = 0.6$ (b) and $p = 0.7$ (c).

Despite the characteristic behaviour of the clusters and system as a whole being similar, shown here by the occurrence of a spanning cluster, the obstacles have effected the quantitative value of one of the fundamental results, p_c. To confirm this alteration, the percolation threshold will be re-evaluated. To calculate this, we recall the variable

L_{max}, which is defined as the maximum dimension (i.e. height or width) of the biggest cluster present in the system at the chosen probability. In order to understand how L_{max} changes for the obstacle ridden system, it was calculated at each value of p. The results were recorded for one iteration and for the averaged data over 1,000 iterations. These results can be seen in Figure 5.4.



(a) (b)

Figure 5.4: L_{max} vs p for one iteration (a) and averaged over 1,000 iterations (b) for a system with obstacles with $r_{ob} = 100$. Figure (a) shows L_{max} vs p (red line) highlighting the value of p_c (black dotted line). Figure (b) shows the averaged data for 100 iterations for L_{max} vs p (red line) along with the standard deviation of L_{max} (blue line), the value of p_c (black dotted line) and the standard deviation of p_c over all iterations (magenta dotted lines)

Considering the results for one iteration, shown in Figure 5.4(a), the data displays a larger amount of noise than the standard system data (Figure 4.2) for $p < p_c$. However a general increasing trend can still be seen. Due to the value of p_c increasing, the gradual increasing trend is stretched to higher values of p before the characteristic sharp increase, within close proximity of p_c, occurs. After p_c has been reached, L_{max} plateaus as the maximum dimension of the system has been reached (in this case 1,000). For this single iteration, p_c has been found to be 0.64. In order to reduce the statistical uncertainty of this data, 1,000 iterations were performed and the averaged data is displayed in Figure 5.4(b).

The figure shows a much smoother curve that follows a similar trend as Figure 5.4(a). However, the averaged data does exhibit some different behaviour. Considering the typical sharp increase in L_{max} close to p_c and comparing this to the increase seen in Figure 5.4(b), we can see the sharp increase has been surpressed. The sharp increase now occurs over a large range just below p_c, with L_{max} tending towards the maximum dimension as $p \to p_c$, instead of the typical abrupt transition. To gain more of an insight into this, the standard deviation over the 1,000 iterations was also calculated. This is represented by the blue solid line. Concentrating on the standard deviation close to p_c, as elsewhere the standard deviation is very small, we can see the peak in the standard deviation occurs approximately 0.04 away from p_c, instead of immediately before p_c as would be expected. This expected behaviour is related to the behaviour shown for the standard system in Figure 4.2. To explain this behaviour, the spanning cluster must be considered. The removal of a single

site of the spanning cluster at p_c can be detrimental to it's connectivity. Consequently, by introducing multiple obstacles in the system, such sites are being removed which causes larger deviations in the connectivity and hence fluctuations in the values that have been obtained for p_c. As the value of p_c is varying more, the standard deviation is higher but is spread over a larger range. From this point onwards, the deviation then quickly decreases back down to zero as L_{max} plateaus, due to the maximum dimension being reached.

Figure 5.4 also displays the values of p_c found from the simulations (see the black dotted lines on both images). In Figure 5.4(a), p_c was found to be 0.64. Comparing this with the value of p_c found for the averaged data in Figure 5.4(b), the deviation discussed above is brought into context. This is further supported by the standard deviation of p_c over the 1,000 iterations, shown by the magenta dotted lines in Figure 5.4(b). After the deviations were accounted for and averaged, it was found that $p_c = 0.66$ for obstacles with $r_{ob} = 100$. This varies significantly from the standard system p_c result of $p_c = 0.592746$, when taking into consideraton the critical behaviour shown either side of this value.

Having explored the effect of obstacles of radius $r_{ob} = 100$ on p_c, it follows to consider the effect changing r_{ob} has. With a view of reviewing p_c for each r_{ob} chosen, the values of r_{ob} that were used were 0 - 100 (increasing in increments of 20), 100 - 120 (increasing in increments of 5) and 124 (as this is the maximum radii possible for a system with size $L = 1,000$). Averaging each system over multiple iterations, to reduce statistical uncertainty and artifacts, the results are shown in Figure 5.5.

Using $p_c = 0.6$ for $r_{ob} = 0$ (obtained in Section 4.1.1) as a comparison the influence of increasing r_{ob} can be determined. Examining lower values of r_{ob}, i.e. 20, 40 and 60, there is very little deviation away from the known p_c result. However at $r_{ob} = 80$, the behaviour seems to change causing an increase in p_c and showing a tendency to behave as $r_{ob} = 100$ did above. Subsequently, from this point onwards, p_c begins to increase. This occurs gradually at first but grows rapidly in speed tending towards 1 as $r_{ob} \to 124$. The percolation threshold finally reaches 1 when $r_{ob} = 124$, which due to the size of the system, leaves a one site thick border around the edge. This reduces down to simulating percolation theory in 1D. The known result for this is $p = 1$, which therefore supports the simulation.

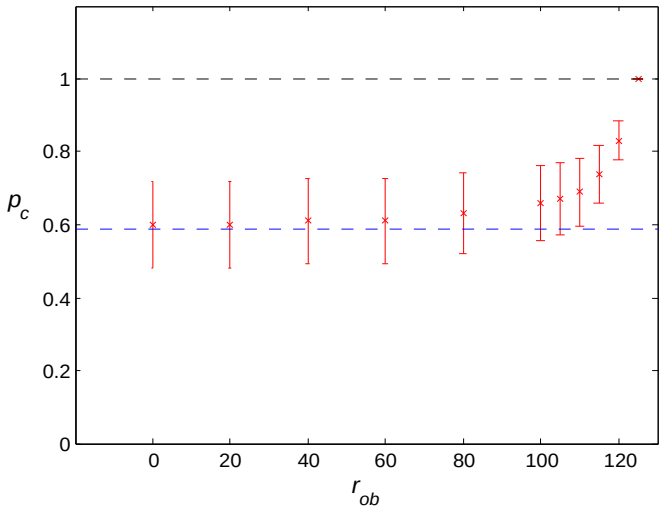


Figure 5.5: Comparing p_c for increasing r_{ob} to determine the effect the obstacles have on the behaviour of the system.

Through this comparison, the effect of introducing obstacles into the system and increasing their size can be observed. It has been found that the same fundamental behaviour is exhibited by the system for low values of r_{ob} however there is a point, that could be thought of as a 'critical value' of r_{ob}, at which p_c begins to change and increase, deviating away from the standard system results. This presents a useful insight into a slightly more natural approach to percolation theory and provides a basis of further study into the area.

Chapter 6

Conclusion

The purpose of this report can be summarised into three main areas. Firstly, to introduce the reader into percolation theory, covering the key concepts that are needed to create the system. This is proceeded by the exploration of the fundamental characteristics and universal scaling behaviour of the conventional, obstacle-free, percolating system. Finally moving onto the introduction of obstacles into the system and their effect on the percolation properties.

Chapter 2 began by introducing the concepts of percolation theory. The tools used to create the system were introduced and the key features were explored, highlighting the importance of the spanning cluster. This was followed by the consideration of the computational cost of simulating each system in Chapter 3. The algorithm of the *bwconncomp* function was explained, as this is the basis of the Matlab code, and the variable choices were clarified. The chapter drew to a close through considering the computational cost of varying the system size.

The concepts introduced in Chapter 2 were built upon in Chapter 4 obtaining an understanding about the behaviour of the system. The calculation of the percolation threshold introduced the idea of critical behaviour and confirmed the first generated result of the simulated systems. Providing this basis of knowledge, the attention was then shifted to the universal scaling behaviour. Critical exponents were explained and the power law behaviour of properties, such as the average cluster size, were formalised.

The intention of Chapter 5 was to recognise a more naturally occurring percolation theory and explore an aspect of percolation theory which, to our knowledge, has not been considered before. The introduction of obstacles into the system was considered in a variety of ways, however one route was chosen to focus on. Utilising the ideas explored in Section 4.1.1, the percolation threshold was recalculated and the effects the obstacles caused were evaluated, determining an increase in p_c. Through this initial evaluation, a foundation has been formed for further work. The first aim is to explore the effect of the obstacles on the universal scaling properties discussed in Chapter 4, such as the average cluster size and fractal dimension. This will then develop into the gradual inclusion of changes in the numbers, shapes, sizes and position of the obstacles, into the model, which provides an amplitude of possibilities into the understanding of networks used across the world. Percolation theory is a vast subject within which the possibilities are endless. From oil extraction to the spread of disease, these models aim to provide a realistic representation of real-life systems, in the hope of producing more accurate results.

Bibliography

[1] Callaway, D. S., Newman, M.E.J, Strogatz, S. H., Watts, D.J. *Network, Robustness and Fragility: Percolation on Random Graphs*, Physical Review Letters, Vol.85(25), American Physical Society, Dec 2000.

[2] Flory, P. J., *Molecular Size Distribution in Three Dimensional Polymers I. Gelation*, Journal of American Chemical Society, Vol. 63, pg. 3083-3090, 1941.

[3] Stockmayer, W. H, *Theory of Molecular Size Distribution and Gel Formation in Branched Polymers II. General Cross Linking*, Journal of Chemical Physics, Vol. 12(4), pg.125-131, 1944.

[4] Broadbent, S., Hammersley, J., *Percolation processes I. Crystals and mazes*, Proceedings of the Cambridge Philosophical Society, Vol. 53, pg. 629-641, 1957.

[5] OpenFlights, *Airport, airline and route data*, OpenFlights.org, Online Resource, Webpage. http://openflights.org/data.html

[6] Dobson, R., *Injection of jelly that can ease back pain - and fix a slipped disc*, Mail on Sunday, Mail Online, Online Resource, Webpage, Aug 2013. http://www.dailymail.co.uk/health/article-2402671/Injection-jelly-ease-pain–fix-slipped-disk.html

[7] Smart, V., *Mearns and Gill - Offshore Achievement Awards 2012*, Smart Design, Blog entry, Online Resource, Webpage, Nov 2011. http://imgkid.com/porosity-of-rocks.shtml

[8] Stauffer, D., Aharony, A., *Introduction to Percolation Theory*, Second Edition, Taylor & Francis Ltd, London, 1991.

[9] Bunde, A., Kantelhardt, J. W., *Introduction to Percolation Theory (Part A)*, Diffusion in Condensed Matter, edited by J. Kärger and R. Haberlandt, pg. 340-352, Vieweg Verlag, Wiesbaden, 1998.

[10] Austin, D., *Percolation: Slipping through the Cracks*, American Mathematical Society Feature Column, Online Resource, Webpage. http:\\www.ams.org/samplings/feature-column/fcarc-percolation

[11] Haralick, R. M., Shapiro, L.G., *Computer and Robot Vision*, Vol. 1, Addison-Wesley, USA, 1992.

[12] Hoshen, J., Kopleman, R., *Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm**, Physcial Review B, Vol. 14(8), pg.3438-3445, American Physical Society, Oct 1976.

[13] Christensen, K., *Percolation Theory*, Lecture Notes, Imperial College London, Online Resource, Webpage, Oct 2002. http:\\www.mit.edu/∼levitov/8.334/notes/percol_notes.pdf

[14] Christensen, K., *Critical Phenomena and Percolation Theory III*, Lecture Slides, Joint CRM-Imperial College School and Workshop, Complexity & Networks Group Imperial College London, Barcelona, Online Resource, Webpage, April 2013. http:\\www.crm.cat/en/Activities/Documents/Lecture03-web.pdf

[15] Christensen, K., Moloney, N. R., *Complexity and Criticality*, Imperial College Press Advanced Physics Texts Vol. 1, Imperial College Press, London, 2005.

[16] Giordano, N., *Introduction to Percolation*, Lecture Slides, Random Media Summer School, Purdue University, Online Resource, Webpage, July 2006. http:\\www.physics.purdue.edu/flow/percolation.pdf