Anders Malthe-Sorenssen*

Percolation and Disordered Systems – A Numerical Approach

Apr 30, 2015

Springer

* Department of Physics, University of Oslo, Norway.

Contents

Introduction to percolation 1

Percolation is the study of connectivity of random media – and of other properties of connected subsets of random media. Fig. 1.1 illustrates a porous material – a material with holes, pores, of various sizes. These are examples of random materials with built-in disorder. In this book, we will address the physical properties of such media, develop the underlying mathematical theory and the computational and statistical methods needed to discuss the physical properties of random media. In order to do that, we will develop a simplified model system – a model porous medium – for which we can develop a well-founded mathematical theory, and then afterwards we can apply this model to realistic random systems.

The porous media illustrated in the figure serves as a useful, fundamental model for random media in general. What characterizes the porous material in fig. 1.1? The porous medium consists of regions with material and without material. It is therefore an extreme, binary version of a random medium. An actual physical porous material will be generated by some physical process, which will affect the properties of the porous medium in some way. For example, if the material is generated by sedimentary deposition, details of the deposition process may affect the shape and connectivity of the pores, or later fractures may generate straight fractures in addition to more round pores. These features are always present in the complex geometries found in nature, and they will generate correlations in the randomness of the material. While these correlations can be addressed in detailed, specific studies of random materials, we will here instead start with a simpler class of materials – uncorrelated random, porous materials.

We will here introduce a simplified model for a random porous material. We divide the material into cubes (*sites*) of size d. Each site can be either filled or empty. We can use this method to characterize an actual porous medium, as illustrated in fig. 1.1, or we can use it as a model for a *random porous medium* if we fill each voxel with a probability p. On average, the

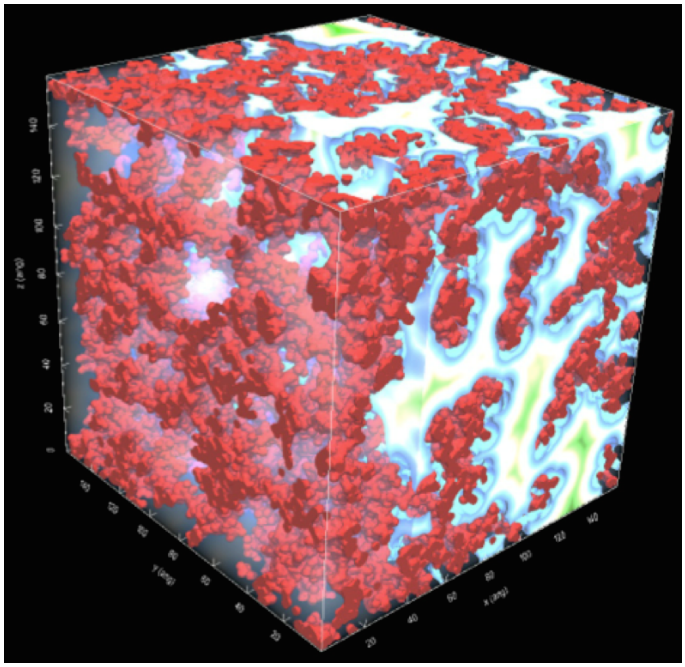

Fig. 1.1 Illustration of a porous material from a nanoporous silicate (SiO_2). The colors inside the pores illustrates the distance to the nearest part of the solid.

volume of the solid part of the material will be $V_s = pV$, where V is the volume of the system, and the volume of the pores will be $V_p = (1-p)V$. We usually call the relative volume of the pores, the **porosity**, $\phi = V_p/V$, of the material. The solid is called **the matrix** and the relative volume of the matrix, V_s/V is called the solid fraction, $c = V_s/V$. In this case, we see that p corresponds to the solid fraction. Initially, we will assume that on the scale of lattice cells, the fill probabilities are statistically independent – we will study an *uncorrelated random medium*.

Fig. 1.2 illustrates a two-dimensional system of 4×4 cells filled with a probability p. We will call the filled cells occupied or set, and they are colored black. This system is a 4×4 matrix, where each cell is filled with probability p. We can generate such a matrix, `m`, in matlab using

```
p = 0.25;
z = rand(4,4);
m = z<p;
imagesc(m);
```

The resulting matrices are shown in the figure for various values of p. The left figure illustrates the matrix, m with its various values. A site i is set as p reaches the value m_i in the matrix. (This is similar to changing the water level and observing what parts of a landscape is above water).

Percolation is the study of connectivity. The simplest question we can ask is when does a path form from one side of the sample to the other?

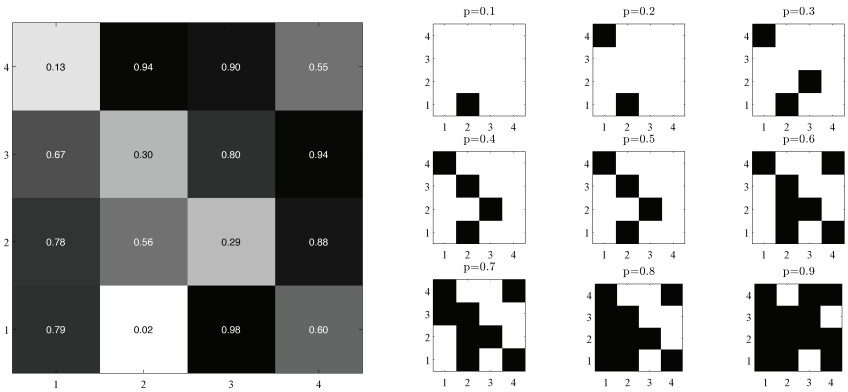

Fig. 1.2 Illustration of an array of 4×4 random numbers, and the various sites set for different values of p.

By when, we mean at what value of p. For the particular realization of the matrix m shown in fig. 1.2 we see that the answer depends on how we define connectivity. If we want to make a path along the set (black) sites from one side to another, we must decide on when two sites are connected. Here, we will typically use nearest neighbor connectivity: Two sites in a square (cubic) lattice are connected if they are nearest neighbors. In the square lattice in fig. 1.2 each site has $Z = 4$ nearest neighbors and $Z = 8$ next-nearest neighbors, where the number Z is called the connectivity. We see that with nearest-neighbor connectivity, we get a path from the bottom to the top when $p = 0.7$, but with next-nearest neighbor connectivity we would get a path from the bottom to the top already at $p = 0.4$. We call the value p_c, when we first get a path from one side to another (from the top to the bottom, from the left to the right, or both) the *percolation threshold*. For a given realization of the matrix, there is well-defined value for p_c, but for another realization, there would be another p_c. We therefore need to either use statistical averages to characterize the properties of the percolation system, or we need to refer to a theoretical – thermodynamic – limit, such as the value for p_c in an infinitely large system. When we use p_c here, we will refer to the thermodynamic value.

In this book, we will develop theories describing various physical properties of the percolation system as a function of p. We will characterize the sizes of connected regions, the size of the region connecting one side to another, the size of the region that contributes to transport (fluid, thermal or electrical transport), and other geometrical properties of the system. Most of the features we study will be universal – independent of many of the details of the system. From fig. 1.2 we see that p_c depends on the details: It depens on the rule for connectivity. It would also depend on the type of lattice used: square, trianglar, hexagonal, etc. The value of p_c is specific. However, many other properties are general. For example, how the conductivity of the porous medium depends on p near

p_c does not depend on the type of lattice or the choice of connectivity rule. It is universal. This means that we can choose a system which is simple to study in order to gain intuition about the general features, and then apply that intuition to the special cases afterwards. While the connectivity or type of lattice does not matter, some things do matter. For example, the dimensionality matters: The behavior of a percolation system is different in one, two and three dimensions. However, the most important differences occur between one and two dimensions, where the difference is dramatic, whereas the difference between two and three dimension is more of a degree that we can easily handle. Actually, the percolation problem becomes simpler again in higher dimensions. In two dimensions, it is possible to go around a hole, and still have connectivity. But is it not possible to have connectivity of both the pores and the solid in the same direction at the same time – this is possible in three dimensions: A two-dimensional creature would have problems with having a digestive tract, as it would divide the creature in two, but in three dimensions this is fully possible. Here, we will therefore focus on two and three-dimensional systems.

In this book, we will first address percolation in one and infinite dimensions, since we can solve the problems exactly in these cases. We will then address percolation in two dimensions - where there is no exact solutions. However, we will see that if we assume that the cluster density function has a particular scaling form, we can still address the problem in two dimension, and make powerful predictions. We will also see that close to the percolation threshold, the porous medium has a self-affine scaling structure - it is a fractal. This property has important consequences for the physical properties of random systems. We will also see how this is reflected in a systematic change of scales, a renormalization procedure, which is a general tool that can applied to rescaling in many areas.

1.1 Basic concepts in percolation

Let us initially study a specific example of a random medium. We will generate an $L \times L$ lattice of points that are occupied with probability p. This corresponds to a coarse-grained porous medium with a porosity $\phi = p$, if we assume that the occupied sites are considered to be holes in the porous material.

We can generate a realization of a square $L \times L$ system in matlab using

```
L = 20;
p = 0.5;
z = rand(L,L);
m = z<p;
imagesc(m);
```

The resulting matrix is illustrated in fig. **??**. However, this visualization does not provide us with any insight into the connectivity of the sites in this system. Let us instead analyze the connected regions in the system.

Definitions

- two sites are **connected** if they are nearest neighbors (4 neighbors on square lattice)
- a **cluster** is a set of connected sites
- a cluster is **spanning** if it spans from one side to the opposite side
- a cluster that is spanning is called the **spanning cluster**
- a system is **percolating** if there is a spanning cluster in the system

Fortunately, there are built-in functions in matlab and python that finds connected regions in an image. The function `bwlabel` finds clusters based on a given connectivity. For example, with a connectivity corresponding to 4 we find

```
[lw,num] = bwlabel(m,4);
```

This function returns the matrix `lw`, which for each site in the original array tells what cluster it belongs to. Clusters are numbered sequentially, and each cluster is given an index. All the sites with the same index belongs to the same cluster. The resulting array is shown in fig. 1.3, where the index for each site is shown and a color is used to indicate the various clusters. Notice that there is a distribution of cluster sizes, but no cluster is large enough to reach from one side to another, and as a result the system does not percolate.

In order to visualize the clusters effectively, we give the various clusters different colors. This is done by the function `label2rgb`:

```
img = label2rgb(lw','jet','k','shuffle);
image(img);
```

Here, we have specified that we use the color map called `jet`. However, we have also specified that the zero value will be colored black – this is done by including the `k`. Finally, we have specified that the color map needs to be in random order. This is important because of a particular property of the underlying algorithm: Clusters are indexed starting from the bottom-left of the matrix. Hence, clusters that are close to each

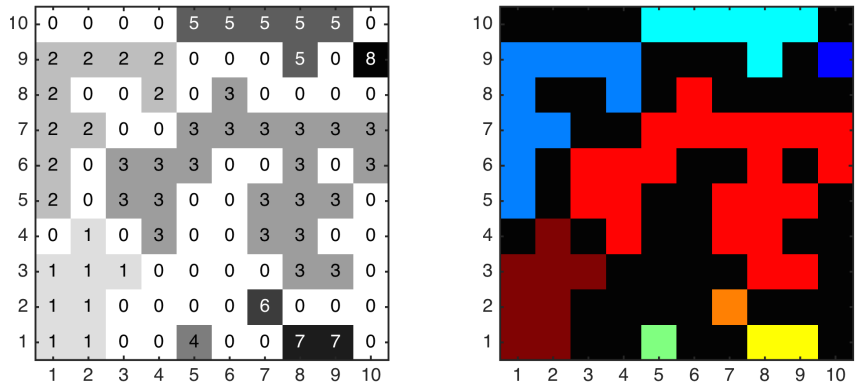

Fig. 1.3 Illustration of the index array returned by the `bwlabel` function for a 10×10 system for $p = 0.45$.

other will get similar colors and therefore be difficult to discern unless we shuffle the colormap. (Try removing the `shuffle` command and see what happens). The resulting image is shown to the right in fig. 1.3. (Notice that in these figures we have reversed the ordering of the y-axis. Usually, the first row is in the top-right corner in your plots – and this will also be the case in most of the following plots).

Let us now study the effect p on the set of connected clusters. We vary the value of p for the same underlying random matrix, and plot the resulting images:

```
L = 100;
pv = [0.2 0.3 0.4 0.5 0.6 0.7];
z = rand(L,L);
for i = 1:length(pv)
    p = pv(i);
    m = z<p;
    [lw,num] = bwlabel(m,4);
    mat = lw;
    img = label2rgb(lw,'jet','k','shuffle');
    subplot(2,3,i)
    tit=sprintf('p=%0.5g',p);
    image(img); axis square;
    title(tit);
    axis off
end
```

Fig. 1.4 shows the clusters for a 100×100 system for p ranging from 0.2 to 0.7 in steps of 0.1. We see that the clusters increase in size as p increases, but at $p = 0.6$, there is just one large cluster spanning the entire region. We have a *percolating cluster*, and we call this cluster that spans the system the **spanning cluster**. However, the transition is very rapid from $p = 0.5$ to $p = 0.6$. We therefore look at this region in more detail in fig. **??**. We see that the size of the largest cluster increases rapidly as p reaches a value around 0.6, which corresponds to p_c for this

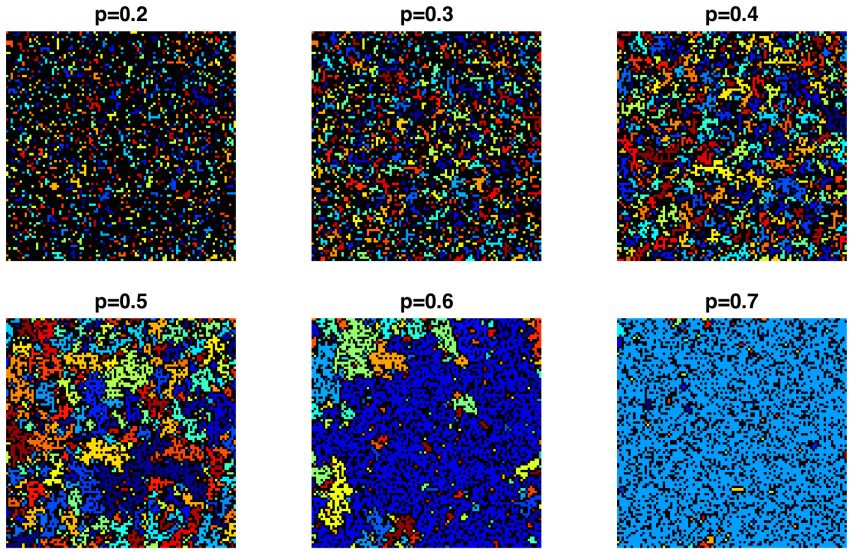

Fig. 1.4 Plot of the clusters in a 100×100 system for various values of p.

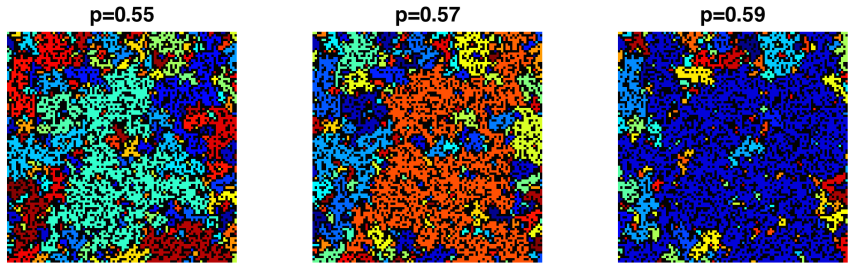

Fig. 1.5 Plot of the clusters in a 100×100 system for various values of p.

system. At this point, the largest cluster spans the entire system. For the two-dimensional system illustrated here we know that in an infinite lattice the percolation threshold is $p_c \simeq 0.5927$.

The aim of this book is to develop a theory to describe how this random porous medium behaves close to p_c. We will characterize properties such as the density of the spanning cluster, the geometry of the spanning cluster, and the conductivity and elastic properties of the spanning cluster. We will address the distribution of cluster sizes and how various parts of the clusters are important for particular physical processes. We start by characterizing the behavior of the spanning cluster near p_c.

1.2 Percolation probability

When does the system percolate? When there exists a path conncting one side to another. This occurs at some value $p = p_c$. However, in a finite system, like the system we simulated above, the value of p_c for a given realization will vary with each realization. It may be slightly above or

slightly below the p_c we find in an infinite sample. Later, we will develop a theory to understand how the effective p_c in a finite system varies from the thermodynamic p_c in an infinitely large system. But already now, we realize that as we perform different experiments, we will measure various values of p_c. We can characterize this behavior by introducing a probability $\Pi(p, L)$:

Percolation probability

The percolation probability $\Pi(p, L)$ is the probability for there to be a connected path from one side to another side as a function of p in a system of size L.

We can measure $\Pi(p, L)$ in a finite sample of size $L \times L$, by generating many random matrices. For each matrix, we perform a cluster analysis for a sequence of p_i values. For each p_i we find all the clusters, and pick out the cluster with the largest extent. If this extent is equal to the system size, there is a spanning cluster, and the system percolates, and we count up how many times a system percolates for a given p_i, N_i, and then divide by the total number of experiment, N to estimate the probability for percolation for a given p_i, $\Pi(p_i, L) \simeq N_i/N$. We implement this as follows. First, we generate a sequence of p_i values from 0.35 to 1.0 in steps of 0.01:

```
p = (0.35:0.01:1.0);
```

Then we prepare an array for N_i with the same number of elements as p_i:

```
nx = length(p);
Pi = zeros(nx,1);
```

We will generate $N = 1000$ samples:

```
N = 1000;
```

We will then loop over all samples, and for each sample we generate a new random matrix. The for each value of p_i we perform the cluster analysis using `bwlabel` as we did above. However, we now need to extract more information from the clusters. This is done by the function `regionprops(lw,'BoundingBox')`, which returns a box enclosing the clusters. For example, for the 6×6 simulation in fig. 1.6, the bounding boxes are:

```
s = regionprops(lw,'BoundingBox');
bbox = cat(1,s.BoundingBox)

bbox =

    0.5000 2.5000 3.0000 3.0000
    0.5000 5.5000 1.0000 1.0000
    2.5000 0.5000 1.0000 1.0000
    4.5000 0.5000 1.0000 2.0000
    4.5000 4.5000 1.0000 1.0000
    5.5000 2.5000 1.0000 1.0000
```

The bounding boxes are given as the top left corner (first two columns) and then the width and height (last two columns). We can therefore simply find the maximum of the two last colums – this corresponds to the maximum extent of any of the clusters

```
max(max(bbox(:,[3 4])))

ans =
     3
```

Now, we are ready to implement this into a complete program:

```
p = (0.4:0.01:1.0);
nx = length(p);
Ni = zeros(nx,1);
N = 10;
L = 100;
for i = 1:N
    z = rand(L,L);
    for ip = 1:nx
        m = z<p(ip);
        [lw,num] = bwlabel(m,4);
        s = regionprops(lw,'BoundingBox');
        bbox = cat(1,s.BoundingBox);
        maxsize = max(max(bbox(:,[3 4])));
        if (maxsize==L) % Percolation
            Ni(ip) = Ni(ip) + 1;
        end
    end
end
Pi = Ni/N;
plot(p,Pi);
```

The resulting plot of $\Pi(p, L)$ is seen in fig. 1.7. The figure shows the resulting plots as a function of system size L. We see that as the system size increases, $\Pi(p, L)$ approaches a step function at $p = p_c$.

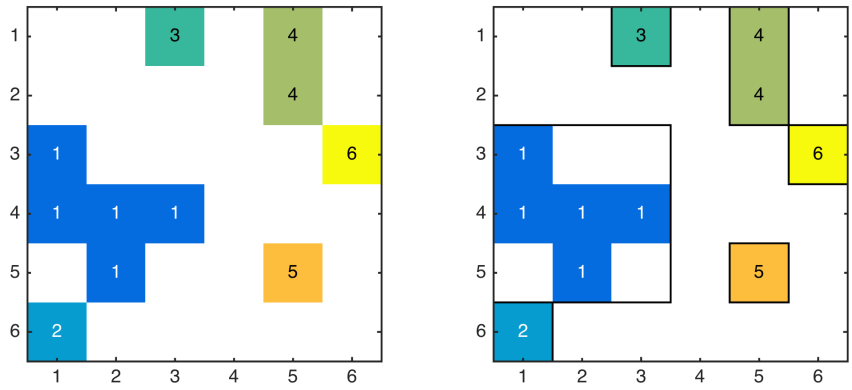

Fig. 1.6 Illustration of the BoundingBox for the clusters in a 6×6 simulation.

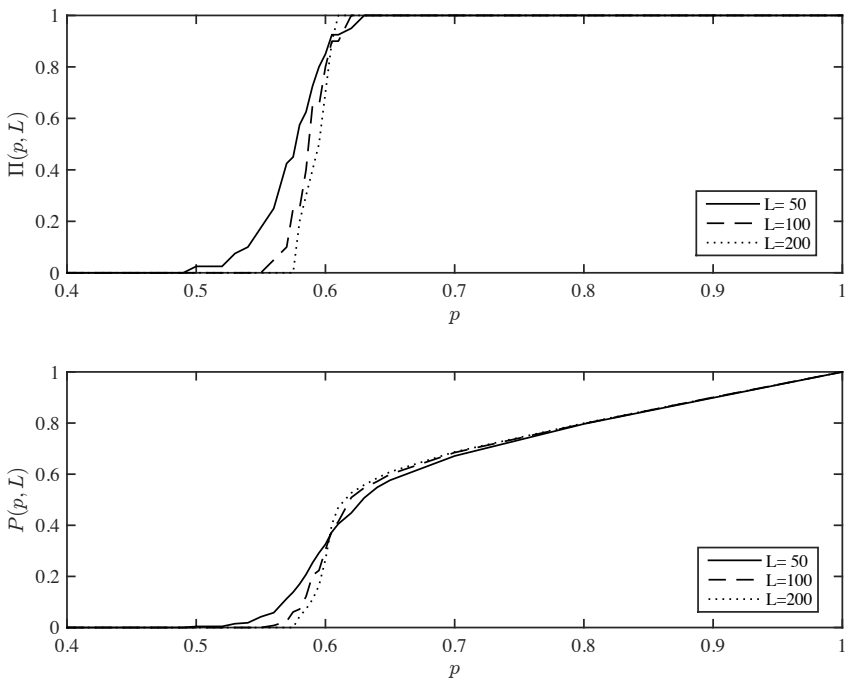

Fig. 1.7 Plot of $\Pi(p, L)$, the probability for there to be a connected path from one side to anther, as a function of p for various system sizes L.

1.3 Spanning cluster

The probability $\Pi(p, L)$ described the probability for there to be a spanning cluster, but what about the spanning cluster itself, how can we characterize it? We see from fig. 1.4 that the spanning cluster grows quickly around $p = p_c$. Let us therefore characterize the cluster by its size, M_S, or by its density, $P(p, L) = M_S/L^2$, which also corresponds to the probability for a site to belong the spanning cluster.

Density of the spanning cluster

The probability $P(p, L)$ for a site to belong to a spanning cluster is called the *density of the spanning cluster*, or the order parameter for the percolation problem.

We can measure $P(p, L)$ by couting the mass M_i of the spanning cluster as a function of p_i for various values of p_i. We can find the mass of the spanning cluster, by finding a cluster that spans the system (there may be more than one) as we did above, and then measuring the number of sites in the cluster using the `Area` property of `regionprops(lw,'Area')`.

First, we find a list of clusters that span the system. These are the clusters that have an extent in the x or the y direction which are equal to L, the system size. We find these using the `find` command, which returns an array with a list of all the cluster numbers that span the system. The list is empty of no clusters span, and may contain more than one element if more than one element spans. We need to find clusters that span either in the x-direction or in the y-direction, which is the union of the arrays of elements that span in the respective direction, as illustrated in this code:

```
[lw,num] = bwlabel(m,4);
s = regionprops(lw,'BoundingBox');
bbox = cat(1,s.BoundingBox);
jx = find(bbox(:,3)==L);
jy = find(bbox(:,4)==L);
j = union(jx,jy);
if (length(j)>0) % Percolation
   Ni(ip) = Ni(ip) + 1;
   for jj = 1:length(j)
      Mass(ip) = Mass(ip) + area(j(jj));
   end
end
```

Here, we loop through all the clusters that span, and include the mass of each cluster in the total mass of the spanning cluster. This is one possible definition of the spanning cluster – you could also have selected only one of these, for example the largest, to be the spanning cluster.

We implement these features in the following program, which measures both $\Pi(p, L)$ and $P(p, L)$ for a given value of L:

```
p = (0.4:0.01:1.0);
nx = length(p);
Ni = zeros(nx,1);
Mi = zeros(nx,1);
N = 10;
L = 100;
for i = 1:N
    z = rand(L,L);
```

```
    for ip = 1:nx
        m = z<p(ip);
        [lw,num] = bwlabel(m,4);
        s = regionprops(lw,'BoundingBox');
        bbox = cat(1,s.BoundingBox);
        s = regionprops(lw,'Area');
        area = cat(1,s.Area);
        jx = find(bbox(:,3)==L);
        jy = find(bbox(:,4)==L);
        j = union(jx,jy);
        if length(j)>0 % Percolation
            Ni(ip) = Ni(ip) + 1;
            for jj = 1:length(j)
                Mi(ip) = Mi(ip) + area(j(jj));
            end
        end
    end
end
subplot(2,1,1)
Pi = Ni/N;
plot(p,Pi); xlabel('p'),ylabel('\Pi')
subplot(2,1,2)
P = Mi/(N*L*L);
plot(p,P); xlabel('p'),ylabel('P')
```

The resulting plot of $P(p, L)$ is shown in the bottom of fig. 1.7. We see that $P(p, L)$ changes rapidly around $p = p_c$ and that it grows slowly – approximately linearly – as $p \to 1$. We can understand this linear behavior: When p is near 1 all the set sites are connected and part of the spanning cluster. The density of the spanning cluster is therefore proportional to p in this limit. We will now develop a theory for the observations of $\Pi(p, L)$, $P(p, L)$ and other features of the percolation system. First, we see what insights we can gain from small, finite systems.

1.4 Percolation in small systems

First, we will address the two-dimensional system directly. We will study a $L \times L$ system, and the various physical properties of it. We will start with $L = 1$, $L = 2$, and $L = 3$, and then try to generalize.

First, let us address $L = 1$. In this case, the system percolates if the site is present, which has a probability p, hence the percolation probability is $\Pi(p, 1) = p$. The probability for a site to belong to the spanning cluster is p, therefore $P(p, 1) = 1$.

Then, let us examine $L = 2$. This is still simple, but we now have to develop a more advanced strategy than for $L = 1$. Our strategy will be to list all possible outcomes, find the probability for each outcome, and then use this to find the probability for the various physical properties we are interested in. The possible configurations are listed in fig. 1.8.

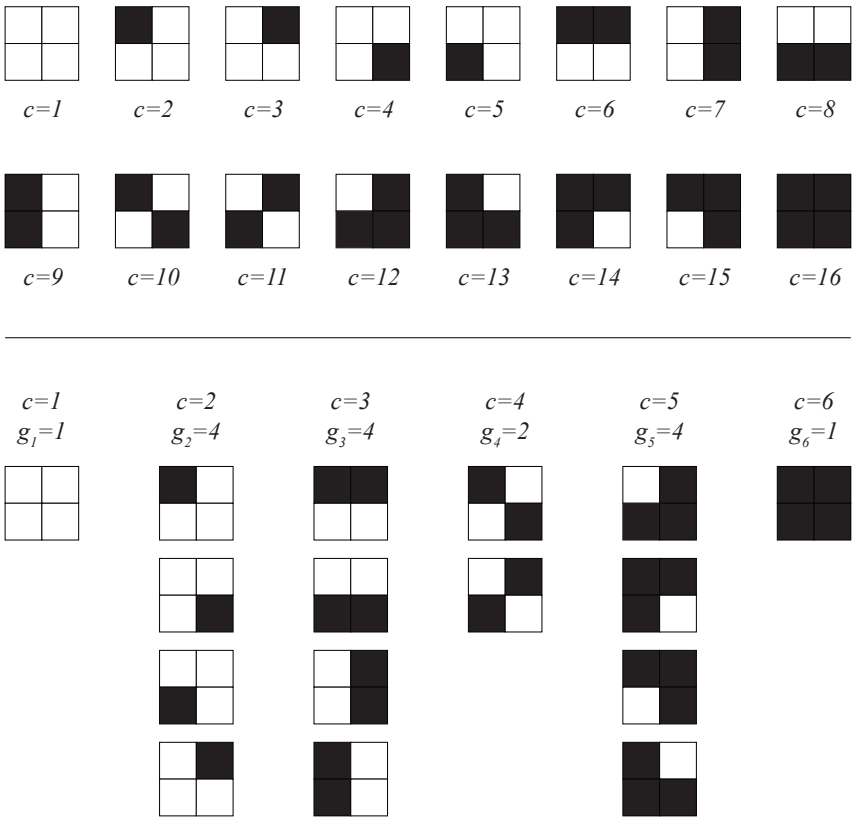

Fig. 1.8 The possible configurations for a $L = 2$ site percolation lattice in two-dimensions. The configurations are indexed using the cluster configuration number c.

The strategy is to use a basic result from probability theory: If we want to calculate the probability of an event A, we can do this by summing the probability of A given B multiplied by the probability for B over all possible outcomes B.

$$P(A) = \sum_B P(A|B)P(B) \ , \tag{1.1}$$

where we have used the notation $P(A|B)$ to denote the probability of A given that B occurs . We can use this to calculate properties such as Π and $P(p, L)$ by summing over all possible configurations c:

$$\Pi(p, L) = \sum_c \Pi(p, L|c)P(c) \ , \tag{1.2}$$

where $\Pi(p, L|c)$ is the value of Π for the particular configuration c, and $P(c)$ is the probability of this configuration.

The configurations for $L = 2$ have been numbered from $c = 1$ to $c = 16$ in fig. 1.8. However, configurations that are either mirror images or rotations of each other will have the same probability and the same physical properties since percolation can take place both in the x and the

y directions. It is therefore only necessary to group the configurations into 6 different classes as illustrated in the bottom of fig. 1.8, but we then need to remember the multiplicity, g_c, for each class when we calculate probabilities. Let us make table of the configurations, the number of such configurations, the probability of *one* such configuration, and the value of $\Pi(p, L|c)$ for this configuration.

c	g_c	$P(c)$	$\Pi(p, L\|c)$
1	1	$p^0(1-p)^4$	0
2	4	$p^1(1-p)^3$	0
3	4	$p^2(1-p)^2$	1
4	2	$p^2(1-p)^2$	0
5	4	$p^3(1-p)^1$	1
6	1	$p^4(1-p)^0$	1

We should check that we have actually listed all possible configurations. The total number of configurations is $2^4 = 16 = 1 + 4 + 2 + 4 + 4 + 1$, which is ok.

We can then find the probability for Π directly:

$$\Pi = 0 \cdot 1 \cdot p^0(1-p)^4 + 0 \cdot 4 \cdot p^1(1-p)^3 + 1 \cdot 4 \cdot p^2(1-p)^2 \tag{1.3}$$

$$+ 0 \cdot 2 \cdot p^2(1-p)^2 + 1 \cdot 4 \cdot p^3(1-p)^1 + 1 \cdot 1 \cdot p^4(1-p)^0 \, . \tag{1.4}$$

We therefore find the exact value for $\Pi(p, L = 2)$:

$$\Pi(p, L = 2) = 4p^2(1-p)^2 + 4p^3(1-p)^1 + p^4(1-p)^0 \, , \tag{1.5}$$

which we can simplify further if we want. The shape of $\Pi(p, L)$ for $L = 1$, and$L = 2$ is shown in fig. 1.9.

We could characterize $p = p_c$ as the number for which $\Pi = 1/2$, which would give $p_c(L = 2) =$, which is better than for $L = 1$, for which we got $p_c(L = 1) = 1/2$. Maybe we can just continue doing this type of calculation for higher and higher L and we will get a better and better approximation for p_c?

We notice that for finite L, $\Pi(p, L)$ will be a polynomial of order $o = L^2$ - it is in principle a function we can calculate. However, the number of possible configurations is 2^{L^2} which increases very rapidly with L. It is therefore not realistic to use this technique for calculating the percolation probabilities. We will need to have more powerful techniques, or simpler problems, in order to perform exact calculations.

However, we can still learn much from a discussion of finite L. For example, we notice that

$$\Pi(p, L) \simeq Lp^L + c_1 p^{L+1} + \ldots + c_n p^{L^2} \, , \tag{1.6}$$

in the limit of $p \ll 1$. The leading order term when $p \to 0$ is therefore Lp^L.

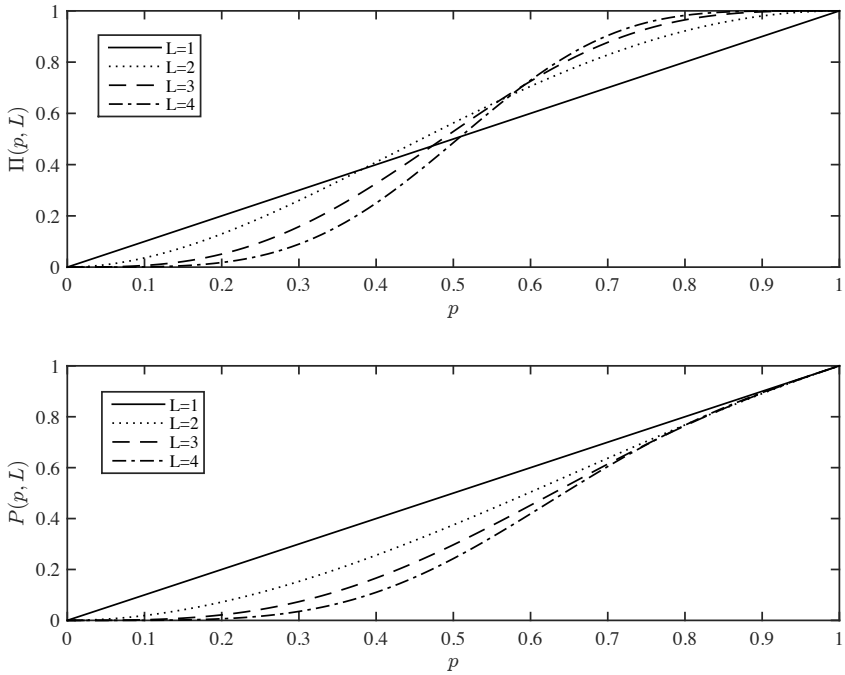

Fig. 1.9 Plot of $\Pi(p, L)$ for $L = 1$ and $L = 2$ as a function of p.

Similarly, we find that for $p \to 1$, the leading order term is approximately

$$\Pi(p, L) \simeq 1 - (1 - p)^L \, . \tag{1.7}$$

These two results gives us an indication about how the percolation probability $\Pi(p, L)$ is approaching the step function when $L \to \infty$.

Similarly, we can calculate $P(p, L)$ for $L = 2$. However, we leave the calculation of the $L = 3$ and the $P(p, L)$ system to the exercises.

1.5 Exercises

Exercise 1.1: Percolation for $L = 3$

a) Find $P(p, L)$ for $L = 1$ and $L = 2$.

b) Categorize all possible configurations for $L = 3$.

c) Find $\Pi(p, L)$ and $P(p, L)$ for $L = 3$.

Exercise 1.2: Counting configurations in small systems

a) Write a program to find all the configurations for $L = 2$.

b) Use this program to find $\Pi(p, L = 2)$ and $P(p, L = 2)$. Compare with the exact results from the previous exercise.

c) Use you program to find $\Pi(p, L)$ and $P(p, L)$ for $L = 3, 4$ and 5.

Exercise 1.3: Percolation in small systems in 3d

In this exercise we will study the three-dimensional site percolation system for small system sizes.

a) How many configurations are there for $L = 2$?

b) Categorize all possible configurations for $L = 2$.

c) Find $\Pi(p, L)$ and $P(p, L)$ for $L = 2$.

d) Compare your results with your result for the two-dimensional system. Comment on similarities and differences.

2 One-dimensional percolation

The percolation problem can be solved exactly in two limits: in the one-dimensional and the infinite dimensional cases. Here, we will first address the one-dimensional system. While the one-dimensional system does not allow us to study the full complexity of the percolation problem, many of the concepts and measures introduced to study the one-dimensional problem can be generalized to higher dimensions.

2.1 Percolation probability

Let us first address a one-dimensional lattice of L sites. In this case, there is a spanning cluster if and only if all the sites are occupied. If only a single site is empty, there will not be any connecting path from one side to the other. The percolation probability is therefore

$$\Pi(p, L) = p^L \tag{2.1}$$

This has a trivial behavior when $L \to \infty$

$$\Pi(p, \infty) = \begin{cases} 0 \; p < 1 \\ 1 \; p = 1 \end{cases} . \tag{2.2}$$

This shows that the percolation threshold is $p_c = 1$ in one dimension. However, the one-dimensional system is anomalous, and higher dimensions, we will always have $p_c < 1$, so that we can study the system both above and below p_c. Unfortunately, for the one-dimensional system we can only study the system below p_c.

2.2 Cluster number density

2.2.1 Definition of cluster number density

In the simulations in fig. 1.4 we saw that the percolation system was characterized by a wide distribution of clusters – regions of connected sites. The clusters have varying shape and size. If we increase p to approach p_c we saw that the clusters increased in size until they reached the system size. We can use the one-dimensional system to learn more about the behavior of clusters as p approaches p_c.

Fig. 2.1 illustrates a realization of an $L = 16$ percolation system in one dimension below $p_c = 1$. In this case there are 5 clusters of sizes 1,1,4,2,1 measured in the number of sites in each cluster. The clusters are numbered - indexed - from 1 to 5 as we did for the numerical simulations in two dimensions. How can we characterize the clusters in a system? In percolation theory we characterize cluster sizes by asking a particular question: If you point at a (random) site in the lattice, what is the probability for this site to belong to a cluster of size s?

$$P(\text{site is part of cluster of size } s) = sn(s,p)\ . \tag{2.3}$$

It is common to use the notation $sn(s,p)$ for this probability for a given site to belong to a cluster of size s. Why is it divided into two parts, s and $n(s,p)$? Because we must divide the question into two parts: (1) What is the probability for a given site to be a *specific site* in a cluster of size s, and (2) how many such specific sites are there? What do we mean by a specific site? For cluster number 3 in fig. 2.1 there are 4 sites. We could therefore ask the question, what is the probability for a site to be the left-most site in a cluster of size s. This is what we mean with a specific site. We could ask the same question about the second left-most, the third left-most and so on. We call the probability for a site to belong to a specific site in a cluster of size s (such as the left-most site in the cluster) the **cluster number density**, and we use the notation $n(s,p)$ for this. To find the probability $sn(s,p)$ for a site to belong to any of the s sites in a cluster of size s we must sum the probabilities for each of the specific sites. This is illustrated for the case of a cluster of size 4:

$$\begin{aligned} &P(\text{site to be in cluster of size 4}) \\ &\quad = P(\text{site to be left-most site in cluster of size 4}) \\ &\quad + P(\text{site to be second left-most site in cluster of size 4}) \\ &\quad + P(\text{site to be third left-most site in cluster of size 4}) \\ &\quad + P(\text{site to be fourth left-most site in cluster of size 4}) \\ &\quad = 4P(\text{site to be left-most site in cluster of size 4}) \end{aligned}$$

; .

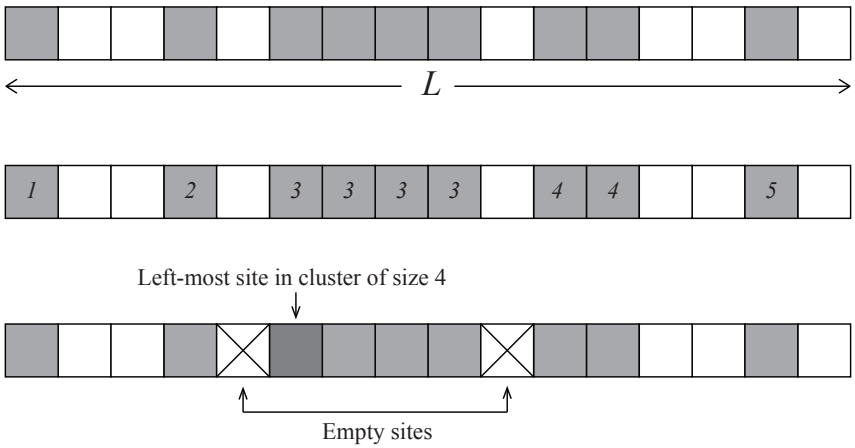

Fig. 2.1 Realization of a $L = 16$ percolation system in one dimension. Occupied sites are marked with black squares.

Because each of these probabilities are the same. What is the probability for a site to be the left-most site in a cluster of site s in one dimension? In order for it to be in a cluster of size s, the site must be present, which has probability p, and then $s-1$ sites must also be present to the right of it, which has probability p^{s-1}. In addition, the site to the left must be empty (illustrated by an X in fig. 2.1 bottom part), which has probability $(1-p)$ and the site to the right of the fourth site (illustrated by an X in fig. 2.1 bottom part), which also has probability $(1-p)$. Since the occupation probabilities for each site are independent, the probability for the site to be the left-most site in a cluster of size s is:

$$n(s,p) = (1-p)^2 p^s \ . \tag{2.4}$$

This is the cluster number density in one dimension.

Cluster number density

The cluster number density $n(s,p)$ is the probability for a site to be a particular site in a cluster of size s. For example, in 1d, $n(s,p)$ is the probability for a site to be the left-most site in a cluster of size s.

We should check that $sn(s,p)$ really is a normalized probability. How should it be normalized? We know that if we point at a random site in the system, the probability for that site to be occupied is p. An occupied site is then either a part of a finite cluster of some size s or it is part of the inifinite cluster. The probability for a site to be a part of the infinite cluster is P. This means that we have the following normalization condition:

Normalization of the cluster number density

A site is occupied with probability p. An occupied site is either part of a finite cluster of size s with probability $sn(s,p)$ or it is part of the infinite (spanning) cluster with probability P:

$$p = \sum_{s=1}^{\infty} sn(s,p) + P \; . \tag{2.5}$$

Let us check that this is indeed the case for the one dimensional result we have found by calculating the sum:

$$\sum_{s=1}^{\infty} sn(s,p) = \sum_{s=1}^{\infty} sp^s(1-p)^2 = (1-p)^2 p \sum_{s=1}^{\infty} sp^{s-1} \; , \tag{2.6}$$

where we will now employ a common trick:

$$\sum_{s=1}^{\infty} sp^{s-1} = \frac{d}{dp} \sum_{s=0}^{\infty} p^s = \frac{d}{dp} \frac{1}{1-p} = (1-p)^{-2} \; , \tag{2.7}$$

which gives

$$\sum_{s=1}^{\infty} sn(s,p) = (1-p)^2 \, p \sum_{s=1}^{\infty} sp^{s-1} = (1-p)\, p\, (1-p)^{-2} = p \; . \tag{2.8}$$

Since $P = 0$ when $p < 0$ we see that the probability is normalized. We can use similar tricks to calculate moments of any order.

2.2.2 Measuring the cluster number density

In order to gain further insight into the distribution of cluster sizes, let us look study fig. 2.1 in more detail. There are 3 clusters of size $s = 1$, one cluster of size $s = 2$, and one cluster of size $s = 4$. We could therefore introduce a histogram of cluster sizes, which is what we would do if we studied the cluster distribution numerically. Let us write N_s as the number of clusters of size s.

s	N_s	$n(s,p)$
1	3	3/16
2	1	1/16
3	0	0/16
4	1	1/16

How can we now estimate $sn(s,p)$, the probability for a given site to be part of a cluster of size s, from N_s? The probability for a site to belong

to cluster of size s can be estimated by the number of sites belonging to a cluster of size s divided by the total number of sites. The number of sites belonging to a cluster of size s is sN_s, and the total number of sites is L^d, where L is the system size and d is the dimensionality. (Here, $d = 1$). This means that we can estimate the probability $sn(s,p)$ from

$$\overline{sn(s,p)} = \frac{sN_s}{L^d} \,, \tag{2.9}$$

where we use a bar to show that this is an estimated quantity and not the actual probability. We divide by s on both sides, and find

$$\overline{n(s,p)} = \frac{N_s}{L^d} \,. \tag{2.10}$$

This argument and the result is valid in any dimension, not only for $d = 1$. We have therefore found a method to estimate the cluster number density:

Measuring the cluster number density

We can measure $n(s,p)$ in a simulation by measuring N_s, the number of clusters of size s, and then calculate $n(s,p)$ from

$$\overline{n(s,p)} = \frac{N_s}{L^d} \,. \tag{2.11}$$

For the clusters in fig. 1.8 we find that

$$\overline{n(1,p)} = \frac{N_1}{L^1} = \frac{3}{16} \,, \tag{2.12}$$

$$\overline{n(2,p)} = \frac{N_2}{L^1} = \frac{1}{16} \,, \tag{2.13}$$

$$\overline{n(3,p)} = \frac{N_3}{L^1} = \frac{0}{16} \,, \tag{2.14}$$

$$\overline{n(4,p)} = \frac{N_4}{L^1} = \frac{1}{16} \,, \tag{2.15}$$

which is our estimate of $n(s,p)$ based on this single realization.

We check the consistency of the result by ensuring that the estimated probabilities also are normalized:

$$\sum_s \overline{sn(s,p)} = 1 \cdot \frac{3}{16} + 2 \cdot \frac{1}{16} + 3 \cdot 0 + 4 \cdot \frac{1}{16} = \frac{9}{16} = \overline{p} \,, \tag{2.16}$$

where $\overline{p}$ is estimated from number of present sites divided by the total number of sites.

In order to produce good statistical estimates for $n(s,p)$ we must sample from many random realization of the system. If we sample from M realizations, and then measure the total number of clusters of size s, $N_s(M)$, summed over all the realizations, we estimate the cluster number density from

$$\overline{n(s,p)} = \frac{N_s(M)}{ML^d} \ . \tag{2.17}$$

Notice that all simulations are for finite L, and we would therefore expect deviations due to L as well as randomness due to the finite number of samples. However, we expect the estimated $\overline{n(s,p;L)}$ to approach the underlying $n(s,p)$ as M and L approaches infinity.

2.2.3 Shape of the cluster number density

We found that the cluster number density in one dimension is

$$n(s,p) = (1-p)^2 p^s \ . \tag{2.18}$$

In fig. 2.2 we have plotted $n(s,p)$ for various values of p. In order to compare see the s-dependence of the plot directly for various p-values we plot

$$G(s) = (1-p)^2 \, n(s,p) = p^s \ , \tag{2.19}$$

as a function of s. We notice that $(1-p)^2 n(s,p)$ is approximately constant for a wide range of s, and then falls off rapidly for some characteristic value s_ξ which increases as p approaches $p_c = 1$. We can understand this behavior better by rewriting $n(s,p)$ as

$$n(s,p) = (1-p)^2 e^{s \ln p} = (1-p)^2 e^{-s/s_\xi} \ , \tag{2.20}$$

where we have introduced the cut-off cluster size

$$s_\xi = \frac{-1}{\ln p} \ . \tag{2.21}$$

What we are seeing in fig. 2.2 is therefore the exponential cut-off curve, where the cut-off $s_\xi(p)$ increases as $p \to 1$. We call it a *cut-off* because the value of $n(s,p)$ decays very rapidly (exponentially) when s is larger than s_ξ.

How does s_ξ depend on p?. We see from eq. 2.21 that as p approaches $p_c = 1$, the characteristic cluster size s_ξ will diverge. The form of the divergence can be determined in more detail through a Taylor expansion:

$$s_\xi = -\frac{1}{\ln p} \tag{2.22}$$

when p is close to 1, $1 - p \ll 1$ and we can write

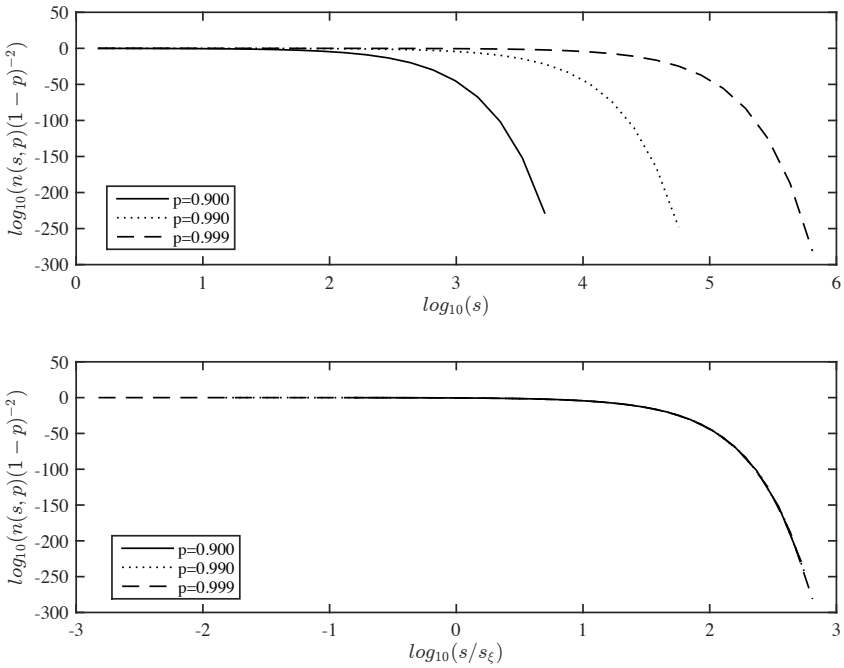

Fig. 2.2 (Top) A plot of $n(s,p)(1-p)^2$ as a function of s for various values of p for a one-dimensional percolation system shows that the cut-off increases as a function of s. (Bottom) When the s axis is rescaled by s/s_ξ all the curves fall onto a common scaling function, that is, $n(s,p) = (1-p)^2 F(s/s_\xi)$.

$$\ln p = \ln(1-(1-p)) \simeq -(1-p)\ , \tag{2.23}$$

where we have used that $\ln(1-x) = -x + o(x^2)$, which is simply the Taylor expansion of the logarithm. As a result

$$s_\xi \simeq \frac{1}{1-p} = \frac{1}{p_c - p} = |p - p_c|^{-1/\sigma}\ . \tag{2.24}$$

This shows that the divergence of s_ξ as p approaches p_c is a power-law with exponent -1. This is a feature which is general in percolation theory.

Scaling behavior of the characteristic cluster size

The characteristic clustes size s_ξ diverges as

$$s_\xi \propto |p - p_c|^{-1/\sigma}\ , \tag{2.25}$$

when $p \to p_c$. In one dimension, $\sigma = 1$.

The value of the exponent σ depends on the lattice dimensionality, but it does not depend on the details of the lattice. It would, for example, be the same also for next-nearest neighbor connectivity - a problem we leave for the reader to solve as an exercise.

The functional form we have found is also an example of a **data collapse**. We see that if we plot $(1-p)^{-2}n(s,p)$ as a function of s/s_ξ, all data-points for various values of p should fall onto a single curve, as illustrated in fig. 2.2:

$$n(s,p) = (1-p)^2 e^{-s/s_\xi} \ , \tag{2.26}$$

This is what we call a data-collapse. We have one behavior for small s and then a rapid cut-off when s reaches s_ξ. We can rewrite $n(s,p)$ so that all the s_ξ dependence is in the cut-off function by realizing that since $s_\xi \simeq (1-p)^{-1}$ we have that $(1-p)^2 = s_\xi^{-2}$. This gives

$$n(s,p) = s_\xi^{-2} e^{-s/s_\xi} = s^{-2} \left(\frac{s}{s_\xi}\right)^2 e^{-\frac{s}{s_\xi}} = s^{-2} F\left(\frac{s}{s_\xi}\right) \ . \tag{2.27}$$

where $F(u) = u^2 e^{-u}$. We will see later that this form is general – it is valid for percolation in any dimesion, although with other values for the exponent -2. In parcoluation theory, we call this exponent τ:

$$n(s,p) = s^{-\tau} F(s/s_{xi}) \ , \tag{2.28}$$

where $\tau = 2$ in two dimensions. The exponent τ is another example of a universal exponent that does not depend on details such as the connectivity rule, but it does depend on the dimensionality of the system.

2.2.4 Numerical measurement of the cluster number density

Let us now test the measurement method and the theory through a numerical study of the cluster number density. According to the theory developed above we can estimate the cluster number density $n(s,p)$ from

$$\overline{n(s,p)} = \frac{N_s(M)}{L^2 \cdot M} \ , \tag{2.29}$$

where $N_s(M)$ is the number of clusters of size s measured in M realizations of the percolation system. We generate a one-dimensional percolation system and index the clusters using

```
L = 20;
p = 0.90;
z = rand(L,1);
m = z<p;
[lw,num] = bwlabel(m,4);
```

Now, `lw` contains the indecies for all the clusters. We can extract the size of the clusters using the `Area`-property of the `regionprops` command:

```
s = regionprops(lw,'Area');
area = cat(1,s.Area);
```

The resulting list of areas for one sample is

```
>> lw'

ans =
  Columns 1 through 20

   1 0 2 0 3 3 3 3 0 0 4 0 0 5 5 5 5 0 6 0

>> area'

ans =

   1 1 4 1 4 1
```

We need to collect all the areas of all the clusters for many realizations, and then calculate the number of cluster of each size s based on this long list of areas. This is all brought together by continuously appending the `area`-array to the end of an array `allarea` that contains the areas of all the clusters.

```
nsamp = 1000;
L = 1000;
p = 0.90;
allarea = [];
for i = 1:nsamp
    z = rand(L,1);
    m = z<p;
    [lw,num] = bwlabel(m,4);
    s = regionprops(lw,'Area');
    area = cat(1,s.Area);
    allarea = cat(1,allarea,area);
end
[n,s]=hist(allarea,L);
nsp = n/(L*nsamp);
sxi = -1.0/log(p);
nsptheory = (1-p)^2*exp(-s/sxi);
i = find(n>0);
plot(s(i),nsp(i),'ok',s,nsptheory,'-k');
xlabel('s'); ylabel('n(s,p)');
```

This scipt also calculates N_s using the histogram function with enough bins to ensure that there is at least one bin for each value of s:

```
[n,s]=hist(allarea,L);
```

And we then calculate $\overline{n(s,p)}$ from

```
nsp = n/(L*nsamp);
```

We find the theoretically predicted form for $n(s,p)$, which is $n(s,p) = (1-p)^2 \exp(-s/s_\xi)$, where $s_\xi = -1/lnp$. This is calculated for the same values of s as found from the histogram using:

```
sxi = -1.0/log(p);
nsptheory = (1-p)^2*exp(-s/sxi);
```

When we use the histogram function with many bins, we risk that many of the bins contain zero elements. To remove these elements from the plot, we can use the `find` function to find the indecies of the elements of `n` that are non-zero:

```
i = find(n>0);
```

And then we only plot the values of $\overline{n(s,p)}$ at these indicides. The values for the theoretical $n(s,p)$ are calculated for all values of s.

```
plot(s(i),nsp(i),'ok',s,nsptheory,'-k');
```

The resulting plot is shown in fig. 2.3. We see that the measured results and the theoretical values fit nicely, even though the theory is for infinite system sizes, and the simulations where performed at $L = 1000$. We also see that for larger values of s there are fewer observed values. It may therefore be a good idea to make the bins used for the histogram larger for larger values of s. We will return to this when we measure the cluster number density in two-dimensional systems in chapter 4.

2.2.5 Average cluster size

Since we have a precise form of the cluster number density, $n(s,p)$ we can use it to calculate the average cluster size. However, what do we mean by the average cluster size in this case? In percolation theory it is common to define the average cluster size as the average size of a cluster connected to a given (random) site in our system. That is, we will use the cluster number density, $n(s,p)$, as the basic distribution for calculating the moments.

Average cluster size

The average cluster size $S(p)$ is defined as

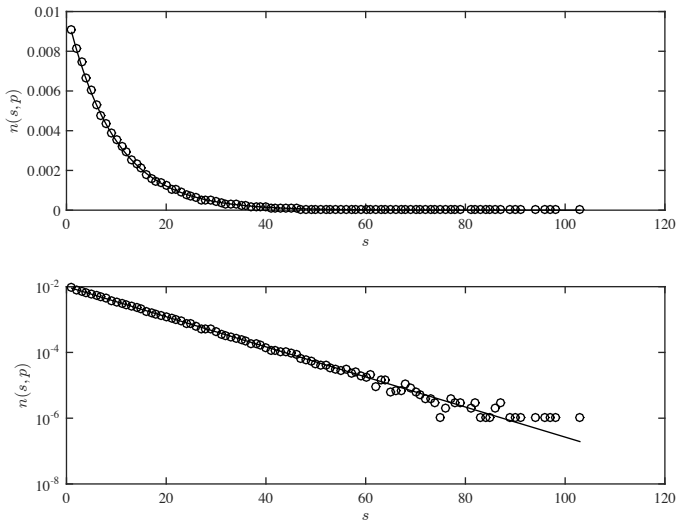

Fig. 2.3 Plot of the predicted $\overline{n(s,p)}$, based on $M = 1000$ samples of a $L = 1000$ system with $p = 0.9$, and the theoretical $n(s,p)$ curve on a linear scale (top) and a semilogarithmic scale (bottom). The semilogarithmic plot clearly shows that $\overline{n(s,p)}$ follows an exponential curve.

$$S(p) = \langle s \rangle = \sum_s s(\frac{sn(s,p)}{\sum_s sn(s,p)}) \ , \tag{2.30}$$

The normalization sum in the denominator is equal to p when $p < p_c$. We can therefore write this as

$$S(p) = \sum_s s(\frac{sn(s,p)}{p}) \ . \tag{2.31}$$

Similarly, we can define the k-th moment to be

$$S_k = \langle s^k \rangle = \sum s^k(\frac{sn(s,p)}{p}) \ . \tag{2.32}$$

Let us calculate the first moment, corresponding to $k = 1$, the average cluster size.

$$S = \frac{1}{p}\sum_s s^2 n(s,p) \tag{2.33}$$

$$= \frac{(1-p)^2}{p}\sum_s s^2 p^s \tag{2.34}$$

$$= \frac{(1-p)^2}{p}\sum_s p\frac{\partial}{\partial p}p\frac{\partial}{\partial p}p^s \tag{2.35}$$

$$= \frac{(1-p)^2}{p}p\frac{\partial}{\partial p}p\frac{\partial}{\partial p}\sum_s p^s \tag{2.36}$$

$$= \frac{(1-p)^2}{p}p\frac{\partial}{\partial p}\frac{p}{(1-p)^2} \text{ (from } \sum_s sn(s,p) \text{)} \tag{2.37}$$

$$= (1-p)^2\frac{\partial}{\partial p}\frac{p}{(1-p)^2} \tag{2.38}$$

$$= (1-p)^2\left(\frac{1}{(1-p)^2} + \frac{2p}{(1-p)^3}\right) \tag{2.39}$$

$$= \frac{1+p}{1-p} \tag{2.40}$$

where we have used the trick introduced in eq. 2.7 to move the derivation out through the sum. In addition, we have also used our previous result from $\sum_s sn(s,p)$ directly.

This shows that we can write

$$S = \frac{1+p}{1-p} = \frac{\Gamma}{|p-p_c|^\gamma}\ , \tag{2.41}$$

with $\gamma = 1$ and $\Gamma(p) = 1+p$. That is, the average cluster size also diverges as a power-law when p approaches p_c. The exponent $\gamma = 1$ of the power-law is again universal. That is, it depends on features such as dimensionality, but not on details such as the lattice structure.

Later, we will observe that we have a similar behavior for percolation in any dimension, although with other values of γ.

We will leave it as an exercise for our reader to find the behavior for higher moments, S_k, using a similar argument.

2.3 Spanning cluster

The density of the spanning cluster, $P(p;L)$, is similarly simple to find and discuss. The spanning cluster only exists for $p \geq p_c$. The discussion for $P(p;L)$ is therefore not that interesting for the one-dimensional case. However, we can still introduce some of the general notions.

The behavior of $P(p;\infty)$ in one dimension is given as

$$P(p;\infty) = \begin{cases} 0 \; p < 1 \\ 1 \; p = 1 \end{cases} . \tag{2.42}$$

We could introduce a similar finite size scaling discussion also for $P(p;L)$. However, we will here concentrate on the relation between $P(p;L)$ and the distribution of cluster sizes. The distribution of the size of a finite cluster is described by $sn(s,p)$, which is the probability that a given site belongs to a cluster of size s. If we look at a given site, that site is occupied with probability p. If a site is occupied it is either part of a finite cluster of size s or it is part of the spanning cluster. Since these two events cannot occur at the same time, the probability for a site to be set must be the sum of the probability to belong to a finite cluster and to belong to the infinite cluster. The probability to belong to a finite cluster is the sum of the probability to belong to a cluster of s for all s. We therefore have the equality:

$$p = P(p;L) + \sum_s sn(s,p;L) \; , \tag{2.43}$$

which is not only valid in the one-dimensional case, but also for percolation problems in general.

We can use this relation to find the density of the spanning cluster from the cluster number density $n(s,p)$ through

$$P(p) = p - \sum_s sn(s,p) \; . \tag{2.44}$$

This illustrates that the cluster number density $n(s,p)$ is a fundamental property, which can be used to deduce many of the other properties of the percolation system.

2.4 Correlation length

From the simulations in fig. 1.4 we see that the size of the clusters increases as $p \to p_c$. We expect a similar behavior for the one-dimensional system. We have already seen that the mass (or area) of the clusters diverges as $p \to p_c$. However, the characteristic cluster size s_ξ characterizes the mass (or area) of a cluster. How can we characterize the extent of a cluster?

To characterize the linear extent of a cluster, we find the probability for two sites at a distance r to be part of the same cluster. This probability is called the **correlation function**, $g(r)$:

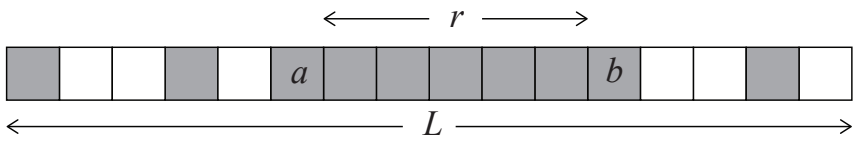

Fig. 2.4 An illustration of the distance r between two sites a and b. The two sites a and b are connected if and only if all the sites between a and b are occupied.

The correlation function $g(r)$ describes the conditional probability that two sites a and b, which both are occupied and are separated by a distance r belong to the same cluster.

For one-dimensional percolation, two sites a and b only can be part of the same cluster if all the points in between a and b are occupied. If r denotes the number of points between a and b (not counting the start and end positions) as illustrated in fig. 2.4, we find that the correlation function is

$$g(r) = p^r = e^{-r/\xi} \,, \tag{2.45}$$

where $\xi = -\frac{1}{\ln p}$ is called the correlation length. The correlation length diverges as $p \to p_c = 1$. We can again find the way in which it diverges through by using that when $p \to 1$

$$\ln p = \ln(1 - (1 - p)) \simeq -(1 - p) \,. \tag{2.46}$$

We find that the correlation length is

$$\xi = \xi_0 (p_c - p)^{-\nu} \,, \tag{2.47}$$

with $\nu = 1$. The correlation length therefore diverges as a power-law when $p \to p_c = 1$. This behavior is general for percolation theory, although the particular value of the exponent ν depends on the dimensionality.

We can use the correlation function to strengthen our interpretation of when a finite system size becomes relevant. As long as $\xi \ll L$, we will not notice the effect of a finite system, because no cluster is large enough to notice the finite system size. However, when $\xi \gg L$, the behavior is dominated by the system size L, and we are no longer able to determine how close we are to percolation.

2.5 (Advanced) Finite size effects

We have so far not discussed the effects of a finite lattice size L. We have implicitly assumed that the lattice size L is so large that the corrections will be small and can be ignored. However, we have now observed that the average cluster size S, the characteristic cluster size s_ξ, and the

correlation length ξ diverges when p approaches p_c. We will therefore eventually start observing effects of the finite system size as p approaches p_c.

We have essentially ignored two effects:

- **(a)** the upper limit for cluster sizes is L and not ∞
- **(b)** there are corrections to $n(s, p; L)$ due to the finite lattice size

The effect of (b) becomes clear as p approaches p_c: As s_ξ increases it will eventually be larger than L, which in one dimension also provides an upper limit for s. This is indeed observed in the scaling collapse plot for $n(s, p)$, where we for finite lattice sizes will find a cross-over cluster size s_L, which depends on the lattice size L.

What will be the effect of including a finite upper limit L for all the sums? This will imply that the result of the sum $\sum_s p^s$ will be

$$\sum_{s=1}^{L} p^s = \frac{1-p^L}{1-p} \ , \tag{2.48}$$

instead of $1/(1-p)$ when L is infinite. Indeed, this sum approaches $1/(1-p)$ as $L \to \infty$. This implies that S will approach L when $p \to p_c$, as can be seen by applying l'Hopital's rule to find the limit as $p \to p_c$. However, as long as $\xi \ll L$, we will still observe that $S \propto 1/(1-p)$. We will make these types of arguments more precise when we discuss finite size scaling further on.

2.5.1 Finite size effects in $\Pi(p, L)$ and p_c

So far we have only addressed the behavior of an infinite system, We have found that $\Pi(p, L) = p^L$. From this, we find that

$$\Pi' = \frac{d\Pi}{dp} = Lp^{L-1} \ . \tag{2.49}$$

What is the interpretation of Π'? We can write

$$\Pi'(p, L)dp = \Pi(p + dp, L) - \Pi(p, L) \ , \tag{2.50}$$

where the right hand term is the probability that the system became spanning when p increased from p to $p + dp$. That is, it is the probability that the spanning cluster appeared for the first time for p between p and $p + dp$. We can therefore interpret Π' as the probability density for p', which is the p when a spanning cluster appears.

What can we learn from the form of Π'? If we perform numerical experiments to find p_c, we see that for finite system sizes L, we might observe a p_c which is lower than 1. We can use Π' to find the average p' found - this will be done generally further on. Here, we will only study

the width of the distribution Π', which will give us an idea about the possible deviation when we measure p_c by a measurement of p'. We define the width as the value p_x for which Π' has reached $1/2$ (or some other value you like).

$$\Pi'(p_x, L) = L p_x^{L-1} = 1/2 \; . \tag{2.51}$$

This gives

$$\ln p_x = -\frac{\ln 2}{L-1} \; , \tag{2.52}$$

We will now use a standard approximation for $\ln x$, when x is close to 1, by writing

$$\ln p_x = \ln(1-(1-p_x)) \simeq -(1-p_x) \; , \tag{2.53}$$

where we have used that $\ln(1-x) \simeq -x$, when $x \ll 1$. This gives us that

$$(1-p_x) \simeq \frac{\ln 2}{L-1} \; , \tag{2.54}$$

and consequently,

$$p_x = p_c - \frac{\ln 2}{L-1} \; . \tag{2.55}$$

We will therefore have an L dependence in the effective p_c which is measured for a finite system. We will address this topic in much more depth later on under finite size scaling in chap. 6.

We can also find a similar scaling for $\Pi(p, L)$, because

$$\Pi(p, L) = p^L = e^{L \ln p} = e^{-L/\xi} \; , \tag{2.56}$$

where we have defined $\xi = -1/\ln p$. We notice that $\xi \to \infty$ when $p \to p_c = 1$. We can therefore classify the behavior of Π according to the relative sizes of the length ξ and L:

$$\Pi(p, L) = \begin{cases} 1 & L \ll \xi \\ 0 & L \gg \xi \end{cases} \; , \tag{2.57}$$

We have therefore found an important length scale ξ in our problem that appears whenever the length L appears.

2.6 Exercises

Exercise 2.1: Next-nearest neighbor connectivity in 1d

Assume that connectivity is to the next-nearest neighbors for an infinite one-dimensional percolation system.

a) Find $\Pi(p, L)$ for a system of length L.

b) What is p_c for this system?

c) Find $n(s, p)$ for an infinite system.

Exercise 2.2: Higher moments of s

The k'th moment of s is defined as

$$\langle s^k \rangle = \sum_s s^k \left(\frac{sn(s,p)}{p}\right) . \tag{2.58}$$

a) Find the second moment of s as a function of p.

b) Calculate the first moment of s numerically from $M = 1000$ samples for $p = 0.90$, 0.95, 0.975 and 0.99. Compare with the theoretical result.

c) Calculate the second moment of s numerically from $M = 1000$ samples for $p = 0.90$, 0.95, 0.975 and 0.99. Compare with the theoretical result.

Infinite-dimensional percolation

3

We have now seen how the percolation problem can be solved exactly for a one-dimensional system. However, in this case the percolation threshold is $p_c = 1$, and we were not able to address the behavior of the system for $p > p_c$. There is, however, another system in which many features of the percolation problem can be solved exactly, and this is percolation on a regular tree structure on which there are no loops. The condition of no loops is essential. This is also why we call this system a system of infinite dimensions, because we need an infinite number of dimensions in Euclidean space in order to embed a tree without loops. In this section, we will provide explicit solution to the percolation lattice on a particular tree structure called the Bethe lattice.

The Bethe lattice, which is also called the Cayley tree, is a tree structure in which each node has Z neighbors. This structure has no loops. If we start from the central point and draw the lattice, the perimeter grows as fast as the bulk. Generally, we will call Z the coordination number. The Bethe lattice is illustrated in fig. 3.1.

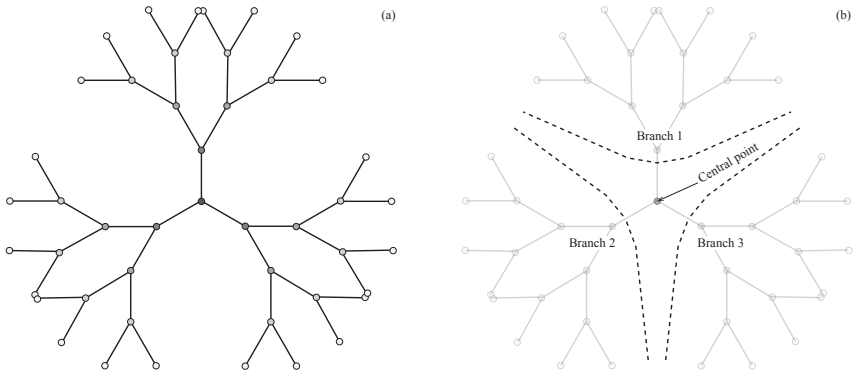

Fig. 3.1 Illustration of four generations of the Bethe lattice with number of neighbors $Z = 3$.

3.1 Percolation threshold

If we start from the center and move along a branch, we will generate $(Z-1)$ new neighbors from each of the branches. To get a spanning cluster, we need to ensure that at least one of the $Z-1$ sites are occupied on average. That is, the occupation probability, p, must be:

$$p(Z-1) \geq 1 \,, \tag{3.1}$$

in order for this process to continue indefinitely.

We associate p_c with the value for p where the cluster is on the verge of dying out, that is

$$p_c = \frac{1}{Z-1} \,. \tag{3.2}$$

For $Z = 2$ we regain the one-dimensional system, with percolation threshold $p_c = 1$. However, when $Z > 2$, we obtain a finite percolation threshold, that is, $p_c < 1$, which means that we can observe the behavior both above and below p_c.

In the following, we will use a set of standard techniques to find the density of the spanning cluster, $P(p)$, the average cluster size S, before we address the full scaling behavior of the cluster density $n(s,p)$.

3.2 Spanning cluster

We will use a standard approach to find the density $P(p)$ of the spanning cluster when $p > p_c$. The technique is based on starting from a "central" site, and then address the probability that a given branch is connected to infinity.

We can use a strictly technical approach to find P by noting that P can be found from

$$p = P + \sum_s sn(s,p) \ , \tag{3.3}$$

where the sum is the probability that the site is part of a finite cluster, that is, it is the probability that the site is not connected to infinity. Let us use Q to denote the probability that a branch does not lead to infinity. The concept of a central point and a branch is illustrated in fig. 3.1.

We can arrive at this result by noticing that the probability that at site is not connected to infinity in a particular direction is Q. The probability that the site is not connected to infinity in any direction is therefore Q^Z. The probability that the site *is* connected to infinity is therefore $1 - Q^Z$. In addition, we need to include the probability p that the site is occupied. The probability that a given site is connected to infinity, that is, that it is part of the spanning cluster, is therefore

$$P = p(1 - Q^Z) \ . \tag{3.4}$$

It now remains to find an expression for $Q(p)$. We will determine Q through a consistency equation. Let us assume that we are moving along a branch, and that we have come to a point k. Then, Q gives the probability that this branch does not lead to infinity. This can occur by either the site k not being occupied, with probability $(1-p)$, or by site k being occupied with probability p, and all of the $Z-1$ branches leading out of k not being connected to infinity, with probability Q^{Z-1}. The probability Q for the branch not to be connected to infinity is therefore

$$Q = (1-p) + pQ^{Z-1} \ . \tag{3.5}$$

We can check this equation by looking at the case when $Z = 2$, which should correspond to the one-dimensional system. In this case we have $Q = 1 - p + pQ$, which gives, $(1-p)Q = (1-p)$, where we see that when $p \neq 1$, $Q = 1$. That is, when $p < 1$ all branches are not connected to infinity, implying that there is no spanning cluster. We regain the results from one-dimensional percolation theory.

We could solve this equation for general Z. However, for simplicity we will restrict ourselves to $Z = 3$, which is the smallest Z that gives a behavior different from the one-dimensional system. In this case

$$Q = 1 - p + pQ^2 \ , \tag{3.6}$$

$$pQ^2 - Q + 1 - p = 0 \ . \tag{3.7}$$

The solution of this second order equation is

$$Q = \frac{1 \pm \sqrt{(2p-1)^2}}{2p} = \begin{cases} 1 & p < p_c \\ \frac{1-p}{p} & p > p_c \end{cases} \ . \tag{3.8}$$

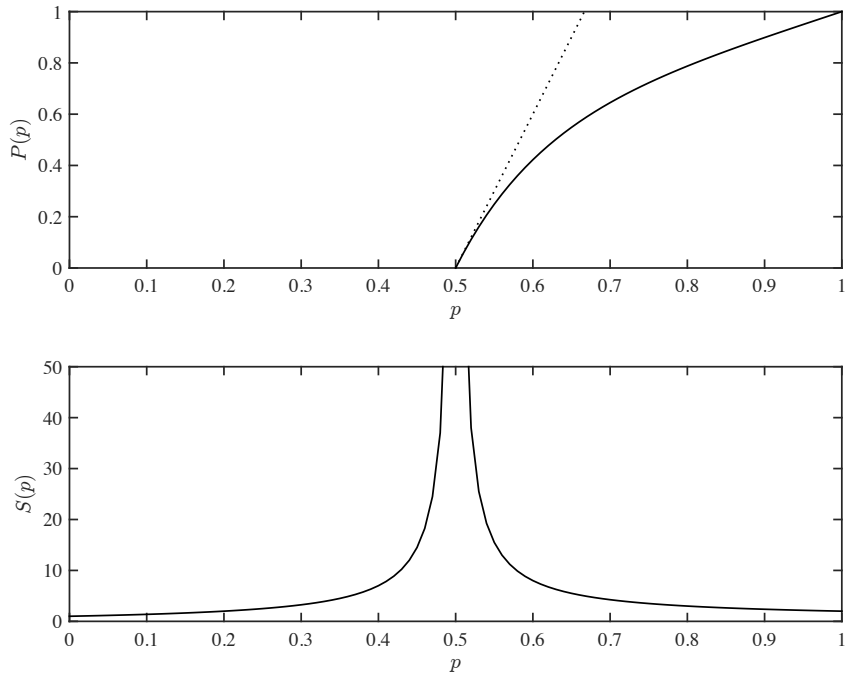

Fig. 3.2 (Top) A plot of $P(p)$ as a function of p for the Bethe lattice with $Z = 3$. The tangent at $p = p_c$ is illustrated by a straight line. (Bottom) A plot of the average cluster size, $S(p)$, as a function of p for the Bethe lattice with $Z = 3$. The average cluster size diverges when $p \to p_c = 1/2$ both from below and above.

There are two possible solutions. We recognize that the solution $(1-p)/p$ is 1 for $p = p_c = 1/2$, and is larger than 1 for smaller values of p, we must therefore use the other solution of $Q = 1$ for $p < p_c = 1/2$. These results confirm the value $p_c = 1/2$ as the percolation threshold. When $p \leq p_c$, we find that $Q = 1$, that is, no branch propagates to infinity. Whereas, when $p > p_c$, Q becomes smaller than 1, and there is a finite probability for a branch to continue to infinity.

We insert this back into the equation for $P(p)$ and find that for $p > p_c$:

$$P = p(1 - Q^3) \tag{3.9}$$

$$= p(1 - (\frac{1-p}{p})^3) \tag{3.10}$$

$$= p(1 - \frac{1-p}{p})(1 + \frac{1-p}{p} + (\frac{1-p}{p})^2) \; . \tag{3.11}$$

This result is illustrated in fig. 3.2.

From this we observe the expected result that when $p \to 1$, $P(p) \propto p$. We can rewrite the equation as

$$P = 2(p - \frac{1}{2})(1 + \frac{1-p}{p} + (\frac{1-p}{p})^2) \; , \tag{3.12}$$

From this we can immediately find the leading order behavior when $p \to p_c = 1/2$. In this case we have

$$P \simeq 6(p - p_c) + o((p - p_c)^2) \ . \tag{3.13}$$

We have therefore found that for $p > p_c$

$$P(p) \simeq B(p - p_c)^\beta \ , \tag{3.14}$$

where $B = 6$, and the exponent $\beta = 1$. The density of the spanning cluster is therefore a power-law in $(p-p_c)$ with exponent β. The exponent depends on the dimensionality of the lattice, but should not depend on lattice details, such as the number of neighbors Z. We will leave it to the reader as an exercise to show that β is the same for $Z = 4$.

We notice in passing that our approach is an example of a mean field solution, or a self-consistency solution: We assume that we know Q, and then solve to find Q. We will use similar methods further on in this course.

3.3 Average cluster size

We will use a similar method to find the average cluster size, $S(p)$. Let us introduce $T(p)$ as the average number of sites connected to a given site on a specific branch, such as in branch 1 in fig. 3.1. The average cluster size S is then given as

$$S = 1 + ZT \ , \tag{3.15}$$

where the 1 represents the central point, and T is the average number of sites on each branch. We will again find a self-consistent equation for T, starting from a center site. The average cluster size T is found from summing the probability that the next site k is empty, $1 - p$, multiplied with the contribution to the average in this case (0), plus the probability that the next site is occupied, p, multiplied with the contribution in this case, which is the contribution from the site (1) and the contribution of the remaining $Z - 1$ subbranches. In total:

$$T = (1 - p)0 + p(1 + (Z - 1)T) \ , \tag{3.16}$$

We can solve this directly for T, finding

$$T = \frac{p}{1 - p(Z - 1)} \ , \tag{3.17}$$

where we recognize that the value $p_c = 1/(Z - 1)$ plays a special role because the average size of the branch diverges when $p \to p_c$. We find the average cluster size S to be:

$$S = 1 + ZT = \frac{1+p}{1-(Z-1)p} = \frac{p_c(1+p)}{p_c - p} \,, \tag{3.18}$$

which is illustrated in fig. 3.2. The expression for $S(p)$ can therefore be written on the general form

$$S = \frac{\Gamma}{(p_c - p)^\gamma} \,, \tag{3.19}$$

where our argument determines $p_c = 1/(Z-1)$, and the exponent $\gamma = 1$. The average cluster size S therefore diverges as a power-law when p approaches p_c. The exponent γ characterizes the behavior, and the value of γ depends on the dimensionality, but not on the details of the lattice. Here, we notice in particular that γ does not depend on Z.

3.4 Cluster number density

In order to find the cluster number density for the Bethe lattice, we need to address how we in general can find the cluster number density. In general, in order to find the cluster number density for a given s, we need to find all possible configurations of clusters of size s, and sum up their probability:

$$n(s,p) = \sum_{c(s)} p^s (1-p)^{t(c)} \tag{3.20}$$

Here we have included the term p^s, because we know that we must have all the s sites of the cluster present, and we have included the term $(1-p)^t$, because all the neighboring sites must be unoccupied, and there are $t(c)$ neighbors for configuration c. Based on this, we realize that we could instead make a sum over all t, but then we need to include the effect that there are several clusters that can have the same t. We will then have to introduce the degeneracy factor $g_{s,t}$ which gives the number of different clusters that have size s and a number of neighbors equal to t. The cluster number density can then be written as

$$n(s,p) = p^s \sum_t g_{s,t} (1-p)^t \,. \tag{3.21}$$

This can be illustrated for two-dimensional percolation. Let us study the case when $s = 3$. In this case there are 6 possible clusters for size $s = 3$, as illustrated in fig. 3.3.

There are two clusters with $t = 8$, and four clusters with $t = 7$. There are no other clusters of size $s = 3$. We can therefore conclude that for the two-dimensional lattice, we have $g_{3,8} = 2$, and $g_{3,7} = 4$, and $g_{3,t} = 0$ for all other values of t.

For the Bethe lattice, there is a particularly simple relation between the number of sites, and the number of neighbors. We can see this by

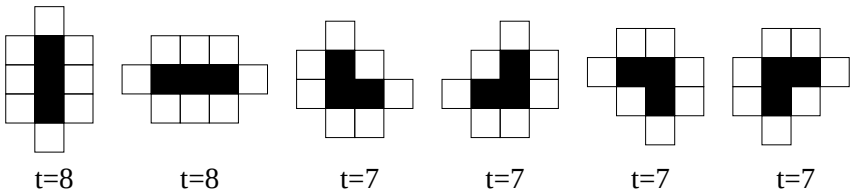

Fig. 3.3 Illustration of the 6 possible configurations for a two-dimensional cluster of size $s = 3$.

looking at the first few generation of a Bethe lattice grown from a central seed. For $s = 1$, the number of neighbors are $t_1 = Z$. When we add one more site, we remove one neighbor from what we had previously, in order to add a new site, and then we add $Z - 1$ new neighbors: $s = 2$, and $t_2 = t_1 + (Z - 2)$. Consequently,

$$t_k = t_{k-1} + (Z - 2) \, , \tag{3.22}$$

and therefore:

$$t_s = s(Z - 2) + 2 \, . \tag{3.23}$$

The cluster number density, given by the sum over all t, is therefore reduced to only a single term for the Bethe lattice

$$n(s, p) = g_{s,t_s} p^s (1 - p)^{t_s} \, , \tag{3.24}$$

For simplicity, we will write $g_s = g_{s,t_s}$. In general, we do not know g_s, but we will show that we still can learn quite a lot about the behavior of $n(s, p)$.

The cluster density can therefore be written as

$$n(s, p) = g_s p^s (1 - p)^{2+(Z-2)s} \, . \tag{3.25}$$

We rewrite this as a common factor to the power s:

$$n(s, p) = g_s [p(1 - p)^{Z-2}]^s (1 - p)^2 \, , \tag{3.26}$$

which, for $Z = 3$ becomes

$$n(s, p) = g_s [p(1 - p)]^s (1 - p)^2 \, . \tag{3.27}$$

However, we can use a general Z for our argument. We will study $n(s, p)$ for p close to p_c. In this range, we will do a Taylor expansion of the term $f(p) = p(1 - p)^{Z-2}$, which is raised to the power s in the equation for $n(s, p)$. The shape of $f(p)$ as a function of p is shown in fig. 3.4. The maximum of $f(p)$ occurs for $p = p_c = 1/(Z - 1)$. This is also easily seen from the first derivative of $f(p)$.

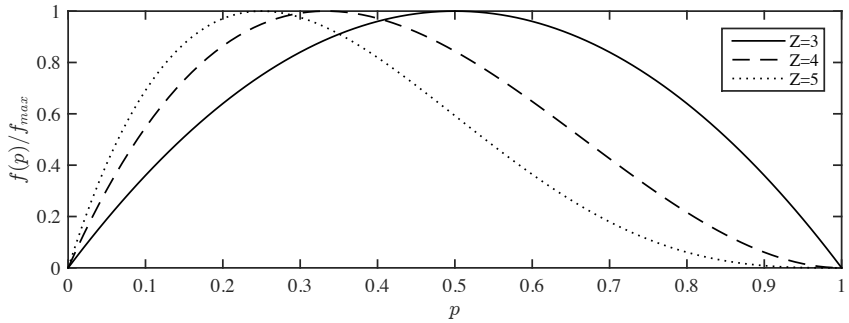

Fig. 3.4 A plot $f(p) = p(1-p)^{Z-2}$, which is a term in the cluster number density $n(s,p) = g_s[p(1-p)^{Z-2}]^s(1-p)^2$ for the Bethe lattice. We notice that $f(p)$ has a maximum at $p = p_c$, and that the second derivative, $f''(p)$, is zero in this point. A Taylor expansion of $f(p)$ around $p = p_c$ will therefore have a second order term in $(p - p_c)$ as the lowest-order term - to lowest order it is a parabola at $p = p_c$. It is this second order term which determines the exponent σ, which consequently is independent of Z.

$$f'(p) = (1-p)^{Z-2} - p(Z-2)(1-p)^{Z-3} = \tag{3.28}$$

$$= (1-p)^{Z-3}(1-p-p(Z-2)) = \tag{3.29}$$

$$= (1-p)^{Z-3}(1-(Z-1)p) \tag{3.30}$$

which shows that $f'(p_c) = 0$. We leave it to the reader to show that $f''(p_c) < 0$.

The Taylor expansion can be written as

$$f(p) = f(p_c) + f'(p_c)(p-p_c) + \frac{1}{2}f''(p_c)(p-p_c)^2 + o((p-p_c)^3)\ , \tag{3.31}$$

where we already have found the the first order term, $f'(p_c) = 0$. We can therefore write

$$f(p) \simeq f(p_c) - \frac{1}{2}f''(p_c)(p-p_c)^2 = A(1-B(p-p_c)^2)\ . \tag{3.32}$$

The cluster number density is

$$n(s,p) = g_s[f(p)]^s(1-p)^2 = g_s e^{s\ln f(p)}(1-p)^2\ , \tag{3.33}$$

where we now insert $f(p) \simeq A(1-B(p-p_c)^2)$ to get

$$n(s,p) \simeq g_s A^s e^{s\ln(1-B(p-p_c)^2)}(1-p)^2\ . \tag{3.34}$$

We use the first order of the Taylor expansion of $\ln(1-x) \simeq -x$, to get

$$n(s,p) \simeq g_s A^s e^{-sB(p-p_c)^2}(1-p)^2\ . \tag{3.35}$$

Consequently, for $p = p_c$ we get

$$n(s,p_c) = g_s A^s (1-p)^2\ . \tag{3.36}$$

As a result, we can rewrite the cluster density in terms of $n(s, p_c)$, giving

$$n(s,p) = n(s,p_c)e^{-sB(p-p_c)^2} \ , \tag{3.37}$$

when p is close to p_c. The exponential term we could again rewrite as

$$n(s,p) = n(s,p_c)e^{-s/s_\xi} \ , \tag{3.38}$$

where the characteristic cluster size s_ξ is

$$s_\xi = B^{-1}(p-p_c)^{-2} \ , \tag{3.39}$$

which implies that the characteristic cluster size diverges as a power-law with exponent $1/\sigma = 2$. The general scaling form for the characteristic cluster size s_ξ is

$$s_\xi \propto |p-p_c|^{-1/\sigma} \ , \tag{3.40}$$

where the exponent σ is universal, meaning that is does not depend on lattice details such a Z, as we have demonstrated here, but it does depend on lattice dimensionality. It will therefore be a different value for two-dimensional percolation.

The next step is to address the behavior at $p = p_c$, when the characteristic cluster size is diverging.

We have already found some limits on the behavior of the cluster density $n(s, p)$, because we have found S and $P(p)$, which can be related to the cluster number density. We will use these relations two find limits on the behavior of $n(s, p_c)$.

The average cluster size at $p = p_c$ is

$$S = \frac{\Gamma}{p_c - p} \ , \tag{3.41}$$

which should diverge, that is

$$S = \sum s^2 n(s,p_c) \to \infty \ , \tag{3.42}$$

if we go the the limit of a continuous $n(s, p_c)$, the integral

$$S = \int_0^\infty s^2 n(s,p_c) ds \to \infty \ , \tag{3.43}$$

should diverge. We can therefore conclude that $n(s, p_c)$ is not an exponential, since that would lead to convergence. We can make a scaling ansatz

$$n(s,p_c) \simeq Cs^{-\tau} \ , \tag{3.44}$$

for $s \gg 1$. We can include this into the restrictions that

$$\sum_s sn(s,p) = p - P \ , \tag{3.45}$$

which should converge, and

$$\sum_s s^2 n(s, p_c) \to \infty \, , \tag{3.46}$$

which should not converge. This provides a set of limits on the possible values of τ, because

$$\sum_s s n(s, p_c) \simeq \sum_s s^{1-\tau} < \infty \;\Rightarrow\; \tau - 1 > 1 \, , \tag{3.47}$$

and

$$\sum s^2 n(s, p_c) \simeq \sum_s s^{2-\tau} > \infty \;\Rightarrow\; \tau - 2 \leq 1 \, , \tag{3.48}$$

which therefore implies that

$$2 < \tau \leq 3 \, . \tag{3.49}$$

We can therefore sum up our arguments so far in the relation

$$n(s, p) = n(s, p_c) e^{-B(p-p_c)^2 s} = C s^{-\tau} e^{-B(p-p_c)^2 s} = C s^{-\tau} e^{-s/s_\xi} \, . \tag{3.50}$$

We will now use this expression to calculate S, for which we know the exact scaling behavior, and then again use this to find the value for τ

$$S = C \sum_s s^{2-\tau} e^{-s/s_\xi} \to C \int_1^\infty s^{2-\tau} e^{-s/s_\xi} ds \, . \tag{3.51}$$

We could now make a very rough estimate. This is useful, since it is in the spirit of this course, and it also provides the correct behavior. We could assume that

$$S = C \int_1^\infty s^{2-\tau} e^{-s/s_\xi} ds \sim C \int_1^{s_\xi} s^{2-\tau} ds \sim s_\xi^{3-\tau} \, , \tag{3.52}$$

which actually provides the correct result. We can do it slightly more elaborately:

$$S \simeq C \int_1^\infty s^{2-\tau} e^{-s/s_\xi} ds \, , \tag{3.53}$$

we change variables by introducing, $u = s/s_\xi$, which gives

$$S \simeq s_\xi^{3-\tau} \int_{1/s_\xi}^\infty u^{2-\tau} e^{-u} du \, . \tag{3.54}$$

Where the integral is now a number, since $1/s_\xi \to 0$, when $p \to p_c$. The asymptotic scaling behavior in the limit $p \to p_c$ is therefore

$$S \sim s_\xi^{3-\tau} \sim (p - p_c)^{-2(3-\tau)} \sim (p - p_c)^{-1} \, , \tag{3.55}$$

where we have used that

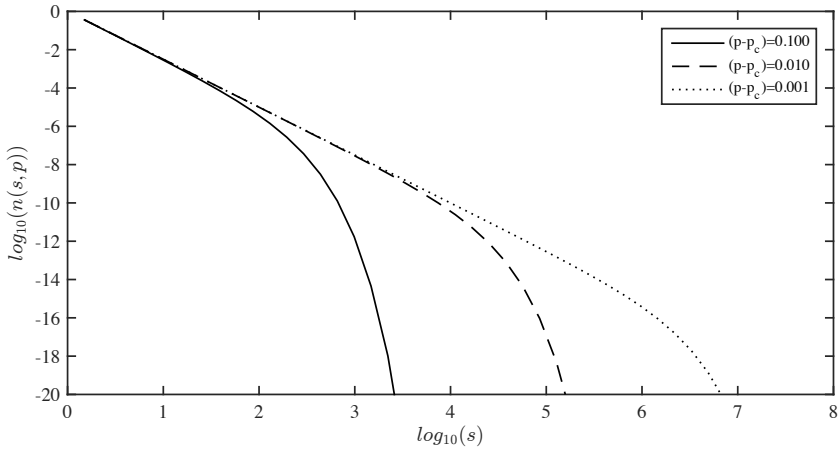

Fig. 3.5 A plot of $n(s,p) = s^{-\tau}\exp(-s(p-p_c)^2)$ as a function of s for various values of p illustrates how the characteristic cluster size s_ξ appears as a cut-off in the cluster number density that scales with $p - p_c$.

$$s_\xi \sim (p - p_c)^{-2} \,, \tag{3.56}$$

and that

$$S \sim (p - p_c)^{-1} \,. \tag{3.57}$$

Direct solution therefore shows that

$$\tau = \frac{5}{2} \,. \tag{3.58}$$

This relation also satisfies the exponent relations we found above, since $2 < 5/2 \leq 3$. A plot of the scaling form is shown in fig. 3.5.

This provides us with a preliminary scaling theory for the cluster density. We will spend time now trying to verify this scaling relation for percolation in other dimensionalities. We have found that in the vicinity of p_c, we do not expect deviations until we reach large s, that is, before we reach a characteristic cluster size s_ξ that increases as $p \to p_c$. We therefore expect a general form of the cluster density

$$n(s,p) = n(s,p_c)F(\frac{s}{s_\xi}) \,, \tag{3.59}$$

where

$$n(s,p_c) = Cs^{-\tau} \,, \tag{3.60}$$

and

$$s_\xi = s_0|p - p_c|^{-1/\sigma} \,. \tag{3.61}$$

In addition, we have the following scaling relations:

$$P(p) \sim (p - p_c)^\beta \,, \tag{3.62}$$

$$\xi \sim |p - p_c|^{-\nu} \,, \tag{3.63}$$

and

$$S \sim |p - p_c|^{-\gamma} \,, \tag{3.64}$$

with a possible non-trivial behavior for higher moments of the cluster density.

3.5 Advanced: Embedding dimension

Why is it difficult to embed such a structure in a $d+1$-dimensional space? Because for an Euclidean structure of dimension d, the volume, V grows as

$$V \propto L^d \,, \tag{3.65}$$

and the surface, S, grows as

$$S \propto L^{d-1} \,, \tag{3.66}$$

where L is the linear dimension of the system. This means that

$$S \propto V^{1-\frac{1}{d}} \,. \tag{3.67}$$

However, for the Bethe lattice, the surface is proportional to the volume, $S \propto V$, which would imply that $d \to \infty$.

3.6 Exercises

Exercise 3.1: $P(p)$ for $Z = 4$

Find $P(p)$ for $Z = 4$ and determine β for this value of Z.

Finite-dimensional percolation

4

For the one-dimensional and the infinite-dimensional systems we have been able to find exact results for the percolation probability, $\Pi(p)$, for $P(p)$, the probability for a site to belong to an infinite cluster, and we have characterized the behavior using the distribution of cluster sizes, $n(s,p)$ and its cut-off, s_ξ. In both one and infinite dimensions we have been able to calculate these functions exactly. However, in two and three dimensions – which are the most relevant for our world – we are unfortunately not able to find exact solutions. We saw above that the number of configurations in a L^d system in d-dimensions increases very rapidly with L – so rapidly that a complete enumeration is impossible. But can we still use what we learned from the one and inifinite-dimensional systems?

In the one-dimensional case it was simple to find $\Pi(p,L)$ because there is only one possible path from one side to another. We cannot generalize this to two dimensions, since in two-dimensions there are many paths from one side to another – and we need to include all to estimate the probability for percolation. Similarly, it was simple to find $n(s,p)$, because all clusters only have two neighboring sites – the surface is always of size 2. This is also not generalizable to higher dimensions.

In the infinite-dimensional system, we were able to find $P(p)$ because we could separate the cluster into different paths that never can intersect expect in a single point, because there are no loops in the Bethe lattice. This is not the case in two and three dimensions, where there will always be the possibility for loops. When there are loops present, we cannot use the arguments we used for the Bethe lattice, because a branch cut off at one point may be connected again further out. For the Bethe lattice, we could also estimate the multiplicity $g(s,t)$ of the clusters, the number of possible clusters of size s and surface t, since t was a function of s. In a two- or three-dimensional system this is not similarly simple, because the multiplicity $g(s,t)$ is not simple even in two dimensions, as illustrated in fig. 4.1.

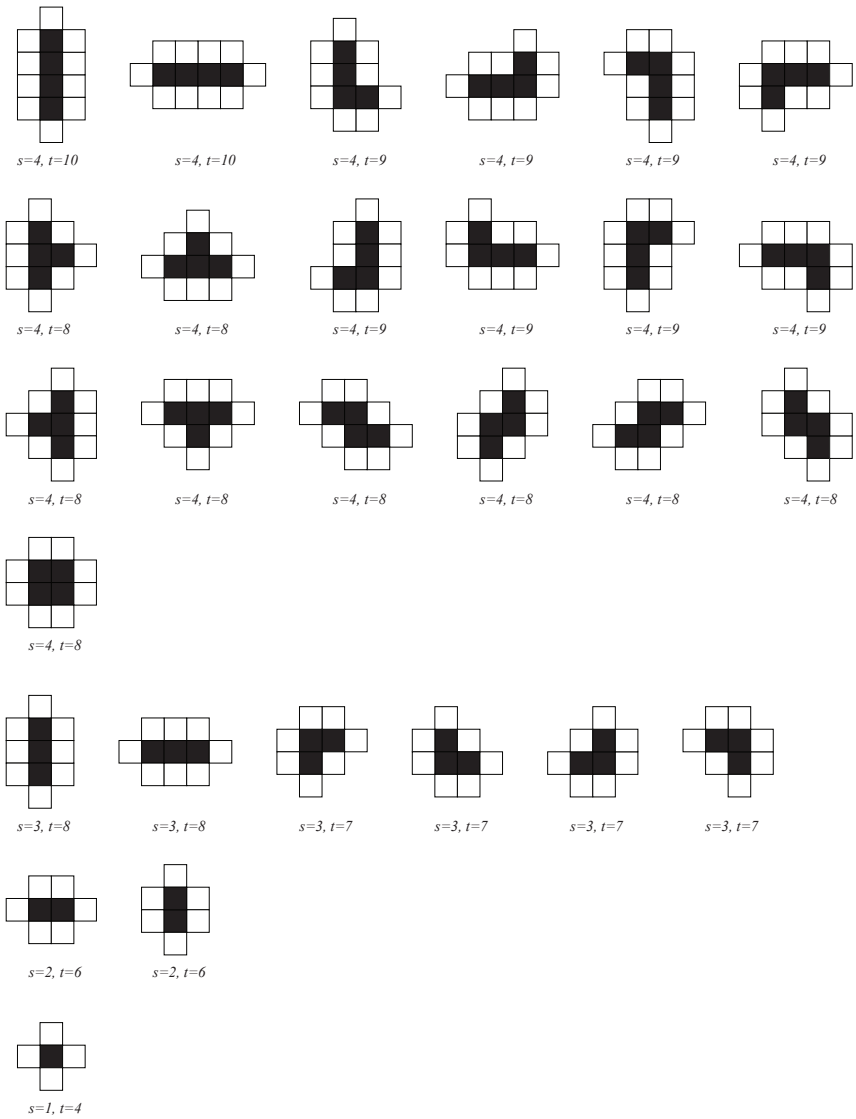

Fig. 4.1 Illustration of the 6 possible configurations for a two-dimensional cluster of size $s = 3$.

This means that the solution methods used for the one and the infinite dimensional systems cannot be exteded to address two or three dimensional systems. However, several of the techniques and observations we have made for the one-dimensional and the Bethe lattice systems, can be used as the basis for a generalized theory that can be applied in any dimension. Here, we will therefore pursue the more general features of the percolation system, starting with the cluster number density, $n(s, p)$.

4.1 Cluster number density

We have found that the cluster number density plays a fundamental role in our understanding of the percolation problem, and we will use it here as our basis for the scaling theory for percolation.

When we discussed the Bethe lattice, we found that we could write the cluster number density as a sum over all possible configurations of cluster size, s:

$$n(s,p) = \sum_j p^s (1-p)^{t_j} \ , \tag{4.1}$$

where j runs over all different configurations, and t_j denotes the number of neighbors for this particular configuration. We can simplify this by rewrite the sum to be over all possible number of neighbors, t, and include the degeneracy $g_{s,t}$, the number of configurations with t neighbors:

$$n(s,p) = \sum_t g_{s,t} p^s (1-p)^t \ . \tag{4.2}$$

The values of $g_{s,t}$ have been tabulated up to $s = 40$. However, while this may give us interesting information about the smaller cluster, and therefore for smaller values of p, it does not help us to develop a theory for the behavior for p close to p_c.

In order to address the cluster number density, we will need to study the characteristics of $n(s,p)$, for example by generating numerical estimates for its scaling behavior, and then propose a general scaling form which will be tested in various settings.

4.1.1 Numerical estimation of $n(s,p)$

We discussed how to measure $n(s,p)$ from a set of numerical simulations in chap. 2. We can use the same method in two and higher dimensions. We estimate $n(s,p;L)$ using

$$\overline{n(s,p;L)} = \frac{N_s}{M \cdot L^d} \ , \tag{4.3}$$

where N_s is the total number of clusters of size s measured for M simulations in a system of size L^d and for a given value of p. We perform these simulations just as we did in one dimension, using the following program:

```
M = 1000;
L = 200;
p = 0.58;
allarea = [];
for i = 1:M
    z = rand(L,L);
```

```
    m = z<p;
    [lw,num] = bwlabel(m,4);
    s = regionprops(lw,'Area');
    area = cat(1,s.Area);
    allarea = cat(1,allarea,area);
end
[n,s]=hist(allarea,L^2);
nsp = n/(L^2*M);
i = find(n>0);
subplot(2,1,1)
plot(s(i),nsp(i),'ok');
xlabel('s'); ylabel('n(s,p)');
subplot(2,1,2)
loglog(s(i),nsp(i),'ok');
xlabel('s'); ylabel('n(s,p)');
```

The resulting plot of $\overline{n(s,p;L)}$ for $L = 200$ is shown in fig. 4.2a,b. Unfortunately, this plot is not very useful. The problem is that there are many values of s for which we have little or no data at all! For small values of s we have many clusters for each value of s and the statistics is good. But for large values of s, such as for clusters of size $s = 10^4$ and above, we have less than one data point for each value of s. Our measured distribution $\overline{n(s,p;L)}$ is therefore a poor representation of the real $n(s,p;L)$ in this range.

4.1.2 Measuring probabilty densities of rare events

The problem with the measured results in fig. 4.2 occur because we have chosen a very small bin size for the histogram. However, we see that for small values of s we want to have a small bin size, since the statistics here is good, but for large values of s we want to have larger bin sizes. This is often solved by using logarithmic binning: We make the bin edges a^i, where a is the basis for the bins and i is bin number. If we chose $a = 2$ as the basis for the bins, the bin edges will be $2^0, 2^1, 2^2, 2^3, \ldots$, that is $1, 2, 4, 8, \ldots$. (Maybe we should instead have called the method *exponential binning*). We then count how many events occur in each such bin. If we number the bins by i, then the edges of the bins are $s_i = a^i$, and the width of bin i is $\Delta s_i = s_{i+1} - s_i$. We then count how many event, N_i, occuring in the range from s_i to $s_i + \Delta s_i$, and we use this to find the cluster number density $n(s,p;L)$. However, since we now look at ranges of s values, we need to be precise: We want to measure the probability for a cluster to belong to a specific site of a cluster in the range from s to $s + \Delta s$, that is, we want to measure $n(s,p;L)\Delta s$, which we estimate from

$$\overline{n(s_i,p;L)}\Delta s_i = \frac{N_i}{ML^d} \,, \tag{4.4}$$

and we find $n(s,p;L)$ from

$$\overline{n(s_i, p; L)} = \frac{N_i}{ML^d \Delta s_i} \ . \tag{4.5}$$

It is important to remember to divide by Δs_i when the bin sizes are not all the same! We implement this by generating an array of all the bin edges. First, we find an upper limit to the bins, that is, we find an i_m so that

$$a^{i_m} > \max(s) \ \Rightarrow \ \log_a a^{i_m} > \log_a \max(s) \ , \tag{4.6}$$

$$i_m > \log_a \max(s) \ . \tag{4.7}$$

We can for example round the right hand side up to the nearest integer

```
imax = ceil(log(max(allarea))/log(a));
```

where `allarea` corresponds to all the s-values. We can then generate an array of indecies from 1 to this maximum value

```
bins = a.^(0:1:logamax);
```

And we can further generate the histogram with this set of bin edges

```
nl = histc(allarea,bins);
```

And we must then find the bin sizes and the bin centers

```
ds = diff(bins);
sl = (bins(1:end-1)+bins(2:end))*0.5;
```

And we calculate the estimated value for $\overline{n(s, p; L)}$:

```
nsl = nl(1:end-1)'./(M*L^2*ds);
```

Finally we plot the results. The complete code for this analysis is found in the following script

```
a = 1.2;
logamax = ceil(log(max(s))/log(a));
bins = a.^(0:1:logamax);
nl = histc(allarea,bins);
ds = diff(bins);
sl = (bins(1:end-1)+bins(2:end))*0.5;
nsl = nl(1:end-1)'./(M*L^2*ds);
subplot(3,1,3)
loglog(sl,nsl,'ok');
```

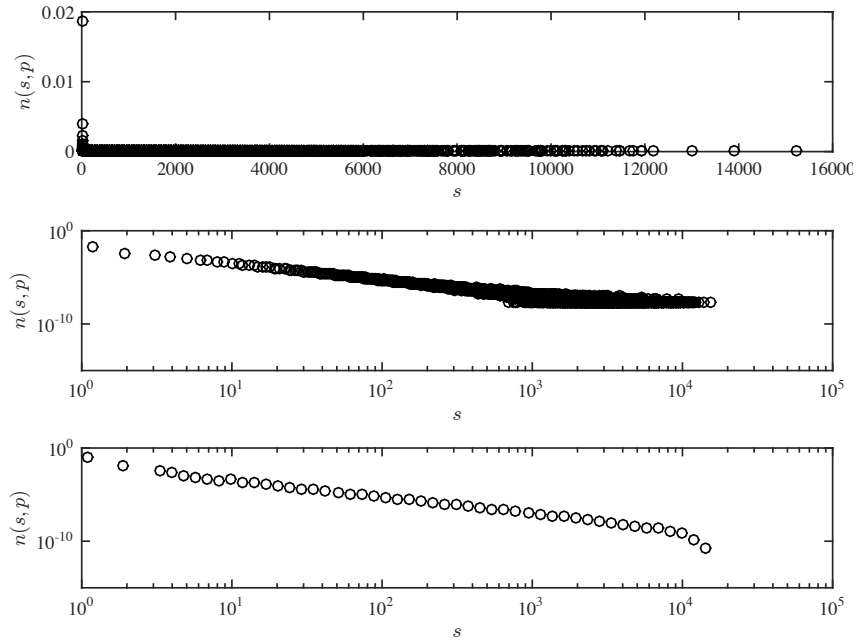

Fig. 4.2 Plot of $n(s,p;L)$ estimated from $M = 1000$ samples for $p = 0.58$ and $L = 200$. (a) Direct plot. (b) Log-log plot. (c) Plot of the logarithmically binned distribution.

The resulting plot for $a = 1.2$ is shown in fig. 4.2c. Notice that the resulting plot now is much easier to interpret than the linearly binned plot. (You should, however, always reflect on whether your binning method may influence the resulting plot in some way, since there may be cases where your choice of binning method may affect the results you get. Although this is not expected to play any role in your measurements in this book.) We will therefore in the following adapt logarithmic binning strategies whenever we measure a dataset which is sparse.

4.1.3 Measurements of $n(s,p)$ when $p \to p_c$

What happens to $n(s,p:L)$ when we change p so that it approaches p_c. We perform a sequence of simulations for various values of p_c and plot the resulting values for $\overline{n(s,p;L)}$. The resulting plot is shown in fig. 4.3.

Since the plot is double-logarithmic, a straight line corresponds to a power-law type behavior, $n(s,p) \propto s^{-\tau}$. We see that as p approaches p_c the cluster number density $n(s,p)$ more and more approaches a power-law behavior. For a value of p which is away from p_c, the $n(s,p)$ curve follows the power-law behavior for some time, but then deviates by dropping rapidly. This is an effect of the characteristic cluster size, which also can be visually observed in fig. 1.4 and fig. 1.5, where we see that the characteristic cluster size increases as p approaches p_c. How can we characterize the characteristic cluster size based on this measurement of

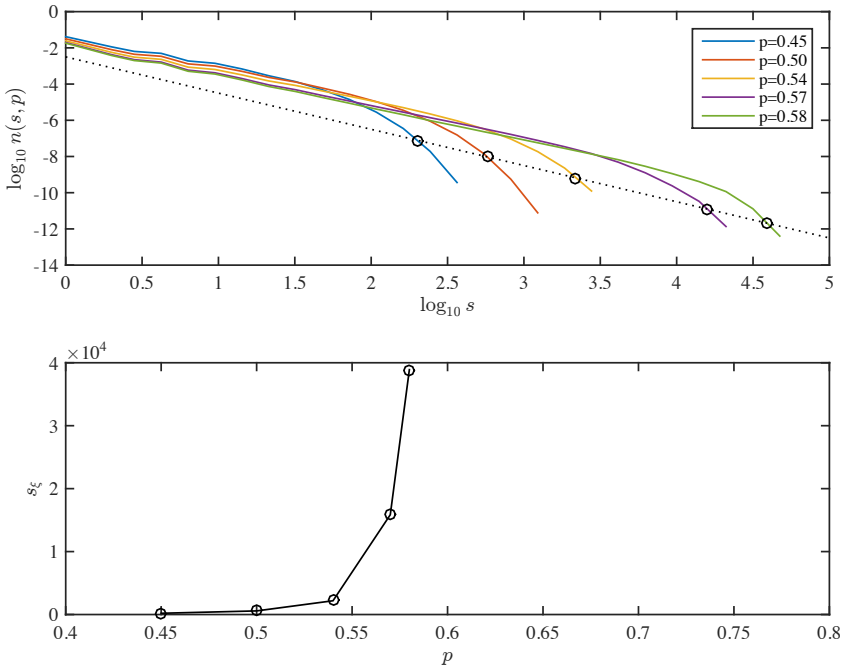

Fig. 4.3 (a) Plot of $n(s,p;L)$ as a function of s for various values of p for a 512×512 lattice. (b) Plot of $s_\xi(p)$ measured from the plot of $n(s,p)$ corresponding to the points shown in circles in (a).

$n(s,p)$? When s reaches s_ξ, it falls off from the power-law type behavior observed as $p \to p_c$. So, we could measure s_ξ directly from the lot, by drawing a straight line parallel to the behavior of $n(s,p_c)$, but below the $n(s,p_c)$ line, as illustrated in fig. 4.3. When the measured, $n(s,p)$ intersects this drawn line, $n(s,p)$ has fallen by a constant factor below $n(s,p_c)$ and we *define* this as s_ξ, and we measure it by reading the values from the s-axis. The resulting set of s_ξ values are plotted as a function of p in fig. 4.3. We see that s_ξ increases and possibly diverges as p approaches p_c. This is an effect we also found in the one-dimensional and the infinite-dimensional case, where we found that

$$s_\xi \propto |p - p_c|^{-1/\sigma} \tag{4.8}$$

where σ was 1 is one dimension. We will now use this to develop a theory for both $n(s,p;L)$ and s_ξ based on our experience from one and infinite dimensional percolation.

4.1.4 Scaling theory for $n(s,p)$

When we develop a theory, we realize that we are only interested in the limit $p \to p_c$, that is $|p - p_c| \ll 1$, and $s \gg 1$. In this limit, we expect that s_ξ marks the cross-over between two different behaviors. There is a

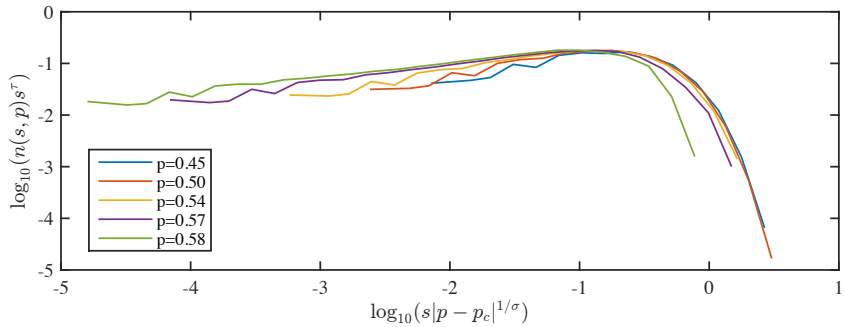

Fig. 4.4 A plot of $n(s,p)s^\tau$ as a function of $|p-p_c|^{1/\sigma}s$ shows that the cluster number density satisfies the scaling ansatz of eq. 4.12.

common behavior for small s, up to a cut-off, s_ξ, as we also observe in fig. 4.3: The curves for all p are approximately equal for small s.

Based on what we observed in one-dimension and infinite-dimensions, we expect and propose the following form for $n(s,p)$:

$$n(s,p) = n(s,p_c)F(\frac{s}{s_\xi}) \ , \tag{4.9}$$

$$n(s,p_c) = Cs^{-\tau} \ , \tag{4.10}$$

$$s_\xi = s_0|p-p_c|^{-1/\sigma} \ . \tag{4.11}$$

The best estimates for the exponents for various systems are listed in the following table: .

d	β	τ	σ	γ	ν	D	μ	D_{min}	D_{max}	D_B
1		2	1	1	1					
2	5/36	187/91	36/91	43/18	4/3	91/48	1.30	1.13	1.4	1.6
3	0.41	2.18	0.45	1.80	0.88	2.53	2.0	1.34	1.6	1.7
4	0.64	2.31	0.48	1.44	0.68	3.06	2.4	1.5	1.7	1.9
Bethe	1	5/2	1/2	1	1/2	4	3	2	2	2

We will often simplify the scaling form by writing it on the form:

$$n(s,p) = s^{-\tau}F(s/s_\xi) = s^{-\tau}F((p-p_c)^{1/\sigma}s) \ . \tag{4.12}$$

What can we expect from the scaling function $F(x)$?

This is essentially the prediction of a data-collapse . If we plot $s^\tau n(s,p)$ as a function of $s|p-p_c|^{1/\sigma}$ we would expect to get the scaling function $F(x)$, which should be a universal curve, as illustrated in fig. 4.4.

An alternative scaling form is

$$n(s,p) = s^{-\tau}\hat{F}((p-p_c)s^\sigma) \ , \tag{4.13}$$

where we have introduced the function $\hat{F}(u) = F(u^\sigma)$. These forms are equivalent, but in some cases this form produces simpler calculations.

This scaling form should in particular be valid for both the 1d and the Bethe lattice cases - let us check this in detail.

4.1.5 Scaling ansatz for 1d percolation

In the case of one-dimensional percolation, we know that we can write the cluster density exactly as

$$n(s,p) = (1-p)^2 e^{-s/s_\xi} \ . \tag{4.14}$$

We showed that we could rewrite this as

$$n(s,p) = s^{-2} F(\frac{s}{s_\xi}) \ , \tag{4.15}$$

where $F(u) = u^2 e^{-u}$. This is indeed in the general scaling form with $\tau = 2$.

4.1.6 Scaling ansatz for Bethe lattice

For the Bethe lattice we found that the cluster density was approximately on the form

$$n(s,p) \propto s^{-\tau} e^{-s/s_\xi} \ , \tag{4.16}$$

which is already on the wanted form, so that

$$n(s,p) = s^{-\tau} F(s/s_\xi) \ . \tag{4.17}$$

4.2 Consequences of the scaling ansatz

The scaling ansatz is simple, but it has powerful consequences. Here, we address the consequences of the scaling ansatz, and test the validity of the scaling ansatz by comparing the consequences of the scaling ansatz with known and measured results.

4.2.1 Average cluster size

Let us first use the scaling ansatz to calculate the scaling of the average cluster size, and then also of other moments of the cluster size.

The average cluster size is found from

$$S(p) = \sum_s s^2 n(s,p) = \int s^2 n(s,p) ds \ , \tag{4.18}$$

where we now will insert the scaling form for $n(s,p)$

$$n(s,p) = s^{-\tau}\hat{F}((p-p_c)s^{\sigma}) \ , \tag{4.19}$$

where we are now studying the system for $p < p_c$, although an identical calculation can be made for $p > p_c$.

$$S(p) = \int_1^{\infty} s^{2-\tau}\hat{F}((p-p_c)s^{\sigma})ds \ , \tag{4.20}$$

where we substitute $y = s(p_c - p)^{1/\sigma}$:

$$S(p) = (p_c-p)^{\frac{\tau-3}{\sigma}} \int_{(p_c-p)^{1/\sigma}}^{\infty} y^{2-\tau}\hat{F}(-y^{\sigma})dy \ . \tag{4.21}$$

Our scaling assumption is that the scaling function $\hat{F}(u)$ goes exponentially fast to zero - we can therefore replace the integral by an integral with an upper limit 1, and in this range we can replace $\hat{F}$ by a constant.

This implies that the value of the integral is

$$\int_{(p_c-p)^{1/\sigma}}^{1} y^{2-\tau}dy \simeq \Gamma\,(p_c-p)^{\frac{3-\tau}{\sigma}} \ . \tag{4.22}$$

And the result for the average cluster size is therefore

$$S(p) \propto \Gamma(p_c-p)^{\frac{\tau-3}{\sigma}} \propto \frac{\Gamma}{(p_c-p)^{\gamma}} \ , \tag{4.23}$$

which gives a scaling relation for γ:

$$\gamma = \frac{3-\tau}{\sigma} \ . \tag{4.24}$$

We recall that we found that $2 \leq \tau < 3$, which is a result that is valid in all dimensions. Consequently, we notice that γ is positive. As a simple exercise, you can check that this scaling relation holds for the Bethe lattice and one-dimensional percolation.

4.2.2 Density of spanning cluster

We can use a similar argument to find the behavior of $P(p)$, because we have the general relation

$$\sum_s sn(s,p) + P(p) = p \ , \tag{4.25}$$

which is just a general way for formulating that if we pick a site at random, that site is occupied with probability p (right hand side), and this corresponds to a site picked at random to either be in a finite cluster of size s or to be in the infinite cluster.

We can therefore find $P(p)$ from

$$P(p) = p - \sum_s sn(s,p) \ . \tag{4.26}$$

We will now use a standard trick, which is that

$$P(p_c) = p_c - \sum_s sn(s,p_c) = 0 \ . \tag{4.27}$$

Subtracting eq. **??** from eq. 4.27, we find that

$$P(p) = p - p_c - \sum_s s[n(s,p) - n(s,p_c)] \ . \tag{4.28}$$

In this case we are only interested in $p > p_c$, and we can write the sum using our scaling ansatz

$$\sum_s sn(s,p) = \sum_s s^{1-\tau} \hat{F}((p-p_c)s^\sigma) \ . \tag{4.29}$$

$$\sum_s sn(s,p) \simeq (p-p_c)^{\frac{\tau-2}{\sigma}} \int_{(p-p_c)^{1/\sigma}}^{\infty} y^{1-\tau} \hat{F}(y^\sigma) dy \ , \tag{4.30}$$

where we will again use our assumption that $\hat{F}$ has a rapid cross-over and an exponential decay for large y, so that we can write the integral as

$$\int_{(p-p_c)^{1/\sigma)}}^{1} y^{1-\tau} dy \propto c_1 - c_2 (p-p_c)^{\frac{2-\tau}{\sigma}} \ , \tag{4.31}$$

where we again remember that $2 \leq \tau < 3$.

The sum is therefore

$$\sum_s sn(s,p) \propto c_1 (p-p_c)^{\frac{\tau-2}{\sigma}} + c_2 \ . \tag{4.32}$$

Inserting into $P(p)$ produces:

$$P(p) = c_a (p-p_c)^1 + c_b (p-p_c)^{\frac{\tau-2}{\sigma}} \ , \tag{4.33}$$

where the second term is dominating, giving a scaling relation for β since

$$P(p) \propto (p-p_c)^\beta \ . \tag{4.34}$$

$$\beta = \frac{\tau-2}{\sigma} \ . \tag{4.35}$$

We have demonstrated the use of the scaling ansatz for the cluster number density to calculate several measures of interest. Similar calculations can also be made of higher moments of the cluster number density.

4.3 Percolation thresholds

While the exponents are universal – independent of the details of the lattice but dependent on the dimensionality – the percolation threshold, p_c, depends on all the details of the system. The percolation threshold depends on the lattice type and the type of percolation. We typically discern between site percolation, where percolation is on the sites of a lattice, and bond percolation, where the bonds between the sites determines the connectivity. The following table provides our best know values for the percolation thresholds for various dimenions and lattice types. (For $d = 1$, the percolation threshold is $p_c = 1$ for all lattice types.)

Lattice type	Site	Bond
$d = 2$		
Square	0.592746	0.50000
Triangular	0.500000	0.34729
$d = 3$		
Cubic	0.3116	0.2488
FCC	0.198	0.119
BCC	0.246	0.1803
$d = 4$		
Cubic	0.197	0.1601
$d = 5$		
Cubic	0.141	0.1182
$d = 6$		
Cubic	0.107	0.0942
$d = 7$		
Cubic	0.089	0.0787

4.4 Exercises

Exercise 4.1: Generating percolation clusters

In this exercise we will use Matlab to generate and visualize percolation clusters. We generate a $L \times L$ matrix of random numbers, and will examine clusters for a occupation probability p.

We generate the percolation matrix consisting of occupied (1) and unoccupied (0) sites, using

```
L = 100;
r = rand(L,L);
p = 0.6;
z = r<p; % This generates the binary array
[lw,num] = bwlabel(z,4);
```

We have then produced the array `lw` that contains labels for each of the connected clusters, and the variable `num` that contains the number of clusters.

a) Familiarize yourself with labeling by looking at `lw`, and by studying the second example in the Matlab help system on the image analysis toolbox.

We can examine the array directly by mapping the labels onto a color-map, using `label2rgb`.

```
img = label2rgb(lw);
image(img);
```

We can extract information about the labeled image using `regionprops`, for example, we can extract an array of the areas of the clusters using

```
s = regionprops(lw,'Area');
area = cat(1,s.Area);
```

You can also extract information about the `BoundingBox` and other properties of clusters using similar commands

```
s = regionprops(lw,'BoundingBox');
bbox = cat(1,s.BoundingBox);
```

b) Using these features, you should make a program to calculate $P(p, L)$ for various p.

Hint: you can use either `BoundingBox` or `intersect` and `union` to find the spanning cluster.

c) How robust is your algorithm to changes in boundary conditions? Could you do a rectangular grid where $L_x \gg L_y$? Could you do a more complicated set of boundaries? Can you think of a simple method to ensure that you can calculate P for any boundary geometry?

Exercise 4.2: Finding $\Pi(p, L)$ and $P(p, L)$

a) Write a program to find $P(p, L)$ and $\Pi(p, L)$ for $L = 2, 4, 8, 16, 32, 64, 128$. Comment on the number of samples you need to make to det a good estimate for P and Π.

b) Test the program for small L by comparing with the exact results from above. Comment on the results?

Exercise 4.3: Determining β

We know that when $p > p_c$, the probability $P(p, L)$ for a given site to belong to the percolation cluster, has the form

$$P(p, L) \sim (p - p_c)^{\beta} \ . \tag{4.36}$$

Use the data from above to find an expression for β. For this you may need that $p_c = 0.59275$.

Exercise 4.4: Determining the exponent of power-law distributions

In this exercise you will build tools to analyse power-law type probability densities.

Generate the following set of data-points in Matlab:

```
z = rand(1e6,1).^(-3+1);
```

Your task is to determine the distribution function $f_Z(z)$ for this distribution. Hint: the distribution is on the form $f(u) \propto u^{\alpha}$.

a) Find the cumulative distribution, that is, $P(Z > z)$. You can then find the actual distribution from

$$f_Z(z) = \frac{dP(Z > z)}{dz} \ . \tag{4.37}$$

b) Generate a method to do logarithmic binning in Matlab. That is, you estimate the density by doing a histogram with bin-sizes that increase exponentially in size. Hint: Remember to divide by the correct bin-size.

Exercise 4.5: Cluster number density $n(s, p)$

We will generate the cluster number density $n(s, p)$ from the two-dimensional data-set.

a) Estimate $n(s, p)$ for a sequence of p values approaching $p_c = 0.59275$ from above and below.

Hint 1: The cluster sizes are extracted using `.Area` as described in a previous exercise.

Hint 2: Remember to remove the percolating cluster.

Hint 3: Use logarithmic binning.

b) Estimate $n(s, p_c; L)$ for $L = 2^k$ for $k = 4, \ldots, 9$. Use this plot to estimate τ.

c) Can you estimate the scaling of $s_\xi \sim |p - p_c|^{-1/\sigma}$ using this data-set?

Hint 1: Use $n(s, p)/n(s, p_c) = F(s/s_\xi) = 0.5$ as the definition of s_ξ.

Exercise 4.6: Average cluster size

a) Find the average (finite) cluster size $S(p)$ for p close to p_c, for p above and below p_c.

b) Determine the scaling exponent $S(p) \sim |p - p_c|^{-\gamma}$.

c) In what ways can you generate $S^{(k)}(p)$? What do you think is the best way?

Geometry of clusters

5

We have seen how we can characterize clusters by their mass, s. Fig. **??** shows that as p approaches p_c, the typical cluster size s increases. From this figure we also see that the characteristic diameter of the clusters increase. In this chapter we will discuss the geometry of clusters, and by geometry we will mean how the number of sites in a cluster is related to the linear size of the cluster. We will introduce a measure to characterize the spatial extent, the characteristic diameter, of clusters; how the characteristic length behaves as p approaches p_c; and how the characteristic length is related to the characteristic mass, s, of a cluster.

5.1 Characteristic cluster size

We have so far studied the clusters in our model porous material, the percolation system, through the distribution of cluster sizes, $n(s, p)$, and derivatives of this, such as the average cluster size, S and the characteristic cluster size, s_ξ. However, clusters with the same mass, s, can have very different shapes. Fig. 5.1 illustrates three clusters all with $s = 20$ sites. (The linear and the compact clusters are unlikely, but possible realizations). How can we characterize the diameter or radius of these clusters?

There are many ways to define the extent of a cluster. We could, for example, define the maximum distance between any two points in the cluster ($R_{\max}$) to be the extent of the cluster, or we could use the average distance between two points in the cluster. However, we usually introduce a measure which is similar to the standard deviation used to characterize the spread in a random variable: We use the standard deviation in the position, which is also known as the radius of gyration of a cluster:

The radius of gyration, R_i for a particular cluster i of size s_i, with sites $\mathbf{r}_j$ for $j = 1, \ldots, s_i$, is defined as

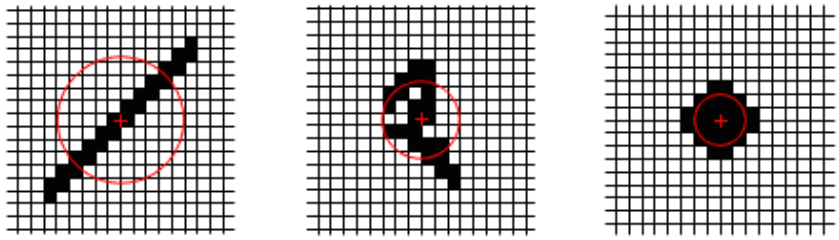

Fig. 5.1 Illustrations of three clusters all with $s = 24$.

$$R_i^2 = \frac{1}{s_i} \sum_{j=1}^{s_i} (\mathbf{r}_j - \mathbf{r}_{cm,i})^2 \ , \tag{5.1}$$

where $\mathbf{r}_{cm,j}$ is the center of mass of cluster i. An equivalent definition is

$$R_i^2 = \frac{1}{2s_i^2} \sum_{n,m} (\mathbf{r}_n - \mathbf{r}_m)^2 \ , \tag{5.2}$$

where the sum is over all sites n and m in cluster i, and we have divided by $2s_i^2$ because each site is counted twice and the number of components in the sum is s_i^2. The radius of gyration of the clusters in Fig. 5.1 is illustrated by the circles in the figures[1].

This provides a measure of the radius of a cluster i. As we see from Fig. 5.1, clusters of the same size s can have different radii. How can we then find a characteristic size for a given cluster size s? We find that by averaging over all clusters of the same size s.

$$R_s^2 = \langle R_i^2 \rangle_i \ . \tag{5.3}$$

where the average is over all clusters of the same size.

5.1.1 Analytical results in one dimension

We can use the one-dimensional percolation system to gain insight into how we expect R_s to depend on s. For the one-dimensional system, there is just one cluster for a given size s corresponding to a line of length s. If the cluster runs from 1 to s, the center of mass is at $s/2$, and the sum over all sites runs from 1 to s:

$$R_s^2 = \frac{1}{s} \sum_{i=1}^{s} (i - s/2)^2 \ , \tag{5.4}$$

[1] Notice that we could have used another moment q to define the radius. Higher moments will put more emphasis on the sites that are far from the center of mass. As the order q approaches infinity, the radius will approach the maximum size of the cluster, $R_{\max}$.

where we assume that s is so large that we only need to address the leading term in s, and we do not have to treat even and odd s separately. This can be expanded to

$$R_s^2 = \frac{1}{s}[\sum_{i=1}^{s} i^2 - is + \frac{s^2}{4}] \tag{5.5}$$

$$= \frac{1}{s}[\frac{s^3}{3} - s\frac{s(s+1)}{2} + s\frac{s^2}{4} \tag{5.6}$$

$$\propto \qquad\qquad\qquad\qquad s^2 \tag{5.7}$$

We have therefore found the result that $s \propto R_s$ in one dimension — which is what we expected.

5.1.2 Numerical results in two dimensions

For the one-dimensional system we found that $s \propto R_s$. How does this generalize to higher dimensions? We start by measuring the behavior for a finite system of size L and with a percolation threshold p. Our strategy is to generate clusters on a $L \times L$ lattice, analyze the clusters, for each cluster, i, of size s_i we will find the center of mass and the radius of gyration, R_i^2. For each value of s we will find the average radius, R_i^2, by a linear average. However, for larger values of s we will collect the data in bins, following the same approach we used to determine $n(s,p)$ — using logarithmic binning.

First, we introduce a function to calculate the radius of gyration of all the clusters in a lattice. This is done in two steps, first we find the center of mass of all clusters, and then we find the radius of gyration. The center of mass for a cluster i with positions $\mathbf{r}_{i,j}$ for $j = 1, \ldots, s_i$, is

$$\mathbf{r}_{cm,i} = \frac{1}{s_i} \sum_{j=1}^{s_i} \mathbf{r}_{i,j} \ , \tag{5.8}$$

We assume that the clusters are numbered and marked in the lattice with their index, as done by the `[lw,num] = bwlabel(m,4)` command. We run through all the sites, `ix,iy`, in the lattice. For each site, we find what cluster i the site belongs to: `i=lw(ix,iy)`. If the site belongs to a cluster, that is if `i>0`, we increase the area of that cluster by one:

```
area(i) = area(i) + 1;
```

and we add the coordinates for this part of the cluster to the sum for the center of mass of the cluster

```
rcm(i,:) = rcm(i,:) + [ix iy];
```

Finally, we find the center of mass for each of clusters by dividing `rcm` by the corresponding area for each of the clusters:

```
rcm(:,1) = rcm(:,1)./area;
rcm(:,2) = rcm(:,2)./area;
```

Second, we follow a similar approach to find the radius of gyration. We run through all the sites in the cluster, and for each site, we find the cluster number i it belongs to, and add the sum of the square of the distance for this site:

```
dr = [ix iy]-rcm(i,:);
rad2(i) = rad2(i) + dot(dr,dr);
```

After running through all the sites, we divide by the area, s_i, to find the radius of gyration according to the formula

$$R_i^2 = \frac{1}{s_i} \sum_{j=1}^{s_i} (\mathbf{r}_{i,j} - \mathbf{r}_{cm,i})^2 \ , \tag{5.9}$$

This is implemented in the following function:

```
function [area,rcm,rad2] = radiusofgyration(lw,num)
%RADIUSOFGYRATION Calculates the radius of gyration of all the clusters
area = zeros(num,1);
rad2 = zeros(num,1);
rcm = zeros(num,2);
% Find center of mass for all the clusters
lwsize = size(lw);
nx = lwsize(1);
ny = lwsize(2);
for ix = 1:nx
    for iy = 1:ny
        ilw = lw(ix,iy);
        if (ilw>0)
            area(ilw) = area(ilw) + 1;
            rcm(ilw,:) = rcm(ilw,:) + [ix iy];
        end
    end
end
rcm(:,1) = rcm(:,1)./area;
rcm(:,2) = rcm(:,2)./area;
% Find radius of gyration for all the clusters
for ix = 1:nx
    for iy = 1:ny
        ilw = lw(ix,iy);
        if (ilw>0)
            dr = [ix iy]-rcm(ilw,:);
            dr2 = dot(dr,dr);
```

```
            rad2(ilw) = rad2(ilw) + dr2;
        end
    end
end
rad2 = rad2./area;
end
```

We use this function to calculate the average radius of gyration for each cluster size s and plot the results using the following script:

```
M = 20; % Nr of samples
L = 400; % System size
p = 0.58; % p-value
allr2 = [];
allarea = [];
for i = 1:M
    z = rand(L,L);
    m = z<p;
    [lw,num] = bwlabel(m,4);
    [area,rcm,rad2] = radiusofgyration(lw,num);
    allr2 = cat(1,allr2,rad2);
    allarea = cat(1,allarea,area);
end
loglog(allarea,allr2,'k.')
```

The resulting plots for several different values of p are shown in Fig. 5.2. We see that that there is an approximately linear relation between R_s^2 and s in this double-logarithmic plot, which indicates that there is a power-law relationship between the two:

$$R_s^2 \propto s^x \ . \tag{5.10}$$

How can we interpret this relation? Equation (5.10) relates the radius R_s and the area (or mass) of the cluster. We are more used to the inverse relation:

$$s \propto R_s^D \ , \tag{5.11}$$

where $D = 2/x$ is the exponent relating the radius to the mass of a cluster. This corresponds to our intuition from geometry. We know that for a cube of size L, the mass (or volume) of the cube is $M = L^3$. For a square of length L, the mass (or area) is $M = L^2$, and similarly for a circle $M = \pi R^2$, where R is the radius of the circle. For a line of length L, the mass is $M = L^1$. We see a general trend, $M \propto R^d$, where R is a characteristic length for the object, and d describes the dimensionality of the object. If we extend this intuition to the relation in (5.11), which is an observation based on Fig. 5.2, we see that we may interpret D as the dimension of the cluster. However, the value of D is not an integer. We have indicated the value of $D = 1.89$ with a dotted line in Fig. 5.2. (The value of D is well know for the two-dimensional percolation problem, see Table 4.1.4). This non-integer value of D may seem strange, but it

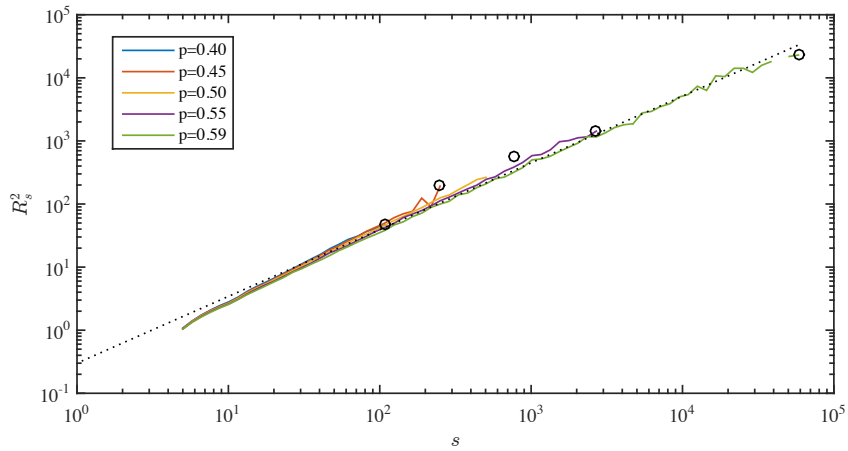

Fig. 5.2 Plot of R_s^2 as function of s for simulations on two-dimensional percolation system with $L = 400$. The largest cluster for each value of p is illustrated by a circle. The dotted line shows the curve $R_s^2 \propto s^{2/D}$ for $D = 1.89$.

is fully possible, mathematically, to have non-integer dimensions. This is a feature frequently found in fractal structures, and the percolation clusters as p approaches p_c is indeed a good example of a self-similar fractal. We will return to this aspect of the geometry of the percolation system in Sect. **??**.

The largest cluster and its corresponding radius of gyration is indicated by a circle for each p value in Fig. 5.2. We see that as p approaches p_c, both the area and the radius of the largest cluster increases. Indeed, this corresponds to the observation we have previously made for the characteristic cluster size, s_ξ. We may define a corresponding characteristic cluster radius, R_{s_ξ}. This gives:

$$s_\xi \propto R_{s_\xi}^D \ . \tag{5.12}$$

This length is a characteristic length for the system at a given value of p, corresponding to the largest cluster size or the typical cluster size in the system. In Sect. 5.2 we see how we can relate this length directly to the cluster size distribution.

5.1.3 Scaling behavior in two dimensions

We have already found that the characteristic cluster size s_ξ diverges as a power law as p approaches p_c:

$$s_\xi \simeq s_0 \left(p - p_c\right)^{-1/\sigma} \ , \tag{5.13}$$

when $p < p_c$. The behavior is similar when $p > p_c$, but the prefactor s_0 may have a different value. How does R_{s_ξ} behave when p approaches p_c? We can find this by combining the scaling relations for s_ξ and R_{s_ξ}. We

remember that $R_{s_\xi} \propto s_\xi^{1/D}$. Therefore

$$R_{s_\xi} \propto s_\xi^{1/D} \propto \left((p-p_c)^{-1/\sigma}\right)^{1/D} \propto (p-p_c)^{-1/\sigma D} \ , \tag{5.14}$$

where we introduce the symbol $\nu = 1/(\sigma D)$. For two-dimensional percolation the exponent ν is a universal number, just like σ and D. This means that it does not depend on details such as the lattice type or the connectivity of the lattice, although it does depend on the dimensionality of the system. We know the value of ν reasonably well in two dimensions, $\nu = 4/3$. For values in other dimensions see Table 4.1.4.

The arguments we have provided here is again an example of scaling argument. In these arguments we are only interested in the exponent in the scaling relation, the functional form, and not in the values of the constant prefactors.

5.2 Geometry of finite clusters

We have defined the characteristic length R_{s_ξ} through the definition of the characteristic cluster size, s_ξ, and the scaling relation $s \propto R_s^D$. However, it may be more natural to define the characteristic length of the system as the *average* radius and not the *cut-off* radius. We have introduced several averages for the radius of gyration. For each cluster i we can calculate the radius of gyration, R_i. We can then find the average radius of gyration for a cluster of size s by averaging over all clusters i of size s:

$$R_s^2 = \langle R_i^2 \rangle_i \ , \tag{5.15}$$

where the average is over all clusters i of the same size s. This gives us the radius of curvature R_s which we found to scale with cluster mass s as $s \propto R_s^D$.

For the cluster sizes, we introduced an average cluster size S, which is

$$S = \frac{1}{Z_S} \sum_s s\, sn(s,p) \ , \ Z_S = \sum_s sn(s,p) \ . \tag{5.16}$$

We can also similarly introduce an average radius of gyration, R, by averaging R_s over all cluster sizes:

$$R = \frac{1}{Z_R} \sum_s R_s^2 s^k sn(s,p) \ , \ Z_R = \sum_s s^k sn(s,p) \ . \tag{5.17}$$

Here, we have purposely introduced an unknown exponent k. We are to some extent free to choose this exponent, although the average needs to be finite, and the exponent will determine how we small and large clusters are weighed in the sum. A natural choise may be to choose $k = 1$ so that we get terms $R_s^2 s^2 n(s,p)$ in the sum. However, the results we

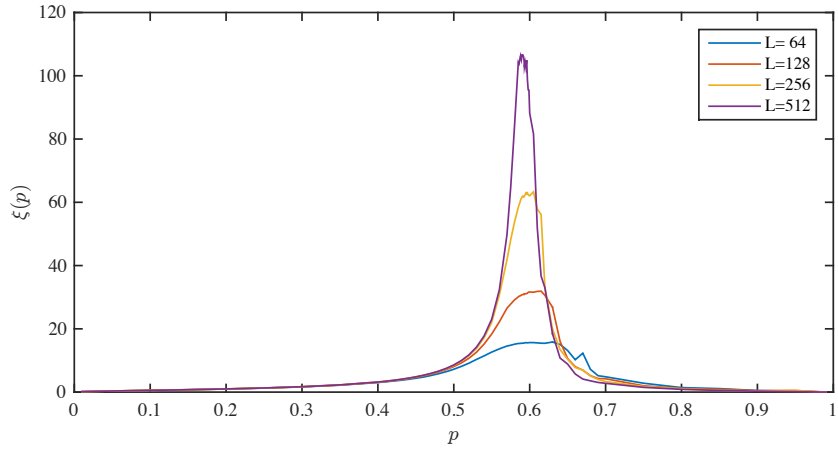

Fig. 5.3 A plot of ξ as a function of p for a $L = 64$, 128, 256 and 512 system as a function of p. We observe that ξ diverges when $p \to p_c$. We notice that the correlation length does not diverge, but crosses over as a result of the finite system size.

present here will not change in any significant way, expect for different prefactors to the scaling relations, if you choose a larger value of k. Our definition of the average radius of gyration is therefore:

$$R = \frac{1}{Z_R} \sum_s R_s^2 s^2 n(s,p) \ , \ Z_R = \sum_s s^2 n(s,p) \ , \tag{5.18}$$

where we notice that the normalization sum $Z_R = S$ is the average cluster size.

Fig. 5.3 shows a plot of the average R as a function of p for various systems sizes L. We see that R diverges as p approaches p_c. How can we develop a theory for this behavior?

We know that the cluster number density $n(s,p)$ has the approximate scaling form

$$n(s,p) = s^{-\tau} F\left(s/s_\xi\right) \ , \ s_\xi \propto |p - p_c|^{-1/\sigma} \ . \tag{5.19}$$

We can use this to calculate the average radius of gyration, R, when p is close to p_c.

The average radius of gyration is

$$R^2 = \frac{\sum_s R_s^2 s^2 n(s,p)}{\sum_s s^2 n(s,p)} = \frac{\int_1^\infty R_s^2 s^{2-\tau} F(s/s_\xi) ds}{\int_1^\infty s^{2-\tau} F(s/s_\xi) ds} \tag{5.20}$$

$$\propto \frac{\int_1^\infty s^{2/D} s^{2-\tau} F(s/s_\xi) ds}{\int_1^\infty s^{2-\tau} F(s/s_\xi) ds} \ , \tag{5.21}$$

where we have inserted $R_s^2 \propto s^{2/D}$. This expression is valid when $s < s_\xi$. We insert it here since $F(s/s_\xi)$ goes rapidly to zero when $s > s_\xi$, and therefore only the $s < s_\xi$ values will contribute significantly to the

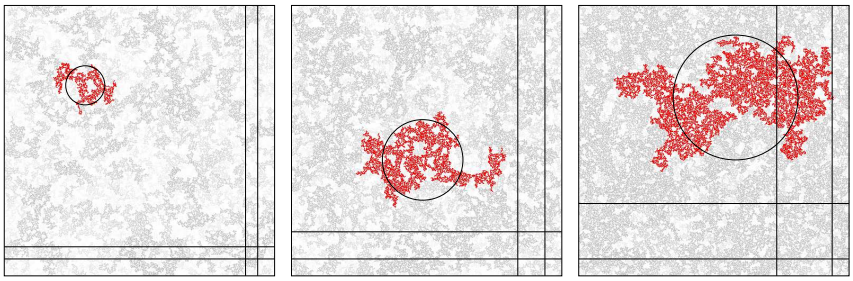

Fig. 5.4 Illustration of the largest cluster in 512×512 systems for $p = 0.55$, $p = 0.57$, and $p = 0.59$. The circles illustrate the radius of gyration of the largest cluster, and the boxes show the size of the average radius of gyration, $R = \langle R_s \rangle$. We observe that both lengths increase approximately proportionally as p approaches p_c.

integral. We change variables to $u = s/s_\xi$, getting:

$$R^2 \propto \frac{s_\xi^{2/D+3-\tau} \int_{1/s_\xi}^{\infty} u^{2/D+2-\tau} F(u) du}{s_\xi^{3-\tau} \int_{1/s_\xi}^{\infty} u^{2-\tau} F(u) du} \tag{5.22}$$

$$\propto s_\xi^{2/D} \frac{\int_0^{\infty} u^{2/D+2-\tau} F(u) du}{\int_0^{\infty} u^{2-\tau} F(u) du} \propto s_\xi^{2/D} \ , \tag{5.23}$$

where the two integrals over $F(u)$ simply are numbers, and therefore have been included in the constant of proportionality.

This shows that $R^2 \propto s_\xi^{2/D}$. We found above that $R_{s_\xi} \propto s_\xi^{2/D}$. Therefore, $R \propto R_{s_\xi}$! These two characteristic lengths therefore have the same behavior. They are only different by a constant of proportionality, $R = c\, R_{s_\xi}$. We can therefore use either length to characterize the system — they are effectively the same.

Fig. 5.4 illustrates the radius of gyration of the largest cluster with a circle and the average radius of gyration, R, indicated by the length of the side of the square. As p increases, both the maximum cluster size and the average cluster size increases — according to the theory they are indeed proportional to each other and therefore increase in concert.

5.2.1 Correlation length

We can also measure the typical size of a cluster from the correlation function. The correlation function $g(r, p)$, which is the probability that two sites, which are a distance r apart, are connected and part of the same cluster for a system with occupation probability p. We can use this to define the average squared distance between two sites i and j belonging to the same cluster as

$$\xi = \langle \frac{\sum_j r_{ij}^2 g(r_{ij}; p)}{\sum_j g(r_{ij}; p)} \rangle_i \ . \tag{5.24}$$

where the sum is over all sites j and the average is also over all sites i. The denominator is a normalization sum, which corresponds to the average cluster size, S. You can think of this sum in the following way: For a site i, we sum over all other sites j in the system. The probability that site j belongs to the same cluster as site i is $g(r_{ij};p)$, and the mass of the site at j is 1. The average number of sites connected to site at i is therefore:

$$S(p) = \langle \sum_j g(r_{ij};p) \ = \sum_j g(r_{ij};p) \rangle_i \ , \tag{5.25}$$

where we average over all the the sites i in the system.

This means that we can connect $g(r;p)$ and ξ to the average cluster size S. Let us now see if we can calculate the behavior of $g(r;p)$ in a one-dimensional system, how to measure it in a two-dimensional system, and how to develop a theory for it for any dimension.

One-dimensional system

In Sect. we found that for the one-dimensional system the correlation function $g(r)$ is

$$g(r) = p^r = e^{-r/\xi} \ , \tag{5.26}$$

where $\xi = -\frac{1}{\ln p} =\simeq 1/(1-p_c)$ is called the correlation length. The correlation length diverges as $p \to p_c = 1$, $\xi \simeq (1-p_c)^{-\nu}$, where $\nu = 1$.

We can generalize this behavior by writing the correlation function in a more general scaling form for the one-dimensional system

$$g(r) = r^0 f(r/\xi) \ , \tag{5.27}$$

where $f(u)$ decays rapidly when u is larger than 1. We will assume that this behavior is general. Also for other dimensions, we expect the correlation function to decay rapidly beyond a length, which corresponds to the typical extent of clusters in the system.

Measuring the correlation function

For the two- or three-dimensional system, we cannot find an exact solution for the correlation function. However, we can still measure it from our simulations, although such measurements typically are computationally intensive. How can we measure it? We can loop through all sites i and j and find their distance r_{ij}. We estimate the probability for two sites at a distance r_{ij} to be connected to count how many of the sites that are a distance r_{ij} apart are connected, compared to how many sites in total are a distance r_{ij} apart. This is done through the following implementation, which returns the correlation function `g(r)` estimated for a lattice `lw` which contains the cluster indexes for each site, similar

to what is returned by the `[lw,num] = bwlabel(m,4)` command. First, we write a subroutine to find the correlation function for a given lattice `lw`:

```
function [r,pr] = perccorrfunc(lw)
%PERCCORRFUNC calculates the correlation function based on of all
%the clusters
    s = size(lw); nx = s(1); ny = s(2);
    L = max([nx ny])
    r = (1:2*L)';
    pr = zeros(2*L,1);
    npr = zeros(2*L,1);
    for ix1 = 1:nx
        for iy1 = 1:ny
            lw1 = lw(ix1,iy1);
            for ix2 = 1:nx
                for iy2 = 1:ny
                    lw2 = lw(ix2,iy2);
                    dx = (ix2-ix1);
                    dy = (iy2-iy1);
                    rr = hypot(dx,dy);
                    nr = ceil(rr)+1;
                    pr(nr) = pr(nr) + (lw1==lw2)*(lw1>0);
                    npr(nr) = npr(nr) + 1;
                end
            end
        end
    end
    pr = pr./npr;
end
```

and then we use this function to find the correlation function for several values of p and for several values of L:

```
M = 1; % Nr of samples
L = 100; % System size
pp = [0.55 0.57 0.59]; % p-value
lenpp = length(pp);
pr = zeros(2*L,lenpp);
rr = zeros(2*L,lenpp);
for i = 1:M
    z = rand(L,L);
    for ip = 1:lenpp
        p = pp(ip)
        m = z<p;
        [lw,num] = bwlabel(m,4);
        [r,g] = perccorrfunc(lw);
        pr(:,ip) = pr(:,ip) + g;
        rr(:,ip) = rr(:,ip) + r;
    end
end
pr = pr/M;
r = r/M;
% Plot data - linearly binned
for ip = 1:lenpp
    loglog((1:2*L-1),pr(2:end,ip),'+')
    hold all
```

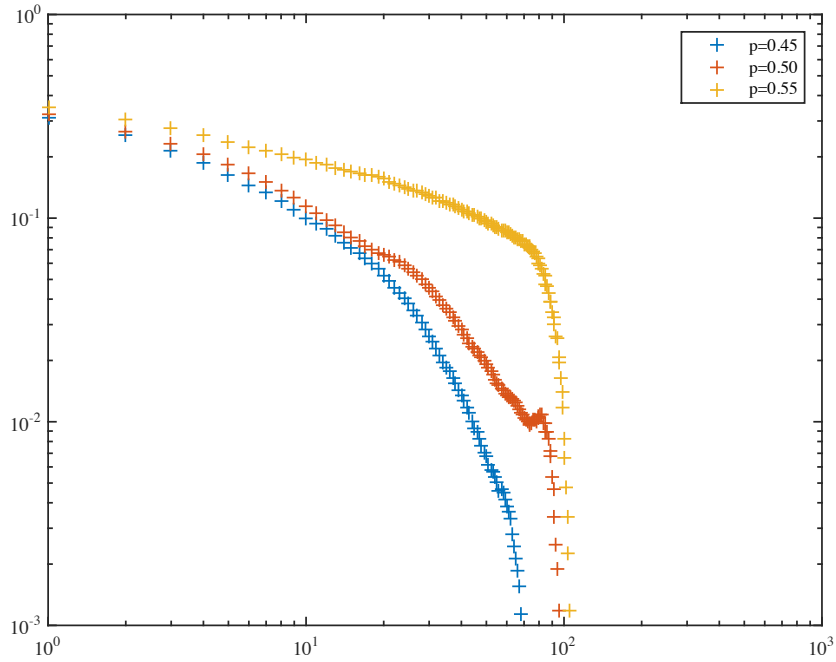

Fig. 5.5 A plot of $g(r;p)$ as a function of r for various values of p. The function approaches a power-law behavior $g(r) \propto r^x$ when p approaches p_c.

```
end
```

Fig. 5.5 shows the resulting plots of the correlation function $g(r;p)$ for various values of p for an $L = 400$ system. This plot shows that there is indeed a cross-over length ξ, beyond which the correlation function falls rapidly to zero. But there appears to be a scaling regime for $r > \xi$ where the correlation function is approximately a power-law. The plot suggests the following functional form

$$g(r;p) = r^x f(r/\xi) \ , \tag{5.28}$$

where the cross-over function $f(u)$ falls rapdily to zero when $u > 1$ and is approximately constant when $u < 1$. When p approaches p_c, the correlation length ξ grows to infinity, and the correlation function $g(r;p_c)$ approaches a power-law r^x for all values of r.

Theory for the correlation function

Based on these observations, we are motivated to develop a theory for the behavior of the correlation function. First, we know that when $p = p_c$, the average cluster size, S diverges. We can express S using the correlation function as

$$S = \sum_j g(r, p_c) = \int g(r)\, dr^d = \iint g(r) r^{d-1} dr d\Omega \ , \tag{5.29}$$

where the integral is written in spherical coordinates in a d-dimensional space, and the integration over Ω indicates an integration over all angles. For this integral to diverge, the function $g(r)$ cannot have an exponential cut-off, and it needs to diverge slower than a power-law with exponent $-d$. That is, in order for S to diverge at $p = p_c$, we know that at $p = p_c$:

$$g(r; p_c) \propto r^{-(d-2+\eta)} \ , \tag{5.30}$$

where η is a positive number, randing from $\eta = 0$ for the Bethe lattice (infinite dimensions) to $\eta = 1$ for one-dimensional percolation, as we found above.

This corresponds both to the results we found for the one-dimensional system, and to the results we found from numerical measurements for the two-dimensional system. In addition, we know that for $p \neq p_c$, the correlation function should have a cut-off proportional to ξ, because the probability for two sites to be connected goes exponentially to zero with distance when the distance is significantly larger than ξ. These features indicates that $g(r, p)$ has a scaling form, and we propose the following scaling ansatz for $g(r, p)$:

$$g(r, p) = r^{-(d-2+\eta)} f(\frac{r}{\xi}) \ . \tag{5.31}$$

The scaling function $f(r/\xi)$ should be a constant when $r \ll \xi$, and in this range we cannot discern the behavior from the behavior of a system at p_c. For $r \gg \xi$, we expect the function to have an exponential form. The scaling function will therefore have the following behavior:

$$f(u) = \begin{cases} \text{constant} & \text{when } u \ll 1 \\ \exp(-u) & \text{when} u \gg 1 \end{cases} \ . \tag{5.32}$$

We can use this scaling form to determine the exponent η. We know that the average cluster size S is given as an integral over $g(r; p)$, that is

$$S = \sum_j g(r; p) = \int g(r; p) dr \ . \tag{5.33}$$

Let us use the scaling form for $g(r; p)$ to calculate this integral when p approaches p_c, but is not equal to p_c.

$$S = \int g(r;p)dr = \int_1^\infty r^{-(d-2+\eta)} f(r/\xi)dr^d \tag{5.34}$$

$$= \int_1^\infty r^{-(d-2+\eta} r^{d+1} \exp(-r/\xi)drd\Omega \propto \int_1^\infty r^{1-\eta} \exp(-r/\xi)dr \tag{5.35}$$

$$= \xi^{2-\eta} \int (\frac{r}{\xi})^{1-\eta} \exp(-r/\xi)\frac{dr}{\xi} = \xi^{2-\eta} \int u^{1-\eta} \exp(-u)du \propto \xi^{2-\eta} \tag{5.36}$$

We already know the scaling behavior of S when $p \to p_c$:

$$S \propto |p - p_c|^{-\gamma} \propto \xi^{2-\eta} , \tag{5.37}$$

Consequently, we now know the behavior of ξ:

$$\xi \propto |p - p_c|^{-\gamma/(2-\eta)} , \tag{5.38}$$

where η is a number between 0 (for the infinite-dimensional system) and 1 (for the one-dimensional system). Indeed we remember that for the one-dimensional system we found that $\xi \propto |p - p_c|^{-1}$ and that $\gamma = 1$, which is indeed consistent with $\eta = 1$.

What does this teach us about the two- and three-dimensional system? For these systems, we already have related the average cluster size to the average radius of gyration, R:

$$S \propto s_\xi^{3-\tau} \propto R^{(3-\tau)/D} , \tag{5.39}$$

and we know that the average radius of gyration behaves as

$$R \propto R_{s_\xi} \propto s_\xi^{1/D} \propto |p - p_c|^{-1/\sigma D} . \tag{5.40}$$

We interpret both ξ and R (and R_{s_ξ}) as characteristic lengths. Let us now make a daring assumption! Let us assume that ξ and R are proportional — that there is only one characteristic length in the system. This allows us to write:

$$R \propto |p - p_c|^{-1/\sigma D} \propto |p - p_c|^{-\gamma/(2-\eta)} . \tag{5.41}$$

We can use this relation to find η, given that the assumption of $\xi \propto R$ is correct, or to demonstrate that $\xi \propto R$ by measuring η and checking for consistency with this equation.

We have already done this for the one-dimensional system, where $\sigma = D = 1$ and $\gamma = 1$, and therefore $\eta = 1$, which is indeed what we found above. Similarly, we can check this result for the Bethe-lattice, where we also find that the assumption holds. Simulations and theoretical arguments indeed support the assumption. We will therefore in the following only use one symbol for all the characteristic lengths since they are proportional to each other and therefore only differ (scaling-wise) by a constant of proportionality:

$$\xi \propto R \propto R_{s_\xi} \propto |p - p_c|^{-\nu} \, . \tag{5.42}$$

We will typically only use the symbol ξ for this characteristic length of the system.

The characteristic length ξ and system size L

The introduction of a single characteristic length ξ, corresponding to the characteristic cluster size s_ξ through $s_\xi \propto \xi^D$, allows us to discuss what happens to a system that is close to, but not exactly at, p_c. Fig. **??** shows a plot of $\xi(p)$ for two-dimensional systems with $L = 100$, 200, and 400. Notice that since ξ diverges as p approaches p_c, and we are in a finite system of size L, we will not observe clusters that are larger than L. This means that if we measure $\xi(p)$ and we try to estimate p_c we only know that it is somewhere in the region where $\xi(p) > L$, but we do not really know where. This also means that if we are studying a system where p is different from, but close to p_c, we need to study clusters that are at least of the size of ξ in order to notice that we are not at $p = p_c$.

If we study a system of size $L \ll \xi$, we will typically observe a cluster that spans the system, since the typical cluster size, ξ, is larger than the system size. We are therefore not able to determine if we observe a spanning cluster because we are at p_c or only because we are sufficiently close to p_c. We will start to observe a spanning cluster when $\xi \simeq L$, which corresponds to

$$\xi_-(p_c - p)^{-\nu} = \xi \simeq L \, , \tag{5.43}$$

and therefore that

$$(p_c - p) \simeq (L/\xi_-)^{-(1/\nu)} \, , \tag{5.44}$$

when $p < p_c$, and a similar expression for $p > p_c$. This means that when we observe spanning we can only be sure that p is within a certain range of p_c:

$$|p - p_c| = cL^{-1/\nu} \, . \tag{5.45}$$

The correlation length ξ is therefore the natural length characterizing the geometry of the cluster. At distances smaller than ξ, the system behaves as if it is at $p = p_c$. However, at distances much larger than ξ, the system is essentially homogeneous.

As we can observe in fig. 5.6 the system becomes more and more homogeneous when p goes away from p_c. We will now address this feature in more detail when $p > p_c$.

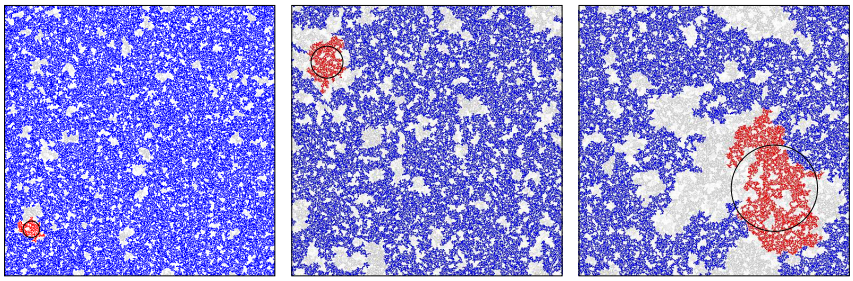

Fig. 5.6 Illustration of the largest cluster in 512×512 systems with $p > p_c$, for $p = 0.593$, $p = 0.596$, and $p = 0.610$. The circles illustrate the radius of gyration of the largest cluster. We observe that the radius of gyration increases as p approaches p_c.

5.3 Geometry of the spanning cluster

How can we develop a scaling theory for the spanning cluster? As p is increased from below towards p_c, the characteristic cluster size ξ diverges. However, the size of a characteristic cluster of size ξ is expected to follow the scaling relation $s_\xi \propto \xi^D$. For a given value of p we can therefore choose the system size L to be equal to ξ, $L = \xi(p)$. In this case, a cluster of size ξ would correspond to a cluster of size L, which is a spanning cluster in this system. For this system of size $L = \xi$, we therefore expect the mass of the spanning cluster to be $M(p, L) \propto \xi^D \propto L^D$. This suggests (but does not really prove) that the mass of the spanning cluster in a system close to or at p_c scales as $M(p, L) \propto L^D$.

The density of the spanning cluster at $p = p_c$ therefore has the following behavior:

$$P(p, L) = \frac{M(p, L)}{L^d} \propto L^D / L^d \propto L^{D-d} \,. \tag{5.46}$$

Because we know that $P(p, L) \to 0$ when $L \to \infty$, we deduce that $D < d$. The value of D in two-dimensional percolation is $D = 91/48 \simeq 1.90$. Values for other systems can be found in Table 4.1.4.

Fractal geometry of the spanning cluster

What does this result tell us about the geometry of the percolation cluster? First, we observe that the density of the cluster depends on the system size, L, on which we are observing it. This is a general feature of a fractal with a dimension different from the Euclidean dimension in which it is embedded. For any object that obeys the scaling relation

$$M \propto L^D \,, \tag{5.47}$$

where $D < d$, and d is the dimension of the Euclidean dimension, we have that the density ρ is

$$\rho \propto \frac{M}{L^d} \propto L^{D-d} \,, \tag{5.48}$$

which depends on system size L. We also notice that the density decreases as the system size increases.

Notice that these features do not represent something new, but are simply extensions of features we are very well familiar with. For example, consider a thin, flat sheet of thickness h, and dimensions $\mathcal{L} \times \mathcal{L}$, placed in a three-dimensional space. If we cut out a volume of size $L \times L \times L$, so that $L \ll \mathcal{L}$, the mass of the sheet inside that volume is

$$M = hL^2 \,, \tag{5.49}$$

which implies that the density of the sheet is

$$\rho = \frac{hL^2}{L^3} = hL^{-1} \,. \tag{5.50}$$

It is only in the case when we use a two-dimensional volume $L \times L$ with a third dimension of constant thickness H larger than h, that we recover a constant density ρ independent of system size.

5.4 Spanning cluster above p_c

Let us now return to the discussion of the mass $M(p, L)$ of the spanning cluster for $p > p_c$ in a finite system of size L. The behavior of the percolation system for $p > p_c$ is illustrated in fig. 5.6. We notice that the correlation length ξ diverges when p approaches p_c. At lengths larger than ξ, the system is effectively homogeneous because there are no holes significantly larger than ξ. There are two types of behavior, depending on whether L is larger than or smaller than the correlation length ξ.

When $L \ll \xi$, we are again in the situation where we cannot discern p from p_c because the size of the holes (empty regions described by ξ when $p > p_c$) in the percolation cluster is much larger than the system size. In this case, the mass of the percolation cluster will follow the scaling relation $s \propto R_s^D$, and the finite section of size L of the cluster will follow the same scaling if we assume that the radius of gyration of the cluster inside a region of size L is proportional to L:

$$M(p, L) \propto L^D \text{ when } L \ll \xi \,. \tag{5.51}$$

In the other case, when $L \gg \xi$, and $p > p_c$, the typical size of a hole in the percolation cluster is ξ, as illustrated in fig. 5.6. This means that on lengths much larger than ξ, the percolation cluster is effectively homogeneous. We can therefore divide the $L \times L$ system into $(L/\xi)^d$ regions of size ξ, so that for each such region, the mass if $m \propto \xi^D$. The

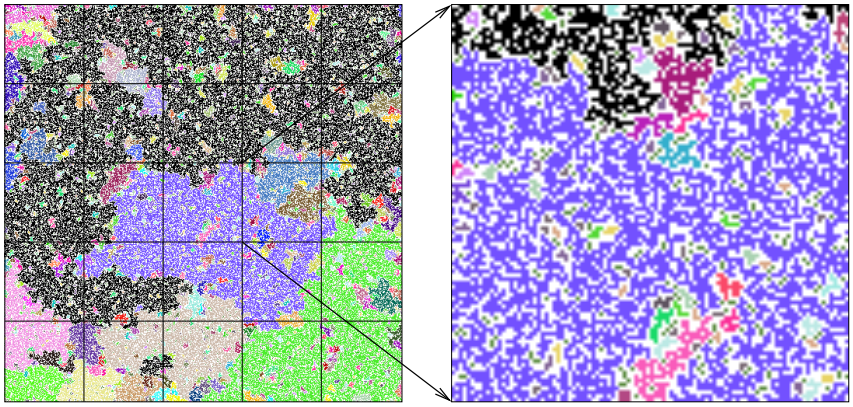

Fig. 5.7 Illustration of the spanning cluster in a 512×512 system at $p = 0.595 > p_c$. In this case, the correlation length is $\xi = 102$. The system is divided into regions of size ξ. Each such region has a mass $M(p, \xi) \propto \xi^D$, and there are $(L/\xi)^d \simeq 25$ such regions in the system.

total mass of the spanning cluster is therefore the mass of one such region multiplied with the number of regions:

$$M(p, L) \propto (\xi^D)(L/\xi)^d \propto \xi^{D-d} L^d \, . \tag{5.52}$$

We can now introduce the complete behavior of the mass, $M(p, L)$, of the spanning cluster for $p > p_c$:

$$M(p, L) \propto \begin{cases} L^D & L \ll \xi \\ \xi^{D-d} L^d & L \gg \xi \end{cases} \, . \tag{5.53}$$

This form can be rewritten in the standard scaling form as:

$$M(p, L) = L^D Y(\frac{L}{\xi}) \, , \tag{5.54}$$

where

$$Y(u) = \begin{cases} \text{constant} & u \ll 1 \\ u^{d-D} & u \gg 1 \end{cases} \, . \tag{5.55}$$

5.5 Fractal cluster geometry

What happens to the scaling behavior of the system if we change the effective length-scale by a factor b?. That is, what happens if we introduce a new set of variables $\xi' = \xi/b$, and $L' = L/b$.

We can use our scaling form $M(p, L) = L^D Y(L/\xi)$, to find that

$$M(p', L') = (L')^D Y(L'/\xi') = (L/b)^D Y(L/\xi) = b^{-D} M(p, L) \, , \tag{5.56}$$

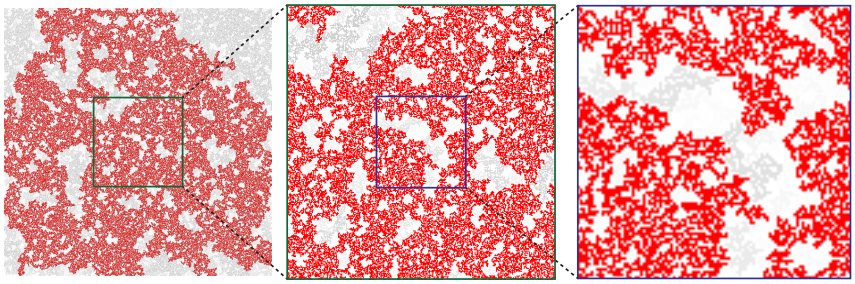

Fig. 5.8 Illustrations of the spanning cluster (shown in red), and the other clusters (shown in gray) at $p = p_c$ in a $L = 900$ site percolation system. **a** The 900×900 system. **b** The central 300×300, and part. **c** The central 100×100. Each step represents a rescaling by a factor 3. However, at $p = p_c$, the correlation length is infinite, so a rescaling of the length-scales should not influence the geometry of the cluster, which is evident from the pictures: The percolation clusters are indeed similar in a statistical manner.

where we have written p' to indicate that a rescaling of the correlation length corresponds to a change in p - reducing the correlation length corresponds to moving p further away from p_c.

This shows that the mass displays a simple rescaling when the system size is rescaled - functions that display this simple form of rescaling are called homogeneous functions.

The change of length-scale results in a change of correlation length, except for the cases when the correlation length is either zero or infinity. The correlation length is zero for $p = 0$, and for $p = 1$. These two values of p therefore corresponds to trivial fix-points for the rescaling: The scaling behavior does not change under this rescaling. The correlation length is infinite for $p = p_c$, which implies that the correlation length does not change when the system size is rescaled by a factor b. This is illustrated in fig. 5.8, which shows that the structure of the percolation cluster at $p = p_c$ does not change significant.

Self-similar fractals

The spanning cluster shows a particular simple scaling behavior at $p = p_c$. That is when the correlation length increases to infinity — there is therefore no other length-scale in our system except the system size L and the lattice unit a. When $p = p_c$ we found that the mass of the spanning cluster displayed the scaling relation:

$$M(L) = b^{-D} M(bL) \,, \tag{5.57}$$

corresponding to a rescaling by a factor b. This is an example of self-similar scaling.

Let us address self-similar scaling in more detail by addressing an example of a deterministic fractal, the Sierpinski gasket [**?**]. The Sierpinski gasket can be defined iteratively. We start with a unit equilateral triangle

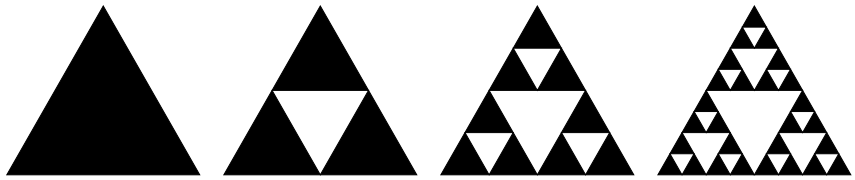

Fig. 5.9 Illustration of three generations of the Sierpinski gasket starting from an equilateral triangle.

as illustrated in Fig. 5.9. We divide the triangle into 4 identical triangles, and remove the center triangle. For each of the remaining triangles, we continue this process. The result set of points after infinitely many iterations is called the Sierpinski gasket. This set contains a hierarchy of holes. We also notice that the structure is identical under (a specific) dilational rescaling. If we take one of the tree triangles generated in the first step and rescale it to fit on top of the initial triangle, we see that it reproduces the original identically. This structure is therefore a fractal.

The dimensionality of the structure is related to the relation between the rescaling of the mass and the length. If we take one of the three triangles from the first iteration, we need to rescale the x and the y axes by a factor 2. We can write this as a rescaling of the system size, L, by a factor 2

$$L' = 2L \; . \tag{5.58}$$

Through this rescaling we get three triangles, each with the same mass as the original triangle. The mass is therefore rescaled by a factor 3.

$$M' = 3M \; . \tag{5.59}$$

If we write the mass as a function of length, $M(L)$, we can formulate the scaling as

$$M(2L) = 3M(L) \; , \tag{5.60}$$

or, equivalently,

$$M(L) = 3^{-1} M(2L) \; . \tag{5.61}$$

If we compare this with the general relation,

$$M(L) = b^{-D} M(bL) \; , \tag{5.62}$$

we see that

$$2^{-D} = 3^{-1} \; , \tag{5.63}$$

giving

$$D = \frac{\ln 3}{\ln 2} \; . \tag{5.64}$$

We will use this rescaling relation as our definition of fractal dimension. The relation corresponds to the relation $M = L^D$ for the mass. Let us also show that this relation is indeed consistent with our notion of Euclidean

dimension. For a cube of size L, the mass is L^3. If we look at a piece of size $(L/2)^3$, we see that we need to rescale it by a factor of 2 in all direction to get back to the original cube, but the mass must be rescaled by a factor 8. We will therefore find the dimension from

$$D = \frac{\ln 8}{\ln 2} = 3 \ , \tag{5.65}$$

which is, as expected, the Euclidean dimension of the cube.

Typically, the mass dimension is measured by box counting . The sample is divided into regular boxes where the size of each side of the box is δ. The number of boxes, $N(\delta)$, that contain the cluster are counted as a function of δ. For a uniform mass we expect

$$N(\delta) = (\frac{L}{\delta})^d \ , \tag{5.66}$$

and for a fractal structure we expect

$$N(\delta) = (\frac{L}{\delta})^D \ , \tag{5.67}$$

We leave it as an exercise for the reader to address what happens when $\delta \to 1$, and when $\delta \to L$.

5.6 Exercises

Exercise 5.1: Mass scaling of percolating cluster

a) Find the mass $M(L)$ of the percolating cluster at $p = p_c$ as a function of L, for $L = 2^k$, $k = 4, \ldots, 11$.

b) Plot $\log(M)$ as a function of $\log(L)$.

c) Determine the exponent D.

Exercise 5.2: Correlation function

a) Write a program to find the correlation function, $g(r, p, L)$ for $L = 256$.

b) Plot $g(r, p, L)$ for $p = 0.55$ to $p = 0.65$ for $L = 256$.

c) Find the correlation length $\xi(p, L)$ for $L = 256$ for the p-values used above.

d) Plot ξ as a fuction of $p - p_c$, and determine ν.

Finite size scaling

6

6.1 Overview

From our discussion of the correlation length, we learned that the percolation system has three intrinsic length-scales: the size of a site, the system size L, and the correlation length ξ. Finite size scaling provides a theoretical framework to address the effect of finite system sizes in percolation systems. However, the techniques developed here can be extended also to phase transitions and critical phenomena in general, to the study of fractals, and to other systems that display scaling.

Finite size scaling addresses the change in behavior of a system with a finite system size. Typically, we divide the behavior into two main categories: when the system size L is much smaller than the correlation length ξ, ($L \ll \xi$), the system appears to be on the percolation threshold. However, when L is much larger than ξ, the geometry is essentially homogeneous at lengths longer than ξ.

Typically, we are interested studying the thermodynamic limit ($L \to \infty$) of a quantity $X(p)$ which behaves as a power-law when $p \to p_c$. Examples are: $P(p)$, $S(p)$, and $M(p)$, and we will later see that also most physical properties such as the conductivity of the system behaves similarly. That is

$$X(p) \propto (p - p_c)^{-\gamma_x} \ , \tag{6.1}$$

where the exponent γ_x determines the behavior close to p_c.

We will study such a system by making the finite size scaling ansatz:

$$X(p, L) = L^{\frac{\gamma_x}{\nu}} \mathcal{X}\left(\frac{L}{\xi}\right) \ , \tag{6.2}$$

or, equivalently,

$$X(p, L) = \xi^{\frac{\gamma_x}{\nu}} \tilde{\mathcal{X}}\left(\frac{L}{\xi}\right) \ , \tag{6.3}$$

where we have left it as an exercise for the reader to show the equivalence of the two expressions.

We will then infer the behavior in the limits when $\xi \gg L$, and $\xi \ll L$ to determine the form of the scaling function $\mathcal{X}(u)$, and use this functional form as a tool to study the behavior of the system.

6.2 Finite size

So far, we have chosen to ignore the effects of finite lattice sizes. However, our discussion of correlation lengths shows us that we cannot ignore a finite lattice size, even if the lattice size is large. When p approaches p_c the percolation system will eventually reach a correlation length ξ that exceeds the finite system size, and from that point we cannot discern a finite from an infinite cluster. We will therefore need a systematic way to handle the effect of a finite system size. We will develop such a systematic approach using the methods we now have developed to address the cross-over of the cluster mass.

6.3 Spanning cluster

Let us now redo the discussion for the mass M and the spanning cluster density P, but with a finite system size L. We will look at the spanning cluster on a finite lattice , and address the mass $M(p, L)$, which is related to $P(p, L)$ according to:

$$P(p, L) = \frac{M(p, L)}{L^d} \ . \tag{6.4}$$

Our approach to finite size scaling problems will always be to assume that the finite size enters through a scaling function modifying the asymptotic behavior. That is, we will assume that

$$P(p, L) = (p - p_c)^\beta X(\frac{L}{\xi}) \ . \tag{6.5}$$

In the case when $L \to \infty$ we want to recover the result from the thermodynamic limit, that is, we expect that $X(u)$ is a constant when $u \gg 1$.

What behavior do we expect in the limit when $\xi \gg L$. This is the case when the correlation length is much larger than the system size, and we expect the behavior to be independent of the correlation length, but only depend on L. That is, we expect that

$$P(p, L) = (p - p_c)^\beta X(\frac{L}{\xi}) = \xi^{-\beta/\nu} X(\frac{L}{\xi}) = Y(L) \ , \tag{6.6}$$

where we have used that

$$\xi = \xi_0 (p - p_c)^{-\nu} \,, \tag{6.7}$$

when $p > p_c$. In order to get rid of the factor $\xi^{-\beta/\nu}$ in front of the function, we require that the function $X(u)$ has a similar power-law behavior in the limit of small u, so that the ξ in front of X and the ξ inside the X cancel each other. We achieve this by setting

$$X(u) \propto u^{-\beta/\nu} \,. \tag{6.8}$$

We have therefore found that the scaling function X has the form

$$X(u) = \begin{cases} \text{constant} & u \gg 1 \\ u^{-\beta/\nu} & u \ll 1 \end{cases} \,. \tag{6.9}$$

We notice that in the limit when $L \ll \xi$ we get that $P(p, L) \propto L^{-\beta/\nu}$. We can then recover the full behavior of the order parameter:

$$P(p, L) \propto \begin{cases} (p - p_c)^{\beta} & L \gg \xi \\ L^{-\beta/\nu} & L \ll \xi \end{cases} \,. \tag{6.10}$$

We could also rewrite this as a function of ξ, to get

$$P(p, L) \propto \begin{cases} \xi^{-\beta/\nu} & L \gg \xi \\ L^{-\beta/\nu} & L \ll \xi \end{cases} \,, \tag{6.11}$$

which the reader should be able to relate to the results we found previously.

We also get the finite size scaling behavior of the mass $M(p, L) = P(p, L)L^d$

$$M(p, L) \propto \begin{cases} (p - p_c)^{\beta} L^d & L \gg \xi \\ L^{d-\beta/\nu} & L \ll \xi \end{cases} \,. \tag{6.12}$$

Normally we write

$$M \propto L^D \,, \tag{6.13}$$

where D denotes the fractal dimension, and we see that the fractal dimension can be related to the exponents ν and β:

$$D = d - \beta/\nu \,, \tag{6.14}$$

which call a hyper-scaling relation , since it also includes the spatial dimension d.

We could have carried through the same argument by initially assuming that the finite-size scaling form of the mass is

$$M(p, L) = L^D Y(\frac{L}{\xi}) \,, \tag{6.15}$$

which we did in detail above.

6.4 Average cluster size

Let us address the behavior of $S(p, L)$, the average cluster size in a finite system of size L. When p approaches p_c, the cluster cannot grow larger than the system size. The average cluster size will therefore not diverge, but reach a maximum determined by L, as illustrated in fig. 6.1.

What behavior do we expect for S? We know that in the thermodynamic limit, $S \propto |p - p_c|^{-\gamma}$. However, for finite L and $\xi \to \infty$, we expect S only to depend on L, $S \propto L^X$, where the exponent X is unknown. We will therefore make the scaling ansatz that

$$S(p, L) = L^X \Sigma(\frac{L}{\xi}) \; . \tag{6.16}$$

Our assumption that S only depends on L when $L \ll \xi$, implies that $\Sigma(u)$ approaches a constant when $u \ll 1$.

However, we also know that as long as $L \gg \xi$, S diverges when p approaches p_c according to:

$$S \propto |p - p_c|^{-\gamma} \propto ((|p - p_c|^{-\nu})^{-1/\nu})^{-\gamma} \propto \xi^{\gamma/\nu} \; , \tag{6.17}$$

because we know that $\xi \propto |p - p_c|^{-\nu}$. In order to cancel the L dependence when $L \gg \xi$, and $\xi \to \infty$, the function $\Sigma(u)$ must be a power-law with exponent $-X$: $\Sigma(u) \propto u^{-X}$, when u
$gg1$. Using the finite size scaling ansatz when $L \gg \xi$ gives

$$S = L^X \Sigma(\frac{L}{\xi}) \propto L^X (\frac{L}{\xi})^{-X} \propto \xi^X \; . \tag{6.18}$$

Comparison with the exponent in eq. 6.17 shows that $X = \gamma/\nu$, and we can therefore write the finite size scaling behavior as

$$S(p, L) = L^{\gamma/\nu} \Sigma(\frac{L}{\xi}) \; , \tag{6.19}$$

where the scaling function Σ has the following behavior

$$\Sigma(u) = \begin{cases} \text{constant} & u \ll 1 \\ u^{-\frac{\gamma}{\nu}} & u \gg 1 \end{cases} \; . \tag{6.20}$$

This result shows that if we plot $L^{-\gamma/\nu} S(p, L)$ as a function of $L|p - p_c|^{\nu}$ we would expect to have all the data collapse onto a common functional form corresponding to $\Sigma(u)$. However, we could make the data-collapse even simpler, by integrating the exponent of $|p - p_c$ into the scaling function:

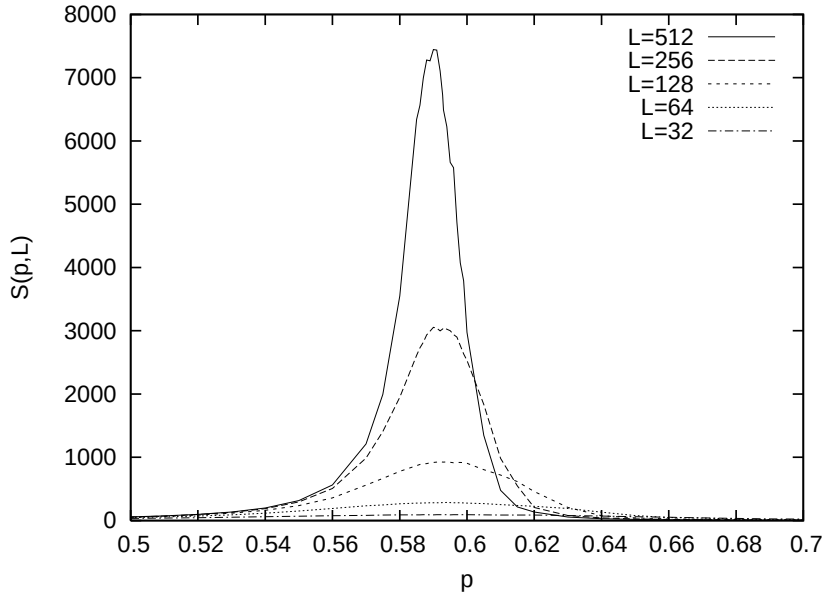

Fig. 6.1 (a) A plot of the average cluster size $S(p, L)$ as a function of p for various L. (b) A data-collapse plot of the rescaled average cluster size $L^{\gamma/\nu} S(p, L)$ as a function of $L^{1/\nu}(p - p_c)$ for various L.

$$S(p, L) = L^{\gamma/\nu} \Sigma(L(p - p_c)^{\nu}) = L^{\gamma/\nu} \bar{\Sigma}(L^{1/\nu}(p - p_c)) \; . \tag{6.21}$$

The data-collapse plot in fig. 6.1 shows that our theory indeed provides a reasonable data-collapse for two-dimensional percolation.

6.5 Percolation threshold

Let us see see how we can use finite size scaling to understand a finite system size will affect the effective percolation threshold .

Let us make a finite size scaling theory for the percolation probability $\Pi(p, L)$ for a finite system of size L.

We will start with the assumption that

$$\Pi(p, L) = \xi^0 f(\frac{L}{\xi}) \text{ (No power dependence on } \xi) \tag{6.22}$$
$$= f(\frac{L}{\xi}) \tag{6.23}$$
$$= f(\frac{L}{\xi_0(p-p_c)^{-\nu}}) \tag{6.24}$$
$$= f(\frac{L(p-p_c)^{\nu}}{\xi_0}) \tag{6.25}$$
$$= \hat{f}([L^{1/\nu}(p-p_c)]^{\nu}) \tag{6.26}$$
$$= \Phi[(p-p_c)L^{1/\nu}] \tag{6.27}$$

where we have used that $\xi = \xi_0(p-p_c)^{-\nu}$ when $p > p_c$, and where $\hat{f}(u) = f(u/\xi_0)$, and $\Phi(u) = \hat{f}(u^{\nu})$.

We can use this finite size scaling ansatz (theory) to find a way to estimate p_c. Let us fix a value $x = 0.8$, and find the p which gives $\Pi(p) = x$. This value of p is a function of L and is denoted:

$$p_{\Pi=x}(L) \tag{6.28}$$

Our scaling ansatz gives us that

$$x = \Phi[(p_{\Pi=x}(L) - p_c)L^{1/\nu}] \,, \tag{6.29}$$

which can be solved as

$$(p_{\Pi=x} - p_c)L^{1/\nu} = \Phi^{-1}(x) = C_x \,, \tag{6.30}$$

where it is important to realize that the right hand side is now a number which only depends on the x and not on L. We can therefore rewrite this as

$$p_{\Pi=x} - p_c = C_x L^{-1/\nu} \,, \tag{6.31}$$

If we know ν, we see that this is a method to estimate the value of p_c. However, we can also do this without knowing the value of ν, and it will be a good way to estimate both ν and p_c at the same time. We generate plots of $p_{\Pi=x}$ as a function of $L^{-1/\nu}$ for several values of x, and we modify the values of ν until we get a straight line, in that case we can read of the intersect with the x axis as the value for p_c.

Actually, we can do even better by noticing that for two x values x_1 and x_2, we get

$$dp = p_{\Pi=x_1}(L) - p_{\Pi=x_2}(L) = (C_{x_1} - C_{x_2})L^{-\nu} \,, \tag{6.32}$$

and we can therefore plot $\log(dp)$ as a function of $\log(L)$ to get ν, and then use this to estimate p_c.

As an exercise, the reader is encouraged to demonstrate that this scaling ansatz is valid for $d = 1$, and in this case find C_x explicitly.

Advanced

We can make our theory even more elegant by looking at the derivative of Π instead of Π.

What is the interpretation of $\Pi' = d\Pi/dp$? Since we know that

$$\int_0^1 \Pi dp = p \ , \tag{6.33}$$

we find that Π' is normalized

$$\int_0^1 \Pi' dp = 1 \ . \tag{6.34}$$

Our interpretation of Π' is that $\Pi'(p)dp$ is the probability that the system percolates for the first time in the interval from p to $p+dp$. In an infinite system we know that Π is a step-function which goes abruptly from 0 to 1 at p_c. The derivative is the delta-function, which is zero everywhere, except in a small region around p_c.

We use our scaling ansatz to find the derivative:

$$\Pi' = L^{1/\nu}\Phi'[(p - p_c)L^{1/\nu}] \ . \tag{6.35}$$

In particular we find that

$$\Pi'(p_c) = L^{1/\nu}\Phi'[0] \ . \tag{6.36}$$

We also find that position of the maximum of Π' is given by the second derivative

$$\Pi'' = L^{2/\nu}\Phi''[(p - p_c)L^{1/\nu}] \ . \tag{6.37}$$

and we will be looking for where $\Pi'' = 0$. Let us suppose that the value x_0 makes the second derivate zero, that is, suppose that $\Phi'(x)$ has a maximum at $x = x_0$.

At the maximum of Φ' we have that

$$(p_{\max} - p_c)L^{1/\nu} = x_0 \ , \tag{6.38}$$

and therefore

$$p_{\max} = p_c + \frac{x_0}{L^{1/\nu}} \ . \tag{6.39}$$

In each numerical experiment we are really measuring an effective p_c, but as $L \to \infty$ we see that $p_{eff} \to p_c$. The way it goes to p_c tells us something about ν.

Because Π' is a probability density, we can also calculate the average p of this distribution, that is the average p at which we first get a percolation cluster in a system of size L. Let us call this quantity $\langle p \rangle$.

$$\langle p \rangle = \int_0^1 p \Pi'(p) dp \tag{6.40}$$

$$= L^{-1/\nu} \int_0^1 p L^{1/\nu} \Phi'[(p - p_c) L^{1/\nu}] dp L^{1/\nu} \tag{6.41}$$

$$= \int_0^1 (p - p_c) L^{1/\nu} \Phi'[(p - p_c) L^{1/\nu}] dp + p_c \int_0^1 \Pi' dp \tag{6.42}$$

where the last integral is the normalization integral, and is 1.

We therefore get the result that

$$\langle p \rangle = p_c + L^{-1/\nu} \int x \Phi'[x] dx \ , \tag{6.43}$$

where the last integral is simply a constant, so that we can write the average critical percolation threshold in a finite system size as

$$\langle p \rangle = p_c + C L^{-1/\nu} \tag{6.44}$$

Which is not located exactly at p_c but the shift decreases with L.

Renormalization

7

We have now learned that when p approaches p_c, the correlation length grows to infinity, and the spanning cluster becomes a self-similar fractal structure. This implies that the spanning cluster has statistical self-similarity: if we cut out a piece of the spanning cluster, and rescale the lengths in the system, the rescaled system will have the same geometrical properties as the original system. In particular, the rescaled system will have the same mass scaling relation: it will also be a self-similar fractal with the same scaling properties.

What happens when $p \neq p_c$? In this case, there will be a finite correlation length, ξ, and a rescaling of the lengths in the system implies that the correlation length is also rescaled. A rescaling by a factor b corresponds to making an average over b^d sites in order to form the new lattice. Now, we will simply assume that this also implies that the correlation length is reduced by a factor b: $\xi' = \xi/b$. After a few iterations of this rescaling procedure, the correlation length will correspond to the lattice size, and the lattice is uniform.

We could have made this argument even simpler by initially stating that we divide the system into parts that are larger than the correlation length. Again, this would lead to a system that is homogeneous from the smallest lattice spacing an upwards. We can conclude that when $p < p_c$, the system behaves as a uniform, unconnected system. and when $p > p_c$, the system is uniform and connected.

The argument we have sketched above is the essence of the renormalization group argument . It is only exactly at $p = p_c$ that an iterative rescaling is a non-trivial fix point: the system iterates onto itself because it is a self-similar fractal. When p is away from p_c, rescaling iterations will make the system progressively more homogeneous, and effectively bring the rescaled p towards either 0 or 1.

In this section we will provide an introduction to the theoretical framework for renormalization. This is a powerful set of techniques,

introduced for equilibrium critical phenomena by Kadanoff [**?**] in 1965 and by Wilson [**?**] in 1971. Wilson later received the Nobel prize for this work on critical phenomena.

7.1 The renormalization mapping

Let us return to our theoretical model for our study of disorder: the model porous medium with occupation probability p. We will study a system of size L with a correlation length ξ, which is a function of p. We will call the length of a side of a single site a, and we ensure that

$$L \gg \xi \gg a \ . \tag{7.1}$$

We will not address whether it is possible to average over some of the sites in such a way that the macroscopic behavior does not change significantly. That is, we want to replace cells of b^d sites with new, "renormalized" single sites. This averaging procedure is illustrated in fig. 7.1.

In the original lattice the occupation probability is p. However, through our averaging procedure, we may change the occupation probability for the new, averaged sites. We will therefore call the new occupation probability p' - the probability to occupy a renormalized site. We write the mapping between the original and the new occupation probabilities as

$$p' = R(p) \ , \tag{7.2}$$

where the function $R(p)$, which provides the mapping, depends on the details of the rule used for renormalization. It is important to realize that the system size L and the correlation length ξ does not change in real terms, it is only in units of lattice constants they are changing.

There are many choices for the mapping between the original and the renormalized lattice. We have illustrated a particular rule for a mapping with a rescaling $b = 2$ in fig. 7.2. For a site percolation problem with $b = 2$ there are b^d possible configurations. The different configurations are classified into the 6 categories c, where the number of configuration in each category is listed below. In fig. 7.2 we have also illustrated a particular averaging rule. However, we could also have chosen different rules. Usually, we should ensure that the global information is preserved by the mapping. For example, we would want the mapping to conserve connectivity. That is, we would like to ensure that

$$\Pi(p, L) = \Pi(p', \frac{L}{b}) \ . \tag{7.3}$$

However, even though we may ensure this on the level of the mapping, this does not ensure that the mapping actually conserves connectivity when applied to a large cluster - it may, for example, connect clusters

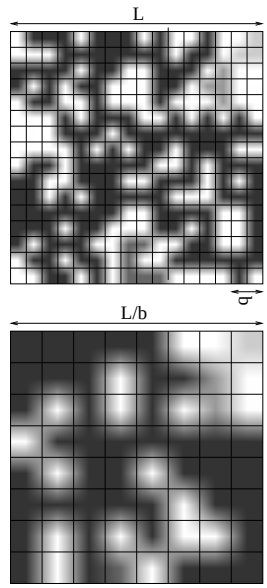

Fig. 7.1 Illustration of averaging using a rescaling $b = 2$, so that a cell of size $b \times b = 2 \times 2$ is reduced to a single site, producing a "renormalized" system of size $L/2$. The original pattern was generated with $p = 0.625$ for a $L = 16$ lattice.

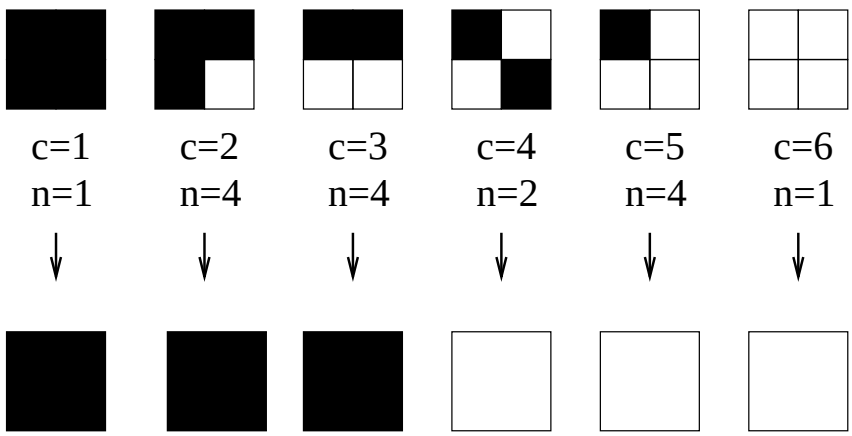

Fig. 7.2 Illustration of a renormalization rule for a site percolation problem with a rescaling $b = 2$. The top row indicates various clusters categorized into 6 classes c. The number of different configurations n in each class is also listed. The mapping ensures that connectivity is preserved. However, this renormalization mapping is not unique: we could have chosen many different averaging schemes.

that were unconnected in the original lattice, or disconnect clusters that were connected, as illustrated in fig. 7.3.

Currently, we will not consider the details of the renormalization mapping $p' = R(p)$, we will only assume that such a map exists and study its qualitative features. Then we will address the renormalization mapping through two worked examples. For any choice of mapping, the rescaling must result in a change in the correlation length ξ:

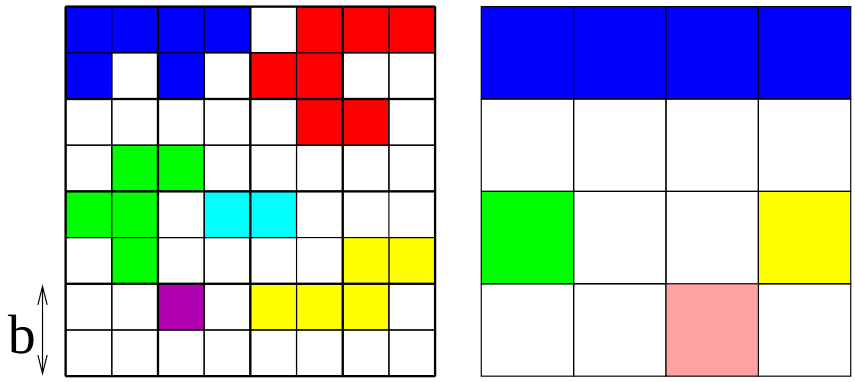

Fig. 7.3 Illustration of a single step of renormalization on an 8×8 lattice of sites. We see that the renormalization procedure introduces new connections: the blue cluster is now much larger than in the original. However, the procedure also removes previously existing connections: the original yellow cluster is split into two separate clusters.

$$\xi' = \xi(p') = \frac{1}{b}\xi(p) \; . \tag{7.4}$$

We will use this relation to address the fixpoints of the mapping. A fixpoint is a point p^* that does not change when the mapping is applied . That is

$$p^* = R(p^*) \; . \tag{7.5}$$

There are two trivial fixpoints : $p = 0$ and $p = 1$. At a fixpoint, the iteration relation for the correlation length becomes:

$$\xi(p^*) = \frac{\xi(p^*)}{b} \; . \tag{7.6}$$

This relation is satisfied at the two trivial fixpoints, because the correlation length is zero here, $\xi(0) = \xi(1) = 0$. The only possible solutions for $\xi(p^*) = \xi(p^*)/b$ is for $\xi = 0$ or for $\xi = \infty$.

Let us assume that there exists a non-trivial fixpoint p^*, and let us address the behavior for p close to p^*. We notice that for any finite ξ, iterations by the renormalization relation will reduce ξ. That is, both for $p < p^*$ and for and for $p > p^*$ iterations will make ξ smaller. This implies that iterations will take the system further away from the non-trivial fixpoint, where the correlation length is infinite. The non-trivial fixpoint is therefore an unstable fixpoints. Similarly, for p close to a trivial fixpoint, where $\xi = 0$, iterations will decrease p, and the renormalized system will move closer to the fixpoint in each iteration. The trivial fixpoint is therefore stable.

Iterations by the renormalization relation $p' = R(p)$ may be studied through on the graph $R(p)$, as illustrated in fig. 7.4. Consecutive iterations takes the system along the arrows illustrated in the figure, as the reader should convince himself of by following the mapping. Notice that the line $p' = p$ is drawn as a dotted reference line. In the figure, the two end

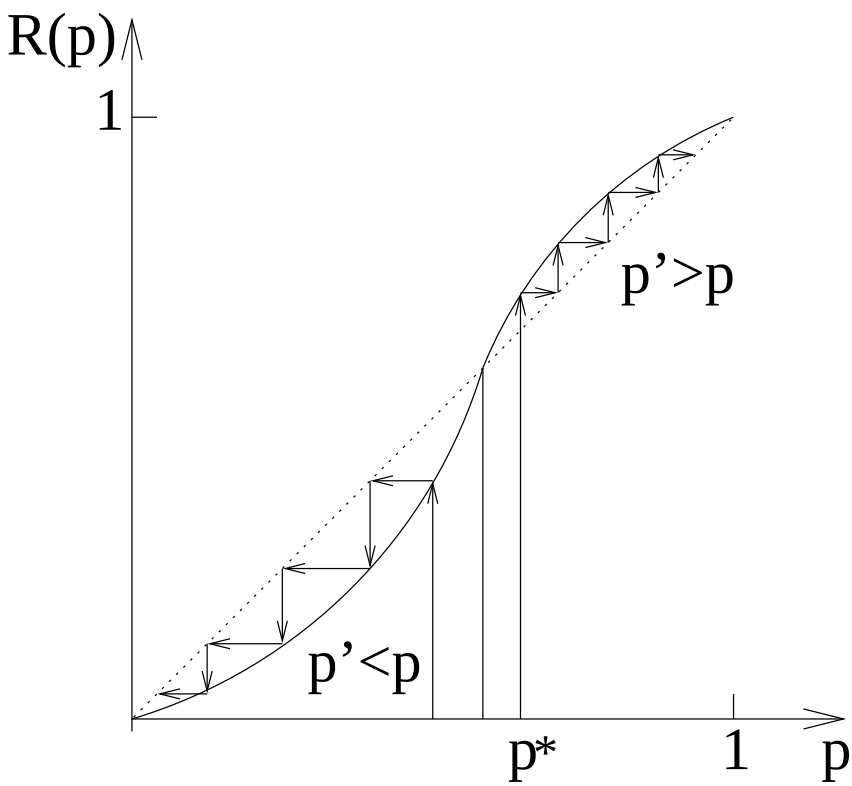

Fig. 7.4 Illustration the renormalization mapping $p' = R(p)$ as a function of p. The non-trivial fixpoint $p^* = R(p^*)$ is illustrated. Two iterations sequences are illustrated by the lines with arrows. Let us look at the path starting from $p > p^*$. Through the first application of the mapping, we read off the resulting value of p'. This value will then be the input value for the next application of the renormalization mapping. A fast way to find the corresponding position along the p axis is to reflect the p' value from the line $p' = p$ shown as a dotted line. This gives the new p value, and the mapping is applied again producing yet another p' which is even further from p^*. With the drawn shape of $R(p)$ there is only one non-trivial fixpoint, which is unstable.

points, $p = 0$ and $p = 1$ are the only stable fixpoints, and the points p^* is the only unstable fixpoints. The actual shape of the function $R(p)$ depends on the renormalization rule, and the shape may be more complex than what is illustrated in fig. 7.4.

Advanced

Let us make small diversion, and see what consequences the renormalization of the correlation length has if we know that there exists a percolation threshold p_c. In that case, we know that

$$\xi(p) = \xi_0(p - p_c)^{-\nu} , \tag{7.7}$$

and

$$\xi(p') = \xi_0(p' - p_c)^{-\nu} . \tag{7.8}$$

We can then use the renormalization condition for the correlation length from eq. 7.4 to obtain:

$$\frac{1}{b}\xi_0(p-p_c)^{-\nu} = \xi_0(p'-p_c)^{-\nu} \ . \tag{7.9}$$

When $p \to p_c$, we see that both $\xi(p)$ and $\xi(p')$ approaches infinity, which implies that if $p = p_c$, then we must also have that $p' = p_c$. That is, we have found that p_c is a fixpoint of the mapping.

We are now ready for a more quantitative argument for the effect of iterations through the renormalization mapping $R(p)$. We can argue that we have found that the non-trivial fixpoint corresponds to the percolation threshold, since the correlation length is diverging for this value of p, and we will indeed assume that we can identify p_c as the fixpoints, as we argued more quantitatively above.

We will now assume that $R(p)$ is an analytic function. This is not a strong assumption, since for any simple $R(p)$ based on polynomials of p and $1-p$ this is trivially fulfilled. We will not Taylor expand the mapping $p' = R(p)$ around $p = p^*$. First, we notice that

$$p' - p^* = R(p) - R(p^*) \ . \tag{7.10}$$

The Taylor expansion of $R(p)$ for a p close to p^* is:

$$R(p) = R(p^*) + R'(p^*)(p-p^*) + o(p-p^*)^2 \ . \tag{7.11}$$

If we define $\Lambda = R'(p^*)$, we can write to first order in $p - p^*$:

$$p' - p^* \simeq \Lambda(p-p^*) \ , \tag{7.12}$$

We see that the value of Λ characterizes the fixpoint. For $\Lambda > 1$ the new point p' will be further away from p^* than the initial point p. Consequently, the fixpoint is unstable . By a similar argument, we see that for $\Lambda < 1$ the fixpoint is stable . For $\Lambda = 1$ we call the fixpoint a marginal fixpoint.

Let us now assume that the fixpoint is indeed the percolation threshold. In this case, when p is close to p_c, we know that the correlation length is

$$\xi(p) = \xi_0(p-p_c)^{-\nu} \ , \tag{7.13}$$

for the initial point, and

$$\xi(p') = \xi_0(p'-p_c)^{-\nu} \tag{7.14}$$

for the renormalized point. We will now use eq. 7.12 for $p^* = p_c$, giving

$$p' - p_c = \Lambda(p-p_c) \ . \tag{7.15}$$

Inserting this into eq. 7.14 gives

$$\xi(p') = \xi_0(p' - p_c)^{-\nu} = \xi_0(\Lambda(p - p_c))^{-\nu} = \xi_0\Lambda^{-\nu}(p - p_c)^{-\nu} \ . \quad (7.16)$$

We can rewrite this using $\xi(p)$

$$\xi(p') = \Lambda^{-\nu}\xi(p) \ . \quad (7.17)$$

However, we also know that

$$\xi(p') = \frac{1}{b}\xi(p) \ . \quad (7.18)$$

Consequently, we have found that

$$b = \Lambda^{\nu} \ . \quad (7.19)$$

This implies that the exponent ν is a property of the fixpoint of the mapping $R(p)$. We can find ν from

$$\nu = \frac{\ln b}{\ln \Lambda} \ , \quad (7.20)$$

where we remember that $\Lambda = R'(p_c)$.

Advanced

We will now show that we can achieve all of these results just from a simple assumption on the effect of renormalization on the correlation length. Trivially, a renormalization procedure will lead to a change in correlation length. Starting at p with a correlation length $\xi(p)$, a renormalization step will produce a new occupation probability p' and a new correlation length $\xi'(p')$. The fundamental assumption in the theory for the renormalization group is that the functional form of ξ and ξ' is the same. That is, that we can write

$$\xi'(p') = \xi(p') \ , \quad (7.21)$$

where $\xi(p)$ was the functional form of the correlation length in the original system. At least we should be able to make this assumption in some small neighborhood around p_c. That is, we assume that $\xi(p) = \xi'(p)$ for $|p - p_c| \ll 1$. In this case, we can write the correlation function as a function of the deviation from p_c: $\epsilon = p - p_c$. Similarly, we define $\epsilon' = p' - p_c$. The relation between the correlation lengths can then be written as

$$\xi(\epsilon') = \frac{\xi(\epsilon)}{b} \ , \quad (7.22)$$

where $\xi(u)$ is a particular function of u. The Taylor expansion of the renormalization mapping $R(p)$ in eq. 7.12 can also be rewritten in terms of ϵ giving

$$\epsilon' = \Lambda\epsilon \ . \tag{7.23}$$

We can therefore rewrite eq. 7.22 as

$$\xi(\epsilon') = \xi(\Lambda\epsilon) = \frac{\xi(\epsilon)}{b} \ , \tag{7.24}$$

or, equivalently

$$\xi(\epsilon) = b\xi(\Lambda\epsilon) \ . \tag{7.25}$$

This implies that $\xi(\epsilon)$ is a homogeneous function. Let us see how this function responds to iterations. We notice that after an iteration, the new value of ϵ is $\Lambda\epsilon$, and we can write

$$\xi(\Lambda\epsilon) = b\xi(\Lambda\Lambda\epsilon) = b\xi(\Lambda^2\epsilon) \ . \tag{7.26}$$

We can insert this value into eq 7.25 to get

$$\xi(\epsilon) = b\xi(\Lambda\epsilon) = b^2\xi(\Lambda^2\epsilon) \ . \tag{7.27}$$

We can continue this process up to any power n, giving

$$\xi(\epsilon) = b^n\xi(\Lambda^n\epsilon) \ , \tag{7.28}$$

for any $n \geq 1$, where we have implicitly assumed that $b > 1$.

Let us now prove that eq. 7.28 implies that $\xi(\epsilon)$ is to leading order a power-law, and let us also find the exponent. We choose a value of n to that

$$\Lambda^n\epsilon = c \ , \tag{7.29}$$

which implies that

$$n = \frac{\ln c/\epsilon}{\ln \Lambda} \ . \tag{7.30}$$

We can always ensure that this produces a value $n > 1$ by selecting c sufficiently small. If we insert this value of n into eq. 7.28 we get

$$\xi(\epsilon) = b^{(\frac{\ln c/\epsilon}{\ln \Lambda})}\xi(c) \tag{7.31}$$

$$= e^{\ln b(\frac{\ln c/\epsilon}{\ln \Lambda}}\xi(c) \tag{7.32}$$

$$= (\frac{c}{\epsilon})^{\frac{\ln b}{\ln \Lambda}}\xi(c) \tag{7.33}$$

$$\propto \qquad\qquad \epsilon^{-\nu} \ , \tag{7.34}$$

where ν is given as

$$\nu = \frac{\ln b}{\ln \Lambda} \ . \tag{7.35}$$

We have now proved that the solution to the equation $\xi(p') = \xi(p)/b$ is a power-law function $\xi \propto |p - p_c|^{-\nu}$ with the exponent ν given by eq. 7.35.

This argument shows that the most important assumption of the renormalization theory, is that the functional form $\xi(p)$ does not change by the renormalization procedure. It is important to realize that this is an assumption, and we will then have to check whether this produces reasonable results.

7.2 Examples

In the following we provide several examples of the application of the renormalization theory. Our renormalization procedure can be summarized in the following points.

- Coarse-grain the system into cells of size b^d.
- Find a rule to determine the new occupation probability, p', from the old occupation probability, p: $p' = R(p)$.
- Determine the non-trivial fixpoints, p^*, of the renormalization mapping: $p^* = R(p^*)$, and use these points as approximations for p_c: $p_c = p^*$.
- Determine the rescaling factor Λ from the renormalization relation at the fixpoint: $\Lambda = R'(p^*)$.
- Find ν from the relation $\nu = \ln b / \ln \Lambda$.

It is important to realize that the renormalization mapping $R(p)$ is not unique. However, in order to obtain useful results we should ensure that the mapping preserves connectivity on average.

7.2.1 One-dimensional percolation

Let us first address the one-dimensional percolation problem using the renormalization procedure. We have illustrated the one-dimensional percolation problem in fig. 7.5. We generate the renormalization mapping by ensuring that it conserves connectivity. The probability for two sites to be connected over a distance b is p^b when the occupation probability for a single site is p. A renormalization mapping that conserves connectivity is therefore:

$$p' = \Pi(p, b) = p^b \ . \tag{7.36}$$

The fixpoints for this mapping are

$$p^* = (p^*)^b \ , \tag{7.37}$$

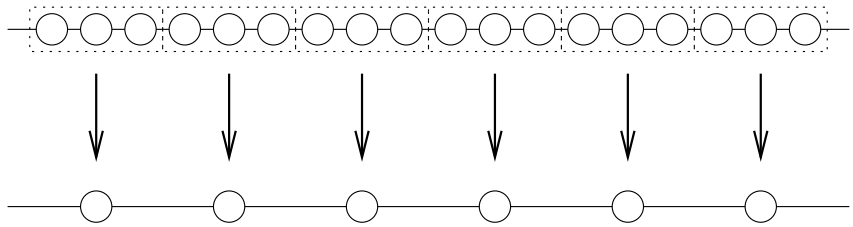

Fig. 7.5 Illustration of a renormalization rule for a one-dimensional site percolation system with $b = 3$.

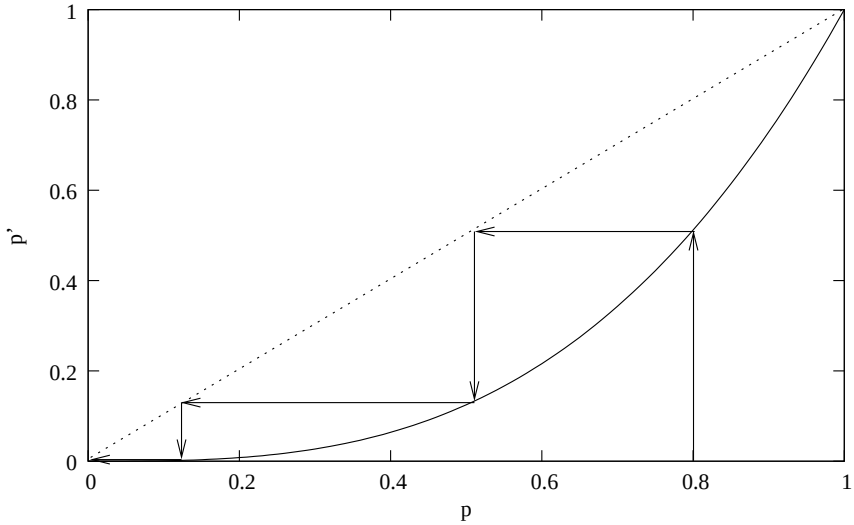

Fig. 7.6 Illustration of a renormalization rule for a one-dimensional site percolation system with $b = 3$.

with only two possible solutions, $p^* = 0$, and $p^* = 1$. An example of a renormalization iteration is shown in fig. 7.6. The curve illustrates that $p^* = 0$ is the only attractive or stable fixpoint, and that $p^* = 1$ is an unstable fixpoint.

We can also apply the theory directly to find the exponent ν. The renormalization relation is $p' = R(p) = p^b$. We can therefore find Λ from:

$$\Lambda = \left.\frac{\partial R}{\partial p}\right|_{p^*} = b(p^*)^{b-1} = b \,, \tag{7.38}$$

where we are now studying the unstable fixpoint $p^* = 1$. We can therefore determine ν from eq. 7.20:

$$\nu = \frac{\ln b}{\ln \Lambda} = 1 \,. \tag{7.39}$$

We notice that b was eliminated in this procedure, which is essential since we do not want the exponent to depend on details such as the size of renormalization cell. The result for the scaling of the correlation length is therefore

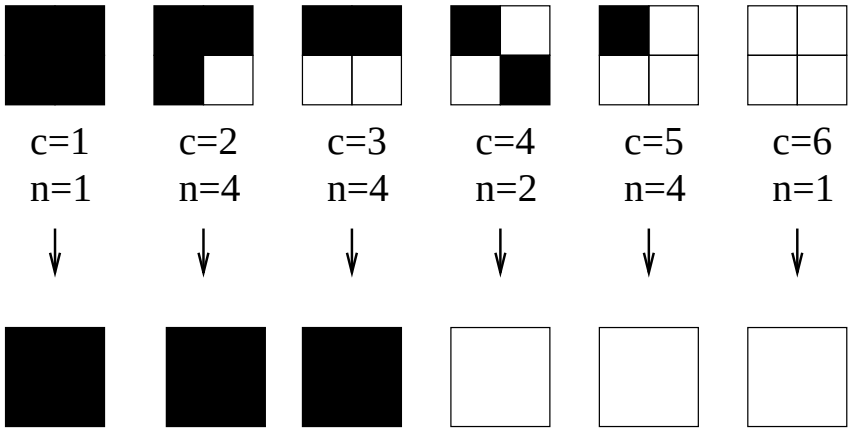

Fig. 7.7 Possible configurations for a 2×2 site percolation system. The top row indicates various clusters categorized into 6 classes c. The number of different configurations n in each class is also listed.

$$\xi \propto \frac{1}{1-p} \,, \tag{7.40}$$

when $1 - p \ll 1$.

7.2.2 Renormalization on 2d site lattice

Let us now use this method to address a renormalization scheme for two-dimensional site percolation. We will use a scheme with $b = 2$. The possible configurations for a 2×2 lattice are shown in fig. 7.7.

In order to preserve connectivity, we need to ensure that configurations $c = 1$ and $c = 2$ are occupied also in the renormalized lattice. However, for configuration $c = 3$, we may choose only to consider spanning in one direction, or spanning in both directions. If we include spanning in only one direction, there are only two of the configurations $c = 3$ that contribute the the spanning probability, and the renormalization relation becomes

$$p' = R(p) = p^4 + 4p^3(1-p) + 2p^2(1-p)^2 \,. \tag{7.41}$$

This is the probability for configurations $c = 1$, $c = 2$, or $c = 3$ to occur. The renormalization relation is illustrated in fig. 7.8.

We will now follow steps 3 and 4. First, in step 3, we determine the fixpoints of the renormalization relation. That is, we find the solutions to the equation

$$p^* = R(p^*) = (p^*)^4 + 4(p^*)^3(1-p^*) + 2(p^*)^2(1-p^*)^2 \,. \tag{7.42}$$

The trivial solution $p^* = 0$ is not of interest. Therefore we divide by p^* to produce

$$(p^*)^3 + 4(p^*)^2(1-p^*) + 2(p^*)(1-p^*)^2 = 1 \,. \tag{7.43}$$

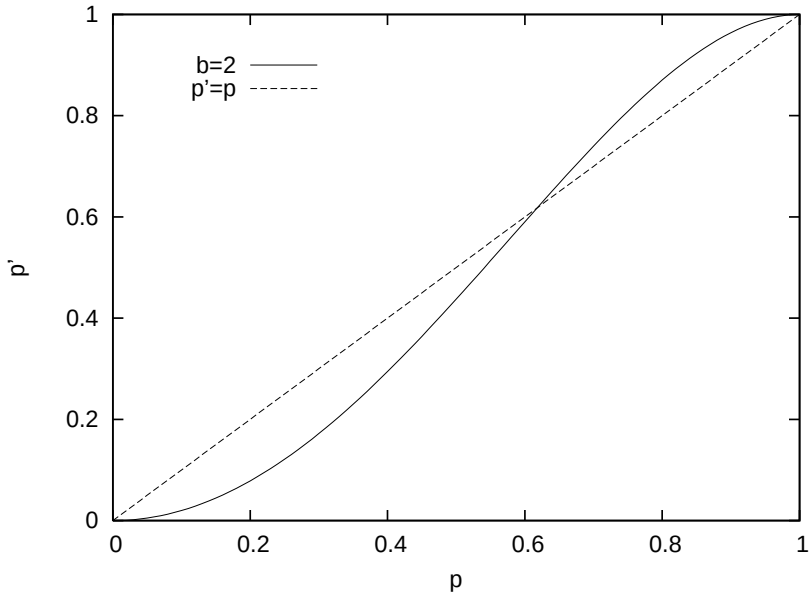

Fig. 7.8 Plot of the renormalization relation $p' = R(p) = p^4 + 4p^3(1-p) + 2p^2(1-p)^2$ for a two-dimensional site percolation problem.

The other trivial fixpoint is $p^* = 1$. We divide the equation by $1 - p^*$ to get

$$(p^*)^2 + p^* - 1 = 0 \, . \tag{7.44}$$

The solutions to this second order equation are

$$p^* = -\frac{1 \pm \sqrt{1+4}}{2} = \frac{\sqrt{5} \pm 1}{2} \simeq 0.62 \, . \tag{7.45}$$

We have therefore found an estimate of p_c by setting $p_c = p^*$. This does not produce the correct value for p_c in a two-dimensional site percolation system, but the result is still reasonably correct. We can similarly find the exponent ν by calculating $R'(p^*)$.

7.2.3 Renormalization on 2d triangular lattice

We will now the same method to address percolation on site percolation on a triangular lattice. A triangular lattice is a lattice where each point has six neighbors. In solid state physics, the lattice is known as the hexagonal lattice because of its hexagonal rotation symmetry. Site percolation on the triangular lattice is particularly well suited for renormalization treatment, because a coarse grained version of the lattice is also a triangular lattice, as illustrated in fig. 7.9, with a lattice spacing $b = \sqrt{3}$ times the original lattice size.

We will use the majority rule for the renormalization mapping. That is, we will map a set of three sites onto an occupied site if a majority

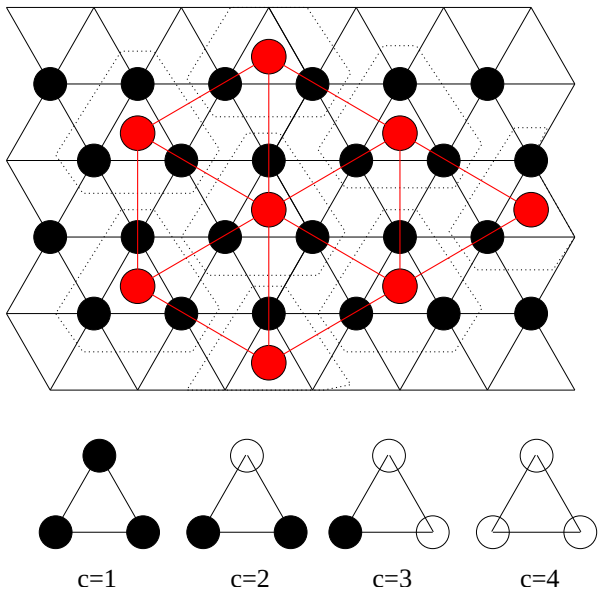

Fig. 7.9 Illustration of a renormalization scheme for site percolation on a triangular lattice. The rescaling factor is $b = \sqrt{3}$, and we use the majority rule for the mapping, that is, configurations $c = 1$ and $c = 2$ are occupied, and configurations $c = 3$ and $c = 4$ are mapped onto empty sites.

of the sites are occupied, meaning that two or more sites are occupied. Otherwise, the renormalized site is empty. This mapping is illustrated in fig. 7.9. This mapping does, as the reader may easily assure himself, on the average conserve connectivity. The renormalization mapping is

$$p' = R(p) = p^3 + 3p^2(1-p) = 3p^2 - 2p^3 \; . \tag{7.46}$$

The fixpoints of this mapping are the solutions of the equation

$$p^* = 3(p^*)^2 - 2(p^*)^3 \; . \tag{7.47}$$

We observe that the trivial fixpoints $p^{ast} = 0$, and $p^* = 1$ indeed satisfy eq. 7.47. The non-trivial fixpoint is $p^* = 1/2$. We are pleased to observe that this is actually the exact solution for p_c for site percolation on the triangular lattice.

We can use this relation to determine the scaling exponent ν. First, we calculate Λ:

$$\Lambda = R'(p^*) = 6p(1-p)|_{p=\frac{1}{2}} = \frac{3}{2} \; . \tag{7.48}$$

As a result we find the exponent ν from

$$\frac{1}{\nu} = \frac{\ln \Lambda}{\ln b} = \frac{\ln 3/2}{\ln \sqrt{3}} \simeq 1.355 \; , \tag{7.49}$$

which is very close to the exact result that $\nu = 4/3$ for two-dimensional percolation.

7.2.4 Renormalization on 2d bond lattice

As our last example of renormalization in two-dimensional percolation problems, we will study the bond percolation problem on a square lattice. The renormalization procedure is shown in fig. 7.10. In the renormalization procedure, we replace 8 bonds by 2 new bonds. We consider connectivity only in the horizontal direction, and may therefore simplify the lattice, by only considering the mapping of the H-cell, a mapping of five bonds onto one bond in the horizontal direction. The various configurations are shown in the figure. In table 7.2.4 we have shown the number of such configurations, and the probabilities for each configuration, which is needed in order to calculate the renormalization connection probability p'.

c	$P(c)$	$n(c)$	$\Pi\|c$
1	p^5	1	1
2	$p^4(1-p)$	1	1
3	$p^4(1-p)$	4	1
4	$p^3(1-p)^2$	2	1
5	$p^3(1-p)^2$	2	1
6	$p^3(1-p)^2$	2	0
7	$p^3(1-p)^2$	4	1
8	$p^2(1-p)^3$	2	1
9	$p^2(1-p)^3$	4	0
10	$p^2(1-p)^3$	2	0
11	$p^2(1-p)^3$	2	0
12	$p^1(1-p)^4$	5	0
13	$p^0(1-p)^5$	1	0

Table

A list of the possible configuration for renormalization of a bond lattice as illustrated in fig. 7.10. The probability for percolation given that the configuration is c is denoted $\Pi|c$. The spanning probability for the whole cell is then $\Pi(p) = p' = \sum_c n(c)P(c)\Pi|c$.

The resulting renormalization equation is given as

$$p' = R(p) = \Pi = \sum_{c=1}^{13} n(c)P(c)\Pi|c \,, \tag{7.50}$$

where we have used c to denote the various configurations, $P(c)$ is the probability for one instance of configuration c, $n(c)$ is the number of different configurations due to symmetry consideration, and $\Pi|c$ is the

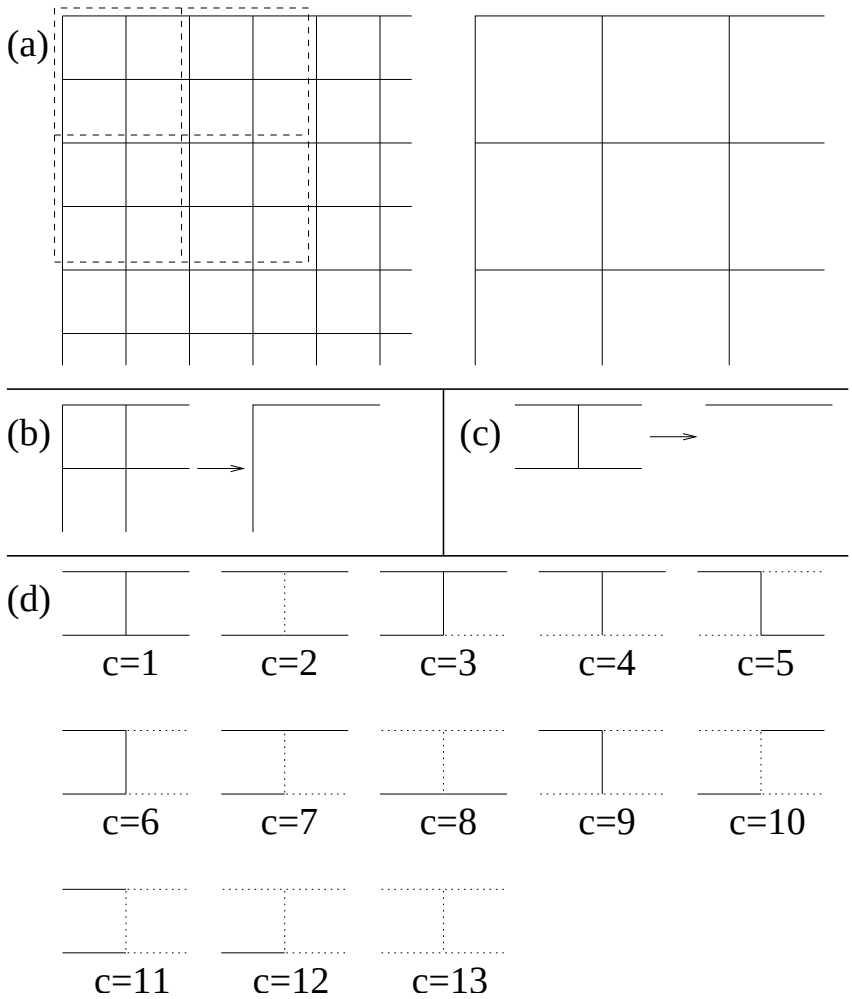

Fig. 7.10 (a) Illustration of a renormalization scheme for bond percolation on a square lattice in two dimensions. The rescaling factor is $b = 2$. (b) In general, the renormalization involves a mapping from 8 to two bonds. However, we will consider percolation only in the horizontal direction. This simplifies the mapping, to the figure shown in (c). For this mapping, the configurations are shown an enumerated in (d).

spanning probability for configuration c given that the configuration is c. The resulting relation is

$$p' = R(p) \tag{7.51}$$

$$= p^5 + p^4(1-p) + 4p^4(1-p) + 2p^3(1-p)^2 \tag{7.52}$$

$$+2p^3(1-p)^2 + 4p^3(1-p)^2 + 2p^2(1-p)^3 \tag{7.53}$$

$$= 2p^5 - 5p^4 + 2p^3 + 2p^2 \; . \tag{7.54}$$

The fixpoints for this mapping are $p^* = 0$, $p^* = 1$, and $p^* = 1/2$. The fixpoints $p^* = 1/2$ provides the exact solution for the percolation

threshold on the bond lattice in two dimensions. We find Λ by derivation

$$\Lambda = R'(p^*) = \frac{13}{8} \, . \tag{7.55}$$

The corresponding estimate for the exponent ν is

$$\nu = \frac{\ln b}{\ln \Lambda} \simeq 1.428 \, , \tag{7.56}$$

which should be compared with the exact result of $\nu = 4/3$ for two-dimensional percolation.

7.3 Universality

Even though we can choose renormalization rules that preserves connectivity statistically, the rule will not preserve connectivity exactly. The renormalization procedure is not exact. This can be illustrated by site renormalization of site percolation in two dimensions are shown in fig. 7.11. We may speculate that various errors of this form, some of them adding together non-connected bonds, and other removing connections, would cancel out on average. However, this is not the case. For the majority rule for two-dimensional site percolation, the connectivity is not preserved, even on the average. The result is that we end up with an error in our estimate of both p_c and ν.

How can we improve this situation? We need to introduce additional bonds between the sites during the renormalization procedure to preserve connectivity, even if the original problem was a pure site problem. This will produce a mixed site-bond percolation problem. The probability q to connect two nearest-neighbors in the original site lattice must be found by counting all possible combinations of spanning between nearest neighbor sites in the original lattice. We may also have to introduce next-nearest neighbor bonds and so on.

Let us describe the renormalized problem by the two renormalized probabilities p' for sites, and x' for bonds. The renormalization procedure will be described by a set of two renormalization relations:

$$p' = R_1(p, x) \tag{7.57}$$

$$x' = R_2(p, x) \tag{7.58}$$

Now, the flow in the renormalization procedure will not simply be along the p axis, but will occur in the two-dimensional p, x-space, as illustrated in fig. 7.12. We will no longer have a single critical points, p_c, but a set of points (p_c, x_c) corresponding to a curve in p, x-space, as shown in the figure. We also notice that when $x = 1$ we have a pure site percolation problem – all bonds will be present and connectivity depends on the

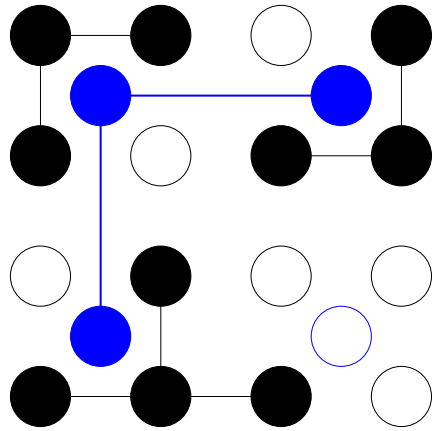

Fig. 7.11 Illustration of renormalization of connectivity for site percolation in two dimensions. The blue sites show the renormalized sites, and the lines shows which clusters are connected. In this case, we see that the renormalized lattice is spanning, even though there are no spanning clusters in the original lattice.

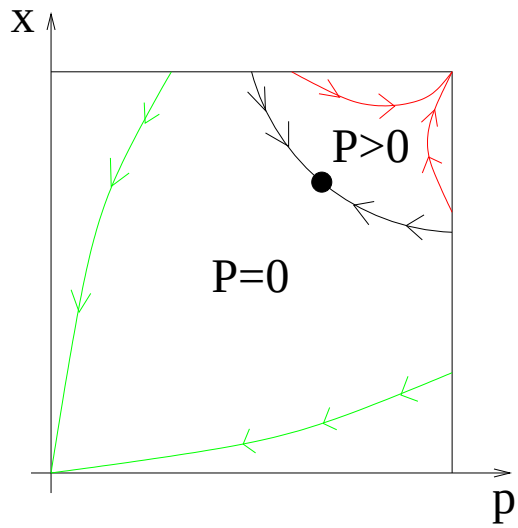

Fig. 7.12 Illustration of the flow due to renormalization in a combined site-bond percolation system. The black line shows the critical line, on which the correlation length is infinite, $\xi = \infty$. Below the critical line, renormalization will lead to the trivial fixpoint at $p, x = 0$ as illustrated by the green lines. Above the line, renormalization will lead to the fixpoint at $p, x = (1, 1)$.

presence of sites alone - and similarly for $p = 1$ we have a pure bond percolation problem.

There are still two trivial fixpoints, for $(p, x) = 0$, and for $(p, x) = (1, 1)$, and we expect these points to be attractors. We will therefore need a line that separates the two trivial fixpoints. If we start on this line, we will remain on this line. We will therefore expect there to be a fixpoint on this line, the non-trivial fixpoints (p^*, x^*). We remark that the fixpoint no longer corresponds to the critical threshold - there will be a whole family of critical values corresponding to the curved, black line in fig. 7.12.

We can find the non-trivial fixpoint from the equations

$$p^* = R_1(p^*, x^*) \tag{7.59}$$
$$x^* = R_2(p^*, x^*) \tag{7.60}$$

Let us linearize the system near the fixpoint. We will do a Taylor expansion for the two functions $R_1(p, x)$, and $R_2(p, x)$, around the point (p^*, x^*):

$$p' - p^* = \Lambda_{11}(p - p^*) + \Lambda_{12}(x - x^*) \tag{7.61}$$
$$x' - x^* = \Lambda_{21}(p - p^*) + \Lambda_{22}(x - x^*) \tag{7.62}$$

where we have defined

$$\Lambda_{11} = \left.\frac{\partial R_1}{\partial p}\right|_{(p^*,x^*)} \Lambda_{12} = \left.\frac{\partial R_1}{\partial x}\right|_{(p^*,x^*)} \tag{7.63}$$
$$\Lambda_{21} = \left.\frac{\partial R_2}{\partial p}\right|_{(p^*,x^*)} \Lambda_{22} = \left.\frac{\partial R_2}{\partial x}\right|_{(p^*,x^*)} \tag{7.64}$$

We can therefore rewrite the recursion relation in matrix form, as

$$\begin{bmatrix} p' - p^* \\ x' - x^* \end{bmatrix} = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix} \begin{bmatrix} p - p^* \\ x - x^* \end{bmatrix} . \tag{7.65}$$

We want to find the behavior after many iterations. This can be done by finding the eigenvector and the eigenvalues of the matrix. That is, we find the vectors $\mathbf{x}_i = (p_i, x_i)$ such that

$$\Lambda \mathbf{x}_i = \lambda_i \mathbf{x}_i . \tag{7.66}$$

We know that we can find two such vectors, and that the vectors are linearly independent, so that any vector $\mathbf{x}$ can be written as a linear combination of the two eigenvectors:

$$\begin{bmatrix} p - p^* \\ x - x^* \end{bmatrix} = \mathbf{x} = a_1 \mathbf{x}_1 + a_2 \mathbf{x}_2 . \tag{7.67}$$

Applying the renormalization mapping will therefore produce

$$\Lambda \mathbf{x} = \lambda_1 a_1 \mathbf{x}_1 + \lambda_2 a_2 \mathbf{x}_2 , \tag{7.68}$$

and after N iterations we get

$$\Lambda^N \mathbf{x} = \lambda_1^N a_1 \mathbf{x}_1 + \lambda_2^N a_2 \mathbf{x}_2 . \tag{7.69}$$

We see that if both $\lambda_1 < 1$ and $\lambda_2 < 1$, then any deviation from the fixpoint will approach zero after many iterations, because the values $\lambda_1^N \to 0$, and $\lambda_2^N \to 0$. We call eigenvalues in the range $0 < \lambda < 1$

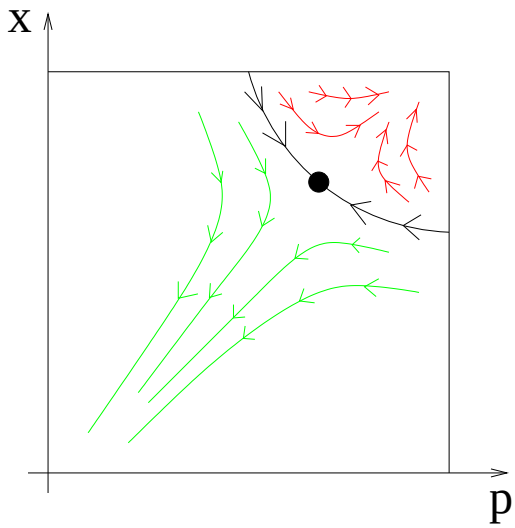

Fig. 7.13 Illustration of the flow around the unstable saddle point corresponding to the fixpoint p^*. The black line shows the critical line, on which the correlation length is infinite, $\xi = \infty$. Below the critical line, renormalization will lead to the trivial fixpoint at $p, x = 0$ as illustrated by the green lines. Above the line, renormalization will lead to the fixpoint at $p, x = (1, 1)$.

irrelevant, and the fixpoint is stable. Eigenvalues with $\lambda > 1$ are termed relevant, because the fixpoint will move away along the direction specified be the corresponding eigenvector. Eigenvalues $\lambda = 1$ are termed marginal - there is no movement along this direction.

Let us look at the case when $\lambda_1 > 1 > \lambda_2$, which corresponds to what we will call a simple critical point. (For a simple critical point, there is only one relevant eigenvalue, and all other eigenvalues are irrelevant.) This corresponds to a stable behavior in the direction $\mathbf{x}_2$, and an unstable behavior in the $\mathbf{x}_1$ direction. That is, the behavior is like a saddle point, as illustrated in fig. 7.13. This is consistent with the picture of a critical line. The flow along the line corresponds to the stable direction, and the flow normal to the line corresponds to the unstable direction, which is the natural generalization of the behavior we found in one dimension. Therefore any point which is originally close to the line, will first flow towards the fixpoint (p^*, x^*), before it flows out in the direction of the the relevant eigenvector.

Let us now study the behavior close to the critical line in detail for a system with $\lambda_1 > 1 > \lambda_2$. We notice that the correlation length ξ is infinite along the whole critical line, because it does not change by iterations along the critical line. That is, we have just a single fixpoint, but infinitely many critical points corresponding to a critical line. Let us start at a point $(p_0, 1)$ close to the critical line, and perform renormalization in order to find the functional shape of ξ and the exponent ν. After k iterations, the point has moved close to the fixpoint, just before it is expelled out from the fixpoint. We can therefore write

$$\begin{bmatrix} p^{(k)} - p^* \\ x^{(k)} - x^* \end{bmatrix} = a_1 \mathbf{x}_1 + a_2 \mathbf{x}_2 \ . \tag{7.70}$$

Since the iteration point is close to the fixpoint, we will assume that we can use the linear expansion around the fixpoint to address the behavior of the system. After a further l iterations we assume that we are still in the linear range, and the renormalized position in phase-space is

$$\begin{bmatrix} p^{(k+l)} - p^* \\ x^{(k+l)} - x^* \end{bmatrix} = \lambda_1^l a_1 \mathbf{x}_1 + \lambda_2^l a_2 \mathbf{x}_2 \ . \tag{7.71}$$

We stop the renormalization procedure at $l = l^*$ when $a_1^(l) \simeq 0.1$ (or some other small value that we can choose). That is

$$\lambda_1^{l^*} a_1 \simeq 0.1 \ . \tag{7.72}$$

The correlation length for this number of iterations is

$$\xi^{(k+l^*)} = \frac{\xi(p_0, 1)}{b^{(k+l^*)}} \ . \tag{7.73}$$

We have therefore found an expression for the correlation length in the point $(p_0, 1)$

$$\xi(p_0, 1) = \xi(a_1 = 0.1) b^{(k+l^*)} \ , \tag{7.74}$$

where the value $\xi(a_1 = 0.1)$ is a constant due to the way we have chosen l^*. The value for l^* is

$$l^* = \frac{\ln(\frac{0.1}{a_1})}{\ln \lambda_1} \ . \tag{7.75}$$

We have therefore found that the correlation length in the original point $(p_0, 1)$ is

$$\xi(p_0, 1) = b^k b^{\frac{\ln(0.1/a_1)}{\ln \lambda_1}} = b^k (\frac{0.1}{a_1})^{\frac{\ln b}{\ln \lambda_1}} \ . \tag{7.76}$$

We can express this further as:

$$\xi(p_0, 1) \propto (\frac{1}{a_1})^{\frac{\ln b}{\ln \lambda_1}} = (\frac{1}{a_1})^{\nu} = a_1^{-\nu} \ . \tag{7.77}$$

Now, what is a_1? This is the value of a_1 at the original point, $(p_0, 1)$, which we can relate the the critical threshold p_c for pure site percolation:

$$\begin{aligned} a_1 &= a_1(p_0, 1) = a_1(p_c + (p_0 - p_c)) && (7.78) \\ &\simeq && a_1(p_c) + a_1'(p_c)(p_0 - p_c) \quad (7.79) \\ &= A(p_0 - p_c) && (7.80) \end{aligned}$$

where we have done a Taylor expansion around p_c. We have used that $a_1(p_c, 1) = 1$, since this is a point on the critical line, and $A = a_1'(p_c)$. If we put this relation back into eq. 7.76, we get

$$\xi(p_0, 1) \propto a_1^{-\nu} \propto (p - p_c)^{-\nu} \, . \tag{7.81}$$

We have therefore shown by renormalization arguments, that ξ has a power-law behavior with exponent ν. However, we can make a similar argument starting at a point $(1, q_0)$ close the the critical point q_c. That is, we could start from a pure bond percolation problem, and we would end up with a similar relation for the correlation length

$$\xi \propto |q_0 - q_c|^{-\nu} \, , \tag{7.82}$$

where the exponent ν depends on λ_1.

We have therefore shown that the exponent ν is the same in these two cases. This is an example of universality. Both pure site and pure bond percolation leads to a power-law behavior for the correlation length ξ with the same power-law exponent ν. We can also use similar arguments to argue that the critical exponent ν is the same below and above the percolation threshold.

7.4 Case: Fragmentation

We will use the concepts and tools we have developed so far to address several problems of interest. First, let us address fragmentation: a large body that is successively broken into smaller part due to fracturing. There can be many processes that may induce and direct the fracturing of the grain. For example. the fracturing may depend on an external load placed on the grain, on a rapid change in temperature in the grain, on a high-amplitude sound wave propagating through the grain, or by stress-corrosion or chemical decomposition processes. Typical examples of fragment patterns are shown in fig. **??**.

Why did I choose D to denote this exponent? Let us look at the scaling properties of the structure generated by these iterations. Let us first assume that we describe the system purely geometrically, and that we are interested in the geometry of the regions that have fragmented. We will therefore assume that areas the are no longer fracturing are removed, and we are studying the mass that is left by this process. Let us start at a length-scale ℓ_n, where the mass of our system is m_n, and let us find what the mass will be when the length is doubled. For $f = 3/4$ we can then generate the new cluster by placing three of the original clusters into three of the four placed in the two-by-two square as illustrated in fig. 7.15. The rescaling of mass and length is therefore: $\ell_{n+1} = 2\ell_n$, and $m_{n+1} = 3m_n$. Similarly, for arbitrary f, the relations are $\ell_{n+1} = 2\ell_n$, and $m_{n+1} = 4fm_n$. As we found in section 5.5, this is consistent with a power-law scaling between the mass and length of the set

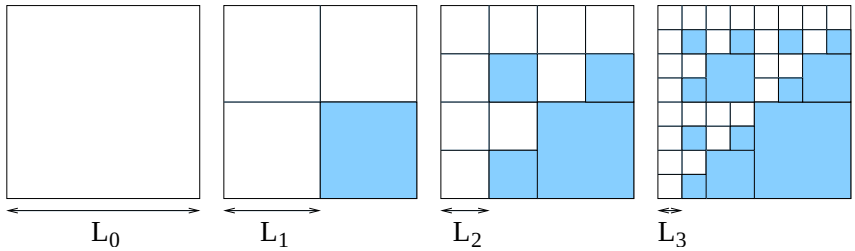

Fig. 7.14 Illustration of a deterministic fragmentation model. The shaded areas indicate the regions that will not fragment any further. That is, this drawing illustrate the case of $f = 3/4$.

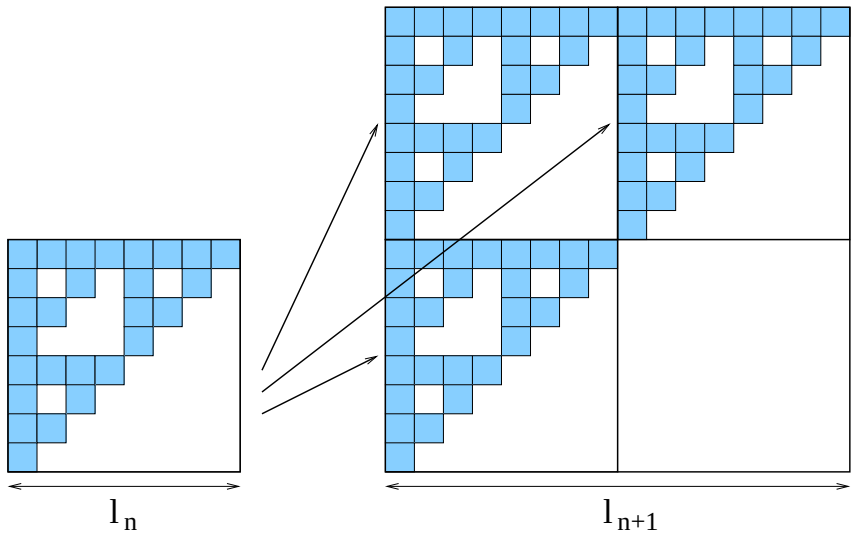

Fig. 7.15 Illustration of construction by length and mass rescaling. Three instances of the fully developed structure with mass m_n and length ℓ_n is used to generate the same structure at a length ℓ_{n+1} and with mass $m_{n+1} = 3m_n$. The mass corresponds to the mass of the regions that are not left unfragmented.

$$m(L) = m_0(\frac{L}{L_0})^D \ , \tag{7.83}$$

where $D = \ln 3/\ln 2$ is the fractal dimension of the structure. The value for the case of general f is similarly $D = \ln(4f)/\ln 2$.

Remember that we now calculated the mass dimension of the part of the system that is present, that is the part of the system that is still fragmenting into smaller pieces. The mass dimension of the part of the system that is no longer fragmenting should be $D' = d$, which is the fractal dimension of the "dust" left by the fragmentation processes.

The methodology that we have introduced to describe fragmentation here, is consistent with the argument of Sammis *et al.* [**?**] for the grains size distribution in fault gouges. Sammis *et al.* argues that during the fragmentation process, two grains of the same size cannot be nearest neighbors without fragmenting. There may be various physics arguments for this assumption, but we will not discuss them in detail here. If this

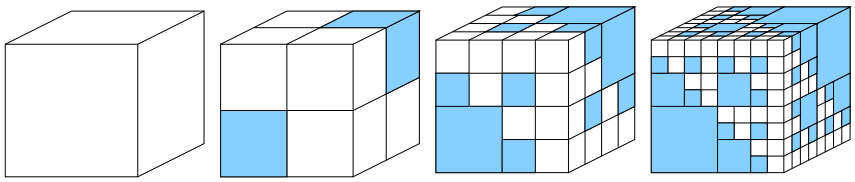

Fig. 7.16 Illustration of the fragmentation model of Sammis *et al.* [**?**]. In each iteration, each cubical grain is divided into 8 identical smaller cubes. The fundamental rule for the model is that if there are two neighboring grains of the same size, one of them will fracture. In the figure we have shaded the regions that are not fragmented in this processes for the first few steps of the iterations.

argument is applied in a simple cubic three-dimensional lattice, the remaining fragments will look like fig. 7.16. However, we realize that this is identical to the fragmentation model introduced here, because features such as the size distribution (and the mass dimension of the unfractured grains), does not depend on where the remaining grains are placed in space, only in the relative density of unfractured grains in each generation. The model of Sammis *et al.* therefore corresponds to the fragmentation model with $f = 6/8$, and with spatial dimensionality $d = 3$. We have therefore found the prediction $D = \ln 6/\ln 2 \simeq 2.58$ for the grain size distribution in fault gouges.

It is important to realize that the argument of Sammis *et al.* depends on the dimensionality of the system, and on the lattice used. For example, in two dimensions the argument leads to a fractal dimension $D = 1$ for a square system (corresponding to a line), whereas a triangular lattice produces a dimension between 1 and 2. We leave it as an exercise for the reader to find the dimension in this case.

Let us now use a simple renormalization type argument from Turcotte [**?**] to find a value for f, the partitioning between fractured and unfractured material.

Subset geometry

8

We have now developed a good understanding of the behavior of percolation systems. We have found that the distribution of cluster sizes is described by the cluster number density, $n(s,p)$, and that the scaling ansatz

$$n(s,p) = s^{-\tau} f((p-p_c)s^{\sigma}) \ , \tag{8.1}$$

works surprisingly well. The exponent τ is a property of the distribution at $p = p_c$, whereas the function f describes the behavior of the system away from $p = p_c$. When $p \neq p_c$, we know that the system is described by the characteristic cluster size s_ξ

$$s_\xi \propto |p-p_c|^{-1/\sigma} \ , \tag{8.2}$$

and by the correlation length ξ

$$\xi \propto |p-p_c|^{-\nu} \ . \tag{8.3}$$

That is, the percolation system is described by the two exponents σ and ν.

In addition, we have developed an understanding for the scaling properties of the spanning cluster, and the scaling properties of finite clusters. However, various physical processes picks out subsets of the clusters that are more important. As an example consider the conductivity of the spanning cluster - this property will only depend on the parts of the spanning cluster that contribute to flow from one side to another side, that is, it will not depend on dangling ends – part of the cluster that are blind alleys. In this chapter, we will discuss the scaling properties of such subsets of the spanning cluster when $p = p_c$. The scaling will be discussed using the renormalization group, and by examples from numerical simulations. This discussion will also lead us to a geometrical

interpretation of the exponent ν, and a better geometrical picture of the percolation system.

8.1 Subsets of the spanning cluster

One of the simplest examples of interesting subsets of the spanning cluster, is the set of singly connected sites (or bonds). A singly connected site is a site with the property that if it is removed, the spanning cluster will no longer be spanning. We can relate this to a physical property: If we study fluid flow in the spanning cluster, all the fluid has to go through the singly connected sites. These sites are also often referred to as red sites, because if we were studying a set of random resistors, the highest current would have to go through the singly connected bonds, and they would therefore heat up and become "red". Several examples of subsets of the spanning cluster, including the singly connected bonds, are shown in fig. 8.1.

We have learned that the spanning cluster may be described by the mass scaling relation $M \propto L^D$, where D is termed the fractal dimension of the spanning cluster. We propose that subsets of the spanning cluster also obey similar scaling relations. For example, we expect the mass of the singly connected sites (M_{SC}) to have the scaling form

$$M_{SC} \propto L^{D_{SC}} \ , \tag{8.4}$$

where we call the dimension D_{SC} the fractal dimension of the singly connected sites. Because the set of singly connected sites is a subset of the spanning cluster, we know that $M_{SC} \leq M$, and therefore that

$$D_{SC} \leq D \ . \tag{8.5}$$

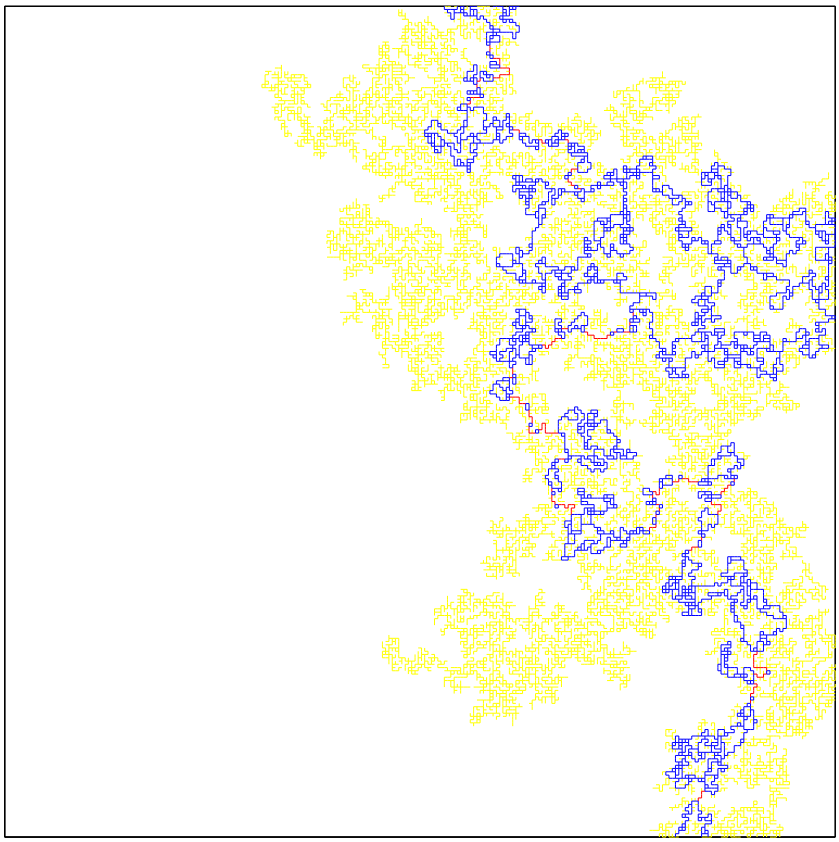

Fig. 8.1 Illustration of the spanning cluster, the singly connected bonds (red), the backbone (blue), and the dangling ends (yellow) for a 256×256 bond percolation system at $p = p_c$. (Figure from Martin Søreng).

8.2 Walks on the cluster

The study of percolation is the study of connectivity, and many of the physical properties that we are interested in depends on various forms of the connecting paths between the two opposite edges of the spanning cluster. We can address the structure of connected paths between the edges by studying self-avoiding walks (SAWs) on the cluster going from one side to the opposite side.

The shortest path between the two edges is given as the shortest SAW between the two edges. We call this the minimal path with a length L_{min}. The scaling exponent of the minimal path is usually written as d_{min}. That is, we have the scaling form

$$L_{min} \propto L^{D_{min}} \ . \tag{8.6}$$

We term the longest SAW between the two edges the longest path with length L_{max}, and term the corresponding exponent D_{max}. We notice that $L_{min} \leq L_{max}$. Consequently, a similar relation holds for the exponents

$D_{min} \leq D_{max}$. We could introduce the term, the average path , meaning the average length of all possible SAWs going between opposite sides of the system, $\langle L_{SAW} \rangle \propto L^{D_{SAW}}$, and the dimension would lie between the dimensions of the minimal and the maximal path.

However, the notion of SAWs can also be used to address the physical properties of the cluster, such as the singly connected bonds. The singly connected bonds, is the intersections between all SAWs connecting the two paths. That is, the singly connected bonds is the set of points that any path must go through in order to connect the two sides. From this definition, we notice the the dimension $D_{SC} < D_{min}$, and as we will see further on, $D_{SC} = 1/\nu$ which is smaller than 1 for two-dimensional systems.

Another useful set is the union of all SAWs that connect the two edges of the cluster. This set is called the backbone with dimension D_B. This set has a simple physical interpretation for a random porous material, since it corresponds to the sites that are accessible to fluid flow if a pressure is applied accross the material. The remaining sites are called dangling ends. The backbone are all the sites that have at least two different paths leading into them, one path from each side of the cluster. The remaining sites only have one (self-avoiding) path leading into them, and we call this set of sites the dangling ends . The spanning cluster consists of the backbone plus the dangling bonds, as illustrated in fig. 8.2. The dangling ends are therefore pieces of the cluster that can be cut away by the removal of a single bond.

We have arrived at the following hierarchy of exponents describing various subsets of paths through the cluster:

$$D_{SC} \leq D_{min} \leq D_{SAW} \leq D_{max} \leq D_B \leq D \leq d \,, \tag{8.7}$$

We have also arrived at a particular geometrical representation of clusters in the percolation system. The cluster can be subdivided into three parts: the dangling ends, a set of blobs where there are several parallel paths, and a set of points, the singly connected points, connecting the blobs to each other and the blobs to the dangling ends. Each of the blobs and the dangling ends will again have a similar substructure of dangling ends, blobs with parallel paths, and singly connected bonds as illustrated in fig. 8.3. This cartoon image of the clusters will show to provide useful intuition on the geometrical structure of percolation clusters.

The exponents can be calculated either by numerical simulations, where the masses of the various subsets are measured as a function of system size at $p = p_c$. Numerical results based on computer simulations are listed in table 8.2. Another approach is to use the renormalization group method to estimate the critical exponents.

d	D_{SC}	D_{min}	D_{max}	D_B	D	D_{DE}
2	3/4	1.1	1.5	1.6	1.89	

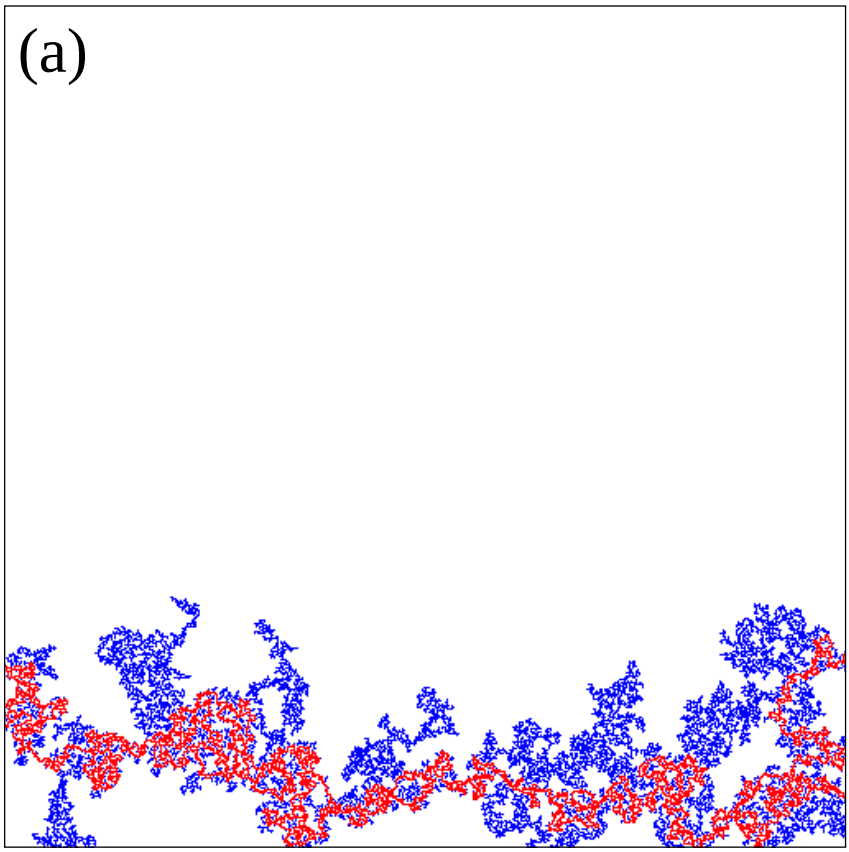

Fig. 8.2 Illustration of the spanning cluster consisting of the backbone (red) and the dangling ends (blue) for a 512×512 site percolation system for (a) $p = 0.58$, (b) $p = 0.59$, and (c) $p = 0.61$.

Table

Numerical exponents for the exponent describing various subsets of the spanning cluster defined using the set of Self-Avoiding Walks going from one side to the opposite side of the cluster.

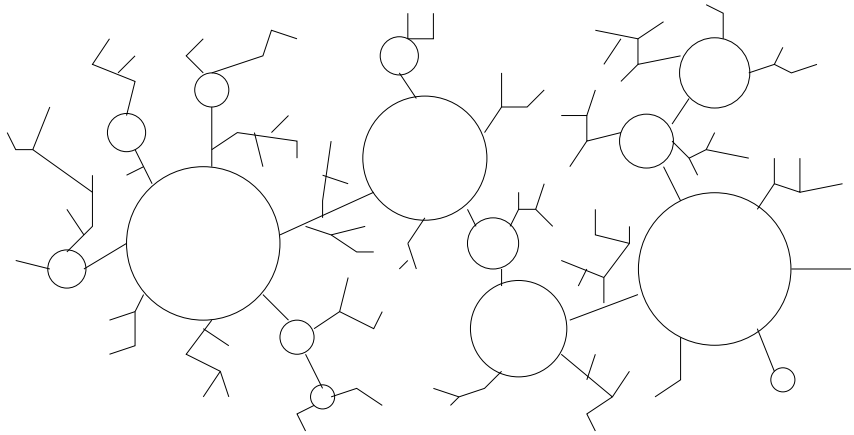

Fig. 8.3 Illustration of the hierarchical blob-model for the percolation cluster (See [**?**] for a more detailed discussion.)

8.3 Renormalization calculation

We will now use the renormalization group approach to address the scaling exponent for various subsets of the spanning cluster at $p = p_c$. For this, we will here use the renormalization procedure for bond percolation on a square lattice in two dimensions, where we have found that the renormalization procedure produces the exact result for the percolation threshold, $p_c = p^* = 1/2$. This is a fixpoint of the mapping.

Our strategy will be to assume that all the bonds have a mass $M = 1$ in the original lattice, and then find the mass M' in the renormalized lattice, when the length has been rescaled by b. For a property that displays a self-similar scaling, we will expect that

$$M' \propto b^{D_x} M \ , \tag{8.8}$$

where D_x denotes the exponent for the particular subset we are looking at. We can use this to determine the fractal exponent D_x from

$$D_x = \frac{\ln M'/M}{\ln b} \ . \tag{8.9}$$

We will do this be calculating the average value of the mass of the H-cell, by taking the mass of the subset we are interested in for each configuration, $M_x(c)$, and multiplying it by the probability of that configuration, summing over all configurations:

$$\langle M \rangle = \sum_c M_x(c) P(c) \ . \tag{8.10}$$

We have now calculated the average mass in the original 2 by 2 lattice, and this should correspond to the average renormalized mass, $\langle M' \rangle = p' M'$, which is the mass of the renormalized bond, M' multiplied with the

probability for that bond to be present p'. That is, we will find M' from:

$$p'M' = \sum_c M(c)P(c) \ , \tag{8.11}$$

We will study our system at the nontrivial fixpoint $p = p^* = 1/2 = p_c$. The spanning configurations c for bond renormalization in two dimensions, are shown together with their probabilities and the masses of various subsets in table **??**.

This use of the renormalization group method to estimate the exponents demonstrates the power of the renormalization arguments. Similar arguments will be used to address other properties of the percolation system.

Advanced

Let us also use this technique to develop an interpretation of the exponent ν and how it is related to the singly connected sites (See Coniglio [**?**] for a detailed argument). Because the exponent ν can be found from the renormalization equation at the fixpoint, which corresponds to the percolation threshold, it is reasonable to assume that the exponent ν can be derived from some property of the fractal structure at $p = p_c$.

Let us address bond percolation in two dimensions, described by the occupation probability p for bonds, and let us introduce an additional variable $1 - \pi$: the probability to remove an occupied bond from the system. We will consider the percolation problem to be described by these two values. When we are at $p = p_c$, we would expect that for any $1 - \pi > 0$, the spanning cluster will break into a set of unconnected clusters. The only fixpoint value when $p = p_c$ is therefore for $1 - \pi = 0$, that is, $\pi = 1$.

Can we derive a recursion relation for $1 - \pi$? For our renormalized cell of size b, the probability to break connectivity between the end nodes should be $1 - \pi'$. This corresponds to the probability that the renormalized bond is broken, because after renormalization there is only one bond in the box of size b. We write the recursion relation as a Taylor expansion around the fix-point $1 - \pi = 0$, or $\pi = 1$:

$$1 - \pi' = A(1 - \pi) + \mathcal{O}((1 - \pi)^2) \ , \tag{8.12}$$

where A is given as

$$A = \left. \frac{\partial \pi'}{\partial \pi} \right|_{\pi=1} \ . \tag{8.13}$$

We realize that the new p in the system after the introduction of π is given by $p = \pi p_c$, when the ordinary percolation system is at p_c. Similarly, the renormalized occupation probability is $p' = \pi' p_c$, and we have therefore found that

$$A = \left.\frac{\partial \pi'}{\partial \pi}\right|_{\pi=1} = \left.\frac{\partial p'}{\partial p}\right|_{p=p_c} = b^{1/\nu} \ . \tag{8.14}$$

8.4 Deterministic fractal models

infinite-dimensional and one-dimensional systems exactly. However, for finite dimensions such as for $d = 2$ or $d = 3$ we must rely on numerical simulations and renormalization group arguments to determine the exponents and the behavior of the system. However, in order to learn about physical properties in systems with scaling behavior, we may be able to construct simpler models that contain many of the important features of the percolation cluster. For example, we may be able to introduce deterministic, iterative fractal structures that reproduce many of the important properties of the percolation cluster at $p = p_c$, but that is deterministic and not a random system. The idea is that we can use this system to study other properties of the physics on fractal structures.

An example of an iterative fractal structure that has many of the important features of the percolation clusters at $p = p_c$ is the Mandelbrot-Given curve [**?**]. The curve is generated by the iterative procedure described in fig. 8.4. Through each generation, the length is rescaled by a factor $b = 3$, and the mass is rescaled by a factor 8. That is, for generation l, the mass is $m(l) = 8^l$, and the linear size of the cluster is $L(l) = 3^l$. If we assume a scaling on the form $m = L^D$, we find that

$$D = \frac{\ln 8}{\ln 3} \simeq 1.89 \ . \tag{8.15}$$

This is surprisingly similar to the fractal dimension of the percolation cluster. We can also look at other dimensions, such as for the singly connected bonds, the minimum path, the maximum path and the backbone.

Let us first address the singly connected bonds. In the zero'th generation, the system is simply a single bond, and the length of the singly connected bonds, L_{SC} is 1. In the first generation, there are two bonds that are singly connecting, and in the second generation there are four bonds that are singly connecting. The general relation is that

$$L_{SC} = 2^l \ , \tag{8.16}$$

where l is the generation of the structure. The dimension, D_{SC}, of the singly connected bonds is therefore $D_{SC} = \ln 2 / \ln 3 \simeq 0.63$, which should be compared with the exact value $D_{SC} = 3/4$ for two-dimensional percolation.

The minimum path will for all generations be the path going straight through the structure, and the length of the minimal path will therefore

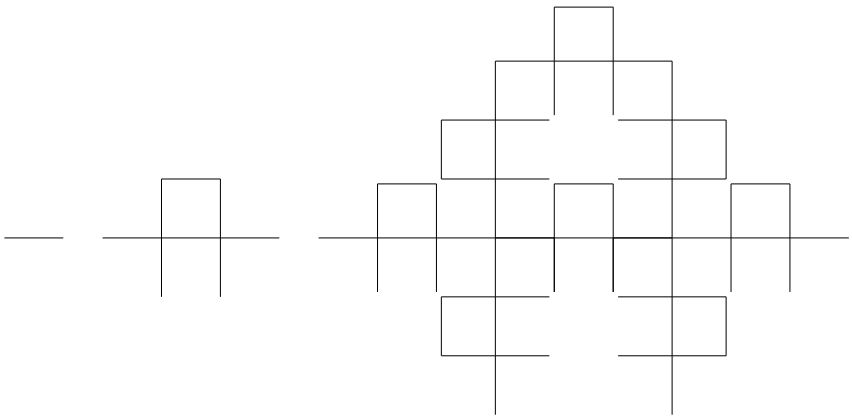

Fig. 8.4 Illustration of first three generations of the Mandelbrot-Given curve [**?**]. The length is rescaled by a factor $b = 3$ for each iteration, and the mass of the whole structure is increased by a factor of 8. The fractal dimension is therefore $D = \ln 8/\ln 3 \simeq 1.89$.

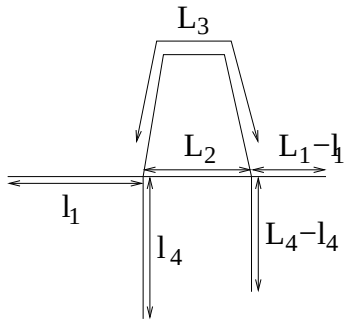

Fig. 8.5 The Mandelbrot-Given construction can be optimized by choosing particular values for the lengths l_1, L_2, L_3, l_4, and L_4. Here, L_3 gives the length around the whole curved path. The choice of l_1 and l_4 does not affect the scaling properties we have been addressing, and are therefore not relevant parameters. The ordinary Mandelbrot-given curve corresponds to $b = 3$, $L_1 = 2$, $L_2 = 1$, $L_3 = 3$, $L_4 = 2$, $l_1 = 1$, and $l_4 = 1$.

be equal to the length of the structure. The scaling dimension D_{min} is therefore $D_{min} = 1$.

The maximum path increases by a factor 5 for each iteration. The dimension of the maximum path is therefore $D_{max} = \ln 5/\ln 3 \simeq 1.465$.

We can similarly find that the mass of the backbone increases by a factor 6 for each iteration, and the dimension of the backbone is therefore $D_B = \ln 6/\ln 3 \simeq 1.631$.

The Mandelbrot-Given curve can be optimized by selecting the lengths L_i illustrated in fig. 8.5 in the way that provides the best estimate for the exponents of interest.

This deterministic iterative fractal can be used to perform quick calculations of various properties on a fractal system, and may also serve as a useful hierarchical lattice on which to perform simulations when we are studying processes occurring on a fractal structure.

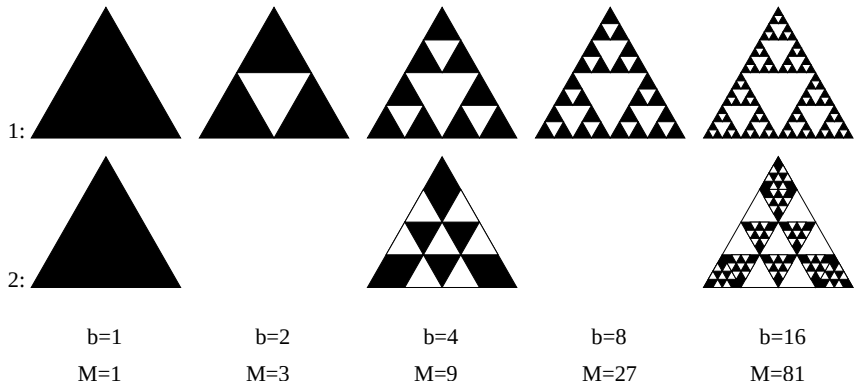

Fig. 8.6 Two versions of the Siepinski gasket. In version 1, the next generation is made from 3 of the structures from the last generation, and the spatial rescaling is by a factor $b = 3$. In version 2, the next generation is made from 9 of the structures from the last generation, and the spatial rescaling is by a factor $b = 6$. The resulting fractal dimension is $D_2 = \ln 9/\ln 4 = \ln 3^2/\ln 2^2 = \ln 3/\ln 2 = D_1$. The two structures therefore have the same fractal dimension. However, version 1 have large fluctuations that version 2.

8.5 Lacunarity

The fractal dimension describes the scaling properties of structures such as the percolation cluster at $p = p_c$. However, structures that have the same fractal dimension, may have a very different appearance. As an example, let us study several variations of the Sierpinski gasket introduced in section 5.5. As illustrated in fig. 8.6, we can construct several rules for the iterative generation of the fractal that all result in the same fractal dimension, but have different visual appearance. The fractal dimension $D = \ln 3/\ln 2$ for both of the examples in fig. 8.6, but by increasing the number of triangles that are used in each generation, the structures become more homogeneous. How can we quantify this difference?

In order to quantify this difference, Mandelbrot invented the concept of lacunarity . We measure lacunarity from the distribution of mass-sizes. We can characterize and measure the fractal dimension of a fractal structure using box-counting, as explained in section 5.5. The structure, such as the percolation cluster, is divided into boxes of size ℓ. In each box, i, there will be a mass $m_i(\ell)$. The fractal dimension was found by calculating the average mass per box of size ℓ:

$$\langle m_i(\ell)\rangle_i = A\ell^D \ . \tag{8.17}$$

However, there will be a full distribution of masses $m(\ell)$ in the boxes, characterized by a distribution $P(m, \ell)$, which gives the probability for mass m in a box of size ℓ. We can characterize this distribution by its moments:

$$\langle m^k(\ell)\rangle = A_k\ell^{kD} \ , \tag{8.18}$$

where this particular scaling form implies that the structure is unifractal: the scaling exponents for all the moments are linearly related.

For a unifractal structure, we expect the distribution of masses to have the scaling form

$$P(m,\ell) = \ell^x f(\frac{m}{\ell^D}) \ , \tag{8.19}$$

where the scaling exponent x is yet undetermined. In this case, the moments can be found by integration over the probability density

$$\langle m^k \rangle = \int P(m,\ell) m^k dm \tag{8.20}$$

$$= \int m^k \ell^x f(\frac{m}{\ell^D}) dm \tag{8.21}$$

$$= \ell^{(kD+x+D} \int (\frac{m}{\ell^D})^k f(\frac{m}{\ell^D}) d(\frac{m}{\ell^D}) \tag{8.22}$$

$$= \ell^{D(k+1)+x} \int x^k f(x) dx \tag{8.23}$$

We can determine the unknown scaling exponent x from the scaling of the zero'th moment, that is, from the normalization of the probability density: $\langle m^0 \rangle = 1$ implies that $D(0+1)+x = 0$, and therefore that $x = -D$. The scaling ansatz for the distribution of masses is therefore

$$P(m,\ell) = \ell^{-D} f(\frac{m}{\ell^D}) \ . \tag{8.24}$$

And we found that the moments can be written as

$$\langle m^k \rangle = \ell^{D(k+1)-D} \int x^k f(x) dx = A_k \ell^{kD} \ , \tag{8.25}$$

as we assumed above.

We therefore see that the distribution of masses is characterized by the distribution $P(m,\ell)$, which in turn is described by the fractal dimension, D, and the scaling function $f(u)$, which gives the shape of the distribution.

The distribution of masses can be broad, which would correspond to "large holes", or narrow, which would correspond to a more uniform distribution of mass. The width of the distribution can be characterized by the mean-square deviation of the mass from the average mass:

$$\Delta = \frac{\langle m^2 \rangle - \langle m \rangle^2}{\langle m \rangle^2} = \frac{A_2 - A_1^2}{A_1^2} \ . \tag{8.26}$$

This number describes another part of the mass distribution relation than the scaling relation, and can be used to characterize fractal set. For the percolation problem, this number is assumed to be universal, independent of lattice type, but dependent on the dimensionality.

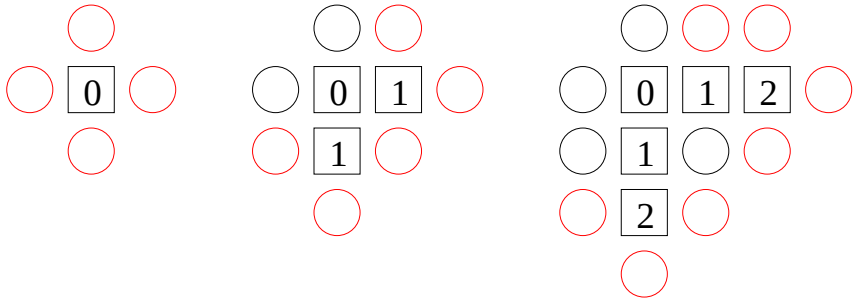

Fig. 8.7 Illustration of the Leath algorithm. Initially, the algorithm starts with the central site occupied. All the neighboring sites are added to a list of sites that should be checked in the next round, marked as red circles in the figure. In the first round, the sites marked with number 1 have been occupied, and some of the neighboring sites have been found not to be occupied, marked by black circles. The newly added neighbors marked with red circles will be checked in the next round. All the sites marked with a 2 have a minimum path of length 2 from them to the origin.

8.6 Numerical methods

The Leath algorithm [**?**] is particularly suited to study the minimal path in a percolation cluster. In this algorithm, the clusters are grown from an occupied site in the origin. Initially, we start with the site at the origin, and mark that in the next round we will check all neighboring sites. In the next round, we check all the neighboring sites, as illustrated in ref. 8.7, and add the newly generated neighboring sites that have not previously been tested to the list of sites that will be tested in the next round. The round, or generation, of check gives directly the minimal path from the center site to the sites grown, as also illustrated in fig. 8.7. This process continues until there are no more perimeter sites activated.

Using this method it is easy to collect statistics for $r(l)$, where r is the Euclidean length and l is the shortest path on the cluster. We can therefore use this method to numerically find the relation between L_{min} and the cluster size L.

Inter-dimensional cross-overs

9

The behavior of the percolation system depends strongly on dimensionality. We have learned that the two end members - a one-dimensional and an infinite dimensional system - can be treated exactly, but that the most relevant dimensions, $d = 2$ and $d = 3$ can only be addressed by numerical means and by the use of renormalization arguments.

However, many problems of practical interest may not be symmetric. For example, many systems of geological relevance are strongly layered, which may lead to a different treatment of one of the dimensions compared to the other dimensions. Another problem of relevance is the behavior of thin strips, or of thin tubes of material. In this chapter, we will demonstrate how we can use the tools we have developed for percolation theory to address problems with strong anisotropies.

In the first section, we will develop the scaling theory for a narrow strip of material, corresponding to a cross-over from two-dimensional to one-dimension behavior. Then, we will address the case of a strongly layered material, and we will also be discussion the behavior of percolation systems in higher dimensions, with a particular focus on the upper critical dimension for percolation.

9.1 Percolation on a strip

Let us address the behavior of a thin, long strip. We are interested in percolation in the long direction, which has a length L measured in lattice units. The length across the sample is $\ell \ll L$. The size of a site is chosen so that all correlations are included at a smaller scale, and at the scale with study the percolation problem, the occupation probabilities are independent and homogeneously distributed, that is, there is a common distribution for the whole sample.

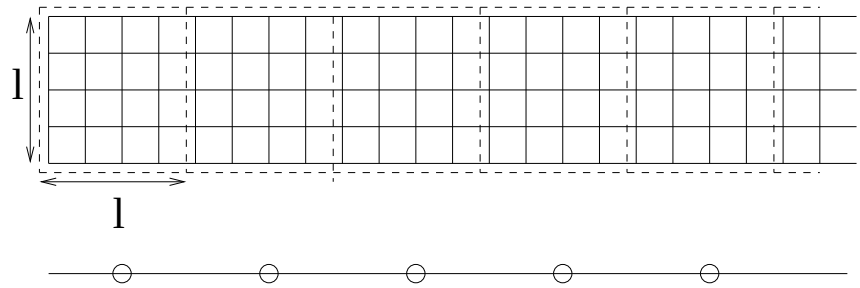

Fig. 9.1 Illustration percolation on a thin strip of width ℓ and length L. In the case when $\ell < \xi$, the system is renormalized until ℓ vanishes, and the problem is reduced to a one-dimensional problem.

Let us address the spanning probability in the x-direction corresponding to the length L. We call the spanning probability $Pi(\ell, L, p)$. The correlation length in the system depends on where we are in the parameter space. If we start close to the two-dimensional fixpoint, we can iterate until we are within the linear regime close to p^*. We can write these iterations into the spanning probability

$$\Pi(\ell, L, p) = \Pi(\ell, L, \xi) = \Pi(\frac{\ell}{b^n}, \frac{L}{b^n}, \frac{\xi}{b^n}) \ , \tag{9.1}$$

where we have used the technique that we can use ξ as the argument instead of p.

There are two possibilities that leads to two different behaviors: Either $\ell < \xi$ or $\ell > \xi$. Let us first address the case when $\ell < \xi$. In this case, we can choose an n^* so that

$$\frac{\ell}{b^n} = 1 \ . \tag{9.2}$$

We insert this into eq. 9.1, getting

$$\Pi = \Pi(1, \frac{L}{\ell}, \frac{\xi}{\ell}) \ . \tag{9.3}$$

We have therefore mapped it onto a one-dimensional problem with blocks of length ℓ as illustrated in fig. 9.1.

The occupation probability in the renormalized one-dimensional lattice will then be

$$p' = R^{(n^*)}(p) \ , \tag{9.4}$$

which corresponds to applying the renormalization iteration equation $R(p)$ a number of times corresponding to n^*. The new spanning probability is

$$\Pi(1, \frac{L}{\ell 0}, p') = (p')^{L/\ell} \ , \tag{9.5}$$

since we have reduced the problem to the one-dimensional percolation problem. For the one-dimensional problem, we can deduce the actual correlation length, because we know that the spanning probability is

related to the correlation length through

$$\Pi = ((p^{(n^*)})^{1/\ell})^L \simeq e^{-L/\xi'} \ , \tag{9.6}$$

where we have used the notation that $p^{(k)}$ corresponds to the p' value after k iterations. The correlation length ξ is given as

$$-\frac{1}{\xi} = \ln[(p^{(n^*)})^{1/\ell}] \ . \tag{9.7}$$

Introduction to disorder

10

We have now developed the tools to address the statistical properties of the geometry of a disordered system such as a model porous medium: the percolation system. In this part, we will apply this knowledge to address physical properties of disordered systems and to study physical processes in disordered materials.

We have learned that the geometry of a disordered system displays fractal scaling close to the percolation threshold. Material properties such as the density of singly connected sites, or the backbone of the percolation cluster, display self-similar scaling. The backbone is the part of the spanning cluster that participates in fluid flow. The mass, M_B, of the backbone scales with the system size, L, according to the scaling relation $M_B = L^{D_B}$, where D_B is smaller than the Euclidean dimension. The density of the backbone therefore decreases with system size. This implies that material properties which we ordinarily would treat as material constants, depend on the size of the sample. In this part we will develop an understanding of the origin of this behavior, and show how we can use the tools from percolation theory to address the behavior in such systems.

The behavior of a disordered system can in principle always be addressed by direct numerical simulation. For example, for incompressible, single-phase fluid flow through a porous material, the effective permeability of a sample can be found to very good accuracy from a detailed numerical model of fluid flow through the system. However, it is not practical to model fluid flow down to the smallest scaling in practical problems such as in oil migration. We would therefore need to extrapolate from the small scale to the large scaling. This process, often referred to as up-scaling, requires that we know the scaling properties of our system. We will address up-scaling in detail in this chapter.

We may argue that the point close to the percolation threshold is anomalous and that any realistic system, such as a geological system,

would typically be far away from the percolation threshold. In this case, the system will only display an anomalous, size-dependent behavior up to the correlation length, and over larger lengths the behavior will be that of a homogeneous material. We should, however, be aware that many physical properties are described by broad distributions of material properties, and this will lead to a behavior similar to the behavior close to the percolation threshold, as we will discuss in detail in this part. In addition, several physical processes ensure that the system is driven into or is exactly at the percolation threshold. One such example is the invasion-percolation process, which gives a reasonable description of oil-water emplacement processes such as secondary oil migration. For such systems, the behavior is best described by the scaling theory we have developed.

In this part, we will first provide an introduction to the scaling of material properties such as conductivity and elasticity. Then we will demonstrate how processes occurring in systems with frozen disorder, such as a porous material, often lead to the formation of fractal structures.

Flow in disordered media

11

In this chapter we will address the flow of an incompressible fluid through a disordered material, such as a percolation system. Traditionally, the conductive properties of a disordered material has been addressed by studying the behavior of random resistor networks. In this case, a voltage V is applied across the disordered material, and the total current, I, through the sample is measured, giving the conductance G of the sample as the constant of proportionality $I = GV$. However, here we will use fluid flow in a porous medium as our basic analogy, but we will study this process in the limit where it is identical to the study of electrical conductivity.

If we study electrical conductivity, we want to find the conductance of the particular sample. However, our basic assumption is that the conductance, G, which describes a particular sample with a given geometry, is related to the conductivity, σ, which is a material property. For an L^d sample in a d-dimensional system, the conductance of a homogeneous material with conductivity σ is $G = L^{d-1}\sigma/L$. That is, the conductance is inversely proportional to the length of the sample in the direction of flow, and proportional to the cross-sectional $d-1$-dimensional area.

When we address incompressible Darcy flow of a sample of length L and cross-sectional area A, Darcy's law provide a relation between the total flux, Φ, that is, the volume per unit time flowing through the sample, and the pressure drop Δp across the sample:

$$\Phi = \frac{kA}{\eta L}\Delta p \ , \tag{11.1}$$

where k is the permeability of the material, and η is the viscosity of the fluid. Again, we would like a description so that k is material property, and all the information about the geometry of the material goes into the flow conductance of the sample through the length L and the cross-

sectional area A. Generalized to a d-dimensional system, the relation is

$$\Phi = \frac{kL^{d-1}}{\eta L}\Delta p = L^{d-2}\frac{k}{\eta}\Delta p \; . \tag{11.2}$$

Consequently, it is clear that the electric conductivity problem is the same and the Darcy-flow permeability problem.

In this chapter we will discuss the behavior of the conductance and the conductivity for percolation systems, where sites are either filled or empty, and for other distributions of local material properties. We will see that the study of this problem introduces a wide range of new scaling behaviors for the percolation problem, and we will in particular address the scaling of the distribution of local fluxes or currents, and introduce the concept of multi-fractality describing their complicated scaling structure. We will also introduce the Effective Medium Theory for the flow problem. Finally, we will given an example of up-scaling of permeabilities and modeling of fluid flow in a porous medium using techniques from renormalization theory.

11.1 Conductivity

Let us first address the conductance of a L^d sample. The sample is a percolation system. The system may be either a site or a bond percolation system, however, many of the concepts we introduce are simpler to explain if we just consider a bond percolation system. Our sample will therefore be a networks of bonds, where all bonds have the same conductance, which we can set to 1 without loss of generality. However, bonds are removed with probability $1 - p$. We can describe this by only addressing the conductances, if we assume that the effective conductance of a removed bonds is 0.

The conductance of the L^d sample, is found by solving the Darcy flow problem as illustrated in fig. 11.1. A pressure drop Δp is applied across the whole sample, and the effective flow conductance G is found by finding the total flux $\Phi = G\Delta p$. The conductance will be a function of p and L: $G = G(p, L)$. We write the conductance between two neighboring sites i and j, corresponding to the conductance of the bond from site i to site j, as $g_{i,j}$. In order to find the effective conductance of the whole sample, the local fluxes between sites i and j, $\phi_{i,j}$, and the local pressures at sites i, p_i, must be found for all the sites i in the sample. The local flux from site i to site j, $\phi_{i,j}$, is related to the local pressure drop through Darcy's law (or Ohm's law for the case of electrical conductivity):

$$\phi_{i,j} = g_{i,j}(p_i - p_j) \; , \tag{11.3}$$

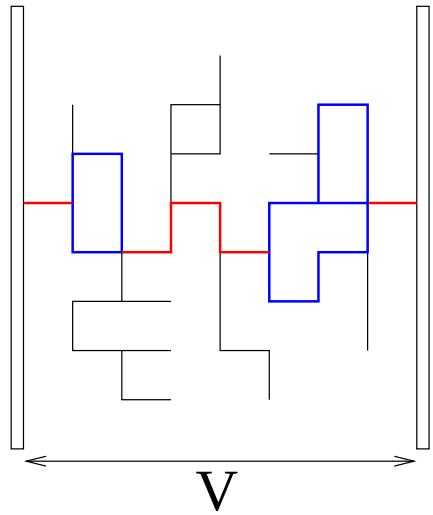

Fig. 11.1 Illustration of flow through a bond percolation system. The bonds shown in red are the singly connected bonds: all the flux has to go through these bonds. The bonds shown in blue are the rest of the backbone: The flow only takes place on the singly connected bonds and the backbone, the remaining bonds are the dangling ends, which do not participate in fluid flow.

where $g_{i,j} = k_{i,j}a/\eta l$, where l is the distance between sites i and j, and a is the cross-sectional area of the bond between sites i and j.

We find the distribution of pressures, by solving the set of local continuity equations: The net flux into site i, Φ_i, is given as the sum of fluxes into site i from all neighbors j:

$$\Phi_i = \sum_j g_{i,j}(p_i - p_j) \ , \tag{11.4}$$

where we know that the net flux into site i can be non-zero only at the boundary sites.

For percolation in an infinite sample, we must address the conductivity, σ, and not the conductance, G of the whole system. We know that for $p < p_c$ there will be no spanning cluster in an infinite sample. The effective conductivity is therefore zero. When p is close to 1, the density of the spanning cluster will be proportional to p, and we also expect the conductivity to be proportional to p in this range. This may lead us to assume that the density of the spanning cluster and the conductivity of the sample is proportional also when p is close to p_c. However, direct measurement by Last and Thouless [**?**] shows that P and σ are not proportional when p approaches p_c. The behavior is illustrated by the computer simulation results shown in fig. 11.2. We see that the density of the spanning cluster has a steep slope close to p_c, corresponding to a power-law behavior $P \propto (p - p_c)^\beta$, where $\beta < 1$. However, the conductivity has a similar scaling behavior, $\sigma \propto (p - p_c)^\mu$, but the

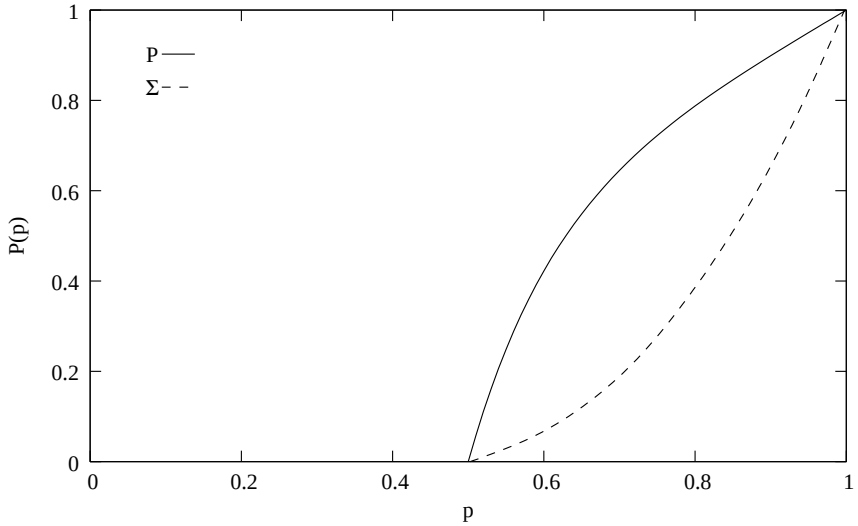

Fig. 11.2 Illustration of the shape of the curves $P(p, L)$ and $\sigma(p.L)$ for a bond percolation system with $L = 128$.

exponent $\mu > 1$. The conductivity therefore rises much slower than the density of the spanning cluster.

We have the tools to understand this behavior. The spanning cluster consists of the backbone and dangling ends. However, it is only the backbone that contributes to conductivity of the sample. We could remove all the dangling ends, and still get the same behavior the the conductivity. This suggests, that it is the scaling behavior of the backbone that is important. However, we have found that the mass-scaling exponent of the backbone, D_B, is smaller than D, the mass scaling exponent for the spanning cluster. This indicates that most of the mass of the spanning cluster is found in the dangling ends. This is the reason for the difference between the behavior of $P(p)$, and $\sigma(p)$ for p close to p_c.

11.2 Scaling arguments

Let us start by addressing the conductance $G(\xi, L)$ of a system with $p > p_c$ and $L \gg \xi$. In this case, we know that over length-scales larger than ξ, the system will be homogeneous. We can see this by subdividing the system into cells of size ξ, so that we have in total $(L/\xi)^d$ such cells. For a homogeneous system of size ℓ^d, we know that the conductance is $G \propto \ell^{d-2} g$, where g is the conductivity of a single box. We apply the same principle to this system: The conductance $G(\xi, L)$ is given as

$$G(\xi, L) = (\frac{L}{\xi})^{d-2} G(\xi, \xi) \ , \tag{11.5}$$

where we have written $G(\xi, \xi)$ for the conductivity within the box. This is the conductance of a system with correlation length equal to the system size ξ.

The conductivity $\sigma(\xi, L)$ is given as

$$\sigma(\xi, L) = L^{-(d-2)} G(\xi, L) = \frac{G(\xi, \xi)}{\xi^{d-2}} \, . \tag{11.6}$$

What is then $G(\xi, \xi)$? A system with correlation length equal to the system size is indistinguishable from a system at $p = p_c$. The conductance $G(\xi, \xi)$ is therefore the conductance of the spanning cluster at $p = p_c$.

If we address the other case, when $L \ll \xi$, the system behaves as if it is at $p = p_c$, and the conductance corresponds to the conductance in a system with infinite correlation length, $G(\infty, L)$.

11.2.1 Conductance of the spanning cluster

This leads us to address the conductance, $G(\infty, L)$, of the spanning cluster at $p = p_c$. We know that the spanning cluster consists of the backbone and the dangling ends, and that only the backbone will contribute to the conductivity. The backbone can be described by the blob model (see section 8.2 for a discussion of the blob model): The backbone consists of blobs of bonds in parallel, and links of singly connected bonds between them. We will assume that the conductance can be described by the scaling exponent $\tilde{\zeta}_R$:

$$G(\infty, L) \propto L^{-\tilde{\zeta}_R} \, . \tag{11.7}$$

Can we find bounds for the scaling of $G(\infty, L)$, and thereby determine bounds for the exponent $\tilde{\zeta}_R$? First, we know that the spanning cluster consists of blobs in series with the singly connected bonds. This implies that the resistivity $R = 1/G$ of the spanning cluster is given as the resistivity of the singly connected bonds R_{SC} plus the resistivity of the blobs, R_{blob} since resistivities are added for a series of resistances:

$$1/G = R = R_{SC} + R_{blob} \, , \tag{11.8}$$

This implies that $R > R_{SC}$. The resistance of the singly connected bonds can easily be found, since the definition of singly connected bonds is that they are coupled in series, one after another. The resistivity of the singly connected bonds is therefore the resistivity of a single bond multiplied with the number of singly connected bonds, M_{SC}. We have therefore found that

$$M_{SC} < R \, . \tag{11.9}$$

Because $M_{SC} \propto L^{D_{SC}}$, and $R \propto L^{\tilde{\zeta}_R}$, we find that

$$D_{SC} \leq \tilde{\zeta}_R \, . \tag{11.10}$$

We have found a lower bound for the exponent.

We can find an upper bound by examining the minimal path. The resistivity of the spanning cluster will be smaller than or equal to the resistivity of the minimal path, since the spanning cluster will have some regions, the blobs, where there are bonds in parallel. Adding parallel bonds will always lower the resistance. Hence, the resistivity is smaller than or equal to the resistivity of the minimal path. Since the minimal path is a series of resistances in series, the total resistance of the minimal path is the mass of the minimal path multiplied by the resistance of a single bonds. Consequently, the resistance of the spanning cluster is smaller than the mass of the minimal path, L_{min}, which we know scales with system size, $L_{min} \propto L^{D_{min}}$. We have found an upper bound for the exponent

$$L^{\tilde{\zeta}_R} \propto R \leq L_{min} \propto L^{D_{min}} \ , \tag{11.11}$$

therefore

$$\tilde{\zeta}_R \leq D_{min} \ . \tag{11.12}$$

We have therefore proved the scaling relation

$$D_{SC} \leq \tilde{\zeta}_R \leq D_{min} \ . \tag{11.13}$$

Because this scaling relation also shows that the scaling of R is bounded by two power-laws in L, we have also proved that the resistance R is a power-law, and that the exponents are within the given bounds. We notice that when dimensionality of the system is high, the probability of loops will be low, and blobs will be unlikely. In this case

$$D_{SC} = \tilde{\zeta}_R = D_{min} = D_{max} \ . \tag{11.14}$$

11.2.2 Conductivity for $p > p_c$

We have established that the conductance $G(\infty, L)$ of the spanning cluster is described by the exponent $\tilde{\zeta}_R$:

$$G(\infty, L) \propto L^{-\tilde{\zeta}_R} \text{ when } L \leq \xi \ . \tag{11.15}$$

We use this to find an expression for $G(\xi, \xi)$, which is the conductance of the spanning cluster at $p = p_c$ in a system of size ξ. Therefore

$$G(\xi, \xi) \propto \xi^{-\tilde{\zeta}_R} \ . \tag{11.16}$$

We use this in eq. 11.6 in order to establish the behavior of the conductivity for $p > p_c$, finding that

$$\sigma = \frac{G(\xi,\xi)}{\xi^{d-2}} \tag{11.17}$$

$$\propto \xi^{-(d-2+\tilde{\zeta}_R)} \tag{11.18}$$

$$\propto (p-p_c)^{\nu(d-2+\tilde{\zeta}_R)} \tag{11.19}$$

$$\propto (p-p_c)^{\mu} \tag{11.20}$$

We have introduced the exponent μ:

$$\mu = \nu(d-2+\tilde{\zeta}_R)\ . \tag{11.21}$$

We notice that for two-dimensional percolation, any value of $\tilde{\zeta}_R$ larger than $1/\nu$ will lead to a value for $\mu > 1$, which was what was observed in figure 11.2. The exponent μ is therefore significantly different from the exponent β that describes the mass of the spanning cluster.

11.2.3 Renormalization calculation

We will use the renormalization group for a square bond lattice in order to estimate the exponent $\tilde{\zeta}_R$. We calculate the average resistance $\langle R' \rangle$ of the H-cell, assuming that the resistance of a single bond is R. The renormalized resistance R' is then given as $p'R' = \langle R' \rangle$. Using the scaling relation for the resistivity, $R \propto L^{\tilde{\zeta}_R}$, we can determine the exponent from

$$\tilde{\zeta}_R = \frac{\ln \frac{\langle R' \rangle}{p'}}{\ln b}\ . \tag{11.22}$$

The renormalization scheme and the values used are shown in table 11.2.3. The resulting value for the renormalized resistance is

$$R' = \frac{1}{p'}\left(\frac{1}{2}\right)^5 \left(1 + 4\cdot\frac{5}{3} + 1 + 2\cdot 2 + 2\cdot 3 + 4\cdot 2 + 2\cdot 2\right) \simeq 1.917\ . \tag{11.23}$$

Consequently, the exponent $\tilde{\zeta}_R$ is given by

$$\tilde{\zeta}_R \simeq \frac{\ln 1.917}{\ln 2} \simeq 0.939\ , \tag{11.24}$$

This value is consistent with the scaling bounds set by the scaling relation in eq. 11.14.

c		$P(c)$	$R(c)$
1			
	$p^5(1-p)^0$	1	
2			
	$p^4(1-p)^1$	1	
3			
	$4p^4(1-p)^1$	5/3	
4			
	$2p^3(1-p)^2$	2	
5			
	$2p^3(1-p)^2$	3	
6			
	$4p^3(1-p)^2$	2	
7			
	$2p^2(1-p)^3$	2	

Table

Renormalization scheme for the scaling of the resistance R in a random resistor network. The value $R(c)$ gives the resistance of configuration c.

11.2.4 Finite size scaling

For the case when $L \gg \xi$, we have concluded that the conductance can be written as

$$G = L^{d-2}\sigma \tag{11.25}$$

where the conductivity, that is, the permeability, is

$$\sigma \propto (p - p_c)^{\mu} \propto \xi^{-\mu/\nu} , \tag{11.26}$$

with the exponent μ given as $\mu = \nu\left(d - 2 + \tilde{\zeta}_R\right)$. However, we want to determine the behavior of $\sigma(\xi, L)$ in both the limit $L \gg \xi$, which we have already addressed, and in the limit $L \ll \xi$. The scaling result, suggest that the conductivity has the scaling form

$$\sigma(\xi, L) = \xi^{-\mu/\nu} G_\sigma(\frac{L}{\xi}) . \tag{11.27}$$

This relation can be developed systematically using the renormalization group approach. Through each renormalization iteration, the conductivity σ is mapped onto itself as all length are rescaled by a factor b, by multiplying the whole expression by b to some exponent x, yet to be determined:

$$\sigma(\xi, L) = b^x \sigma(\frac{\xi}{b}, \frac{L}{b}) . \tag{11.28}$$

The result after l iterations is

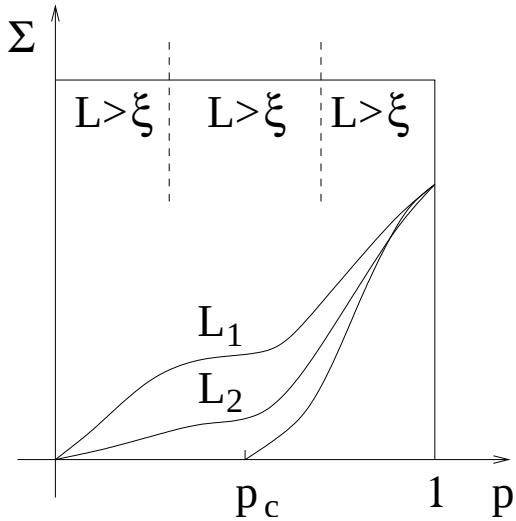

Fig. 11.3 Illustration of the conductivity σ as a function of p for $L_1 < L_2 < \infty$. We see that close to p_c the behavior is scaling with L according to $\sigma \propto L^{-\mu/\nu}$.

$$\sigma(\xi, L) = (b^l)^x \sigma(\frac{\xi}{b^l}, \frac{L}{b^l}) \ . \tag{11.29}$$

For the case $L < \xi$ we choose to continue iterations until $b^l = L$, which gives

$$\sigma(\xi, L) = L^x \sigma(1, \frac{\xi}{L}) \ . \tag{11.30}$$

Similarly, when $L > \xi$, we continue iterations until $b^l = \xi$, which gives

$$\sigma(\xi, L) = \xi^x \sigma(\frac{L}{\xi}, 1) \ . \tag{11.31}$$

Now, we have already established the scaling behavior when $L > \xi$, where we found that $\sigma \propto \xi^{-\mu/\nu}$. We recognize that the exponent $x = -\mu/\nu$, and that the function $\sigma(x, y)$ is a constant both when $x = 1$, and $y \gg 1$, and when $y = 1$, and $x \gg 1$. We have therefore found the limiting scaling behavior

$$\sigma(\xi, L) = \begin{cases} \xi^{-\mu/\nu} & L \gg \xi \\ L^{-\mu/\nu} & L \ll \xi \end{cases} \ . \tag{11.32}$$

The result for $L \ll \xi$ we could also have found by a direct scaling argument. Because, when $L \ll \xi$, the system behaves as if it is at $p = p_c$. We can therefore assume that $G \propto L^{-\tilde{\zeta}_R}$, and we find $\sigma = GL^{d-2}$, which gives

$$\sigma \propto L^{-(d-2+\tilde{\zeta}_R)} \propto L^{-\mu/\nu} \ . \tag{11.33}$$

The behavior is illustrated in figure 11.3.

We can sum these results up into the scaling form for $\sigma(\xi, L)$:

$$\sigma(\xi, L) = \xi^{-\mu/\nu} G_\sigma(\frac{L}{\xi}) \ , \tag{11.34}$$

where the scaling function has the form

$$G_\sigma(u) = \begin{cases} \text{const. } u \gg 1, \ L \to \infty \\ u^{-\mu/\nu} \ u \ll 1, \ \xi \to \infty \end{cases} \tag{11.35}$$

We can measure the conductivity by either measure the conductivity in an infinite sample, or we may use the finite size scaling result, to study the system at $p = p_c$, where we expect $\sigma \propto L^{-\mu/\nu}$. Our conclusion is that the conductivity is a function of p, but also of system size, which implies that the conductivity in a disordered system close to p_c is not a simple material property as we are used to - we need to address the scaling behavior of the system in detail.

11.3 Internal flux distribution

When we solve the incompressible flow problem, such as the set of Darcy's equations, on the percolation cluster, we find a set of fluxes ϕ_b for each bond b on the backbone. For all other bonds, the flux will be identically zero. How can we describe the distribution of fluxes on the backbone?

For the electrical problem, the conservation of energy is formulated in the expression:

$$RI^2 = \sum_b r_b I_b^2 \ , \tag{11.36}$$

where R is the total resistance of the system, I is the total current, r_b is the resistivity of bond b and I_b is the flux in bond b. We can therefore rewrite the total resistance R as

$$R = \sum_b r_b (\frac{I_b}{I})^2 = \sum_b r_b i_b^2 \ , \tag{11.37}$$

where we have introduced the fractional current $i_b = I_b/I$. We have written the total resistance as a sum of the square of the fractional currents in each of the bonds.

The fractional current i_b is assigned to each bond of the fractal backbone (when $p = p_c$). These measures have a probability distribution, describing the number of bonds $n(i)$ having the fractional current i. The total number of bonds is the mass of the backbone:

$$\sum_b 1 = M_B \propto L^{D_B} \ . \tag{11.38}$$

The distribution of fractional currents is therefore given by $P(i) = n(i)/M_B$.

We characterize the distribution $P(i)$ through the moments of the distribution:

$$\langle i^{2q} \rangle = \frac{1}{M_B} \sum_b i_b^{2q} = \frac{1}{M_B} \int i^{2q} n(i) di \ . \tag{11.39}$$

However, there is no general way to simplify this relation, since we do not know whether the function $n(i)$ has a simple scaling form.

However, we can address specific moments of the distribution. We know that the mass of the backbone has a fractal scaling with exponent D_B. This corresponds to the zero'th moment of the distribution. We will now also anticipate that for $p = p_c$, the other moments also has a scaling form:

$$\sum_b i_b^{2q} \propto L^{y(q)} \ , \tag{11.40}$$

and address the scaling exponent $y(q)$.

For $q = 0$, the sum is

$$\sum_b (i_b^2)^0 \propto L^{y(0)} \propto L^{D_B} \ , \tag{11.41}$$

that is, $y(0) = D_B$.

What happens in the limit of $q \to \infty$? In this case, the only terms that will be important in the sum are the terms where $i_b = 1$, because all other terms will be zero. The bonds with $i_b = 1$ are the singly connected bonds: all the current passes through these bonds. Therefore, we have

$$\sum_b (i_b^2)^\infty \propto L^{y(\infty)} \propto M_{SC} \propto L^{D_{SC}} \ , \tag{11.42}$$

and we find that $y(\infty) = D_{SC}$.

When $q = 1$, we find from equation 11.37 that the sum is given as the total resistance of the cluster

$$\sum_b (i_b^2)^1 = R \propto L^{\tilde{\zeta}_R} \ , \tag{11.43}$$

which implies that $y(1) = \tilde{\zeta}_R$.

We can in general argue that because each term in the sum $\sum_b (i_b)^{2q}$ is monotonically decreasing in q, the sum is also monotonically decreasing. We can therefore illustrate the curve $y(q)$ in fig. 11.4.

Advanced

In real resistor-networks, the case is even more complex, because the resistivity is due to impurities, and the impurities diffuse. Therefore, the fluctuations in the resistivity will also have a time-dependent part. This is the origin of thermal noise in the circuit. If we keep the

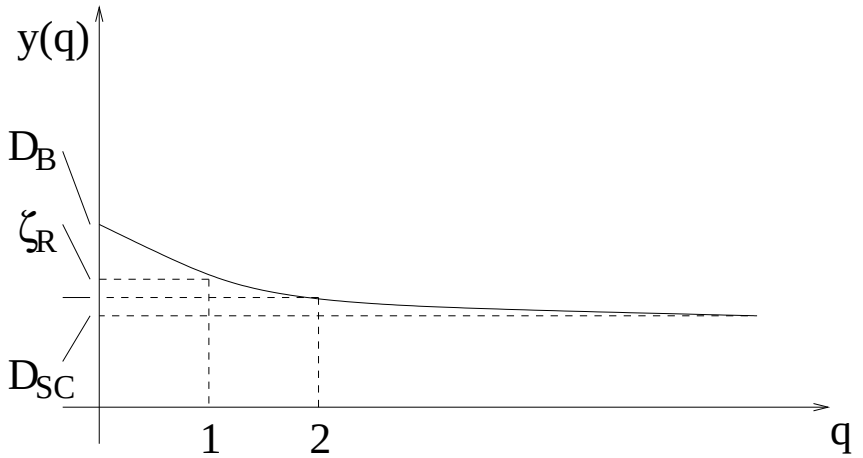

Fig. 11.4 Illustration of the exponents $y(q)$ characterizing the scaling of the moments of the distribution of fractional currents, as a function q, the order of the moment.

total current I constant, fluctuations in the resistivity will lead to fluctuations in the voltage.

What can we learn about the second moment, $q = 2$? We know that the total resistance, R, is

$$R = \sum_b r_b i_b^2 \, . \tag{11.44}$$

So far, we have only addressed the case when $r_b = 1$ for all the bonds on the backbone. However, in reality there will be some variations in the local resistances, so that we can write

$$r_b = 1 + \delta r_b \, , \tag{11.45}$$

where $\langle \delta r_b \rangle = 0$.

Advanced

Let us estimate the fluctuations in the voltage:

$$\delta V = V - \langle V \rangle = \sum_b \delta r_b (i_b)^2 \, . \tag{11.46}$$

However, the fractional currents i_b are now also different, since $i_b = I_b/I$ depends on the overall current I. Therefore we introduce

$$R_0 = \sum_b (i_b^{(0)})^2 \, , \tag{11.47}$$

where

$$i_b = i_b^{(0)} + \delta i_b \, . \tag{11.48}$$

There is a general theorem giving that

$$\sum_b 1 \cdot \delta(i_b^2) \simeq 0 \ , \tag{11.49}$$

to leading order. We can therefore conclude that

$$\delta V = V - \langle V \rangle = \sum_b \delta r_b i_b^2 + \sum_b 1 \cdot (\delta i_b^2) \simeq 0 \ . \tag{11.50}$$

However, what about the fluctuations in the deviations?

$$\langle (\delta V)^2 \rangle = \langle \sum_{b,b'} \langle \delta r_b \delta r_{b'} \rangle i_b^2 i_{b'}^2 \langle = \sum_{b,b'} \langle \delta r_b \delta r_{b'} \rangle i_b^2 (i_b')^2 \ . \tag{11.51}$$

If we assume that the fluctuations are independent:

$$\langle \delta r_b \delta r_{b'} \rangle = \delta_{bb'} \Delta \ , \tag{11.52}$$

where we have introduced

$$\Delta = \langle \delta r_b^2 \rangle \ . \tag{11.53}$$

We therefore find that

$$\langle \delta V^2 \rangle = \Delta \sum_b (i_b)^4 \propto \Delta L^{y(2)} \ . \tag{11.54}$$

Consequently, we find that the noise is related to the second moment. We know that the exponent $y(2)$ is bounded: $D_{SC} \geq y(2) \geq \tilde{\zeta}_R$, which places the value for $y(2) \simeq 0.9$ for a two-dimensional system.

The concept of $1/f$-noise in conductors is related to the fluctuations $\delta r_b(t)$ in $r_b(t)$.

11.4 Multi-fractals

The distribution of fractional currents in the random resistor network is an example of a multi-fractal distribution. The higher moments have the non-trivial scaling relation

$$M_q \propto L^{y(q)} \ , \tag{11.55}$$

Previously, we have studied unifractal distributions, such as the distribution of clusters sizes in the percolation problem when p is close to p_c.

$$\langle s^{k-1} \rangle \propto |p - p_c|^{-\gamma q(\beta+1)} \ , \tag{11.56}$$

Multifractals are typically encountered when measures, such as the fractional current through a bond, is imposed on a fractal structure.

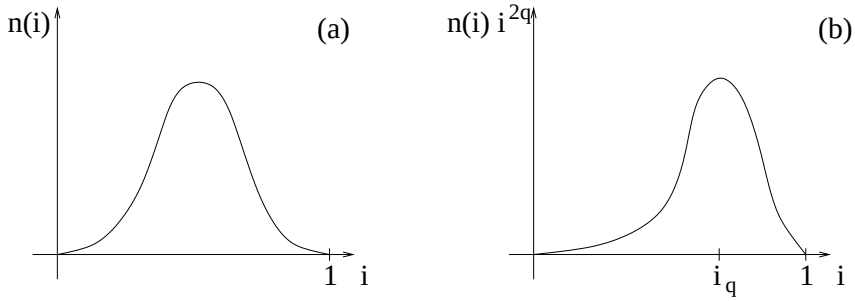

Fig. 11.5 Illustration of the distribution $n(i)$ of fractional currents i in a random resistor network. Part (a) shows the direct distribution, and part (b) shows $n(i)i^{2q}$. The distribution has a maximum at i_q.

We have now studied on example of a multi-fractal: the distribution of fractional currents in the random resistor network. In this case, we found that the moments of the distribution was

$$m_q = \langle i^{2q} \rangle = \frac{\sum_b i_b^{2q}}{M_B} \propto L^{y(q)-D_B} \ . \tag{11.57}$$

This is different from the unifractal case, where the exponent $y(q)$ is linear in k: $y(q) = Dk$.

Let us develop an understanding for what the various moments of order q are measuring in our system. Let us assume that the distribution $n(i)$ of fractional current in the system has the functional form illustrated in figure 11.5(a). How does then the function $n(i)i^{2q}$ look like? We have shown an illustration in figure 11.5(b). The maximum of this function is found at i_q (i_q is at the maximum of $n(i)i^{2q}$). We will assume that we can calculate the moment by only using the values in a small neighborhood of i_q, so that.

$$m_q = \int n(i) i^{2q} di \simeq n(i_q) i_q^{2q} \ . \tag{11.58}$$

This approximation becomes better as $q \to \infty$ since the distribution $n(i)i^{2q}$ is then approaching a delta function around i_q.

This implies that the various moments will focus on various values of i. That is, they will address the structure of points with a current close to i_q. Looking at the different moments of the distribution therefore corresponds to looking at different substructures of the cluster.

Let us now address the L-dependence in $n(i)$ and i_q^{2q}. Let us assume that i_q^2 and $n(i_q)$ is scaling with system size according to

$$i_q^2 \propto L^{-\alpha(q)} \ , \tag{11.59}$$

and

$$n(i_q) \propto L^{f(\alpha(q))} \ . \tag{11.60}$$

And we will assume that the q-th moment depends on the distribution at i_q.

$$m_q \simeq n(i_q) i_q^{2q} \propto L^{f(\alpha(q))-q\alpha(q)} \propto L^{y(q)-D_B} \ . \tag{11.61}$$

However, the value i_q is found from the maximum of $n(i)i^{2q}$. The condition for this maximum is

$$\frac{\partial}{\partial i}\left[n(i)i^{2q}\right]_{i_q} = 0 \ , \tag{11.62}$$

or

$$\frac{\partial}{\partial i}\left[\ln n(i) + 2q \ln i\right]_{i_q} = 0 \ , \tag{11.63}$$

which gives

$$\left.\frac{(\frac{\partial \ln n(i)}{\partial i})}{\ln i}\right|_{i_q} = -2q \ . \tag{11.64}$$

Now we can substitute the L-dependent expressions for $n(i_q)$ and i_q^2, getting

$$\ln n(i_q) = f \ln L \ , \tag{11.65}$$

and

$$\ln i_q^2 = -\alpha \ln L \ , \tag{11.66}$$

and therefore we find that

$$\frac{\partial f}{\partial \alpha} = q \ . \tag{11.67}$$

We have therefore two equations relating $y(q)$ to $f(\alpha)$ and $\alpha(q)$:

$$f(\alpha(q)) = [y(q) - D_B] + q\alpha(q) \tag{11.68}$$

$$\frac{\partial f}{\partial \alpha} = q \tag{11.69}$$

We can also show the reverse equations, using a Legendre transformation:

$$\frac{\partial}{\partial q}(f(\alpha(q)) = \frac{\partial f}{\partial \alpha}\frac{\partial \alpha}{\partial q} \tag{11.70}$$

$$= \frac{\partial y}{\partial q} + \alpha(q) + q\frac{\partial \alpha}{\partial q} \tag{11.71}$$

$$\frac{\partial y}{\partial q} = -\alpha \tag{11.72}$$

What is the interpretation of $f(\alpha)$? Because $n(i_q)L^{D_B}$ is the total number of bonds with current i_q, we can interpret $f(\alpha(q)) + D_B$ as the fractal dimension of the set of bonds with a current i_q.

Numerically, we estimate $f(\alpha)$ by selecting i_q, and then measure the fractal dimension of the subset with $i = i_q$, and plot the relation between i_q and the fractal dimension using as $f(\alpha)$.

11.5 Real conductivity

So far we have addressed conductivity of a percolation cluster. That is a system where the local conductances (or permeabilities) are either zero or a given constant conductance. That is, we have studied a system with local conductivities $\sigma_{i,j}$ so that

$$\sigma_b = \sigma_{i,j} = \begin{cases} 1 & p \\ 0 & 1-p \end{cases} . \tag{11.73}$$

However, in practice, we want to address systems with some distribution of conductances, such as a binary mixture of good and bad conductors, with conductivities:

$$\sigma_b = \sigma_{i,j} = \begin{cases} \sigma_> & p \\ \sigma_< & 1-p \end{cases} . \tag{11.74}$$

However, in order to address this problem, let us first look at the conjugate problem to the random resistor network, the random superconductor network. We will assume that the conductivities are

$$\sigma_b = \sigma_{i,j} = \begin{cases} \infty & p \\ 1 & 1-p \end{cases} . \tag{11.75}$$

In this case, we expect the conductance to diverge when p approaches p_c from below, and that the conductance is infinite when $p > p_c$. It can be shown that the behavior for the random superconductor network is similar to that of the random resistor network, but that the exponent describing the divergence of the conductivity when p approaches p_c is s:

$$\sigma \propto (p_c - p)^{-s} , \tag{11.76}$$

How can we address both these problems? For any system with a finite smallest conductivity, $\sigma_<$, we can always use the smaller conductivity as the unit for conductivity, and write the functional form for the conductivity of the whole system as

$$\sigma(\sigma_<, \sigma_>, p) = (\frac{\sigma(\frac{\sigma_<}{\sigma_<}, \frac{\sigma_>}{\sigma_<}, p)}{\sigma_<}) = \sigma(\frac{\sigma_>}{\sigma_<}, p) , \tag{11.77}$$

We will make a scaling ansatz for the general behavior of σ:

$$\sigma = \sigma_>(p - p_c)^\mu f_\pm(\frac{(\frac{\sigma_<}{\sigma_>})}{(p-p_c)^y}) , \tag{11.78}$$

where the exponent y is yet to be determined.

The random resistor network corresponds to $\sigma_< \to 0$, and $\sigma_> = c_0$. In this case, we retrieve the scaling behavior for p close to p_c, by assuming that $f_+(0)$ is a constant.

For the random superconductor network, the conductivities are $\sigma_> \to \infty$, and $\sigma_< = \text{const.}$. We will therefore need to construct $f_-(u)$ in such a way that the infinite conductivity is canceled from the prefactor. That is, we need $f_-(u) \propto u$. We insert this into equation 11.78, getting

$$\sigma \propto \sigma_>(p-p_c)^\mu \frac{\frac{\sigma_<}{\sigma_>}}{(p-p_c)^y} \propto \sigma_<|p-p_c|^{\mu+y} \ . \tag{11.79}$$

Because we know that the scaling exponent should be $\mu + y = -s$ in this limit, we have determined y: $y = -\mu - s$, where μ and s are determined from the random resistor and random superconductor networks respectively.

When $p \to p_c$ the conductivity σ should approach a constant number when both $\sigma_>$ and $\sigma_<$ are finite. However, $p \to p_c$ corresponds to the argument $x \to +\infty$ in the function $f_\pm(x)$. However, the only way to ensure that the total conductivity is finite, is to require that the two dependencies on $(p-p_c)$ cancel exactly. We achieve this by selecting

$$f_\pm(x) \propto x^{\mu/(\mu+s)} \ . \tag{11.80}$$

We can insert this relation into equation 11.78, getting

$$\sigma = \sigma_>|p-p_c|^\mu \left(\frac{\frac{\sigma_<}{\sigma_>}}{|p-p_c|^{\mu+s}}\right)^{\mu/(\mu+s)} \ , \tag{11.81}$$

which results in

$$\sigma = \sigma_>\left(\frac{\sigma_<}{\sigma_>}\right)^{\frac{\mu}{\mu+s}} \ . \tag{11.82}$$

This expression can again be simplified to

$$\sigma(p=p_c) = \sigma_>^{\frac{s}{\mu+s}} \sigma_<^{\frac{\mu}{\mu+s}} \ , \tag{11.83}$$

In two dimensions, $\mu = s \simeq 1.3$, and the relation becomes:

$$\sigma \propto (\sigma_<\sigma_>)^{\frac{1}{2}} \ , \tag{11.84}$$

11.6 Effective Medium Theory

11.7 Flow in hierarchical systems

Elastic properties of disordered media 12

There are various physical properties that we may be interested in for a disordered material. In the previous chapter, we studied flow problems in disordered materials using the percolation system as a model disordered material. In this chapter we will address mechanical properties of the disordered material, such as the coefficients of elasticity, tensile strength, and the dispersion relation for elastic waves propagating through the material.

We will address the behavior of the disordered material in the limit of fractal scaling. In this limit we expect material properties such as Young's modulus to display a non-trivial dependence on system size. That is, we will expect material properties such as Young's modulus to have an explicit system size dependence. We will use the terminology and techniques already developed to study percolation to address the mechanical behavior of disordered systems.

12.1 Rigidity percolation

First, we will address the elastic properties of the percolation system. Let us assume that we model an elastic material as a bond lattice, where each bond represents a local elastic element. The element will in general have resistance to stretching and bending. Systems with only stretching stiffness are termed central force lattices. Here, we will address systems with both stretching and bending stiffness.

We can formulate the effect of bending and stretching through the elastic energy of the system. The energy will have terms that depend on the elongation of bonds - these will be the terms that are related to stretching resistance. In addition, there will be terms related to the bending of bonds. Here we will introduce the bending terms through the angles between bonds. For any two bonds connected to the same site,

there will be an energy associated with changes in the angle of the bond. This can be expressed as

$$U = \sum_{ij} \frac{1}{2} k_{ij} (\mathbf{u}_i - \mathbf{u}_j)^2 + \sum_{ijk} \frac{1}{2} \kappa_{ijk} \phi_{ijk}^2 \,, \tag{12.1}$$

where U is the total energy, the sums are over all particle pairs ij or all particle triplets ijk. The force constant is $k_{ij} = k$ for bonds in contact and zero otherwise, and $\kappa_{ijk} = \kappa$ for triplets with a common vertice, and zero otherwise. The vector $\mathbf{u}_i$ gives the displacement of node i from its equilibrium position.

Let us address the effective elastic behavior of the percolation system. We would like to describe the material using a material property such as Young's modulus, E, or the shear modulus, G. Let us consider a three-dimensional sample with cross-sectional area $A = L^2$ and length L. Young's modulus, E, relates the tensile stress, σ_{zz}, applied normal to the surface with area A to the elongation ΔL in the z-direction.

$$\sigma_{zz} = \frac{F_z}{A} = E \frac{\Delta L_z}{L} \,, \tag{12.2}$$

We can therefore write the relation between the force F_z and the elongation ΔL_z as

$$F_z = \frac{EA}{L} \Delta L = \frac{EL^2}{L} \Delta L = L^{d-2} E \Delta L \,. \tag{12.3}$$

We recognize this as a result similar to the relation between the conductance and the conductivity of the sample, and we will call $K = L^{d-2}E$ the compliance of the system.

What happens to the compliance of the system as a function of p? When $p < p_c$ there are no connecting paths from one side to another, and the compliance will therefore be zero. It requires zero force F_z to generate an elongation ΔL_z in the system. Notice that we are only interested in the infinitesimal effect of deformation. If we compress the sample we will of course eventually generate a contacting path, but we are only interested in the initial response of the system.

When $p \geq p_c$ there will be at least one path connecting the two edges. For a system with a bending stiffness, there will be a load bearing path through the system, and the deformation ΔL_z of the system requires a finite force, F_z. The compliance K will therefore be larger than zero. We have therefore established that for a system with bending stiffness, the percolation threshold for rigidity coincides with the percolation threshold for connectivity. However, for a central force lattice, we know that the spanning cluster at p_c will contain may singly connected bonds. These bonds will be free to rotate, and as a result a central force network will have a rigidity percolation threshold which is higher than the connectivity

threshold. Indeed, rigidity percolation for central force lattices will have very high percolation thresholds in three dimensions and higher. Here, we will only focus on lattices with bond bending terms.

Based on our experience with percolation systems, we may expect that the behavior of Young's modulus when p approaches p_c from above follows a power-law:

$$E \propto \begin{cases} 0 & p < p_c \\ (p - p_c)^\tau & p > p_c \end{cases} . \tag{12.4}$$

where τ is an exponent describing the elastic system. We will now use our knowledge of the percolation systems to show that this behavior is indeed expected, and to determine the value of the exponent τ.

Let us address the Young's modulus $E(p, L)$ of a percolation system with occupation probability p and a system size L. We could also write E as a function of the correlation length $\xi = \xi(p)$, so that $E = E(\xi, L)$. Young's modulus is in general related to the compliance through $E(\xi, L) = K(\xi, L)L^{d-2}$. We can there address the compliance of the system and then calculate Young's modulus. Let us assume that the correlation length ξ is in the range $a \ll \xi \ll L$, where a is the size of a bond, and L is the system size. We can therefore subdivide the L^d system into boxes of linear size ξ as illustrated in figure 12.1. There will be $(L/\xi)^d$ such boxes. On this scale the system is homogeneous. Each box will have a compliance $K(\xi, \xi)$, and the total compliance will be $K(\xi, L)$. We know that the total compliance of n elements in series is $1/n$ times the compliance of a single element. You can easily convince yourself of this addition rule for spring constants, by addressing two springs in series. Similarly, we know that adding n elements in parallel will make the total system n times stiffer, that is, the compliance will be n times the compliance of an individual element. The total compliance $K(\xi, L)$ is therefore given as

$$K(\xi, L) = K(\xi, \xi)(\frac{L}{\xi})^{d-2} . \tag{12.5}$$

Young's modulus can then be found as

$$E(\xi, L) = L^{-(d-2)}K(\xi, L) = \frac{K(\xi, \xi)}{\xi^{d-2}} . \tag{12.6}$$

We will therefore have to find the compliance $K(\xi, \xi)$. However, we recognize that this is the compliance of the percolation system at $p = p_c$ when the system size L is the correlation length L. We are therefore left with the problem of finding the compliance of the spanning cluster at $p = p_c$ as a function of system size L.

Again, we expect from our experience in the behavior of scaling structures that the compliance will scale with the system size with a

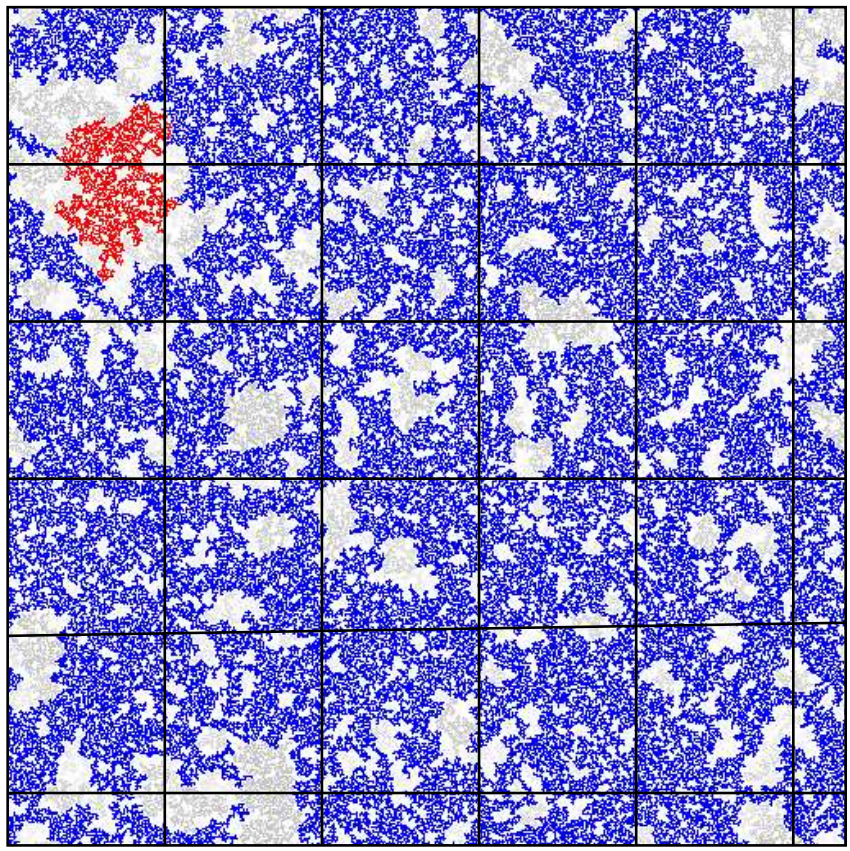

Fig. 12.1 Illustration of subdivision of a system with $p = 0.60$ into regions with a size corresponding to the correlation length, ξ. The behavior inside each box is as for a system at $p = p_c$, whereas the behavior of the overall system is that of a homogeneous system of boxes of linear size ξ.

fractal dimension $\tilde{\zeta}_K$:

$$K \propto L^{\tilde{\zeta}_K} \ . \tag{12.7}$$

Let us now use our standard approach, and first determine a set of bounds for K, which will also serve as a proof of the scaling behavior of K.

We will use arguments based on the total energy of the system. The total energy of a system subjected to a force $F = F_z$ resulting in an elongation ΔL is:

$$U = \frac{1}{2} K (\Delta L)^2 \ , \tag{12.8}$$

where the elongation ΔL is related to the force F through, $\Delta L = F/K$. Consequently,

$$U = \frac{1}{2} K (\frac{F}{K})^2 = \frac{1}{2} \frac{F^2}{K} \ . \tag{12.9}$$

We can therefore relate the elastic energy of a system subjected to the force F directly to the compliance of that system.

Our arguments will be based on the geometrical picture we have of the spanning cluster when $p = p_c$. The cluster consists of singly connected bonds, blobs, and dangling ends. The dangling ends do not influence the elastic behavior, and can be ignored in our discussion. It is only that backbone that contribute to the elastic properties of the spanning cluster. We can find an upper bound for the compliance by considering the singly connected bonds. The system consist of the blob and the singly connected bonds in series. The compliance must include the effect of all the singly connected bonds in series. However, adding the blobs in series as well will only contribute to lowering the compliance. We will therefore get an upper bound on the compliance, by assuming all the blobs to be infinitely stiff, and therefore only include the effects of the singly connected bonds.

Let us therefore study the elastic energy in the singly connected bonds when the cluster is subjected to a force F. The energy, U, can be decomposed in a stretching part, U_s, and a bending part, U_b: $U = U_s + U_b$.

For a singly connected bond from site i to site j, the change in length, $\delta\ell_{ij}$, due to the applied force F is $\delta\ell_{ij} = F/k$, where k is the force constant for a single bond. The energy due to stretching, U_s, is therefore

$$U_s = \sum_{ij} \frac{1}{2} k \delta\ell_{ij}^2 = \sum_{ij} \frac{1}{2} k (\frac{F}{k})^2 = \frac{1}{2} \frac{M_{SC}}{k} F^2 \ , \tag{12.10}$$

where M_{SC} is the mass of the singly connected bonds.

We can find a similar expression for the bending terms. For a bond between sites i and j, the change in angular orientation, $\delta\phi_{ij}$ is due to the torque $T = r_i F$, where r_i is the distance to bond i in the direction normal to the direction of the applied force F: $\delta\phi_{ij} = T/\kappa$. The contribution from bending to the elastic energy is therefore

$$U_b = \sum_{ij} \frac{1}{2} \kappa (\delta\phi_{ij})^2 = \frac{1}{2} \sum_{ij} \kappa (\frac{r_i F}{\kappa})^2 = \frac{1}{2\kappa} M_{SC} R_{SC}^2 F^2 \ , \tag{12.11}$$

where

$$R_{SC}^2 = \frac{1}{M_{SC}} \sum_{ij} r_i^2 \ , \tag{12.12}$$

where the sum is taken over all the singly connected bonds.

The elastic energy of the singly connected bonds is therefore:

$$U_{SC} = (\frac{1}{2k} + \frac{R_{SC}^2}{2\kappa}) M_{SC} F^2 \ , \tag{12.13}$$

and the compliance of the singly connected bonds is

$$K_{SC} = \frac{F^2}{2U} = \frac{1}{(1/k + R_{SC}^2/\kappa) M_{SC}} \ . \tag{12.14}$$

which is an upper bound for the compliance K of the system.

We can make a similar argument for a lower bound for the compliance K of the system. The minimal path on the spanning cluster provides the minimal compliance. The addition of any bonds in parallel will only make the system stiffer, and therefore increase the compliance. We can determine the compliance of the minimal path by calculating the elastic energy of the minimal path. We can make an identical argument as we did above, but we need to replace M_{SC} with the mass, M_{min}, of the minimal path, and the radius of gyration R_{SC}^2 with the radius of gyration of the bonds on the minimal path R_{min}^2.

Kantor [**?**] has provided numerical evidence that both R_{min}^2 and R_{SC}^2 is proportional to ξ^2. When we are studying the spanning cluster at $p = p_c$ this corresponds to R_{min} and R_{SC} being proportional to L. This shows that the dominating term for the energy is the bending and not the stretching energy when p is approaching p_c.

We have therefore determined the scaling relation

$$K_{min} \leq K \leq K_{SC} \ , \tag{12.15}$$

where we have found that when $L \gg 1$, $K_{min} \propto L^{-(D_{min}+2)}$ and $K_{SC} \propto L^{-(D_{SC}+2)}$. That is:

$$L^{-(D_{min}+2)} \leq K(L) \leq L^{-(D_{SC}+2)} \ . \tag{12.16}$$

We have therefore proved that $K(L)$ is indeed a power-law with an exponent $\tilde{\zeta}_K$ satisfying the relation

$$-(D_{min}+2) \leq \tilde{\zeta}_K \leq -(D_{SC}+2) \ . \tag{12.17}$$

We can then use this scaling relation to determine the behavior of Young's modulus from equation 12.6.

$$E(\xi, L) = \frac{K(\xi, \xi)}{\xi^{d-2}} \propto \frac{\xi^{\tilde{\zeta}_K}}{\xi^{d-2}} \propto \xi^{\tilde{\zeta}_K - (d-2)} \ . \tag{12.18}$$

We have therefore found a relation for the scaling exponent τ:

$$E(p, L) = \xi^{-(d-2-\tilde{\zeta}_K)} \propto (p - p_c)^{(d-2-\tilde{\zeta}_K)\nu} \propto (p - p_c)^{\tau} \ . \tag{12.19}$$

The exponent τ is therefore in the range:

$$(d - 2 + D_{SC} + 2)\nu \leq \tau \leq (d - 2 + D_{min} + 2)\nu \ , \tag{12.20}$$

The resulting bounds on the scaling exponents are:

$$(D_{SC} + 2)\,\nu \leq \tau \leq (D_{min} + 2)\,\nu \ , \tag{12.21}$$

For two-dimensional percolation the exponents are approximately

$$3.41 \leq \tau \leq 3.77 \,, \tag{12.22}$$

We see that the bounds are similar to the bounds we found for the exponent $\tilde{\zeta}_R$. This similarity lead Sahimi (1986) and Roux (1986) to conjecture that the elastic coefficients E and G, and the conductivity σ is related through

$$\frac{E}{\sigma} \propto \xi^{-2} \,. \tag{12.23}$$

and therefore that

$$\tau = \mu + 2\nu = (d + \tilde{\zeta}_R)\nu \,. \tag{12.24}$$

which is well supported by numerical studies.

In the limit of high dimensions, $d \geq 6$, the relation $\tau = \mu + 2\nu = 4$ becomes exact. However, we can use as a rule of thumb that the exponent $\tau \simeq 4$ in all dimensions $d \geq 2$.

Diffusion in disordered media

13

For a random walker, as well as for diffusional processes, the average distance $\mathbf{R}$ increases with the square root of time. However, we have previously only considered random walks in free space. How will the random walker behave if it is restricted to move on a structure with a scaling geometry, such as percolation clusters? In this chapter we will address diffusion on scaling structures.

In our previous discussion of random walks, we described the motion in time and space of the random walker through the probability density $P(\mathbf{r}, t)$ so that $P(\mathbf{r}, t)d\mathbf{r}dt$ is the probability for the random walker to be in the volume $\mathbf{r}d\mathbf{r}$ in the time period t to $t + dt$. For a random walker on a grid, the probability to be at a grid position i is given as $P_i(t)$. We argued that the probability for the walker to be at a position i at the time $t = t + \delta t$ was

$$P_i(t + \delta t) = P_i(t) + \sum_j [\sigma_{j,i} P_j(t) - \sigma_{i,j} P_i(t)]\delta t \ , \tag{13.1}$$

where the sum is over all neighbors j of the site i. The term $\sigma_{i,j}$ is the transition probability. The first term in the sum represents the probability that the walker during the time period δt walks into site i from site j, and the second term represents the probability that the walker during the time period δt walks from site i to one of the neighboring sites j.

When $\delta t \to 0$ this equation approaches a differential equation

$$\frac{\partial P_i}{\partial t} = \sum_j [\sigma_{j,i} P_j(t) - \sigma_{i,j} P_i(t)] \ . \tag{13.2}$$

We can now assume that the transition probability is equal for all the neighbors, so that $\sigma_{i,j} = 1/Z$, where Z is the number of neighbors. In this case, the differential equation simplifies to

$$\frac{\partial P}{\partial t} = D\nabla^2 P \ , \tag{13.3}$$

which we recognize as the diffusion equation.

The solution to this equation is

$$P(\mathbf{r},t) = \frac{1}{(2\pi Dt)^{d/2}} e^{-r^2/2Dt} = \frac{1}{(2\pi)^{d/2}|\mathbf{R}|^2} e^{-\frac{1}{2}(\frac{r}{|\mathbf{R}|})^2} \ , \tag{13.4}$$

where we have introduced $|\mathbf{R}| = \sqrt{Dt}$.

We have also found that the moments of this distribution are

$$\langle r^k \rangle = A_k R(t)^k \propto t^{k/2} \ , \tag{13.5}$$

and specifically, that

$$\langle r^2 \rangle = \int P(\mathbf{r},t) r^2 d\mathbf{r} = R^2(t) = Dt \ . \tag{13.6}$$

The structure generated by this process is therefore a unifractal.

13.1 Random walks on clusters

Let us study what happens if we drop a random walker onto a random position in the percolation system and measure the position $\mathbf{r}$ of the walker as a function of time. That is, we set $P(\mathbf{r},0) = \delta(\mathbf{r})$, and ask for the solution to $P(\mathbf{r},t)$.

We know that when $p = 1$, the problem will be the ordinary diffusion problem. We will therefore expect that when $p \simeq 1$, the system behaves similar to ordinary diffusion. When p approaches 0 we expect the maximum value of $\langle r \rangle$ to approach a constant value, because the walker will be trapped on a cluster of finite size.

13.1.1 Diffusion for $p < p_c$

Let us first address the case when $p < p_c$ and let us drop the walker onto a cluster of size s. We will expect that after a long time, $\langle r^2 \rangle \propto R_s^2$. If we repeat this experiment many times, each time dropping the walker onto a random occupied point in the system, we need to take the average over all clusters of size s and over all starting positions, getting

$$\left[\langle r^2 \rangle\right] \propto \left[R_s^2\right] = \frac{\sum_s R_s^2 s n(s,p)}{\sum s n(s,p)} \propto (p_c - p)^{\beta - 2\nu} \ , \tag{13.7}$$

because we know that $R_s \propto s^{1/D}$. This average is different than the average we used as the definition of the correlation length, since we used the weight $s^2 n(s,p)$ for the correlation length. This is the reason for

the scaling exponent to be $\beta - 2\nu$ and not simply 2ν as we got for the correlation length.

This transition occurs after some transient time t_0. What can we say about t_0? We expect t_0 to diverge when $p \to p_c$.

13.1.2 Diffusion for $p > p_c$

Let us then address the case when $p > p_c$. We know that when $p = 1$, $\langle r^2 \rangle = D(1)t$. We will therefore write the general relation for $p > p_c$:

$$\langle r^2 \rangle = D(p)t \ , r \gg \xi \ . \tag{13.8}$$

What behavior do we expect from $D(p)$? We expect $D(p)$ to increase in a way similar to the density of the backbone or the conductivity σ. In fact, the Einstein relation for diffusion relates the diffusion constant to the conductance through:

$$D(p) \propto \sigma(p) \propto (p - p_c)^{\mu} \ . \tag{13.9}$$

We therefore expect that when $p > p_c$, and the time is larger than a crossover time $t_0(p)$, that the behavior is scaling with exponent μ, identical to that of conductivity.

13.1.3 Scaling theory

Let us develop a scaling theory for the behavior of $\langle r^2 \rangle$. We will assume that when the time is smaller than a cross-over time, the behavior is according to a power-law with exponent $2k$, and that when the time is larger than the cross-over time, the behavior is either that of diffusion with diffusion constant $D(p)$, or it reaches a constant plateau for the case when $p < p_c$.

Let us introduce a scaling ansatz with these properties:

$$\langle r^2 \rangle = t^{2k} f[(p - p_c)t^x] \ . \tag{13.10}$$

We could also have started from any of the end-points, such as from the assumption that

$$\langle r^2 \rangle = (p_c - p)^{\beta - 2\nu} G_1(\frac{t}{t_0}) \ , \tag{13.11}$$

or

$$\langle r^2 \rangle = (p - p_c)^{\mu} G_2(\frac{t}{t_0}) \ . \tag{13.12}$$

We have two unknown exponents k and x that must be determined from independent knowledge. We will assume that the function $f(u)$ has the behavior

$$f(u) = \begin{cases} \text{const.} & |u| \ll 1 \\ u^{\mu} & u \gg 1 \\ (-u)^{\beta-2\nu} & u \ll -1 \end{cases} \tag{13.13}$$

Let us now address the various limits in order to determine the scaling exponents k and x in terms of the known exponents. First, let us address the limit when $u \gg 1$, that is, $p > p_c$, and we have found that

$$\langle r^2 \rangle \propto (p - p_c)^{\mu} t \ , \tag{13.14}$$

which should correspond to the functional form from the ansatz:

$$(p - p_c)^{\mu} t \propto t^{2k} f((p - p_c)t^x) \propto t^{2k}[(p - p_c)t^x]^{\mu} \ . \tag{13.15}$$

This resulting the the exponent relation

$$2k = 1 - \mu x \ , \tag{13.16}$$

or

$$k = \frac{1 - \mu x}{2} \ . \tag{13.17}$$

Similarly, we know that the behavior in the limit of $u \ll -1$ should be proportional to $(p_c - p)^{\beta-2\nu}$. Consequently, the scaling ansatz gives

$$(p_c - p)^{\beta-2\nu} \propto t^{2k} f((p - p_c)t^x) \propto t^{2k}[(p_c - p)t^x]^{\beta-2\nu} \ , \tag{13.18}$$

which resulting the exponent relation;

$$2k + x(\beta - 2\nu) = 0 \ . \tag{13.19}$$

We solve the two equations for x and k, finding

$$k = \frac{1}{2}[1 - \frac{\mu}{2\nu + \mu - \beta}] \ , \tag{13.20}$$

and

$$x = \frac{1}{2\nu + \mu - \beta} \ . \tag{13.21}$$

This argument therefore shows that the scaling ansatz is indeed consistent with the limiting behaviors we have already determined, and it allows us to make a prediction for k and x.

When $p = p_c$, we find that

$$\langle r^2 \rangle \propto t^{2k} = t^{\frac{2\nu-\beta}{2\nu+\mu-\beta}} \ , \tag{13.22}$$

We can write this relation in the same way as we wrote the behavior of an ordinary random walk,

$$t \propto r^{d_w} \ , \tag{13.23}$$

where d_w is the dimension of the random walk. Here we have now found that

$$d_w = \frac{1}{k} = 2 + \frac{\mu}{\nu - \frac{\beta}{2}} \ , \tag{13.24}$$

which is a number larger than 2. This means that for a given time, the walk remains more compact, which is consistent with our intuition.

We have introduced a cross-over time, t_0, which is defined so that

$$(p - p_c) t_0^x \simeq 1 \ , \tag{13.25}$$

which gives

$$t_0 \propto |p - p_c|^{-1/x} \propto |p - p_c|^{-(2\nu + \mu - \beta)} \ . \tag{13.26}$$

How can we interpret this relation? We could decompose the relation to be:

$$t_0 \propto \frac{|p - p_c|^{\beta - 2\nu}}{|p - p_c|^{\mu}} \ , \tag{13.27}$$

where we know that the average radius of gyration for clusters are

$$[R_s^2] \propto |p - p_c|^{\beta - 2\nu} \ , \tag{13.28}$$

This gives us an interpretation of the cross-over time for diffusion:

$$t_0(p) \propto \frac{[R_s^2]}{D} \ , \tag{13.29}$$

where D is the diffusion constant. Why is this time not proportional to ξ^2/D, the time it take to diffuse a distance proportional to the correlation length? The difference comes from the particular way we devised the experiment: the walker was dropped onto a randomly selected occupied site.

Let us now address what happens when $p > p_c$. In this case, the contributions to the variance of the position has two main terms: one term from the spanning cluster and one term from the finite clusters.

$$[\langle r^2 \rangle] = Dt = \frac{P}{p} D' t + R_s^2 \ , \tag{13.30}$$

where the first term, $P/pD't$ is the contribution from the random walker on the infinite cluster. This term consists of the diffusion constant D' for a walker on the spanning cluster, and the prefactor P/p which comes from the probability for the walker to land on the spanning cluster: For a random walker placed randomly on an occupied site in the system, the probability for the walker to land on the spanning cluster is P/p, and the probability to land on any of the finite clusters is $1 - P/p$. The second term is due to the finite cluster. This term reaches a constant value for large times t. The only time dependence is therefore in the first term, and we can write:

$$Dt = \frac{P}{p} D' t \ , \tag{13.31}$$

for long times, t. That is:

$$D' = \frac{Dp}{P} \propto (p - p_c)^{\mu-\beta} \propto \xi^{-\frac{\mu-\beta}{\nu}} \propto \xi^{-\theta} \ . \tag{13.32}$$

where we have introduce the exponent

$$\theta = \frac{\mu - \beta}{\nu} \ . \tag{13.33}$$

We have therefore found an interpretation of the cross-over time t_0, and, in particular for the appearance of the β in the exponent. We see that the cross-over time is

$$t_0 \propto \frac{|p - p_c|^{\beta-2\nu}}{|p - p_c|^{\mu}} \propto \frac{\xi^2}{D'} \ . \tag{13.34}$$

The interpretation of t_0 is therefore that t_0 is the time the walker needs to travel a distance ξ when it is diffusing with diffusion constant D' on the spanning cluster.

13.1.4 Diffusion on the spanning cluster

How does the random walker behave on the spanning cluster? We have found that for $p > p_c$ and for $t > t_0$ the deviation increases according to

$$\langle r^2 \rangle = D' t \propto (p - p_c)^{\mu-\beta} t \ , \tag{13.35}$$

and for $t < t_0$, we expect the behavior to be

$$\langle r^2 \rangle \propto t^{2k'} \ , \tag{13.36}$$

as illustrated in figure 13.1. We expect the relations to be valid up to the point (t, ξ^2), where both descriptions should provide the same result. Therefore we expect

$$\xi \propto t_0^{2k'} \propto D' t_0 \ , \tag{13.37}$$

and therefore that

$$t_0 \propto \frac{\xi^2}{D'} \propto \frac{(p - p_c)^{-2\nu}}{(p - p_c)^{\mu-\beta}} \propto (p - p_c)^{-(2\nu+\mu-\beta)} \ . \tag{13.38}$$

Consequently, the value of t_0 is the same on the spanning cluster as for the general process on any cluster. In general, we can interpret t_0 as the time it takes for the walker to diffuse to the end of the cluster when $p < p_c$, and the time it takes to diffuse to a distance ξ on the spanning cluster when $p > p_c$.

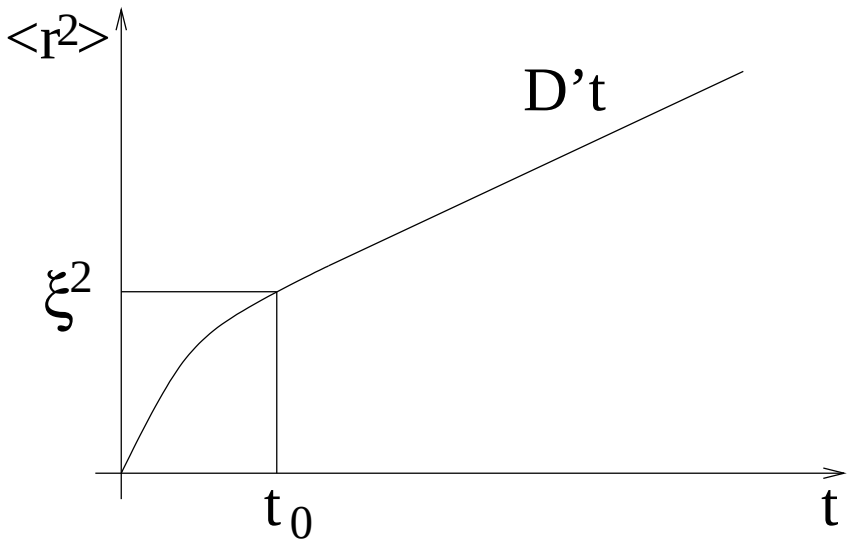

Fig. 13.1 Illustration of the distance $\langle r^2 \rangle$ as a function of time for a random walker on the percolation cluster.

Let us check the other exponent, k'. We find that

$$\xi^2 \propto (p - p_c)^{-2(2\nu+\mu-\beta)k'} \ , \tag{13.39}$$

and therefore that

$$k' = \frac{\nu}{2\nu + \mu - \beta} \ , \tag{13.40}$$

which is not the same as we found in equation 13.20 for all clusters. We find that k' is slightly larger than k.

What is the interpretation of k'? If we consider random walks on the spanning cluster only, the behavior at $p = p_c$ is described by

$$\langle r^2 \rangle \propto t^{2k'} \ , \tag{13.41}$$

this gives

$$r^{1/k'} \propto t \propto r^{d_w} \ , \tag{13.42}$$

where d_w can be interpreted as the dimension of the random walk. For the case of random walkers on the spanning cluster at $p = p_c$ we have therefore found that4

$$d_w = 2 + \frac{\mu - \beta}{\nu} \ . \tag{13.43}$$

The fractal dimension is larger than 2. This corresponds to the walker getting stuck on the percolation cluster, and the structure of the walk is therefore more dense or compact.

13.1.5 The diffusion constant D

We can use the theory we have developed so far to address the behavior of the diffusion constant with time. Fick's law can generally be formulated as

$$\langle r^2 \rangle = \mathcal{D}t \ , \tag{13.44}$$

or, equivalently, we can find the diffusion constant for Fick's law from:

$$\mathcal{D} = \frac{\partial}{\partial t} \langle r^2 \rangle \ . \tag{13.45}$$

Now, we have established that for diffusion on the spanning cluster for $p = p_c$, the diffusion is anomalous. That is, the relation between the square distance and time is not linear, but a more complicated power-law relationship

$$\langle r^2 \rangle \propto t^{2k'} \ . \tag{13.46}$$

As a result, we find that the diffusion constant $\mathcal{D}'$ for diffusion on the spanning cluster defined through Fick's law is

$$\mathcal{D}' \propto \frac{\partial}{\partial t} t^{2k'} \propto t^{2k'-1} \ . \tag{13.47}$$

We can therefore interpret the process as a diffusion process where $\mathcal{D}$ decays with time.

In the anomalous regime, we find that

$$r \propto t^{k'} \ , \tag{13.48}$$

and therefore that

$$r^{1/k'} \propto t \ . \tag{13.49}$$

We can therefore also write the diffusion constant $\mathcal{D}'$ as

$$\mathcal{D}' \propto t^{2k'-1} \propto r^{2-1/k'} \propto r^{-\theta} \ . \tag{13.50}$$

We could therefore also say that the diffusion constant is decreasing with distance.

The reverse is also generally true: Whenever $\mathcal{D}$ depends on the distance, we will end up with anomalous diffusion.

We can also relate these results back to the diffusion equation. The diffusion equation for the random walk was:

$$\frac{\partial P}{\partial t} = \mathcal{D}' \nabla^2 P = \nabla \mathcal{D}' \nabla P \ , \tag{13.51}$$

where the last term is the correct term if the diffusion constant depends on the spatial coordinate.

We can rewrite the dimension, d_w, of the walk to make the relation between the random walker and the dimensionality of the space on which

it is moving more obvious:

$$d_w = 2 - d + \frac{\mu}{\nu} + d - \frac{\beta}{\nu} \,, \tag{13.52}$$

where we recognize the first term as

$$\tilde{\zeta}_R = 2 - d + \frac{\mu}{\nu} \,, \tag{13.53}$$

and the second term as the fractal dimension, D, of the spanning cluster:

$$D = d - \frac{\beta}{\nu} \,. \tag{13.54}$$

We have therefore established the relation

$$d_w = \tilde{\zeta}_R + D \,. \tag{13.55}$$

This relation is actually generalizable, so that for a random walker restricted to only walk on the backbone, the dimension of the walker is

$$d_{w,B} = \tilde{\zeta}_R + D_B \,. \tag{13.56}$$

13.1.6 The probability density $P(\mathbf{r}, t)$

Dynamic processes in disordered media 14

So far we have studied the behavior and properties of systems with disorder, such as the model porous material we call the percolation system. That is, we have studied properties that depend on the existing disorder of the material. In this chapter, we will start to address dynamical processes that generate percolation-like disordered structures, but where the structure evolve, develop, and change in time.

The first dynamic problem we will address is the formation diffusion fronts, and we will demonstrate that the front of a system of diffusing particles can be described as a percolation system.

The second dynamic problem we will address is the slow displacement of one fluid by another in a porous medium. We will in particular demonstrate that the invasion percolation process generates a fractal structure similar to the percolation cluster by itself - it is a process that drives itself to a critical state, similar to the recently introduced notion of Self-Organized Criticality [**?**]. We will then address how we can study similar processes in the gravity field, and, in particular, the influence of stabilizing and destabilizing mechanisms. Invasion percolation in a destabilizing gravity field provides a good model to describe and understand the process of primary migration.

The last example we will address is directed percolation.

14.1 Diffusion fronts

The first dynamical problem we will address is the structure of a diffusion front. Let us address a diffusion process on a square lattice. One example of such a process is the two-dimensional diffusion of particles from a source at $x = 0$ into the $x > 0$ plane, when particles are not allowed to overlap. The system of diffusing particles is illustrated in figure 14.1.

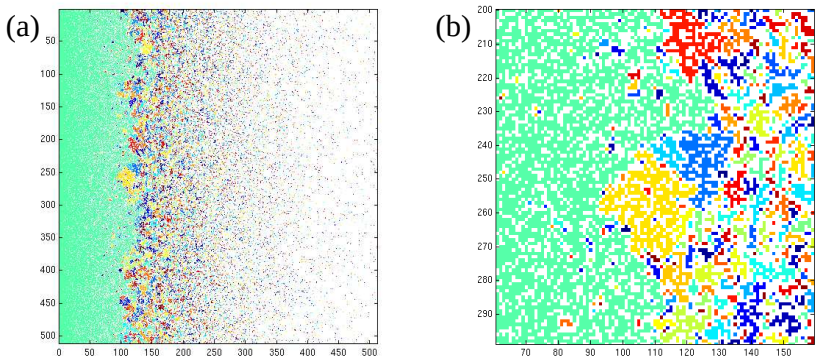

Fig. 14.1 Illustration of the diffusion front. Particles are diffusing from a source at the left side. We address the front separating the particles connected to the source from the particles not connected to the source. The average distance is given by x_c shown in the figure. The width of the front, ξ, is also illustrated in the figure. The different clusters are colored to distinguish them from each other. The close-up in figure (b) illustrates the finer details of the diffusion fronts, and the local cluster geometries.

For this problem we know the exact solution for the concentration, $c(x,t)$, or particles, corresponding to the occupation probability $P(x,t)$. The solution to the diffusion equation with a constant concentration, or $P(x=0,t)=1$, is the error function given as the integral over a Gaussian function:

$$P(x,t) = 1 - erf(\frac{x}{\sqrt{Dt}}) \, , \tag{14.1}$$

where the error function is defined as the integral:

$$erf(u) = \frac{2}{\sqrt{2\pi}} \int_0^u e^{\frac{-v^2}{2}} dv \, . \tag{14.2}$$

This solution produces the expected deviation $\langle x^2 \rangle = Dt$, where D is the diffusion constant for the particles. There is no y (or z) dependence for the solution.

We will address the structure of connected clusters of diffusing particles. Two particles are connected if they are neighbors so that they inhibit each others diffusion in a particular direction. If we fix t, we notice that the system will be compact close to $x=0$, and that there only will be a few thinly spread particles when $x \gg \sqrt{Dt}$. In this system, the occupation probability varies with both time t and spatial position x. However, we expect the system of diffusing particles to be connected to the source out to a distance x_c corresponding to the point where the occupation probability is equal to the percolation threshold p_c for the lattice type studied. That is:

$$P(x_c,t) = p_c \, , \tag{14.3}$$

defines the center of the diffusion front: the front separating the particles that are connected to the source from the particles that are not connected to the source. We notice that $x_c(t) = \sqrt{Dt}$.

What is the width of the diffusion front? For a given time t, the occupation probability decreases with $\delta x = x - x_c$. Similarly, the correlation length will therefore also depend on the distance δx to the average position of the front. We expect that a cluster may be connected to the front if it is within a distance ξ of x_c. Particles that are further away than the local correlation length, ξ, will not be connected over such distances, and will therefore not be connected. Particles that are closer to x_c than ξ will typically by connected through some connecting path. We will therefore introduce ξ as the width of the front, corresponding to the distance at which the local correlation length, due to the occupation probability $P(x,t)$, is equal to the distance from x_c. The local correlation length $\xi(x)$ is given as

$$\xi(x) = \xi_0 |P(x,t) - p_c|^{-\nu} , \tag{14.4}$$

The distance w at which $\xi(x_c + w) = w$ gives the width of the front. We can write this self-consistency equation for w as

$$w = \xi_0 |P(x+w,t) - p_c|^{-\nu} . \tag{14.5}$$

Let us introduce a Taylor expansion of $P(x)$ around $x = x_c$:

$$P(x,t) \simeq P(x_c,t) + \left.\frac{dP}{dx}\right|_{x_c} (x - x_c) , \tag{14.6}$$

where we recognize that $x_c \propto \sqrt{Dt}$ gives

$$\left.\frac{dP}{dx}\right|_{x_c} \propto \frac{1}{\sqrt{Dt}} \propto \frac{1}{x_c} . \tag{14.7}$$

We insert this into the self-consistency equation equation 14.5 getting

$$w = \xi_0 |w \left.\frac{dP}{dx}\right|_{x_c} |^{-\nu} \propto (w/x_c)^{-\nu} , \tag{14.8}$$

which gives

$$w \propto x_c^{\nu/(1+\nu)} . \tag{14.9}$$

The width of the front therefore scales with the average position of the front, and the scaling exponent is related to the scaling exponent of the correlation length for the percolation problem.

What happens in this system with time? Since x_c is increasing with time, we see that the relative width decreases:

$$\frac{w}{x_c} \propto \frac{x_c^{\nu/(1+\nu)}}{x_c} \propto x_c^{-\frac{1}{1+\nu}} . \tag{14.10}$$

This effect will also become apparent under renormalization. Applying a renormalization scheme with length b, will result in a change in the front width by a factor $b^{\nu/(1+\nu)}$, but along the y-direction the rescaling will simply be by a factor b. Successive applications will therefore make the front narrower and narrower. This difference in scaling along the x and the y axis is referred to as self-affine scaling, in contrast to the self-affine scaling where the rescaling is the same in all directions. We will return to this concept when we address surface growth processes.

14.2 Invasion percolation

We will now study the slow injection of a non-wetting fluid into a porous medium saturated with a wetting fluid. In the limit of infinitely slow injection, this process is termed invasion percolation for reasons that will soon become obvious.

When a non-wetting fluid is injected slowly into a saturated porous medium, the pressure in the non-wetting fluid must exceed the capillary pressure in a pore-throat for the fluid to propagate from one pore to the next, as illustrated in fig. 14.2. The pressure difference, δP needed corresponds to the capillary pressure P_c, given as

$$P_c = \frac{\Gamma}{\epsilon} \,, \tag{14.11}$$

where Γ is the interfacial surface tension, and ϵ is the characteristic size of the pore-throats in the porous medium. However, there will be some disorder present in the porous medium corresponding to local variation in the characteristic pore sizes ϵ. This will lead to a distribution of capillary pressures threshold P_c needed to invade a particular pore. We will assume that the medium can be described as a set of pores connected with pore throats with a uniform distribution of capillary pressure thresholds, and we will assume that the capillary pressure thresholds are not correlated but statistically independent. We can therefore rescale the pressure scale, by subtracting the minimum pressure threshold and dividing by the range of pressure thresholds, and describe the system as a matrix of critical pressures P_i required to invade a particular site.

The fluid displacement process can then be modeled by assuming that all the sites on the left side of the matrix are in contact with the invading fluid. The pressure in the invading fluid is increased slowly, until the fluid invades the connected site with the lowest pressure threshold. This generates a new set of invaded sites in contact with the inlet, and a new set of neighboring sites. The invasion process continues until the invading fluid reaches the opposite side. Further injection will then not produce any further fluid displacement, the fluid will flow through the system through the open path generated.

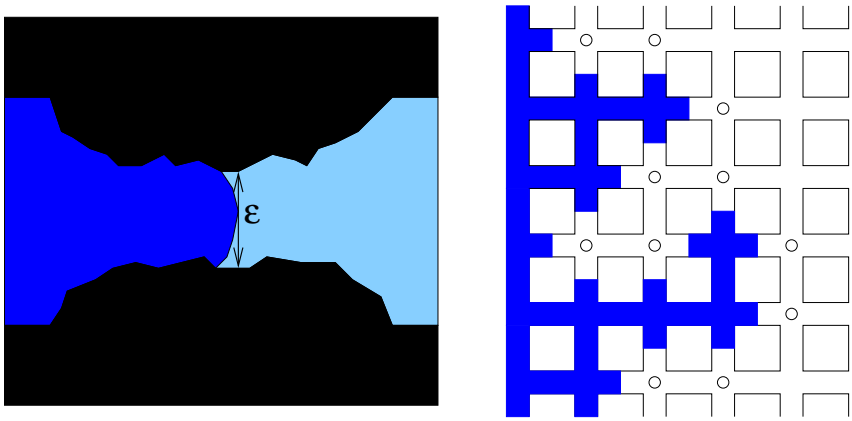

Fig. 14.2 Illustration of the invasion percolation process in which a non-wetting fluid is slowly displacing a wetting fluid. The left figure shows the interface in a pore throat: the pressure in the invading fluid must exceed the pressure in the displaced fluid by an amount corresponding to the capillary pressure $P_c = \Gamma/\epsilon$, where Γ is the interfacial surface tension, and ϵ is a characteristic length for the pore throat. The right figure illustrates the invasion front after injection has started. The fluid may invade any of the sites along the front indicated by small circles. The site with the smallest capillary pressure threshold will be invaded first, changing the front and exposing new boundary sites.

The resulting pattern of injected nodes is illustrated in figure 14.3, where the colors indicate the pressure at which the injection took place. It can be seen from the figure that the injection occurs in bursts. When a site is injected, many new connected neighbors are available as possible sites to invade. As the pressure approaches the pressure needed to percolate to the other side, these newly appearing sites of the front will typically also be invaded, and invasion will occur in gradually larger regions. These bursts have be characterize by Furuberg *et al.* [**?**], and it can be argued that the distribution of burst sizes as well as the time between bursts are power-law distributed.

Based on this algorithmic model for the fluid displacement process, it is also easy to connect the invasion percolation problem with ordinary percolation. For an injection pressure of p, all sites with critical pressure below or equal to p are in principle available for the injection process. However, it is only the clusters of such sites connected to the left side that will actually be invaded, since the invasion process requires a connected path from the inlet to the site for a site to be filled. We will therefore expect that the width of the invasion percolation front corresponds to the correlation length $\xi = \xi_0(p_c - p)^{-\nu}$ as p approaches the percolation threshold p_c, because this is the length at which clusters are connected. That is, cluster that are a distance ξ from the left side will typically be connected to the left side, and therefore connected, whereas clusters that are further away than ξ will typically not be connected and therefore not invaded. This shows that the critical pressure will correspond to p_c. This also shows that when the fluid reaches the opposite side, the system is

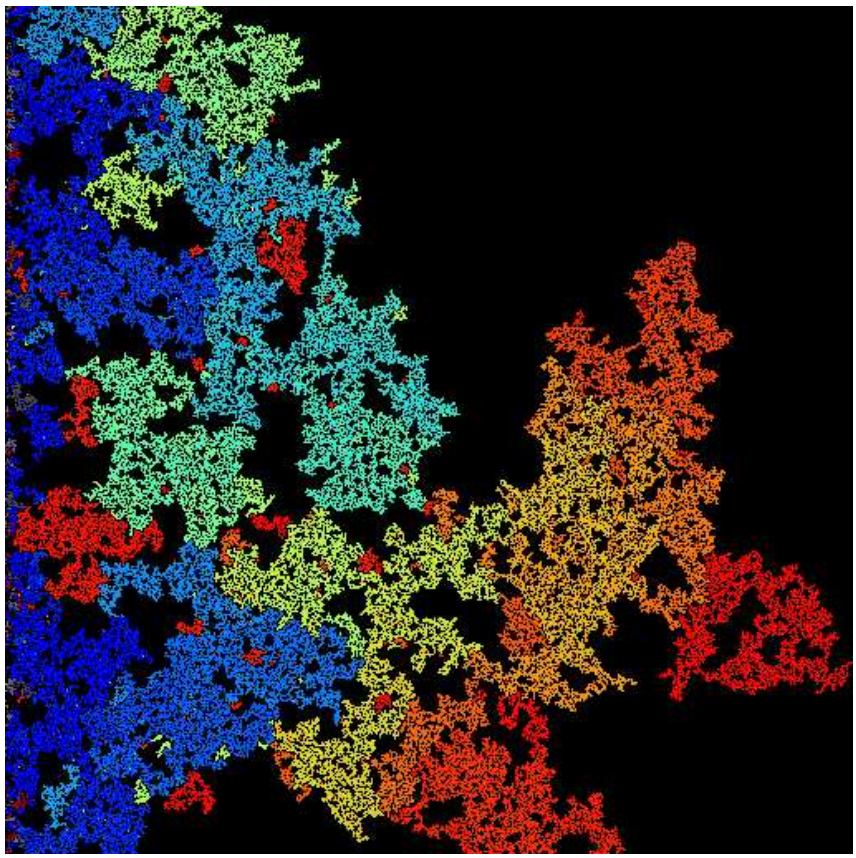

Fig. 14.3 Illustration of the invasion percolation cluster. The color-scale indicates normalized pressure at which the site was invaded.

exactly at p_c, and we expect the invasion percolation cluster to have the same scaling properties as the spanning cluster at $p = p_c$. There will be small differences, because the invasion percolation cluster also contains smaller clusters connected to the left side, but we do not expect these to change the scaling behavior of the cluster. That is, we expect the fractal dimension of the invasion percolation cluster to be D. This implies that the density of the displaced fluid decreases with system size.

The process outlined above does, however, not contain all the essential physics of the fluid displacement process. For displacement of an incompressible fluid, a region that is fully bounded by the invading fluid cannot be invaded, since the displaced fluid does not have any place to go. Instead, we should study the process called invasion percolation with trapping. It has been found that when trapping is included, the fractal dimension of the invasion percolation cluster is slightly smaller [**?**]. In two dimensions, the dimension is $D \simeq 1.82$.

This difference between the process with and without trapping disappears for three-dimensional geometries because trapping become unlikely in dimensions higher than 2. Indeed, direct numerical modeling shows

that the fractal dimension for both the ordinary percolation system and invasion percolation is $D \simeq 2.5$ for invasion percolation with and without trapping [**?**].

14.2.1 Gravity stabilization

The invasion percolation cluster displays self-similar scaling similar to that of ordinary percolation. This implies that the position $h(x,p)$ of the fluid front as a function of the non-dimensional applied pressure p is given as the correlation length - since this is how far clusters connected to the left side typically are connected. That is, when p approaches p_c, the average position of the front is $\bar{h}(x,p) = \xi(p) = \xi_0(p_c - p)^{-\nu}$. The width, $w(p)$ of the front is also given as the correlation length:

$$w(p) = \xi_0(p_c - p)^{-\nu} \,, \tag{14.12}$$

as p approaches p_c both the front position and the front width diverges, that is, both the front position $\bar{h}$ and the width, w, are proportional to the system size L:

$$\bar{h} \propto w \propto L \,, \tag{14.13}$$

However, when the system size increases we would expect other stabilizing effects to become important. For a very small, but finite fluid injection velocity, the viscous pressure drop will eventually become important and comparable to the capillary pressure. Also, any deviation from a completely flat system or for a system with a slight different in densities, the effect of the hydrostatic pressure term will also eventually become important. We will now demonstrate how we may address the effect of such a stabilizing (or destabilizing) effect.

Let us assume that the invasion percolation occurs in the gravity field. This implies that the pressure needed to invade a pore depends both on the capillary pressure, and on a hydrostatic term. The pressure P_i^c needed to invade site i at vertical position x_i in the gravity field is:

$$P_i^c = \frac{\Gamma}{\epsilon} + \Delta\rho g x_i \,, \tag{14.14}$$

We can again normalize the pressures, resulting in

$$p_i^C = p_i^0 + \frac{\Delta\rho g}{\Gamma\epsilon^2} x_i' \,, \tag{14.15}$$

where the coordinates are measured in units of the pore size, ϵ, which is the unit of length in our system. The last term is called the Bond number:

$$Bo = \frac{\Delta\rho g}{\Gamma\epsilon^2} \,, \tag{14.16}$$

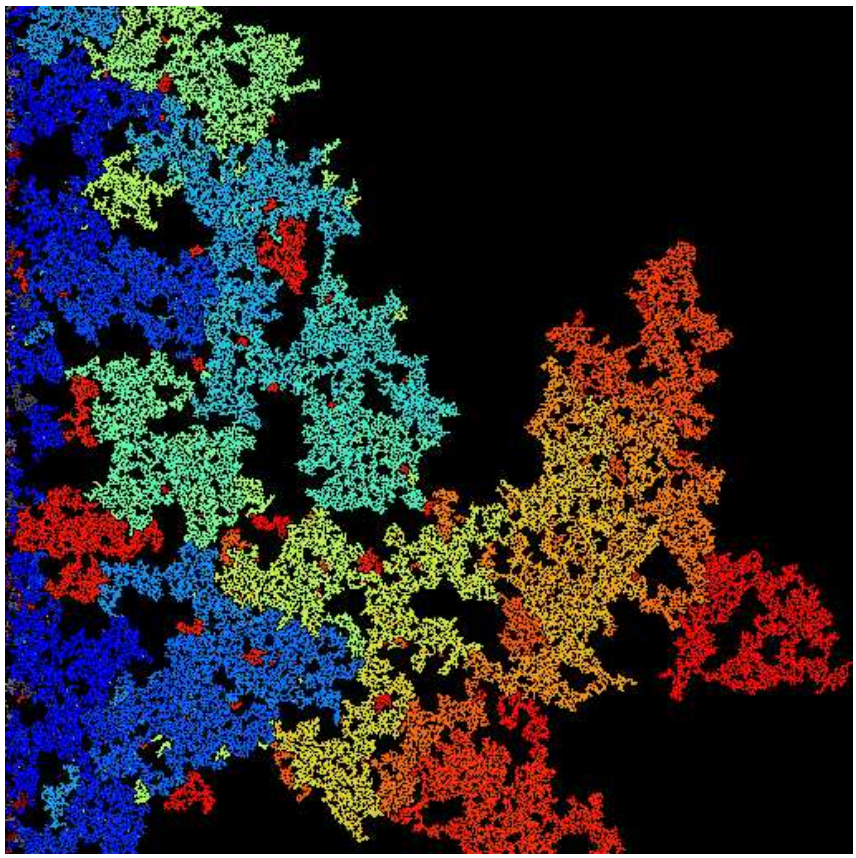

Fig. 14.4 Illustration of the gravity stabilized invasion percolation cluster for $g = 0$, $g = 10^{-4}$, $g = 10^{-3}$, and $g = 10^{-2}$. The color-scale indicates normalized pressure at which the site was invaded.

Here, we will include the effect of the bond number in a single number g, so that the critical pressure at site i is:

$$p_i^c = p_i^0 + g x_i' \,, \tag{14.17}$$

where p_i^0 is a random number between 0 and 1. The invasion percolation front for various numbers of g is illustrated in figure 14.4.

This problem is similar to the diffusion front problem. For an applied pressure p the front will typically be connected up to an average distance x_c given as

$$p = p^0 + x_c g \,. \tag{14.18}$$

The front will also extend beyond the average front position. The occupation probability at a distance a from the front is $p' = p_c - ag$, since fewer sites will be set beyond the front due to the stabilizing term g. A site at a distance a is connected to the front if this distance a is shorter to or equal to the correlation length for the occupation probability p' at this distance. The maximum distance a for which a site is connected to

the front therefore occurs when

$$a = \xi(p') = \xi_0(p_c - p')^{-\nu} \ . \tag{14.19}$$

This gives

$$a = \xi(p') = \xi_0(p_c - p')^{-\nu} = \xi_0(p_c - (p_c - ag))^{-\nu} = \xi_0(ag)^{-\nu}a \ . \tag{14.20}$$

This gives

$$a \propto g^{-\nu/(1+\nu)} \ , \tag{14.21}$$

as the front width. We leave it as an exercise to show find the form of the position $h(p, g)$, and the width, $w(p, g)$, as a function of p and g. We observe that the width has a reasonable dependence on g. When g approaches 0, the width diverges. This is exactly what we expect since the limit $g = 0$ corresponds to the limit of ordinary invasion percolation.

This discussion demonstrates a general principle that we can use to study several stabilizing effect, such as the effect of viscosity or other material or process parameters that affect the pressure needed to advance the front. The introduction of a finite width or characteristic length ξ that can systematically be varied in order to address the behavior of the system when the characteristic length diverges is also a powerful method of both experimental and theoretical use.

14.2.2 Gravity destabilization

The gravity destabilized invasion percolation process corresponds to the case when a less dense fluid is injected at the bottom of a denser fluid. This is similar to the process known as secondary migration, where the produced oil is migrating up through the sediments filled with denser water. Examples of the destabilizing front is shown in figure 14.5.

We can make a similar argument for the case when $g < 0$, but in this case the front is destabilized, and the correlation length $\xi \propto (-g)^{-\nu/(1+\nu)}$ corresponds to the width of the finger extending front the front. The extending finger can be modeled as a sequence of blobs of size ξ extending from the flat surface. This implies that the region responsible for the transport of oil in secondary migration is essentially one-dimensional structures: lines with a finite width w. The amount of hydrocarbons left in the sediments during this process is therefore negligible. However, there will be other effects, such as the finite viscosity and the rate of production compared to the rate of flow, which will induce more than one finger. However, the full process has only to a small degree been addressed. Gravity destabilized invasion percolation is used as a modeling tool in studies of petroleum plays and a commercial software package is available for its simulation. Alternatively the reader may use the 10-line Matlab script found in the exercise section of this book.

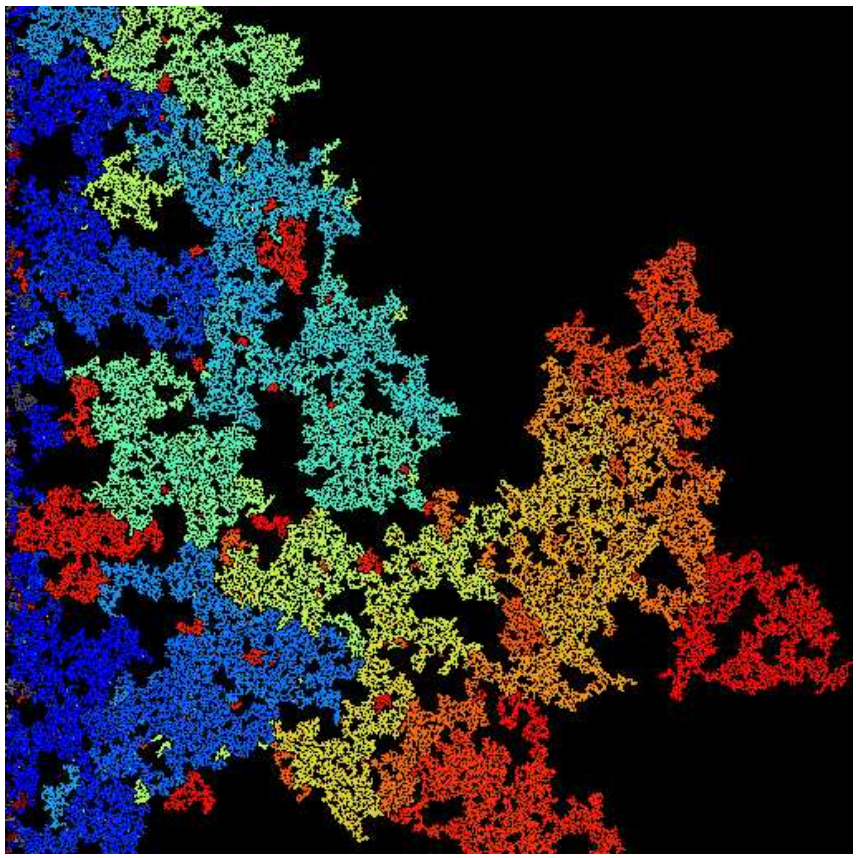

Fig. 14.5 Illustration of the gravity de-stabilized invasion percolation cluster for $g = 0$, $g = -10^{-4}$, $g = -10^{-3}$, and $g = -10^{-2}$. The color-scale indicates normalized pressure at which the site was invaded.

14.3 Directed percolation

Computer Code

15

The course described in this book contains three important, interdependent parts: The basic text of the textbook, the theoretical exercises presented throughout the book, and a set of numerical exercises and tutorials. We believe that the use of the numerical exericises and tutorials is essential in order to develop an intuitive understanding for the concepts presented in the book. In particular, the use of numerical model shows that even though it is difficult to obtain theoretical results for two-dimensional percolation problems, it is easy to generate and study the geometry and physics on two-dimensional percolation problems. We believe the use of numerical methods therefore helps make the concepts more concrete and available to the reader.

We have developed a set of examples for each of the chapters. The examples guide the reader through a set of exercises that will allow the reader to build a toolbox of programs and method that can be used to address, model, and analyze disordered systems. The solution to the exercises can be found separately. However, example codes that are provided as guidance to answer the main quesions are provided.

The numerical exercises have been tested and proved useful for a wide range of students, also students with little or no previous programming experiences. We have found that the use of high-level languages and methods such as Matlab allows us to focus on the important aspects of algorithms and programs, without detracting attention by spending much time on implementation. The program can also be used to generate results

We have decided to use Matlab as the basic language for these exercies. However, we have ensured that the exercises also can be run on non-commerical software, such as Octave, and on student versions of Matlab. The full code will only run on Matlab with the image processing toolbox, however, the alternative codes contain an explicit implementation of the cluster enumeration algorithm, and can be run on either Octave or in

Matlab. We have also provided a set of Fortran77 and Fortran90 programs in order to allow the reader to perform large scale simulations. These programs produce state-of-the-art results, and can be used to reproduce most of the research results presented in this book, in particular since computing power is increasing all the time, old results can now be reproduced with little effort using high-level techniques.

15.1 Percolation

15.1.1 Program `findpi.m`

This program demonstrates how we can find $\Pi(p, L)$ and $P(p, L)$ for two-dimensional site percolation. Notice in particular how we use the functions intersect and union to find the spanning clusters, and how we address the case when several percolation clusters appear at the same time.

```
% Program to generate P(p,L) and Pi(p,L)
nsample = 5;
p = (0.35:0.01:1.0);
nx = size(p,2);
lstart = 5;
lend = 7;
Pi = zeros(nx,lend);
P = zeros(nx,lend);
lvalue = zeros(lend);
clf reset;
for lcount = lstart:lend
    lx = 2^lcount;
    ly = lx;
    ll = lx*ly;
    for ns = 1:nsample
        z=rand(lx,ly);
        for i = 1:nx
            zz = z<p(i);
            [lw,num]=bwlabel(zz,4);
            perc_y = intersect(lw(:,1),lw(:,ly));
            perc_x = intersect(lw(1,:),lw(lx,:));
            perc_u = union(perc_x,perc_y);
            perc = find(perc_u>0);
            if (length(perc)>0)
                Pi(i,lcount) = Pi(i,lcount) + 1;
                s = regionprops(lw,'Area');
                area = cat(1,s.Area);
                ar = sum(area(perc_u(perc)));
                P(i,lcount) = P(i,lcount) + ar/ll;
            end
        end
    end
P(:,lcount) = P(:,lcount)/nsample;
Pi(:,lcount) = Pi(:,lcount)/nsample;
subplot(2,1,1);
plot(p,P(:,lcount));
```

```
xlabel('p'); ylabel('P(p,L)');
hold on
subplot(2,1,2);
plot(p,Pi(:,lcount));
xlabel('p'); ylabel('Pi(p,L)');
hold on
drawnow
end
hold off
```

15.1.2 Function `logbin.m`

This function produces a logarithmically binned data-set. This is used in order to determine the cluster number density $n(s,p)$.

```
function [x,dx,n] = logbin(y,a,binmax)
%
% Returns a logarithmically binned dataset
% y is a vector of the data-set
% a is the bin size, that is, bins are from a^k to a^k+1
% x gives the centers of the bins
% dx gives the width of the bins
% n gives the number of points in each bin
%
% First, general a list of edges, smallest value of y is 1
%ymax = max(y);
ymax = binmax;
yedge = 1.0;
istep = 1;
yedgelast = 0;
while (yedgelast<=ymax)
    edge(istep) = yedge;
    yyedge = floor(yedge*a);
    dy = yyedge - yedge;
    if (dy<=1.0)
        yyedge = yedge + 1.0;
    end
    yedgelast = yedge;
    yedge = yyedge;
    istep = istep + 1;
end
n = histc(y,edge);
dx = diff(edge);
nx = size(edge,2);
x = 0.5*(edge(1:nx-1) + edge(2:nx));
n = n(1:nx-1);
```

15.1.3 Program `findns.m`

This program demonstrates how we can determine $n(s,p)$ for various values of p.

```
lx=256;
ly=lx;
ll=lx*ly;
nsample = 1;
logbinsize = 2;
logbinmax = ll;

p = (0.2:0.05:0.6);
nx = size(p,2);
Pi = zeros(nx,1);
P = zeros(nx,1);

for i = 1:nx
for isample = 1:nsample
    z=rand(lx,ly);
    zz = z<p(i);
    [lw,num]=bwlabel(zz,4);
    perc_y = intersect(lw(:,1),lw(:,ly));
    perc_x = intersect(lw(1,:),lw(lx,:));
    perc_xy = union(perc_x,perc_y);
    perc = find(perc_xy>0);
    s = regionprops(lw,'Area');
    clusterareas = cat(1,s.Area);
    if (length(perc)>0)
        % Set Pi
        Pi(i) = Pi(i) + 1;
        % Find P(p,L)
        ar = sum(clusterareas(perc_xy(perc)));
        P(i) = P(i) + ar/ll;
    end
    % Find the cluster number density, get rid of percolating clusters
    ind = (1:num);
    indnoP = setxor(ind,perc_xy(perc));
    % Do statistics on area(indnoP)
    clusta = clusterareas(indnoP);
    [x,dx,n] = logbin(clusta,logbinsize,logbinmax);
    if (isample==1)
        nnsp = n/ll;
        nnsp = nnsp'./dx;
        nsp = nnsp;
    else
        nnsp = n/ll;
        nnsp = nnsp'./dx;
        nsp = nsp + nnsp;
    end
end
P(i) = P(i)/nsample;
Pi(i) = Pi(i)/nsample;
nsp = nsp/nsample;
ind2 = find(nsp>0);
plot(log10(x(ind2)),log10(nsp(ind2)),'-o');
xlabel('s');ylabel('n(s,p)');
hold on; drawnow;
end
hold off;
```

15.1.4 Program `excoarse.m`

The program `excoarse.m` demonstrates the use of the function `coarse.m` to generate a coarse grained site lattice using an explicitely specified renormalization scheme.

```
% excoarse.m
% Example of use of the coarsening procedure
z = rand(512,512)<0.58;
% Set up array for f
f(1) = 0;
f(2) = 0;
f(3) = 0;
f(4) = 1;
f(5) = 0;
f(6) = 0;
f(7) = 0;
f(8) = 1;
f(9) = 0;
f(10) = 0;
f(11) = 0;
f(12) = 1;
f(13) = 1;
f(14) = 1;
f(15) = 1;
f(16) = 1;

[lz,nz] = bwlabel(z,4);
imgz = label2rgb(lz);
zz = coarse(z,f);
[lzz,nzz] = bwlabel(zz,4);
imgzz = label2rgb(lzz);
zzz = coarse(zz,f);
[lzzz,nzzz] = bwlabel(zzz,4);
imgzzz = label2rgb(lzzz);
zzzz = coarse(zzz,f);
[lzzzz,nzzzz] = bwlabel(zzzz,4);
imgzzzz = label2rgb(lzzzz);

subplot(2,2,1), image(imgz);
axis equal
subplot(2,2,2), image(imgzz);
axis equal
subplot(2,2,3), image(imgzzz);
axis equal
subplot(2,2,4), image(imgzzzz);
axis equal
```

```
function zz = coarse(z,f)
% The original array is z
% The transfer function is f given as a vector with 16 possible places
% f applied to a two-by-two matrix should return
% the renormalized values
%
% The various values of f correspond to the following
% configurations of the two-by-two region that is renormalized,
% where I have used X to mark a present site, and 0 to mark an
```

```
% empty sites
%
% 1 00 5 00 9 00 13 00
% 00 X0 0X XX
%
% 2 X0 6 X0 10 X0 14 X0
% 00 X0 0X XX
%
% 3 0X 7 0X 11 0X 15 0X
% 00 X0 0X XX
%
% 4 XX 8 XX 12 XX 16 XX
% 00 X0 0X XX
%
nx = size(z,1);
ny = size(z,2);
if (mod(nx,2)==1)
  return
end
if (mod(ny,2)==1)
  return
end

nx2 = floor(nx/2);
ny2 = floor(ny/2);

zz = zeros(nx2,ny2);
x=zeros(2,2);

for iy = 1:2:ny
  for ix = 1:2:nx
    x = 1 + z(ix,iy)*1 + z(ix,iy+1)*2 + z(ix+1,iy)*4 + z(ix+1,iy+1)*8;
    xx = f(x);
    zz((ix+1)/2,(iy+1)/2) = xx;
  end
end
```

15.1.5 Program `exwalk.m`

The program `exwalk.m` demonstrates the use of the `walk.m` function which used a left-turning and a right-turning walker to find the perimeter of a particular cluster.

```
% exwalk.m
% Example of use of the walk routine

% Generate spanning cluster (l-r spanning)
lx =64;
ly = 64;
p = 0.585;
ncount = 0;
perc = [];
while (size(perc,1)==0)
  ncount = ncount + 1;
  if (ncount>1000)
```

```
    return
  end
  z=rand(lx,ly)<p;
  [lw,num]=bwlabel(z,4);
  perc_x = intersect(lw(1,:),lw(lx,:));
  perc = find(perc_x>0)
end
s = regionprops(lw,'Area');
clusterareas = cat(1,s.Area);
maxarea = max(clusterareas);
i = find(clusterareas==maxarea);
zz = lw == i;
% zz now contains the spanning cluster
imagesc(zz); % Display spanning cluster

% Run walk on this cluster
[l,r] = walk(zz);
zzz = l.*r; % Find points where both l and r are non-zero
zadd = zz + zzz;

subplot(2,2,1), imagesc(zz);
subplot(2,2,2), imagesc(zadd);
subplot(2,2,3), imagesc(zzz>0);
subplot(2,2,4), imagesc(l+r>0);
```

```
function [left,right] = walk(z);
%
% Left turning walker
%
% Returns left: nr of times walker passes a site
%
% First, ensure that array only has one contact point at left and
% right end: topmost points chosen
%
nx = size(z,1);
ny = size(z,2);
i = find(z(1,:)>0);
iy0 = i(1); % starting point for walker
ix0 = 1; % stopping point for walker
% First do left-turning walker
dirs = zeros(4,2);
dirs(1,1) = -1;
dirs(1,2) = 0;
dirs(2,1) = 0;
dirs(2,2) = -1;
dirs(3,1) = 1;
dirs(3,2) = 0;
dirs(4,1) = 0;
dirs(4,2) = 1;

nwalk = 1;
ix = ix0;
iy = iy0;
dir = 1; % 1=left, 2 = down, 3 = right, 4 = up;
left = zeros(nx,ny);

while (nwalk>0)
  left(ix,iy) = left(ix,iy) + 1;
```

```
  % Turn left until you find an occupied site
  nfound = 0;
  while (nfound==0)
    dir = dir - 1;
    if (dir<1)
      dir = dir + 4;
    end
    % Check this direction
    iix = ix + dirs(dir,1);
    iiy = iy + dirs(dir,2);
    if (iix==nx+1)
      nwalk = 0; % Walker escaped
      iix = nx;
      ix1 = ix;
      iy1 = iy;
    end
    % Is there a site here?
    if (iix>0)
      if (iiy>0)
        if (iiy<ny+1)
          if (z(iix,iiy)>0) % there is a site here, move here
            ix = iix;
            iy = iiy;
            nfound = 1;
            dir = dir + 2;
            if (dir>4)
              dir = dir - 4;
            end
          end
        end
      end
    end
  end
end

%left;

nwalk = 1;
ix = ix0;
iy = iy0;
dir = 1; % 1=left, 2 = down, 3 = right, 4 = up;
right = zeros(nx,ny);

while (nwalk>0)
  right(ix,iy) = right(ix,iy) + 1;
% ix,iy
  % Turn right until you find an occupied site
  nfound = 0;
  while (nfound==0)
    dir = dir + 1;
    if (dir>4)
      dir = dir - 4;
    end
    % Check this direction
    iix = ix + dirs(dir,1);
    iiy = iy + dirs(dir,2);
    if (iix==nx+1)
      if (iy==iy1)
        nwalk = 0; % Walker escaped
        iix = nx;
```

```
      end
    end
    % Is there a site here?
    if (iix>0)
      if (iiy>0)
        if (iiy<ny+1)
          if (iix<nx+1)
            if (z(iix,iiy)>0) % there is a site here, move here
              ix = iix;
              iy = iiy;
              nfound = 1;
              dir = dir - 2;
              if (dir<1)
                dir = dir + 4;
              end
            end
          end
        end
      end
    end
  end
end
```

15.2 Disorder

15.2.1 Program `exflow.m`

The program `exflow.m` demonstrates the use of the programs `coltomat.m`, `sitetobond.m`, `find_cond.m`, and `mk_eqsystem.m` to generate the equations for solving Darcy flow in a porous material using Kirchoff's equations. The code in `find_cond.m` and `mk_eqsystem.m` was provided by Martin Søreng.

```
%
% exflow.m
%
clear all; clf;
% First, find the backbone
% Generate spanning cluster (l-r spanning)
lx = 10;
ly = 10;
p = 0.5927;
ncount = 0;
perc = [];
while (size(perc,1)==0)
  ncount = ncount + 1;
  if (ncount>1000)
    return
  end
  z=rand(lx,ly)<p;
  [lw,num]=bwlabel(z,4);
  perc_x = intersect(lw(1,:),lw(lx,:));
  perc = find(perc_x>0)
```

```
end
s = regionprops(lw,'Area');
clusterareas = cat(1,s.Area);
maxarea = max(clusterareas);
i = find(clusterareas==maxarea);
zz = lw == i;
% zz now contains the spanning cluster
% Transpose
zzz = zz';
% Generate bond lattice from this
g = sitetobond(zzz);
% Generate conductivity matrix
[p c_eff] = find_cond(g,lx,ly);
% Transform this onto a nx x ny lattice
x = coltomat(full(p),lx,ly);
P = x.*zzz;
g1 = g(:,1);
g2 = g(:,2);
z1 = coltomat(g1,lx,ly);
z2 = coltomat(g2,lx,ly);
% Plotting
subplot(2,2,1), imagesc(zzz);
title('Spanning cluster')
axis equal
subplot(2,2,2), imagesc(P);
title('Pressure');
axis equal
f2 = zeros(lx,ly);
for iy = 1:ly-1
  f2(:,iy) = (P(:,iy) - P(:,iy+1)).*z2(:,iy);
end
f1 = zeros(lx,ly);
for ix = 1:lx-1
  f1(ix,:) = (P(ix,:) - P(ix+1,:)).*z1(ix,:);
end
% Find the sum of absolute fluxes into each site
fn = zeros(lx,ly);
fn = fn + abs(f1);
fn = fn + abs(f2);
fn(:,2:ly) = fn(:,2:ly) + abs(f2(:,1:ly-1));
fn(:,1) = fn(:,1) + abs((P(:,1) - 1.0).*(zzz(:,1)));
fn(2:lx,:) = fn(2:lx,:) + abs(f1(1:lx-1,:));
subplot(2,2,3), imagesc(fn);
title('Flux');
axis equal
zfn = fn>limit;
zbb = (zzz + 2*zfn);
zbb = zbb/max(max(zbb));
subplot(2,2,4), imagesc(zbb);
title('BB and DE');
axis equal
```

```
function [P, Ceff] = find_cond(A, X, Y)
%
% Written by Marin Soreng
% (C) 2004
%
%Calculates the effective flow conductance Ceff of the
```

```
%lattice A as well as the pressure P in every site.

P_in = 1;
P_out = 0;

[B C] = mk_eqsystem(A, X, Y);

%Kirchhoff's equations solve for P
P = B\C;

%The pressure at the external sites is added
%(Boundary conditions)
P = [P_in*ones(X, 1); P; P_out*ones(X, 1)];
%Calculate Ceff
Ceff = (P(end-2*X+1:end-X)-P_out)'*A(end-2*X+1:end-X,2)/(P_in-P_out);
```

```
function [B, C] = mk_eqsystem(A, X, Y)
%
% Written by Marin Soreng
% (C) 2004
%
% Sets up Kirchoff's equations for the 2D lattice A.
% A has X*Y rows and 2 columns. The rows indicate the site,
% the first column the bond perpendicular to the flow direction
% and the second column the bond parallel to the flow direction.
%
% The return values are [B, t] where B*x = C. This is solved
% for the site pressure by x = B\C.

% Total no of internal lattice sites
sites = X*(Y-2);

%Allocate space for the nonzero upper diagonals
main_diag = zeros(sites, 1);
upper_diag1 = zeros(sites-1, 1);
upper_diag2 = zeros(sites-X, 1);

%Calculates the nonzero upper diagonals
main_diag = A(X+1:X*(Y-1), 1) + A(X+1:X*(Y-1), 2) + A(1:X*(Y-2), 2) ...
    + A(X:X*(Y-1)-1, 1);
upper_diag1 = A(X+1:X*(Y-1)-1, 1);
upper_diag2 = A(X+1:X*(Y-2), 2);
main_diag(find(main_diag==0)) = 1;

%Constructing B which is symmetric, lower=upper diagonals.
B = sparse(sites, sites); % B*u = t
B = - spdiags(upper_diag1,-1, sites, sites);
B = B + - spdiags(upper_diag2,-X, sites, sites);
B = B + B' + spdiags(main_diag, 0, sites, sites);

%Constructing C
C = sparse(sites, 1);
C(1:X) = A(1:X, 2);
C(end-X+1:end) = 0*A((end-2*X+1:end-X), 2);
```

```
function g = sitetobond(z)
%
% Function to convert the site network z(L,L) into a (L*L,2) bond
% network
% g(i,1) gives bond perpendicular to direction of flow
% g(i,2) gives bond parallel to direction of flow
% z(nx,ny) -> g(nx*ny,2)
%
nx = size(z,1);
ny = size(z,2);
N = nx*ny;
%g = zeros(N,2);

gg_r = zeros(nx,ny); % First, find these
gg_d = zeros(nx,ny); % First, find these
gg_r(:,1:ny-1) = z(:,1:ny-1).*z(:,2:ny);
gg_r(:,ny) = z(:,ny);
gg_d(1:nx-1,:) = z(1:nx-1,:).*z(2:nx,:);
gg_d(nx,:) = 0;

% Then, concatenate gg onto g
ii = 1:nx*ny;
g = zeros(nx*ny,2);
g(:,1) = gg_d(ii)';
g(:,2) = gg_r(ii)';
```

```
function g = coltomat(z,x,y)
% Convert z(x*y) into a matrix of z(x,y)
% Transform this onto a nx x ny lattice
g = zeros(x,y);
for iy = 1:y
  i = (iy-1)*x + 1;
  ii = i + x - 1;
  g(:,iy) = z(i:ii);
end
```

15.2.2 Program `testpercwalk.m`

The program `testpercwalk.m` demonstrates the use of the C-program `percwalk.c` which can be compiled using `mex` in Matlab.

```
%
% testpercwalk.m
%
% Generate spanning cluster (l-r spanning)
lx = 100;
ly = 100;
p = 0.59274;
nstep = 1e5;

nnstep = nstep + 1;
ncount = 0;
perc = [];
```

```
while (size(perc,1)==0)
  ncount = ncount + 1;
  if (ncount>1000)
    return
  end
  z=rand(lx,ly)<p;
  [lw,num]=bwlabel(z,4);
  perc_x = intersect(lw(1,:),lw(lx,:));
  perc = find(perc_x>0)
end
s = regionprops(lw,'Area');
clusterareas = cat(1,s.Area);
maxarea = max(clusterareas);
i = find(clusterareas==maxarea);
zz = lw == i;
% zz now contains the spanning cluster

imagesc(zz),axis equal,axis tight

rz = 1.0*zz;
n = 1;
while (n<=1)
  r = rand(nnstep,1);
  [w,n] = percwalk(rz,r,0);
end
x = w(1,:);
y = w(2,:);
hold on,plot(y,x);
hold off
```

```
/*==================================================================
 *
 * PERCWALK.C Sample .MEX file corresponding to PERCWALK.M
 * Return random walk on percolation cluster
 *
 * The calling syntax is:
 *
 *  [yp,nstep] = percwalk(y,nw,nsaw)
 *
 * Saw mode is currently not implemented
 *
 * You may also want to look at the corresponding M-code, percwalk.m.
 *
 * This is a MEX-file for MATLAB.
 * Copyright 2004 Anders Malthe-Sorenssen
 * Physics of Geological Processes, University of Oslo, Norway
 *
 *================================================================*/
#include <math.h>
#include "mex.h"

/* Input Arguments */

#define Y_IN  prhs[0]
#define NW_IN  prhs[1]
#define NSAW_IN  prhs[2]
```

```
/* Output Arguments */

#define YP_OUT  plhs[0]
#define NSTEP_OUT plhs[1]

#if !defined(MAX)
#define MAX(A, B) ((A) > (B) ? (A) : (B))
#endif

#if !defined(MIN)
#define MIN(A, B) ((A) < (B) ? (A) : (B))
#endif

#define PI 3.14159265

static void percwalk(
                 double yp[],
                 double y[],
                 double nw[],
                 double *nstep,
                 double *nsaw,
                 unsigned int m,
                 unsigned int n,
                 unsigned int nwalk
                 )
{
    double r1,r2,nn,c,pos,sstep;
    int nchange,niter,neighb,idir,nnr;
    int ntot,i,ii,iix,iiy,ix,iy,nnsaw,npos,step,nstop,nwalk1,nc;
    int dir[8];
    int neighblist[8];
    double yy;

    dir[0] = 1;
    dir[1] = 0;
    dir[2] = -1;
    dir[3] = 0;
    dir[4] = 0;
    dir[5] = 1;
    dir[6] = 0;
    dir[7] = -1;

    ntot = m*n;
    nn = *nsaw;
    nnsaw = (int)nn;

       printf("m,n = %i,%i\n",m,n);
          printf("nsaw = %i\n",nnsaw);

    /* First, find random starting position */
    c = nw[0]; /* Get first random number from string, use this
                  for initial position */
    pos = ntot*c;
    npos = floor(pos);
    iy = npos/m;
    ix = npos - m*iy;

    printf("pos,npos = %lf,%i\n",pos,npos);
      printf("ix,iy = %i,%i\n",ix,iy);
```

```
    yp[0] = ix+1;
    yp[1] = iy+1;
    nstop = 0;
    step = 0;
    nwalk1 = nwalk - 1;
    i = iy*m + ix;
    c = y[i];
        printf("z(ix,iy) = %f\n",c);
    if (c>0.1) {
      nstop = 0;
         printf("c0 = %lf\n",c);
    }
    else {
        printf("c1 = %lf\n",c);
      nstop = 1;
    }
    printf("nstop , step, nwalk1 = %i,%i,%i\n",nstop,step,nwalk1);
    while ((nstop==0)&&(step<nwalk1)) {
      /* Check if there is a way out */
      neighb = 0;
      for (idir=0;idir<4;idir++) {
        iix = ix + dir[idir*2];
        iiy = iy + dir[idir*2+1];
        /* Check if inside */
        if ((iix>=0)&&(iix<m)&&(iiy>=0)&&(iiy<n)) {
          ii = iiy*m + iix;
          if (y[ii]>0.1) {
            neighblist[neighb] = idir;
            neighb++;
          }
        }
      }
      /* printf("neighb = %i\n",neighb);*/
      if (neighb>0) { /* Possible way out of this site */
        step++;
        c = nw[step];
        c = c*neighb;
        nc = floor(c);
        if (nc>3) nc = 3;
        if (nc<0) nc = 0;
        /* printf("nc = %i\n",nc);*/
        idir = neighblist[nc];
        ix = ix + dir[idir*2];
        iy = iy + dir[idir*2+1];
        /* printf("ix,iy = %i,%i\n",ix,iy);
               printf("step = %i\n",step);*/
        yp[2*step] = ix+1;
        yp[2*step+1] = iy+1;
      }
      else { /* No way out - stop walker */
        nstop = 1;
      }
    }

    sstep = step + 1.0;

    *nstep = sstep;
    printf("Step = %i\n",step);
    return;
}
```

```
void mexFunction( int nlhs, mxArray *plhs[],
                  int nrhs, const mxArray*prhs[] )

{
    double *yp;
    double *nsaw,*y,*nw,*nstep;
    unsigned int m,n,nwalk,nwalk2;

    /* Check for proper number of arguments */

    if (nrhs != 3) {
        mexErrMsgTxt("Three input arguments required.");
    } else if (nlhs > 2) {
        mexErrMsgTxt("Too many output arguments.");
    }

    /* Find dimensions of Y. */

    m = mxGetM(Y_IN);
    n = mxGetN(Y_IN);

    /* Find dimensions of NW. */
    nwalk = mxGetM(NW_IN);

    /* Create a matrix for the return argument */
    nwalk2 = 2;
    YP_OUT = mxCreateDoubleMatrix(nwalk2, nwalk, mxREAL);
    NSTEP_OUT = mxCreateDoubleMatrix(1, 1, mxREAL);

    /* Assign pointers to the various parameters */
    yp = mxGetPr(YP_OUT);
    nstep = mxGetPr(NSTEP_OUT);

    y = mxGetPr(Y_IN);
    nw = mxGetPr(NW_IN);
    nsaw = mxGetPr(NSAW_IN);

    /* Do the actual computations in a subroutine */
    percwalk(yp,y,nw,nstep,nsaw,m,n,nwalk);
    return;

}
```

15.2.3 Program `invperc.m`

The program `inveperc.m` demonstrates the use of Matlab to study invasion percolation. The program can easily be extended to study gravity stabilized invasion percolation by adding a gradient gi to the pressure threshold $z(i, j)$ at each site.

```
%
% invperc.m
%
% Example program for studying invasion percolation problems
```

```
% NOTE: This is not an optimal but an educational algorithm
%

L = 100; % system size

p = (0.0:0.01:0.7);

perc = 0; % flag to signal if other end is reached

nbetween = 1;
nstep = 0;
nend = numel(p);
nstop = 0;

z = rand(L,L); % Random distribution of thresholds
pcluster = zeros(L,L);

while ((nstop==0)&&(nstep<nend))
  nstep = nstep + 1;
  p0 = p(nstep);
  zz = z<p0;
  [lw,num] = bwlabel(zz,4);
  leftside = lw(:,1);
  i = find(leftside>0);
  leftnonzero = leftside(i);
  uniqueleftside = unique(leftnonzero);
  cluster = ismember(lw,uniqueleftside);
  pcluster = pcluster + cluster;
  if (mod(nstep,nbetween)==0)
    imagesc(pcluster),axis equal, axis tight, colorbar,drawnow
  end
  % Check if it has reached the right hand side
  rightside = lw(:,L);
  ir = find(rightside>0);
  rightnonzero = rightside(ir);
  span = intersect(leftnonzero,rightnonzero);
  if (numel(span)>0)
    nstop = 1; % spanning
  end
end

p0
imagesc(pcluster),axis equal, axis tight, colorbar,drawnow
```

Exercises

16

16.1 Percolation

Exercise 16.1: Percolation in small systems

a) Find $P(p, L)$ for $L = 1$ and $L = 2$.

b) Categorize all possible configurations for $L = 3$.

c) Find $\Pi(p, L)$ and $P(p, L)$ for $L = 3$.

Exercise 16.2: Percolation in small systems

a) Write a program to find all the configurations for $L = 2$.

b) Use this program to find $\Pi(p, L = 2)$ and $P(p, L = 2)$. Compare with the exact results from the previous exercise.

c) Use you program to find $\Pi(p, L)$ and $P(p, L)$ for $L = 3, 4$ and 5.

Exercise 16.3: Next-nearest neighbor connectivity in 1d

Exercise 16.4: Finite-size effects in 1d percolation

Exercise 16.5: Generating percolation clusters

In this exercise we will use Matlab to generate and visualize percolation clusters. We generate a $L \times L$ matrix of random numbers, and will examine clusters for a occupation probability p.

We generate the percolation matrix consisting of occupied (1) and unoccupied (0) sites, using

```
L = 100;
r = rand(L,L);
p = 0.6;
z = r<p; % This generates the binary array
[lw,num] = bwlabel(z,4);
```

We have then produced the array `lw` that contains labels for each of the connected clusters, and the variable `num` that contains the number of clusters.

a) Familiarize yourself with labeling by looking at `lw`, and by studying the second example in the Matlab help system on the image analysis toolbox.

We can examine the array directly by mapping the labels onto a color-map, using `label2rgb`.

```
img = label2rgb(lw);
image(img);
```

We can extract information about the labeled image using `regionprops`, for example, we can extract an array of the areas of the clusters using

```
s = regionprops(lw,'Area');
area = cat(1,s.Area);
```

You can also extract information about the `BoundingBox` and other properties of clusters using similar commands

```
s = regionprops(lw,'BoundingBox');
bbox = cat(1,s.BoundingBox);
```

b) Using these features, you should make a program to calculate $P(p, L)$ for various p.

Hint: you can use either `BoundingBox` or `intersect` and `union` to find the spanning cluster.

c) How robust is your algorithm to changes in boundary conditions? Could you do a rectangular grid where $L_x \gg L_y$? Could you do a more complicated set of boundaries? Can you think of a simple method to ensure that you can calculate P for any boundary geometry?

Exercise 16.6: Finding $\Pi(p, L)$ and $P(p, L)$

a) Write a program to find $P(p, L)$ and $\Pi(p, L)$ for $L = 2, 4, 8, 16, 32, 64, 128$. Comment on the number of samples you need to make to det a good estimate for P and Π.

b) Test the program for small L by comparing with the exact results from above. Comment on the results?

Exercise 16.7: Determining β

We know that when $p > p_c$, the probability $P(p, L)$ for a given site to belong to the percolation cluster, has the form

$$P(p, L) \sim (p - p_c)^{\beta} \, . \tag{16.1}$$

Use the data from above to find an expression for β. For this you may need that $p_c = 0.59275$.

Exercise 16.8: Determining the exponent of power-law distributions

In this exercise you will build tools to analyse power-law type probability densities.

Generate the following set of data-points in Matlab:

```
z = rand(1e6,1).^(-3+1);
```

Your task is to determine the distribution function $f_Z(z)$ for this distribution. Hint: the distribution is on the form $f(u) \propto u^{\alpha}$.

a) Find the cumulative distribution, that is, $P(Z > z)$. You can then find the actual distribution from

$$f_Z(z) = \frac{dP(Z > z)}{dz} \, . \tag{16.2}$$

b) Generate a method to do logarithmic binning in Matlab. That is, you estimate the density by doing a histogram with bin-sizes that increase exponentially in size. Hint: Remember to divide by the correct bin-size.

Exercise 16.9: Cluster number density $n(s, p)$

We will generate the cluster number density $n(s, p)$ from the two-dimensional data-set.

a) Estimate $n(s,p)$ for a sequence of p values approaching $p_c = 0.59275$ from above and below.

Hint 1: The cluster sizes are extracted using `.Area` as described in a previous exercise.

Hint 2: Remember to remove the percolating cluster.

Hint 3: Use logarithmic binning.

b) Estimate $n(s,p_c;L)$ for $L = 2^k$ for $k = 4,\ldots,9$. Use this plot to estimate τ.

c) Can you estimate the scaling of $s_\xi \sim |p-p_c|^{-1/\sigma}$ using this data-set?

Hint 1: Use $n(s,p)/n(s,p_c) = F(s/s_\xi) = 0.5$ as the definition of s_ξ.

Exercise 16.10: Average cluster size

a) Find the average (finite) cluster size $S(p)$ for p close to p_c, for p above and below p_c.

b) Determine the scaling exponent $S(p) \sim |p-p_c|^{-\gamma}$.

c) In what ways can you generate $S^{(k)}(p)$? What do you think is the best way?

Exercise 16.11: Mass scaling of percolating cluster

a) Find the mass $M(L)$ of the percolating cluster at $p = p_c$ as a function of L, for $L = 2^k$, $k = 4,\ldots,11$.

b) Plot $\log(M)$ as a function of $\log(L)$.

c) Determine the exponent D.

Exercise 16.12: Correlation function

a) Write a program to find the correlation function, $g(r,p,L)$ for $L = 256$.

b) Plot $g(r,p,L)$ for $p = 0.55$ to $p = 0.65$ for $L = 256$.

c) Find the correlation length $\xi(p,L)$ for $L = 256$ for the p-values used above.

d) Plot ξ as a fuction of $p-p_c$, and determine ν.

Exercise 16.13: Finite Size Scaling

a) Write a program to find the radius of gyration, $R_g(s,p,L)$ for $L = 256$ and for p close to p_c.

b) Measure the dimension D.

c) Plot $R_s s^{-\nu D}$ as a function of s for $p > p_c$ and for $p < p_c$. Comment on the results.

Exercise 16.14: Finite Size Scaling

In this exercise we will use a finite size scaling ansatz to provide estimates of ν, p_c, and the average percolation probability $\langle p \rangle$ in a system of size L.

We define $p_{\Pi=x}$ so that

$$\Pi(p_{\Pi=x}) = x \ ,$$

notice that $p_{\Pi=x}$ is a function of system size L used for the simulation.

a) Find $p_{\Pi=x}$ for $x = 0.8$ and $x = 0.3$ for $L = 25, 50, 100, 200, 400, 800$. Plot $p_{\Pi=x}$ as a function of L.

According to the scaling theory we have

$$p_{x_1} - p_{x_2} = (C_{x_1} - C_{x_2}) L^{-1/\nu} \ .$$

b) Plot $\log(p_{\Pi=0.8} - p_{\Pi=0.3})$ as a function of $\log(L)$ to estimate the exponent ν. How does it compare to the exact results.

In the following, please use the exact value of ν.

The scaling theory also predicted that

$$p_{\Pi=x} = p_c + C_x L^{-1/\nu} \ .$$

c) Plot $p_{\Pi=x}$ as a function of $L^{-1/\nu}$ to estimate p_c.

d) Generate a data-collapse plot for $\Pi(p, L)$ to find the function $\Phi(u)$ from the lecture notes.

The following parts of the exercise are optional.

e) Plot $\Pi'(p, L)$ as a function of p for the various L values.

f) Generate a data-collapse plot of $\Pi'(p, L)$.

g) Find $\langle p \rangle$ and plot $\langle p \rangle$ as a function of $L^{-1/\nu}$ to find p_c.

Exercise 16.15: Renormalization of nnn-model

a) Develop a renormalization scheme for a two-dimensional site percolation system with next-nearest neighbor connectivity. That is, list the 16 possible configurations, and determine what configration they map onto in the renormalized lattice.

b) Find the renormalized occupation probability $p' = R(p)$.

c) Plot $R(p)$ and $f(p) = p$.

d) Find the fixpoints p^* so that $R(p^*) = p^*$.

e) Find the rescaling factor $\Lambda = R'(p^*)$.

f) Determine the exponent $\nu = \ln \Lambda / \ln b$.

g) How can we improve the estimates of p_c and ν?

Exercise 16.16: Renormalization of three-dimensional site percolation model

a) Find all 2^8 possible configurations for the $2 \times 2 \times 2$ renormalization cell for three-dimensional site percolation.

b) Determine a renormalization scheme - what configurations map onto an occupied site?

c) Find the renormalized occupation probability $p' = R(p)$.

d) Plot $R(p)$ and $f(p) = p$.

e) Find the fixpoints p^* so that $R(p^*) = p^*$.

f) Find the rescaling factor $\Lambda = R'(p^*)$.

g) Determine the exponent $\nu = \ln \Lambda / \ln b$.

Exercise 16.17: Renormalization of three-dimensional bond percolation model

In this exercise we will develop an H-cell renormalization scheme for bond percolation in three dimensions. The three-dimensional H-cell is illustrated in fig. 16.1.

a) Find all 2^12 possible configurations for this H-cell.

b) Determine a renormalization scheme - what configurations map onto an occupied site?

c) Find the renormalized occupation probability $p' = R(p)$.

d) Plot $R(p)$ and $f(p) = p$.

e) Find the fixpoints p^* so that $R(p^*) = p^*$.

f) Find the rescaling factor $\Lambda = R'(p^*)$.

g) Determine the exponent $\nu = \ln \Lambda / \ln b$.

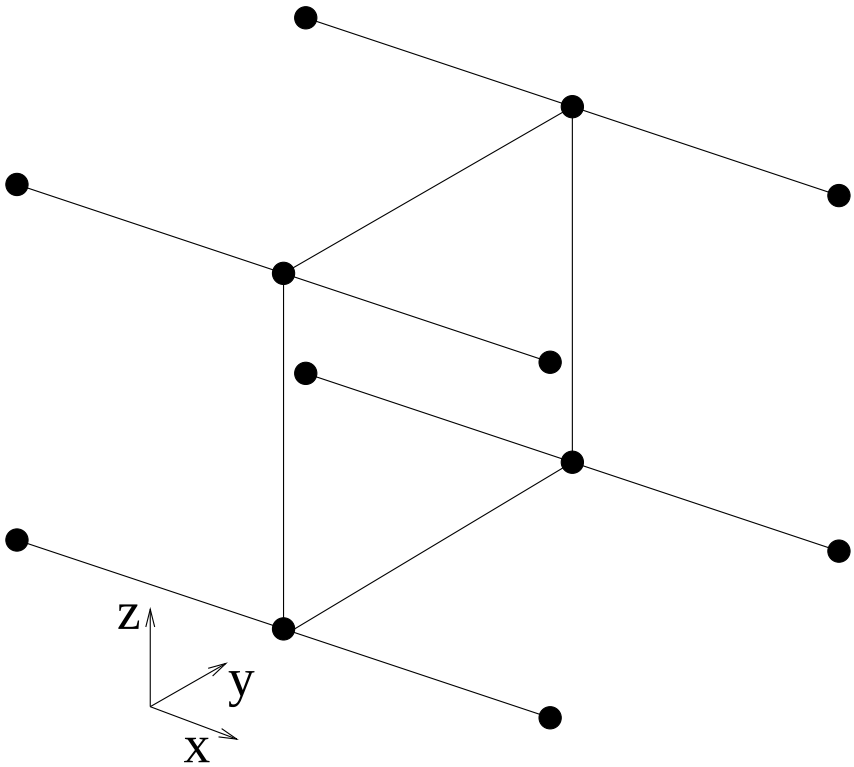

Fig. 16.1 Illustrations of the 3d H-cell.

Exercise 16.18: Numerical study of renormalization

Use the example programs `excoarse.m` and `coarse.m` to study the renormalization of a given sample of a percolation system.

Perform successive iterations for $p = 0.3$, $p = 0.4$, $p = 0.5$, $p = p_c$, $p = 0.65$, $p = 0.70$, and $p0.75$, in order to understand the instability of the fixpoint at $p = p_c$.

Exercise 16.19: Singly connected bonds

Use the example program `exwalk.m` and `walk.m` to find the singly connected bonds.

a) Run the program `exwalk.m` to visualize the singly connected bonds. Can you understand how this algorithms finds the singly connected bonds? Why are some of the bonds of a different color?

b) Find the mass, M_{SC}, of the singly connected bonds as a function of system size L for $p = p_c$ and use this to estimate the exponent D_{SC}: $M_{SC} \propto L^{D_{SC}}$.

c) Can you find the behavior of $P_{SC} = M_{SC}/L^d$ as a function of $p - p_c$?

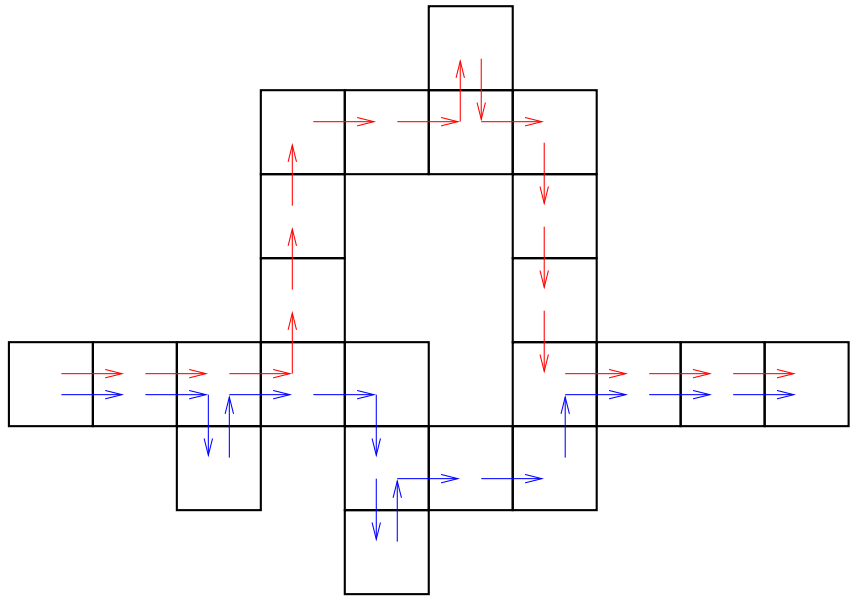

Fig. 16.2 Illustrations of the left-right turning walker.

Exercise 16.20: Left/right-turning walker

We have provided a subroutine and an example program that implements the left/right-turning walker algorithm. The algorithm works on a given clusters. From one end of the cluster, two walkers are started. The walkers can only walk according to the connectivity rules on the lattice. That is, for a nearest-neighbor lattice, they can only walk to their nearest neighbors. The left-turning walker always tries to turn left from its previous direction. If this site is empty, it tries the next-best site, which is to continue straight ahead. If that is empty, it tries to move right, and if that is empty, it moves back along the direction it came. The right-turning walker follows a similar rule, but prefers to turn right in each step. The first walker to reach the other end of the cluster stops, and the other walker stops when it reaches this site.

The path of the two walkers is illustrated in the figure below. The sites that are visited by both walkers consitute the singly connected bonds. The union of the two walks consitutes what is called the external perimeter (Hull) of the cluster.

a) Use the programs `exwalk.m` and `walk.m` to generate and illustrate of the singly connected bonds for a 100×100 system. Check that the illustrated bonds correspond to the singly connected bonds.

b) Measure the dimension D_{SC}.

c) Modify the program `exwalk.m` to find the external perimeter (Hull) of a spanning cluster in a 100×100 system.

d) Measure the dimension D_P of the perimeter.

e) (Advanced) Develop a theory for the behavior of $P_H(p, L)$, the probability for a site to belong to the Hull as a function of p and L for $p > p_c$.

f) (Advanced) Measure the behavior of $P_H(p, L)$ as a function of p for $L = 512 \times 512$.

16.2 Disordered systems

Exercise 16.21: Flow on fractals

Use the example program `exflow.m` to study fluid flow in a percolation system.

This program takes as input an array of (site) conductivities for each individual site, and calculates the local current in each bond connecting two sites in the lattice. Most of the program is used to set up the solution of the linear problem for the local currents, given as the solution to Kirchoffs equations. The programs for vectorizing the setup of the matrices were provided by Martin Søreng.

a) Run the example program `exflow.m` to visualize the currents on the spanning cluster.

b) Modify the program to find the backbone and the dangling ends of the spanning cluster.

c) Use the program to find the singly connected bonds in the spanning cluster.

Exercise 16.22: Conductivity

a) Find the conductivity as a function of $p - p_c$. Determine the exponent $\tilde{\zeta}_R$ by direct measurement.

b) Find the conductivity at $p = p_c$ as a function of system size L.

Exercise 16.23: Current distribution

Use the example program `exflow.m` to find the currents I_b in each bonds b on a spanning cluster at $p = p_c$, $p = 0.585$, and $p = 0.60$.

a) Find the total current I going through the system.

In the following we will study the normalized currents, $i_b = I_b/I$.

b) Find the distribution $P(i)$ of the normalized currents.

c) Measure moments of the distribution.

Exercise 16.24: Bivariate porous media

Rewrite the program `exflow.m` to study a bivariate distribution of conductivities. That is, for each site, the conductivity is 1 with probability p and $\sigma_0 < 1$ with probability $1 - p$.

a) Visualize the distribution of currents for $\sigma_0 = 0.1$.

b) Find the conductivity $\sigma(p)$ for $\sigma_0 = 0.1$, 0.01, and 0.001.

c) Plot $\sigma(p_c)$ as a function of σ_0.

d) (Advanced) Can you find a way to rescale the conductivities to produce a data-collapse?

Exercise 16.25: Random walks on the spanning cluster

In this exercise we will use and modify the program `testpercwalk.m` to study random walks in percolation systems, and on the spanning cluster in particular. We want to find the dimension d_w of a two-dimensional random walk on the spanning cluster.

a) Find the distance $\langle R^2 \rangle$ as a function of the number of steps N for random walks on the spanning cluster for $p = p_c$.

b) Find the dimension, d_w of the walk, from the relation $\langle R^2 \rangle \propto N^{2/d_w}$.

c) Find the distribution $P(R, N)$ for the position R as a function of the number of steps N for a random walker on the percolation cluster.

d) (Advanced) Can you produce a data-collapse for the distribution $P(R, N)$.

e) (Advanced) Can you determine the functional form of the distributoin $P(R, N)$. Is it a Gaussian?

Exercise 16.26: Random walks percolation clusters

In this exercise we will use and modify the program `testpercwalk.m` to study random walks in percolation systems, not restricted to the spanning cluster. We want to find the dimension d_w of a two-dimensional random walk on the spanning cluster.

a) Find the distance $\langle R^2 \rangle$ as a function of the number of steps N for random walks on the spanning cluster for $p < p_c$ and for $p > p_c$.

b) Plot $\log\langle R^2 \rangle$ as a function of N for various values of p.

c) Can you find the behavior of the correlation length ξ from this plot?

d) Discuss the behavior of the characteristic cross-over time t_0 based on the plot.

Exercise 16.27: Self-avoiding walks on fractals

(Advanced) In this exercise we will use the program `testpercwalk.m` to study a self-avoiding random walker on the spanning cluster. In this exercise you will need to collect extensive statistics to be able to determine the scaling behavior.

a) Find the distance $\langle R^2 \rangle$ as a function of the number of steps N for random walks on the spanning cluster for $p = p_c$.

b) Find the dimension, d_w of the walk, from the relation $\langle R^2 \rangle \propto N^{2/d_w}$.

Exercise 16.28: Gravity stabilized invasion percolation

In this exercise we will use and modify the program `invperc.m` to study the gravity stabilized invasion percolation process.

a) Modify the program `invperc.m` to include the effect of a gradient g in the local pressure thresholds. Characterize quantitatively the change in behavior as g is varied.

Hint: use the command `m=meshgrid(g*(1:L),(1:L))` to generate the gradient field and add this to the pressure thresholds `z`.

b) Find the average position, $h(P)$ of the front as a function of the invasion pressure P.

Hint: you can find the position `j` of the front at row `j` by the command `i=max(find(pcluster(j,:) =0);` .

c) Find the width $w(P)$ of the front as a function of P.

d) Generate plots of h and w for various values of g, and generate a data-collapse by a proper rescaling of the axes.

e) (Advanced) Use the `walk.m` routine to find the full shape of the front, and determine the value in h and w when you include the effect of overhangs and the full front shape. Compare these values with the values found above.

f) (Advanced) Determine the fractal dimension D_F of the front using the function `walk.m`.

Project 16.29: Finite size scaling of $n(s, p)$

a) Develop a finite size scaling ansatz/theory for $n(s, p_c, L)$. You should provide arguments for the behavior in the various limits.

b) Plot $n(s, p_c, L)$ as a function of s for $L = 100, 200, 400, 800$.

c) Demonstrate the validity of the scaling theory by producing a data-collapse plot for $n(s, p_c, L)$.

Project 16.30: The Density of the Backbone

The backbone of a spanning cluster is the union of all self-avoiding walks from one side of the cluster to the opposite. The backbone corresponds to the sites the contribute to the flow conductivity of the spanning cluster. The remaining sites are the dangling ends.

We call the mass of the backbone M_B, and the density of the backbone $P_B = M_B/L^d$, where L is the system size, and d the dimensionality of the percolation system. Here, we will study two-dimensional site percolation.

a) Argue that the functional form of $P_B(p)$ when $p \to p_c^+$ is

$$P_B(p) = P_0(p - p_c)^x \ ,$$

and find an expression for the exponent x. You can assume that the fractal dimension of the backbone, D_B, is known.

b) Assume that the functional form of $P_B(p)$ when $p \to p_c^+$ and $\xi \ll L$ is

$$P_B(p) = P_0(p - p_c)^x \ ,$$

Determine the exponent x by numerical experiment. If needed, you may use that $\nu = 4/3$.

16.3 Grand project

This project spans the entire book, and consists of several subprojects. It was used as a compulsory project to guide the students during the corse.

In this project, we will develop the tools and knowhow necessary to study scaling in numerical, experimental and real-world data. You will gain experience with image analysis, discrete models for phase transitions, finite size scaling models, the geometry of percolation clusters including subset geometry, and dynamic processes on fractals, with particular emphasis on the dynamics of a random walker on a self-similar fractal: the percolation cluster.

Project 16.31: Generating percolation clusters

First, we use Matlab to generate and visualize percolation cluster. We generate an $L \times L$ matrix of uniformly distributed random numbers, and introduce the tools necessary to visualize and analyze the clusters.

We generate the percolation matrix consisting of occupied (1) and unoccupied (0) sites, using

```
L = 100;
r = rand(L,L);
p = 0.6;
z = r<p; % This generates the binary array
[lw,num] = bwlabel(z,4);
```

We have then produced the array `lw` that contains labels for each of the connected clusters, and the variable `num` that contains the number of clusters.

We can examine the array directly by mapping the labels onto a color-map, using `label2rgb`.

```
img = label2rgb(lw,'jet','k','shuffle');
image(img);
```

We can extract information about the labeled image using `regionprops`, for example, we can extract an array of the areas of the clusters using

```
s = regionprops(lw,'Area');
area = cat(1,s.Area);
```

You can also extract information about the `BoundingBox` and other properties of clusters using similar commands

```
s = regionprops(lw,'BoundingBox');
bbox = cat(1,s.BoundingBox);
```

a) Using these features, make a program to calculate $P(p, L)$ for various p.

Hint. You can use either `BoundingBox` or `intersect` and `union` to find the spanning cluster.

b) How robust is your algorithm to changes in boundary conditions? Could you do a rectangular grid where $L_x \gg L_y$? Could you do a more complicated set of boundaries? Can you think of a simple method to ensure that you can calculate P for any boundary geometry?

Project 16.32: The behavior of $\Pi(p, L)$ and $P(p, L)$

a) Write a program to find $P(p, L)$ and $\Pi(p, L)$ for $L = 2, 4, 8, 16, 32, 64, 128$. Comment on the number of samples you need to make to get a good estimate for P and Π.

b) We know that when $p > p_c$, the probability $P(p, L)$ for a given site to belong to the percolation cluster, has the form

$$P(p, L) \sim (p - p_c)^{\beta} .$$

Use your program to find an expression for β. For this you may need that $p_c = 0.59275$.

Project 16.33: Determining the exponent of power-law distributions

First, we need to develop tools to analyse power-law type probability densities.

Generate the following set of data-points in Matlab:

```
z = rand(1e6,1).^(-3+1);
```

Your task is to determine the distribution function $f_Z(z)$ for this distribution. Hint: the distribution is on the form $f(u) \propto u^{\alpha}$.

a) Find the cumulative distribution, that is, $P(Z > z)$. You can then find the actual distribution from

$$f_Z(z) = \frac{dP(Z > z)}{dz} .$$

b) Generate a method to do logarithmic binning in Matlab. That is, you estimate the density by doing a histogram with bin-sizes that increase exponentially in size. Hint: Remember to divide by the correct bin-size.

Project 16.34: Cluster number density $n(s, p)$

We will generate the cluster number density $n(s, p)$ from the two-dimensional data-set.

a) Estimate $n(s, p)$ for a sequence of p values approaching $p_c = 0.59275$ from above and below.

Hint 1: The cluster sizes are extracted using `.Area` as described in a previous exercise.

Hint 2: Remember to remove the percolating cluster.

Hint 3: Use logarithmic binning.

b) Estimate $n(s, p_c; L)$ for $L = 2^k$ for $k = 4, \ldots, 9$. Use this plot to estimate τ.

c) Can you estimate the scaling of $s_\xi \sim |p - p_c|^{-1/\sigma}$ using this data-set?

Hint 1: Use $n(s, p)/n(s, p_c) = F(s/s_\xi) = 0.5$ as the definition of s_ξ.

Project 16.35: Mass scaling of percolating cluster

Find the mass $M(L)$ of the percolating cluster at $p = p_c$ as a function of L, for $L = 2^k$, $k = 4, \ldots, 11$. Plot $\log(M)$ as a function of $\log(L)$ and determine the exponent D.

Project 16.36: (Optional) Correlation function

a) Write a program to find the correlation function, $g(r, p, L)$ for $L = 256$. Plot $g(r, p, L)$ for $p = 0.55$ to $p = 0.65$ for $L = 256$.

b) Find the correlation length $\xi(p, L)$ for $L = 256$ for the p-values used above. Plot ξ as a fuction of $p - p_c$, and determine ν.

Project 16.37: (Optional) Finite Size Scaling

a) Write a program to find the radius of gyration, $R_g(s, p, L)$ for $L = 256$ and for p close to p_c.

b) Measure the dimension D.

c) Plot $R_s s^{-\nu D}$ as a function of s for $p > p_c$ and for $p < p_c$. Comment on the results.

Project 16.38: Finite Size Scaling

In this exercise we will use a finite size scaling ansatz to provide estimates of ν, p_c, and the average percolation probability $\langle p \rangle$ in a system of size L.

We define $p_{\Pi=x}$ so that

$$\Pi(p_{\Pi=x}) = x \ ,$$

notice that $p_{\Pi=x}$ is a function of system size L used for the simulation.

a) Find $p_{\Pi=x}$ for $x = 0.8$ and $x = 0.3$ for $L = 25, 50, 100, 200, 400, 800$. Plot $p_{\Pi=x}$ as a function of L.

According to the scaling theory we have

$$p_{x_1} - p_{x_2} = (C_{x_1} - C_{x_2}) L^{-1/\nu} \ .$$

b) $\log(p_{\Pi=0.8} - p_{\Pi=0.3})$ as a function of $\log(L)$ to estimate the exponent ν. How does it compare to the exact results.

In the following, please use the exact value of ν.

The scaling theory also predicted that

$$p_{\Pi=x} = p_c + C_x L^{-1/\nu} \ .$$

c) Plot $p_{\Pi=x}$ as a function of $L^{-1/\nu}$ to estimate p_c. Generate a data-collapse plot for $\Pi(p, L)$ to find the function $\Phi(u)$ from the lecture notes.

d) (Optional) Plot $\Pi'(p, L)$ as a function of p for the various L values. Generate a data-collapse plot of $\Pi'(p, L)$. Find $\langle p \rangle$ and plot $\langle p \rangle$ as a function of $L^{-1/\nu}$ to find p_c.

Project 16.39: Singly connected bonds

We have provided a subroutine and an example program that implements the left/right-turning walker algorithm. The algorithm works on a given cluster. From one end of the cluster, two walkers are released. The walkers can only walk according to the connectivity rules on the lattice. That is, for a nearest-neighbor lattice, they can only walk to their nearest neighbors. The left-turning walker always tries to turn left from its previous direction. If this site is empty, it tries the next-best site, which is to continue straight ahead. If that is empty, it tries to move right, and if that is empty, it moves back along the direction it came. The right-turning walker follows a similar rule, but prefers to turn right in each step. The first walker to reach the other end of the cluster stops, and the other walker stops when it reaches this site.

The path of the two walkers is illustrated in fig. **??** The sites that are visited by both walkers consitute the singly connected bonds. The union of the two walks consitutes what is called the external perimeter (Hull) of the cluster.

a) Run the program `exwalk.m` to visualize the singly connected bonds. Can you understand how this algorithm finds the singly connected bonds? Why are some of the bonds of a different color?

b) Find the mass, M_{SC}, of the singly connected bonds as a function of system size L for $p = p_c$ and use this to estimate the exponent D_{SC}: $M_{SC} \propto L^{D_{SC}}$. Can you find the behavior of $P_{SC} = M_{SC}/L^d$ as a function of $p - p_c$?

Project 16.40: Random walks on the spanning cluster

In this exercise we will use and modify the program `testpercwalk.m` to study random walks in percolation systems, and on the spanning cluster in particular. We want to find the dimension d_w of a two-dimensional random walk on the spanning cluster.

a) Find the distance $\langle R^2 \rangle$ as a function of the number of steps N for random walks on the spanning cluster for $p > p_c$. Plot $\log\langle R^2 \rangle$ as a function of N for various values of p. Can you produce a data-collapse plot for $\langle R^2 \rangle$ as a function of N?

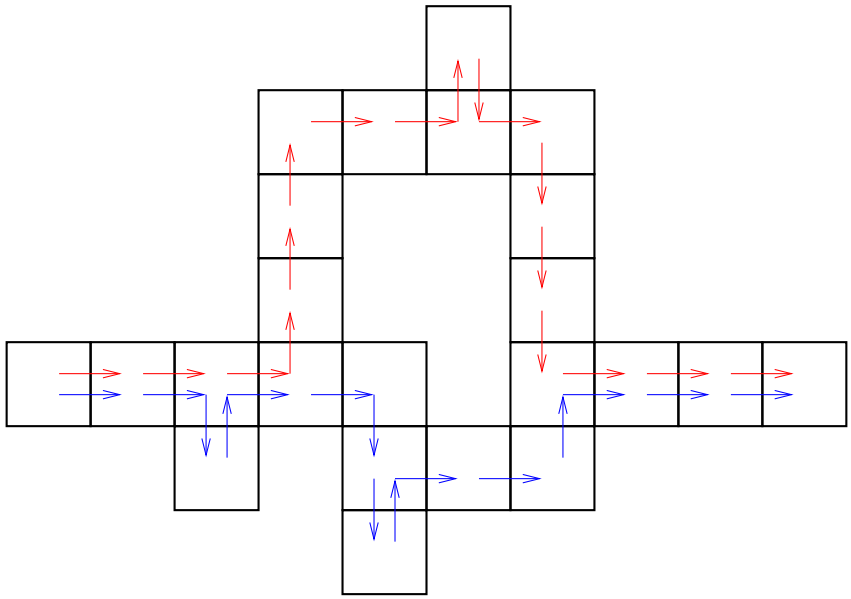

Fig. 16.3 Illustrations of the left-right turning walker.

b) Can you find the behavior of the correlation length ξ from this plot? Discuss the behavior of the characteristic cross-over time t_0 based on the plot. Find the dimension, d_w of the walk, from the relation $\langle R^2 \rangle \propto N^{2/d_w}$.

c) Find the distribution $P(R, N)$ for the position R as a function of the number of steps N for a random walker on the spanning cluster.

d) (Advanced and optional) Can you produce a data-collapse for the distribution $P(R, N)$. Can you determine the functional form of the distributoin $P(R, N)$. Is it a Gaussian?

Index