
Percolation theory using Python

Anders Malthe-Sørensen

Department of Physics, University of Oslo

Apr 1, 2020

Contents

1	Introduction to percolation	1
1.1	Basic concepts in percolation	5
1.2	Percolation probability	9
1.3	Spanning cluster	12
1.4	Percolation in small systems	14
1.5	Exercises	17
2	One-dimensional percolation	19
2.1	Percolation probability	19
2.2	Cluster number density	20
2.2.1	Definition of cluster number density	20
2.2.2	Measuring the cluster number density	23
2.2.3	Shape of the cluster number density	25
2.2.4	Numerical measurement of the cluster number density	27
2.2.5	Average cluster size	29
2.3	Spanning cluster	31
2.4	Correlation length	32
2.5	(Advanced) Finite size effects	34
2.5.1	Finite size effects in $\Pi(p, L)$ and p_c	35

2.6 Exercises	36
3 Infinite-dimensional percolation	37
3.1 Percolation threshold	38
3.2 Spanning cluster	38
3.3 Average cluster size	41
3.4 Cluster number density	42
3.5 Advanced: Embedding dimension.....	49
3.6 Exercises	49
4 Finite-dimensional percolation.....	51
4.1 Cluster number density	54
4.1.1 Numerical estimation of $n(s, p)$	54
4.1.2 Measuring probability densities of rare events	55
4.1.3 Measurements of $n(s, p)$ when $p \rightarrow p_c$	58
4.1.4 Scaling theory for $n(s, p)$	58
4.1.5 Scaling ansatz for 1d percolation	60
4.1.6 Scaling ansatz for Bethe lattice.....	61
4.2 Consequences of the scaling ansatz	61
4.2.1 Average cluster size	61
4.2.2 Density of spanning cluster	62
4.3 Percolation thresholds	64
4.4 Exercises	64
5 Geometry of clusters	69
5.1 Characteristic cluster size	69
5.1.1 Analytical results in one dimension	71
5.1.2 Numerical results in two dimensions	71
5.1.3 Scaling behavior in two dimensions	75
5.2 Geometry of finite clusters	75
5.2.1 Correlation length	78
5.3 Geometry of the spanning cluster	85
5.4 Spanning cluster above p_c	86
5.5 Fractal cluster geometry	88

5.6 Exercises	91
---------------------	----

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.

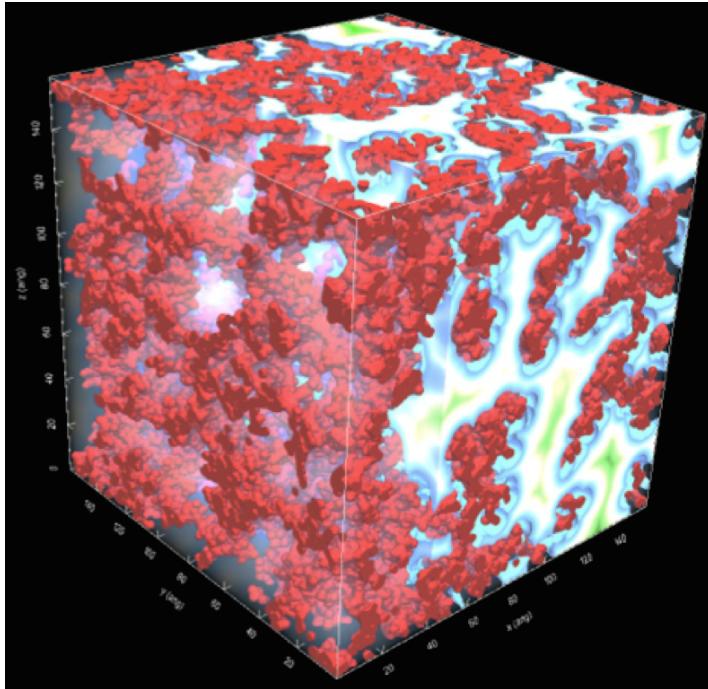


Fig. 1.1 Illustration of a porous material from a nanoporous silicate (SiO_2). The colors inside the pores illustrates the distance to the nearest part of the solid.

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 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, \mathbf{m} , in python using

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

The resulting matrices are shown in the Fig. 1.2 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).

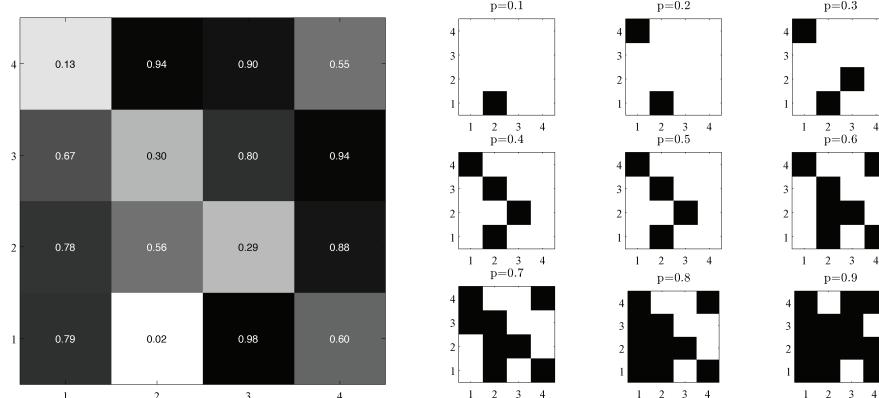


Fig. 1.2 Illustration of an array of 4×4 random numbers, and the various sites set for different values of p .

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? 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 depends on the rule for connectivity. It would also depend on the type of lattice used: square, triangular, 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 be 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 python using

```
from pylab import *
L = 20
p = 0.5
z = rand(L,L)
m = z<p
imshow(m, origin='lower')
show()
```

The resulting matrix is illustrated in Fig. 1.3. 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 python that finds connected regions in an image. The function `measurements.label` finds clusters based on a given connectivity. For example, with a connectivity corresponding to 4 we find

```
from scipy.ndimage import measurements
lw, num = measurements.label(m)
```

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.

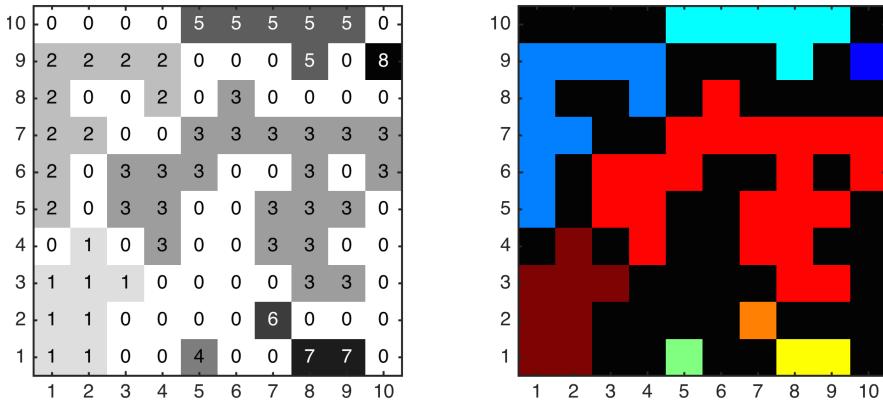


Fig. 1.3 Illustration of the index array for a 10×10 system for $p = 0.45$.

In order to visualize the clusters effectively, we give the various clusters different colors.

```
imshow(lw, origin='lower')
```

Unfortunately, this colors the clusters gradually from the bottom up. This is a property of the underlying algorithm: Clusters are indexed starting from the bottom-left of the matrix. Hence, clusters that are close to each other will get similar colors and therefore be difficult to discern unless we shuffle the colormap. We can fix this by shuffling the labeling:

```
b = arange(lw.max() + 1)
shuffle(b)
shuffledLw = b[lw]
imshow(shuffledLw, origin='lower')
```

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).

It may also be useful to color the clusters based on the size of the clusters, where size refers to the number of sites in a cluster. We can do this using

```
area = measurements.sum(m, lw, index=arange(lw.max() + 1))
areaImg = area[lw]
imshow(areaImg, origin='lower')
colorbar()
```

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:

```
from pylab import *
from scipy.ndimage import measurements
L = 100
pv = [0.2,0.3,0.4,0.5,0.6,0.7]
z = rand(L,L)
for i in range(len(pv)):
    p = pv[i]
    m = z<p
    lw, num = measurements.label(m)
    area = measurements.sum(m, lw, index=arange(lw.max() + 1))
    areaImg = area[lw]
    subplot(2,3,i+1)
    tit = 'p=' + str(p)
    imshow(areaImg, origin='lower')
    title(tit)
    axis()
```

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. 1.5. 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

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 $p_c \simeq 0.5927$.

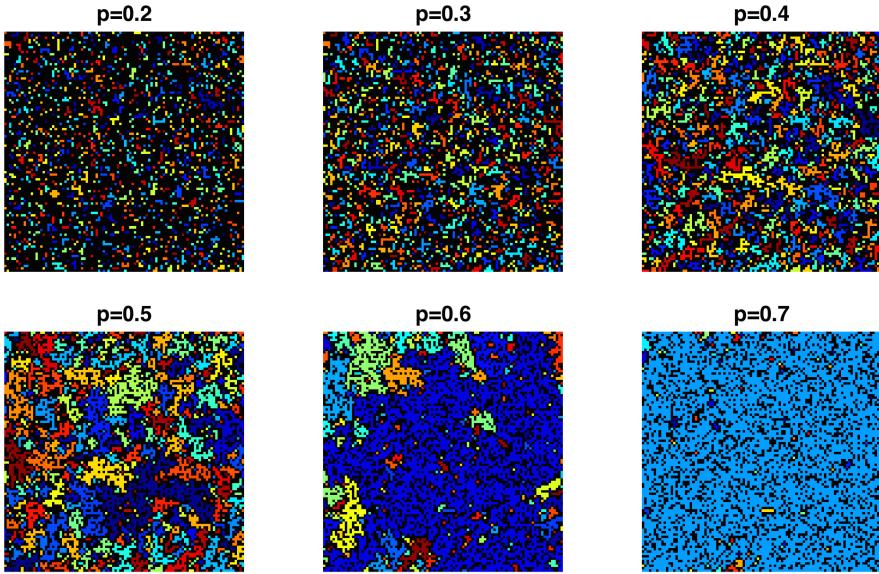


Fig. 1.4 Plot of the clusters in a 100×100 system for various values of p .

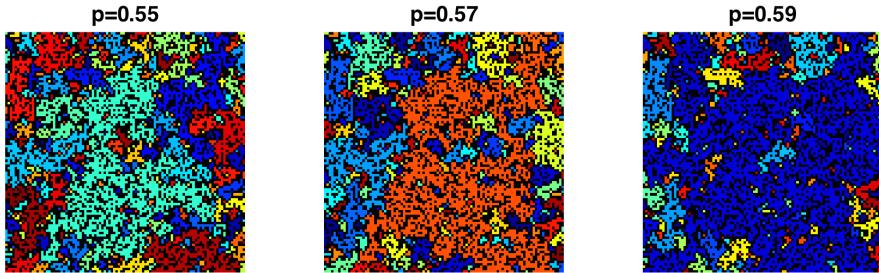


Fig. 1.5 Plot of the clusters in a 100×100 system for various values of p .

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 connecting 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 100 p_i values from 0.35 to 1.0:

```
p = linspace(0.35, 1.0, 100)
```

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

```
nx = len(p)
Pi = zeros(nx)
```

We will generate $N = 1000$ samples:

```
N = 100
```

We will then loop over all samples, and for each sample we generate a new random matrix. Then for each value of p_i we perform the cluster analysis as we did above. However, we now need to extract more information from the clusters. We use the function `measurements.find_objects` to find properties of a given cluster. Let us find the largest cluster and then the extent of this cluster. First, we find the largest cluster by:

```
labelList = arange(lw.max() + 1)
maxLabel = labelList[where(area == area.max())]
```

Now `maxLabel` is the label of the largest cluster. We can extract the properties of this properties using `find_objects`:

```
sliced = measurements.find_objects(lw == maxLabels)
sliceX = sliced[0][1]
sliceY = sliced[0][0]
dx = sliceX.stop - sliceX.start
dy = sliceY.stop - sliceY.start
```

Now `dx` and `dy` is the spatial extent of the largest cluster. If any of these extents are equal to the system size L , there is at least one spanning cluster spanning from one side to another. (Notice that this algorithm is not very robust. Can you think of ways this algorithm may fail?)

Now, we are ready to implement this into a complete program. For a given value of p , we count in how many simulations $N_p(p)$ there is a path spanning from one side to another and estimate $\bar{N}(p) \simeq N_p(p)/N$, where N is the total number of simulations/samples. This is implemented in the following program:

```
from pylab import *
from scipy.ndimage import measurements
p = linspace(0.4,1.0,100)
nx = len(p)
Ni = zeros(nx)
N = 1000
L = 100
for i in range(N):
    z = rand(L,L)
    for ip in range(nx):
        m = z<p[ip]
        lw, num = measurements.label(m)
        labelList = arange(lw.max() + 1)
        area = measurements.sum(m, lw, labelList)
        maxLabel = labelList[where(area == area.max())]
```

```

sliced = measurements.find_objects(lw == maxLabel)
if(len(sliced) > 0):
    sliceX = sliced[0][1]
    sliceY = sliced[0][0]
    dx = sliceX.stop-sliceX.start
    dy = sliceY.stop-sliceY.start
    maxsize = max(dx,dy)
    if (maxsize>=L): # Percolation
        Ni[ip] = Ni[ip] + 1
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 $p = p_c$.

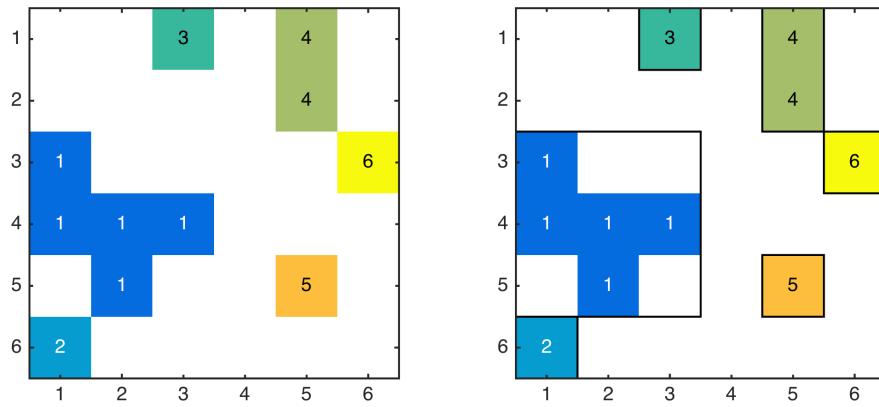


Fig. 1.6 Illustration of the BoundingBox for the clusters in a 6×6 simulation.

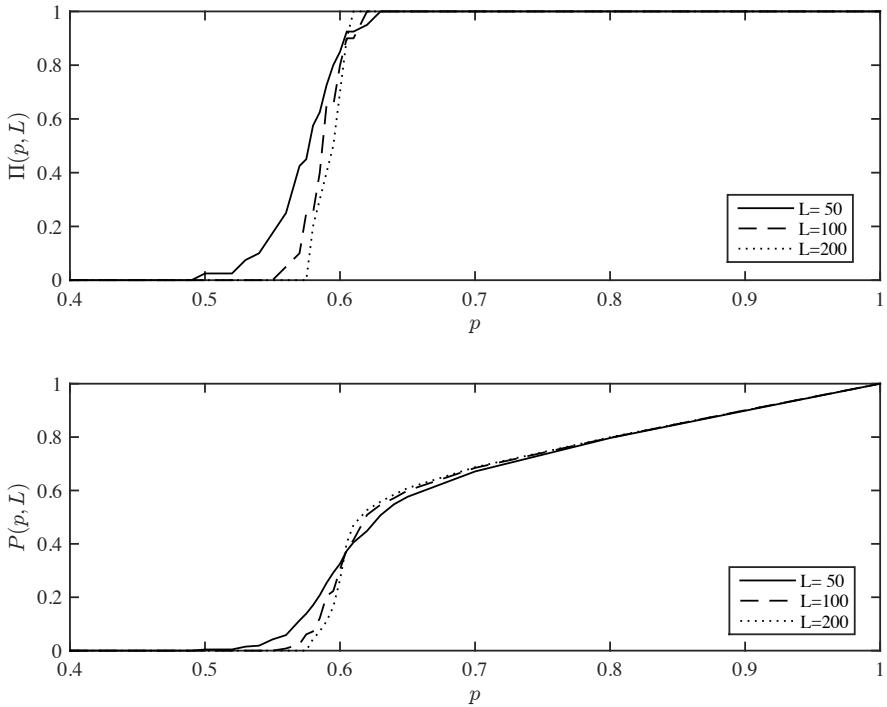


Fig. 1.7 Plot of $\Pi(p, L)$, the probability for there to be a connected path from one side to another, as a function of p for various system sizes L .

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 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 counting 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 = measurements.sum(m, lw, labelList)

We do this in the same program as we developed above. For each p_i , we see if a cluster is spanning from one side to another, and if it is, we add the mass of this cluster to $M_S(p_i)$. We implement these features in the following program, which measures both $\Pi(p, L)$ and $P(p, L)$ for a given value of L :

```
from pylab import *
from scipy.ndimage import measurements
p = linspace(0.4,1.0,100)
nx = len(p)
Ni = zeros(nx)
P = zeros(nx)
N = 1000
L = 100
for i in range(N):
    if (i%100==0):
        print("i = ",i)
    z = rand(L,L)
    for ip in range(nx):
        m = z<p[ip]
        lw, num = measurements.label(m)
        labelList = arange(lw.max() + 1)
        area = measurements.sum(m, lw, labelList)
        maxLabel = labelList[where(area == area.max())]
        sliced = measurements.find_objects(lw == maxLabel)
        if(len(sliced) > 0):
            sliceX = sliced[0][1]
            sliceY = sliced[0][0]
            dx = sliceX.stop-sliceX.start
            dy = sliceY.stop-sliceY.start
            maxsize = max(dx,dy)
            if (maxsize>=L): # Percolation
                Ni[ip] = Ni[ip] + 1
                P[ip] = P[ip] + area.max()
Pi = Ni/N
P = P/(L*L)
subplot(2,1,1)
plot(p,Pi)
subplot(2,1,2)
plot(p,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 $p \rightarrow 1$. 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.

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 .

$$P(A) = \sum_B P(A|B)P(B) , \quad (1.1)$$

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 :

$$\Pi(p, L) = \sum_c \Pi(p, L|c)P(c) , \quad (1.2)$$

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

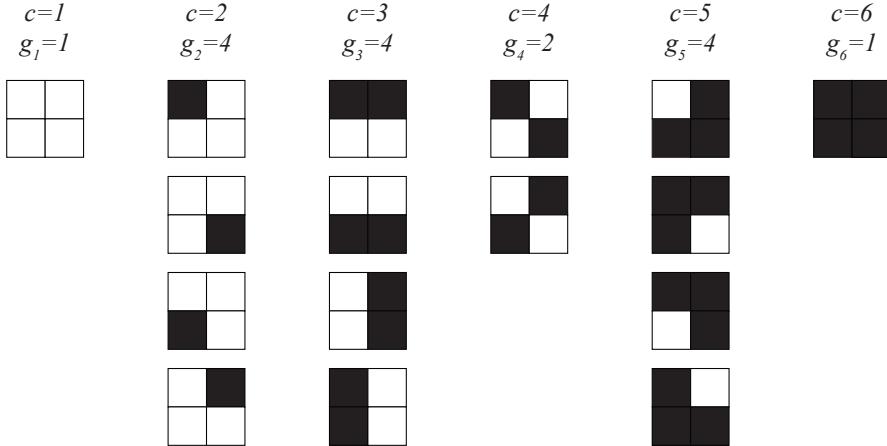
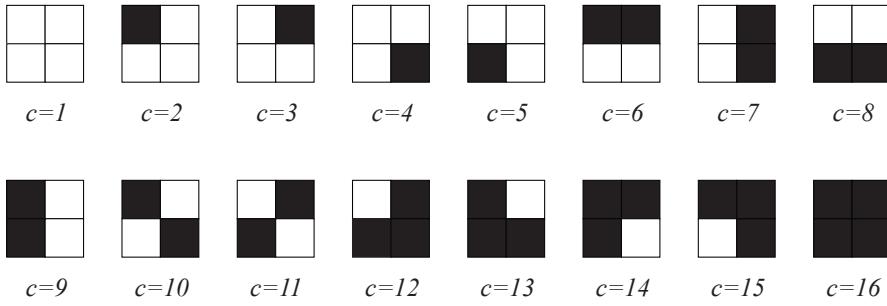


Fig. 1.8 The possible configurations for a $L = 2$ site percolation lattice in two-dimensions. The configurations are indexed using the cluster configuration number c .

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:

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

$$+ 0 \cdot 2 \cdot p^2(1-p)^2 + 1 \cdot 4 \cdot p^3(1-p)^1 + 1 \cdot 1 \cdot p^4(1-p)^0. \quad (1.4)$$

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

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

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

$$\Pi(p, L) \simeq Lp^L + c_1p^{L+1} + \dots + c_n p^{L^2}, \quad (1.6)$$

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

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

$$\Pi(p, L) \simeq 1 - (1-p)^L. \quad (1.7)$$

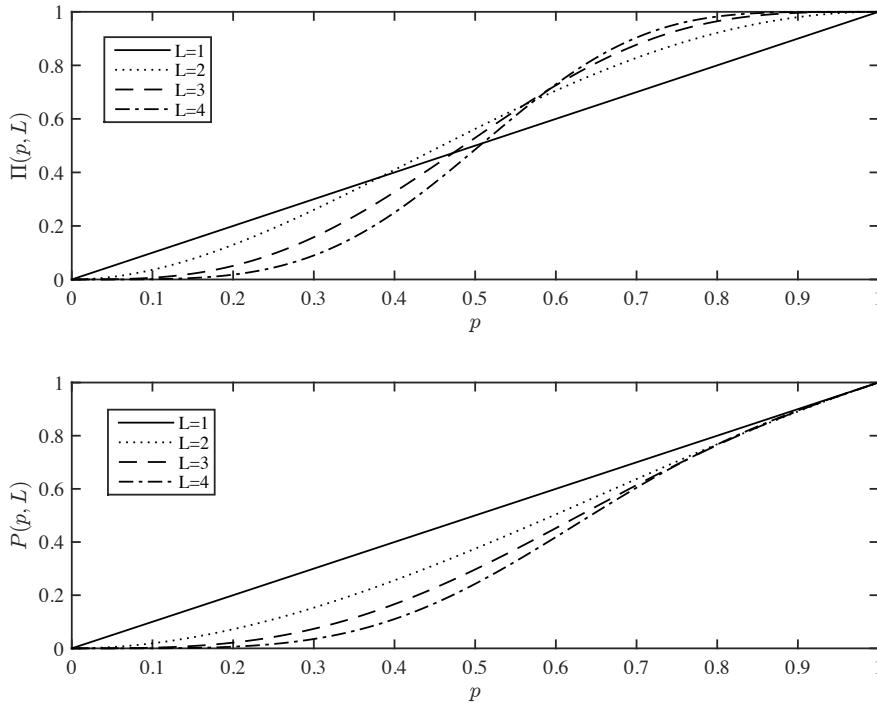


Fig. 1.9 Plot of $\Pi(p, L)$ for $L = 1$ and $L = 2$ as a function of p .

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

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 your 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.

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

$$\Pi(p, L) = p^L \quad (2.1)$$

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

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

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 ?

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

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:

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

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:

$$n(s, p) = (1 - p)^2 p^s . \quad (2.4)$$

This is the cluster number density in one dimension.

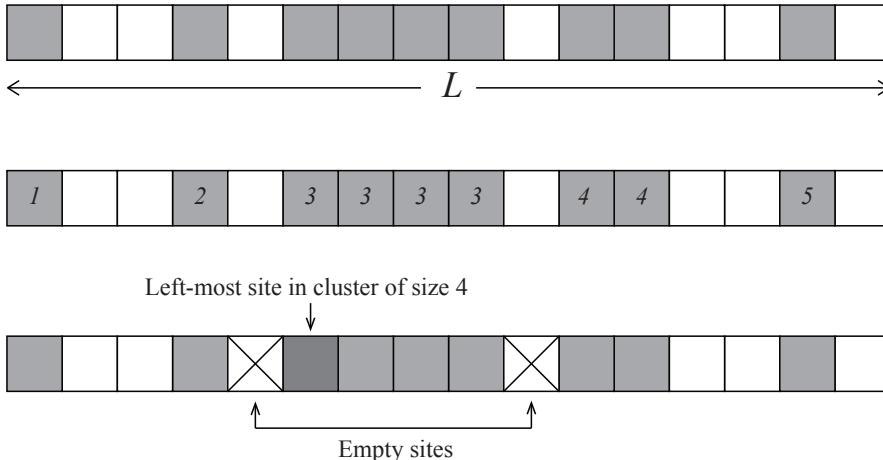


Fig. 2.1 Realization of a $L = 16$ percolation system in one dimension. Occupied sites are marked with black squares.

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 infinite 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 :

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

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

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

where we will now employ a common trick:

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

which gives

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

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

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

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

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

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

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

For the clusters in Fig. 1.8 we find that

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

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

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

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

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:

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

where \bar{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

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

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

$$n(s, p) = (1 - p)^2 p^s . \quad (2.18)$$

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

$$G(s) = (1 - p)^2 n(s, p) = p^s , \quad (2.19)$$

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

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

where we have introduced the cut-off cluster size

$$s_\xi = \frac{-1}{\ln p} . \quad (2.21)$$

What we are seeing in Fig. 2.2 is therefore the exponential cut-off curve, where the cut-off $s_\xi(p)$ increases as $p \rightarrow 1$. 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 (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:

$$s_\xi = -\frac{1}{\ln p} \quad (2.22)$$

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

$$\ln p = \ln(1 - (1 - p)) \simeq -(1 - p) , \quad (2.23)$$

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

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

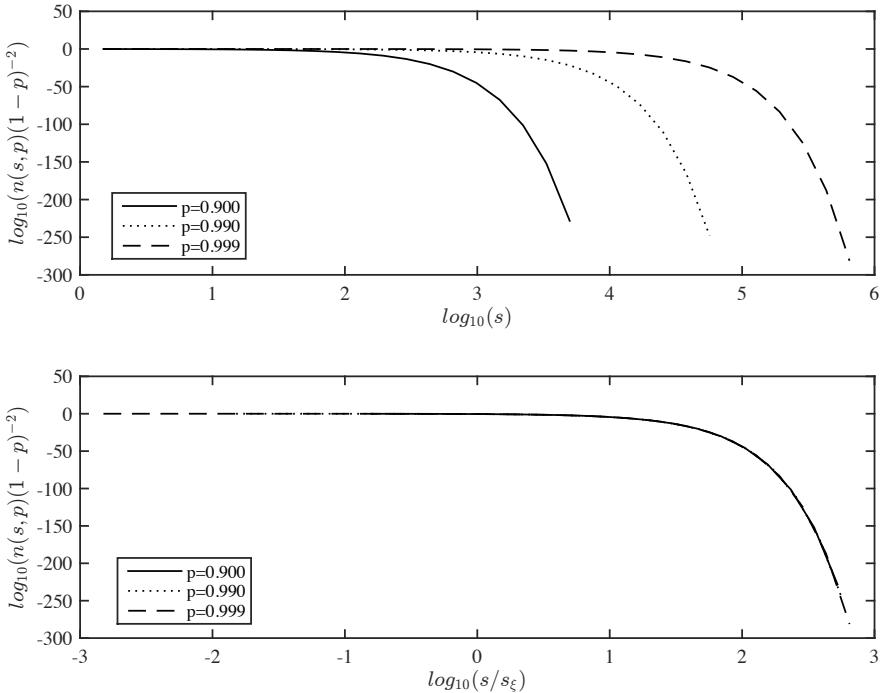


Fig. 2.2 (Top) A plot of $n(s, p)(1 - p)^2$ as a function of s for various values of p for a one-dimensional percolation system shows that the cut-off increases as a function of s . (Bottom) When the s axis is rescaled by s/s_ξ all the curves fall onto a common scaling function, that is, $n(s, p) = (1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

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 cluster size s_ξ diverges as

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

when $p \rightarrow p_c$. 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:

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

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

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

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

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

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

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

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

```
from pylab import *
from scipy.ndimage import measurements
L = 20
p = 0.90
z = rand(L)
```

```
m = z<p
lw, num = measurements.label(m)
```

Now, `lw` contains the indecies for all the clusters. We can extract the size of the clusters by summing the number of elements for each label:

```
labelList = arange(lw.max() + 1)
area = measurements.sum(m, lw, labelList)
```

The resulting list of areas for one sample is

```
>> lw
array([1, 1, 1, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 3, 0, 4, 4, 4, 4],
      dtype=int32)
>> area
array([0., 3., 9., 1., 4.])
```

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.

```
from pylab import *
from scipy.ndimage import measurements
nsamp = 1000
L = 1000
p = 0.90
allarea = array([])
for i in range(nsamp):
    z = rand(L)
    m = z<p
    lw, num = measurements.label(m)
    labelList = arange(lw.max() + 1)
    area = measurements.sum(m, lw, labelList)
    allarea = append(allarea,area)
n,sbins = histogram(allarea,bins=int(max(allarea)))
s = 0.5*(sbins[1:]+sbins[:-1])
nsp = n/(L*nsamp)
sxi = -1.0/log(p)
nsptheory = (1-p)**2*exp(-s/sxi)
plot(s,nsp,'o',s,nsptheory,'-')
xlabel('$s$')
ylabel('$n(s,p)$')
```

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

```
n,sbins = histogram(allarea,bins=int(max(allarea)))
s = 0.5*(sbins[1:]+sbins[:-1])
```

where we find s as the midpoints of the bins returned by the `histogram`-function.

We estimate $\overline{n(s, p)}$ 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 $s_\xi = -1/\ln p$. 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 `nonzero` function to find the indices of the elements of `n` that are non-zero:

```
i = nonzero(n)
```

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],'o',s,nsptheory,'-')
```

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 were 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.

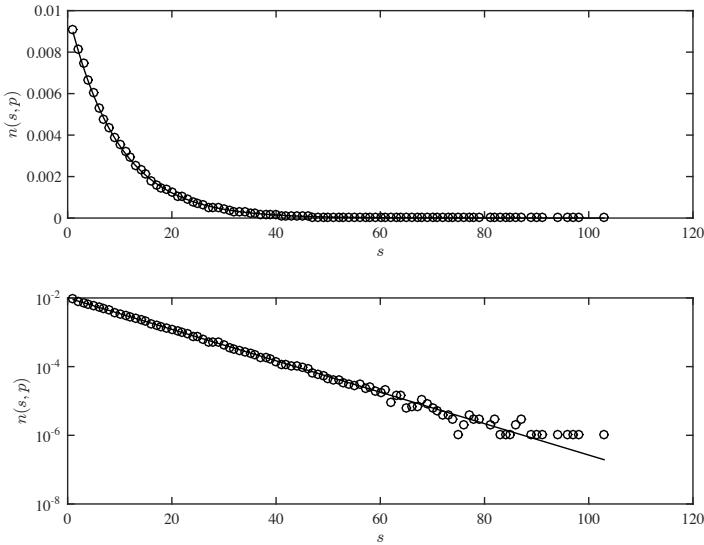


Fig. 2.3 Plot of the predicted $\bar{n}(s, p)$, based on $M = 1000$ samples of a $L = 1000$ system with $p = 0.9$, and the theoretical $n(s, p)$ curve on a linear scale (top) and a semilogarithmic scale (bottom). The semilogarithmic plot clearly shows that $n(s, p)$ follows an exponential curve.

Average cluster size

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$= \frac{1+p}{1-p} \quad (2.40)$$

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

This shows that we can write

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

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

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

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:

$$p = P(p; L) + \sum_s sn(s, p; L), \quad (2.43)$$

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

$$P(p) = p - \sum_s sn(s, p). \quad (2.44)$$

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 $p \rightarrow p_c$. We expect a similar behavior for the one-dimensional system. We have already seen that the mass (or area) of the clusters diverges as $p \rightarrow p_c$. 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**, $g(r)$:

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

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

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

$$\ln p = \ln(1 - (1 - p)) \simeq -(1 - p) . \quad (2.46)$$

We find that the correlation length is

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

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

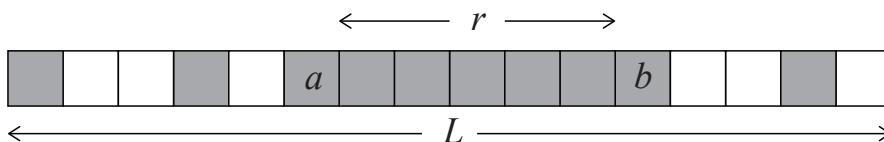


Fig. 2.4 An illustration of the distance r between two sites a and b . The two sites a and b are connected if and only if all the sites between a and b are occupied.

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

$$\sum_{s=1}^L p^s = \frac{1 - p^L}{1 - p}, \quad (2.48)$$

instead of $1/(1 - p)$ when L is infinite. Indeed, this sum approaches $1/(1 - p)$ as $L \rightarrow \infty$. This implies that S will approach L when $p \rightarrow p_c$, as can be seen by applying l'Hopital's rule to find the limit as $p \rightarrow p_c$. 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

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

What is the interpretation of Π' ? We can write

$$\Pi'(p, L)dp = \Pi(p + dp, L) - \Pi(p, L), \quad (2.50)$$

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).

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

This gives

$$\ln p_x = -\frac{\ln 2}{L-1}, \quad (2.52)$$

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

$$\ln p_x = \ln(1 - (1 - p_x)) \simeq -(1 - p_x), \quad (2.53)$$

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

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

and consequently,

$$p_x = p_c - \frac{\ln 2}{L-1}. \quad (2.55)$$

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. ??.

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

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

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

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

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

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

- 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.

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.

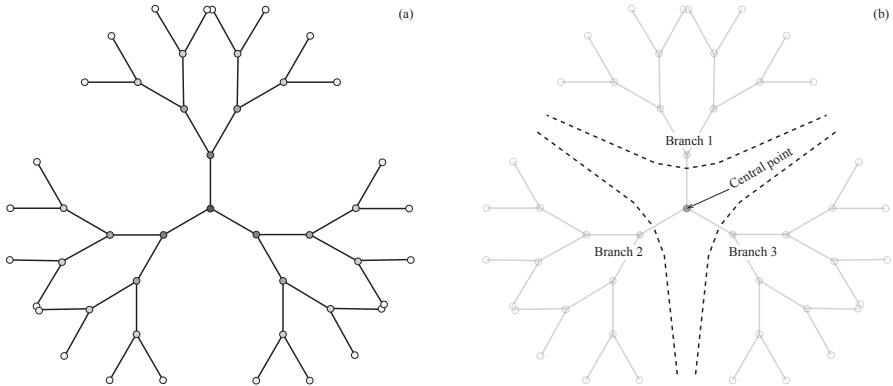


Fig. 3.1 Illustration of four generations of the Bethe lattice with number of neighbors $Z = 3$.

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:

$$p(Z - 1) \geq 1 , \quad (3.1)$$

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

$$p_c = \frac{1}{Z - 1} . \quad (3.2)$$

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

$$p = P + \sum_s sn(s, p) , \quad (3.3)$$

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

$$P = p(1 - Q^Z) . \quad (3.4)$$

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

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

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

$$Q = 1 - p + pQ^2 , \quad (3.6)$$

$$pQ^2 - Q + 1 - p = 0 . \quad (3.7)$$

The solution of this second order equation is

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

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 $p > p_c$:

$$P = p(1 - Q^3) \quad (3.9)$$

$$= p\left(1 - \left(\frac{1-p}{p}\right)^3\right) \quad (3.10)$$

$$= p\left(1 - \frac{1-p}{p}\right)\left(1 + \frac{1-p}{p} + \left(\frac{1-p}{p}\right)^2\right) . \quad (3.11)$$

This result is illustrated in Fig. 3.2.

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

$$P = 2\left(p - \frac{1}{2}\right)\left(1 + \frac{1-p}{p} + \left(\frac{1-p}{p}\right)^2\right) , \quad (3.12)$$

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

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

We have therefore found that for $p > p_c$

$$P(p) \simeq B(p - p_c)^\beta , \quad (3.14)$$

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

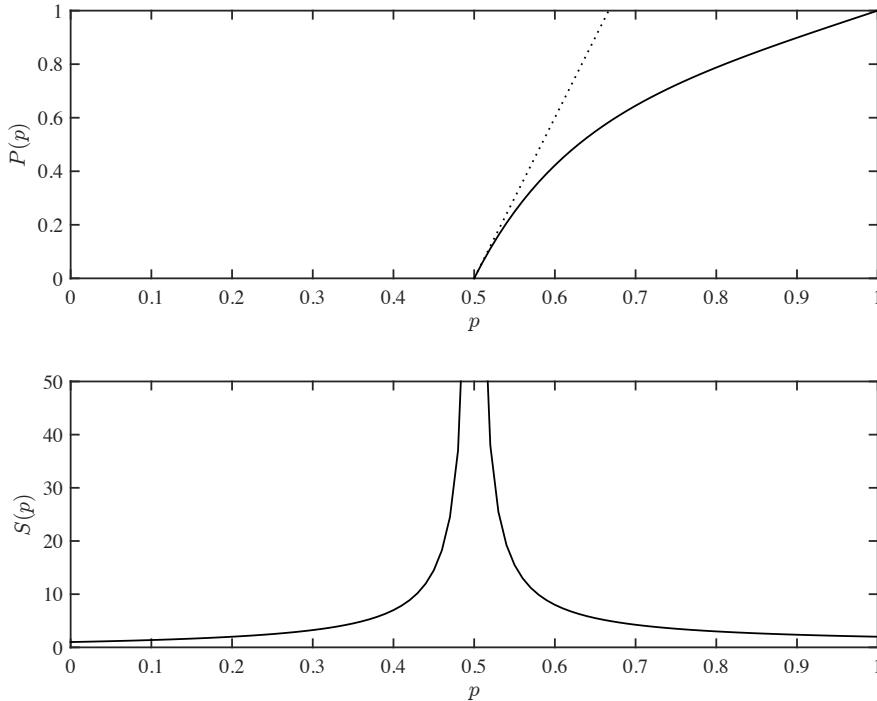


Fig. 3.2 (Top) A plot of $P(p)$ as a function of p for the Bethe lattice with $Z = 3$. The tangent at $p = p_c$ is illustrated by a straight line. (Bottom) A plot of the average cluster size, $S(p)$, as a function of p for the Bethe lattice with $Z = 3$. The average cluster size diverges when $p \rightarrow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

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

$$S = 1 + ZT , \quad (3.15)$$

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:

$$T = (1 - p)0 + p(1 + (Z - 1)T), \quad (3.16)$$

We can solve this directly for T , finding

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

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

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

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

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

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:

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

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

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

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.

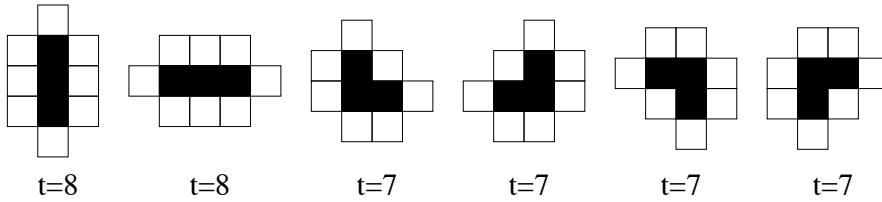


Fig. 3.3 Illustration of the 6 possible configurations for a two-dimensional cluster of size $s = 3$.

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 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,

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

and therefore:

$$t_s = s(Z - 2) + 2 . \quad (3.23)$$

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

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

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

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

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

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

which, for $Z = 3$ becomes

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

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 $f(p)$.

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

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

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

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

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

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

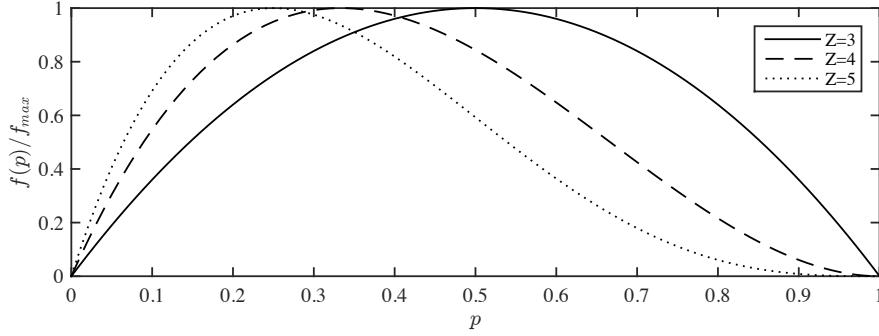


Fig. 3.4 A plot $f(p) = p(1-p)^{Z-2}$, which is a term in the cluster number density $n(s, p) = g_s [p(1-p)^{Z-2}]^s (1-p)^2$ for the Bethe lattice. We notice that $f(p)$ has a maximum at $p = p_c$, and that the second derivative, $f''(p)$, is zero in this point. A Taylor expansion of $f(p)$ around $p = p_c$ 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 .

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

The cluster number density is

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

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

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

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

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

Consequently, for $p = p_c$ we get

$$n(s, p_c) = g_s A^s (1-p_c)^2 . \quad (3.36)$$

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

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

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

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

where the characteristic cluster size s_ξ is

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

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

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

where the exponent σ is universal, meaning that it does not depend on lattice details such as 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 to find limits on the behavior of $n(s, p_c)$.

The average cluster size at $p = p_c$ is

$$S = \frac{\Gamma}{p_c - p} , \quad (3.41)$$

which should diverge, that is

$$S = \sum s^2 n(s, p_c) \rightarrow \infty , \quad (3.42)$$

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

$$S = \int_0^\infty s^2 n(s, p_c) ds \rightarrow \infty , \quad (3.43)$$

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

$$n(s, p_c) \simeq C s^{-\tau} , \quad (3.44)$$

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

$$\sum_s s n(s, p) = p - P , \quad (3.45)$$

which should converge, and

$$\sum_s s^2 n(s, p_c) \rightarrow \infty , \quad (3.46)$$

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

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

and

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

which therefore implies that

$$2 < \tau \leq 3 . \quad (3.49)$$

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

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

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

$$S = C \sum_s s^{2-\tau} e^{-s/s_\xi} \rightarrow C \int_1^\infty s^{2-\tau} e^{-s/s_\xi} ds . \quad (3.51)$$

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

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

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

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

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

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

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

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

where we have used that

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

and that

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

Direct solution therefore shows that

$$\tau = \frac{5}{2}. \quad (3.58)$$

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.

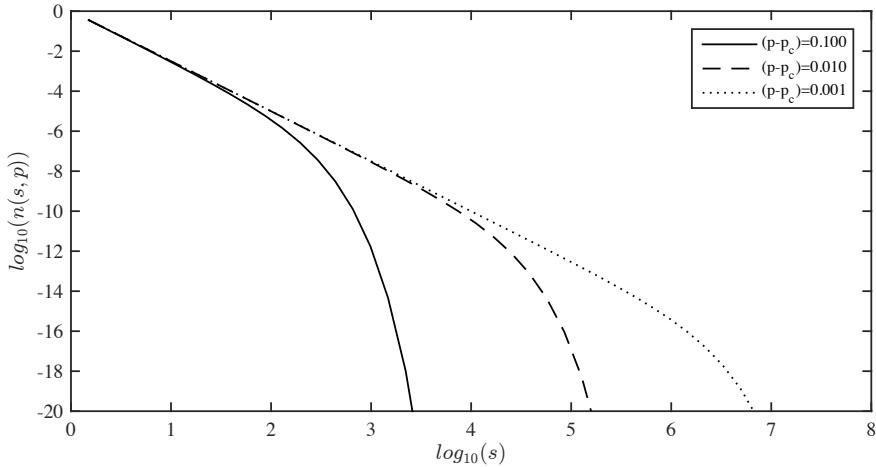


Fig. 3.5 A plot of $n(s, p) = s^{-\tau} \exp(-s(p - p_c)^2)$ as a function of s for various values of p illustrates how the characteristic cluster size s_ξ appears as a cut-off in the cluster number density that scales with $p - p_c$.

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 $p \rightarrow p_c$. We therefore expect a general form of the cluster density

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

where

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

and

$$s_\xi = s_0 |p - p_c|^{-1/\sigma}. \quad (3.61)$$

In addition, we have the following scaling relations:

$$P(p) \sim (p - p_c)^\beta, \quad (3.62)$$

$$\xi \sim |p - p_c|^{-\nu}, \quad (3.63)$$

and

$$S \sim |p - p_c|^{-\gamma}, \quad (3.64)$$

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

$$V \propto L^d, \quad (3.65)$$

and the surface, S , grows as

$$S \propto L^{d-1}, \quad (3.66)$$

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

$$S \propto V^{1-\frac{1}{d}}. \quad (3.67)$$

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

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 .

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 infinite-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 except 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.

This means that the solution methods used for the one and the infinite dimensional systems cannot be extended 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, $n(s, p)$.

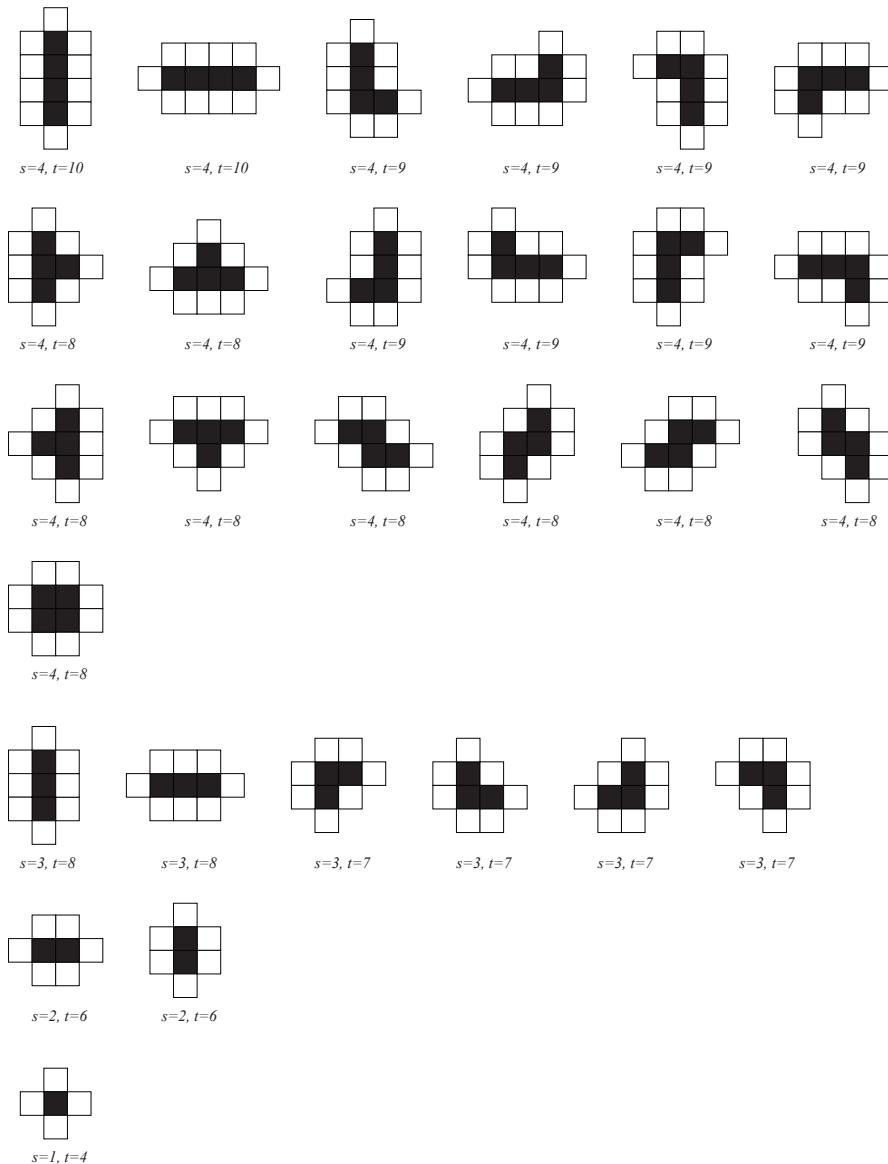


Fig. 4.1 Illustration of the possible configurations for two-dimensional clusters of size $s = 1, 2, 3, 4$.

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 :

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

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:

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

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

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

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:

```

from pylab import *
from scipy.ndimage import measurements
nsamp = 2000
L = 200
p = 0.58
allarea = array([])
for i in range(nsamp):
    z = rand(L)
    m = z<p
    lw, num = measurements.label(m)
    labelList = arange(lw.max() + 1)
    area = measurements.sum(m, lw, labelList)
    allarea = append(allarea,area)
n,sbins = histogram(allarea,bins=int(max(allarea)))
s = 0.5*(sbins[1:]+sbins[:-1])
nsp = n/(L*nsamp)
i = nonzero(n)
subplot(2,1,1)
plot(s[i],nsp[i],'o')
xlabel('$s$')
ylabel('$n(s,p)$')
subplot(2,1,2)
loglog(s[i],nsp[i],'o')
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 probability 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 $2^0, 2^1, 2^2, 2^3, \dots$, that is $1, 2, 4, 8, \dots$ (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 , occurring 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

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

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

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

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

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

$$i_m > \log_a \max(s). \quad (4.7)$$

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

```
a = 1.2
logamax = 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

```
logbins = a**arange(0,logamax)
```

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

```
nl,nlbins = histogram(allarea,bins=logbins)
```

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

```
nl,nlbins = histogram(allarea,bins=logbins)
```

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

```
ns1 = nl/(nsamp*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))
logbins = a**arange(0,logamax)
n1,nlbins = histogram(allarea,bins=logbins)
ds = diff(logbins)
s1 = 0.5*(logbins[1:]+logbins[:-1])
ns1 = n1/(M*L**2*ds)
loglog(s1,ns1,'.b')

```

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.

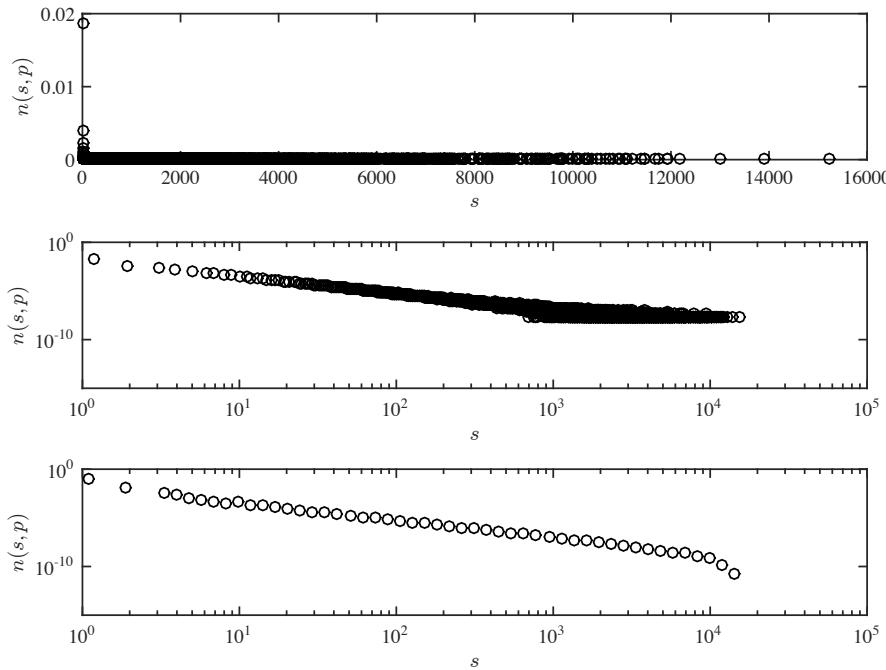


Fig. 4.2 Plot of $n(s, p; L)$ estimated from $M = 1000$ samples for $p = 0.58$ and $L = 200$.
 (a) Direct plot. (b) Log-log plot. (c) Plot of the logarithmically binned distribution.

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

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 $n(s, p)$? When s reaches s_ξ , it falls off from the power-law type behavior observed as $p \rightarrow p_c$. 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, $\overline{n(s, p)}$ 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

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

where σ was 1 in 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 $p \rightarrow p_c$, 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 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 $n(s, p)$:

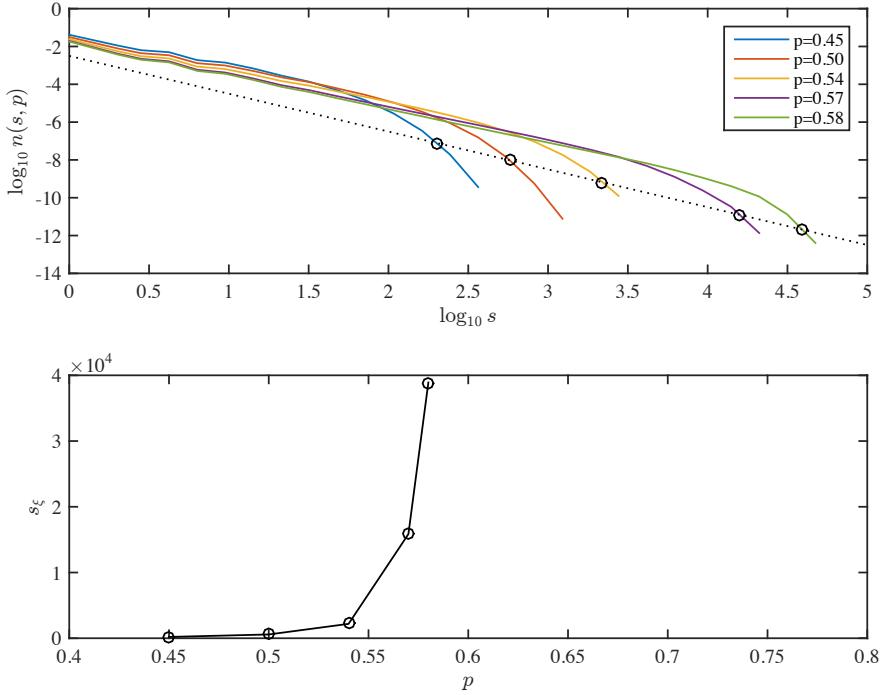


Fig. 4.3 (a) Plot of $n(s, p; L)$ as a function of s for various values of p for a 512×512 lattice. (b) Plot of $s_\xi(p)$ measured from the plot of $n(s, p)$ corresponding to the points shown in circles in (a).

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

$$n(s, p_c) = C s^{-\tau}, \quad (4.10)$$

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

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:

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

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.

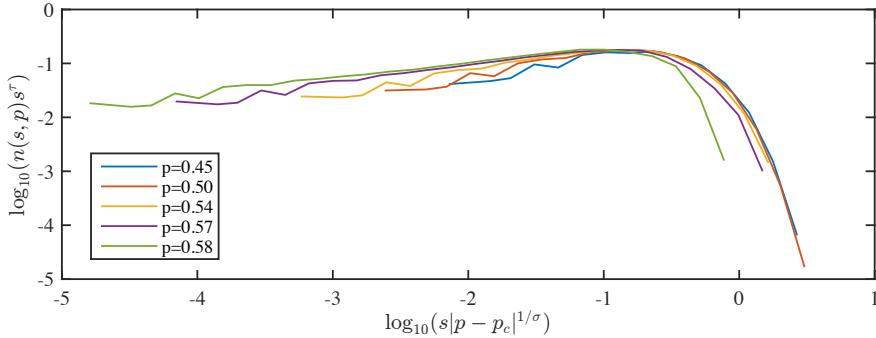


Fig. 4.4 A plot of $n(s, p)s^\tau$ as a function of $|p - p_c|^{1/\sigma}s$ shows that the cluster number density satisfies the scaling ansatz of (4.12).

An alternative scaling form is

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

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

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

We showed that we could rewrite this as

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

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

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

which is already on the wanted form, so that

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

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

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

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

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

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

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

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

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

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

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

And the result for the average cluster size is therefore

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

which gives a scaling relation for γ :

$$\gamma = \frac{3 - \tau}{\sigma} . \quad (4.24)$$

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

$$\sum_s s n(s, p) + P(p) = p , \quad (4.25)$$

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

$$P(p) = p - \sum_s s n(s, p) . \quad (4.26)$$

We will now use a standard trick, which is that

$$P(p_c) = p_c - \sum_s s n(s, p_c) = 0 . \quad (4.27)$$

Subtracting (4.27) from (4.26), we find that

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

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

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

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

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

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

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

The sum is therefore

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

Inserting into $P(p)$ produces:

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

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

$$P(p) \propto (p - p_c)^\beta . \quad (4.34)$$

$$\beta = \frac{\tau - 2}{\sigma} . \quad (4.35)$$

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 known values for the percolation thresholds for various dimensions 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 python 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

```
from pylab import *
from scipy.ndimage import measurements
L = 100
r = rand(L,L)
p = 0.6
```

```
z = r<p # This generates the binary array
lw, num = measurements.label(m)
```

We have then produced the array `lw` that contains labels for each of the connected clusters.

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

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

```
imshow(lw)
```

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

```
labelList = arange(lw.max() + 1)
area = measurements.sum(m, lw, labelList)
```

You can also extract information about the clusters using

```
maxLabel = labelList[where(area == area.max())]
sliced = measurements.find_objects(lw == maxLabel)
if(len(sliced) > 0):
    sliceX = sliced[0][1]
    sliceY = sliced[0][0]
```

as described above.

- b)** Using these features, write a program to calculate $P(p, L)$ for various p for the two-dimensional system.
- 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 get 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

$$P(p, L) \sim (p - p_c)^\beta . \quad (4.36)$$

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 python:

```
from pylab import *
z = rand(1e6)**(-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

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

b) Generate a method to do logarithmic binning in python. 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.

Hint 1. The cluster sizes are extracted using `area = measurements.sum(m, 1w, labelList)` as described in a previous exercise.

Hint 2. Remember to remove the percolating cluster.

Hint 3. Use logarithmic binning.

- a) Estimate $n(s, p)$ for a sequence of p values approaching $p_c = 0.59275$ from above and below.
- b) Estimate $n(s, p_c; L)$ for $L = 2^k$ for $k = 4, \dots, 9$. 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?

We have seen how we can characterize clusters by their mass, s .

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

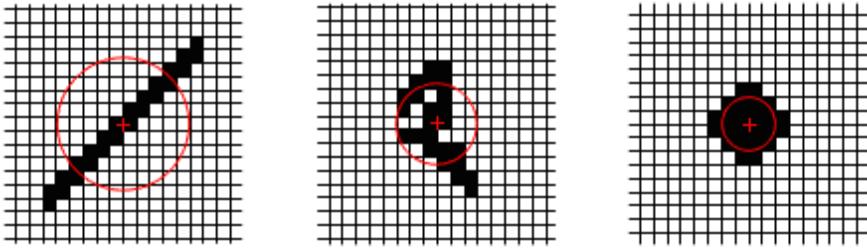


Fig. 5.1 Illustrations of three clusters all with $s = 24$.

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 $j = 1, \dots, s_i$, is defined as

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

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

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

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 figures¹.

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 .

$$R_s^2 = \langle R_i^2 \rangle_i . \quad (5.3)$$

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

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} .

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 :

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

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

$$R_s^2 = \frac{1}{s} \left[\sum_{i=1}^s i^2 - is + \frac{s^2}{4} \right] \quad (5.5)$$

$$= \frac{1}{s} \left[\frac{s^3}{3} - s \frac{s(s+1)}{2} + s \frac{s^2}{4} \right] \quad (5.6)$$

$$\propto s^2 \quad (5.7)$$

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 $j = 1, \dots, s_i$, is

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

We assume that the clusters are numbered and marked in the lattice with their index, as done by the `lw, num = measurements.label(m)` command. We can find the center of mass by a built-in function, such as `cm = measurements.center_of_mass(m, lw, labelList)` or we can calculate the center-of-mass explicitly. This is done by running 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 add the coordinates for this part of the cluster to the sum for the center of mass of the cluster

```
rcm[ilw] = rcm[ilw] + array([ix, iy])
```

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

```
rcm[:,0] = rcm[:,0]/area
rcm[:,1] = rcm[:,1]/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 from the site to the center of mass:

```
dr = array([ix, iy]) - cm[ilw]
rad2[ilw] = rad2[ilw] + 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

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

This is implemented in the following function:

```
from pylab import *
from scipy.ndimage import measurements

def radiusofgyration(m, lw):
    labelList = arange(lw.max() + 1)
    area = measurements.sum(m, lw, labelList)
    cm = measurements.center_of_mass(m, lw, labelList)
    nx = shape(lw)[0]
    ny = shape(lw)[1]
    rad2 = zeros(int(lw.max() + 1))
    for ix in range(nx):
```

```

for iy in range(ny):
    ilw = lw[ix,iy];
    if (ilw>0):
        dr = array([ix,iy])-cm[ilw]
        dr2 = dot(dr,dr)
        rad2[ilw] = rad2[ilw] + dr2
rad = sqrt(rad2/area)

```

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 = array([])
allarea = array([])
for i in range(M):
    z = rand(L,L)
    m = z<p
    lw, num = measurements.label(m)
    area,rcm,rad2 = radiusofgyration(m,lw)
    allr2 = append(allr2,rad2)
    allarea = append(allarea,area)

```

The resulting plots for several different values of p are shown in Fig. 5.2. We see 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:

$$R_s^2 \propto s^x. \quad (5.10)$$

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:

$$s \propto R_s^D, \quad (5.11)$$

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 known for the two-dimensional percolation problem, see Table 4.1.4). This non-integer value of D may seem strange, but it 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. 5.5.

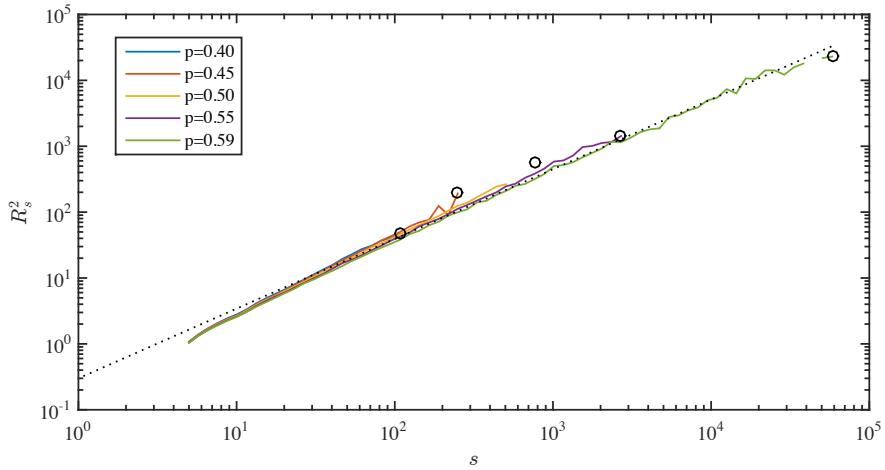


Fig. 5.2 Plot of R_s^2 as function of s for simulations on two-dimensional percolation system with $L = 400$. The largest cluster for each value of p is illustrated by a circle. The dotted line shows the curve $R_s^2 \propto s^{2/D}$ for $D = 1.89$.

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:

$$s_\xi \propto R_{s_\xi}^D. \quad (5.12)$$

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 :

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

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

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

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 :

$$R_s^2 = \langle R_i^2 \rangle_i , \quad (5.15)$$

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

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

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

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

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 choice 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 present here will not change in any significant way, except 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:

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

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

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

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

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

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

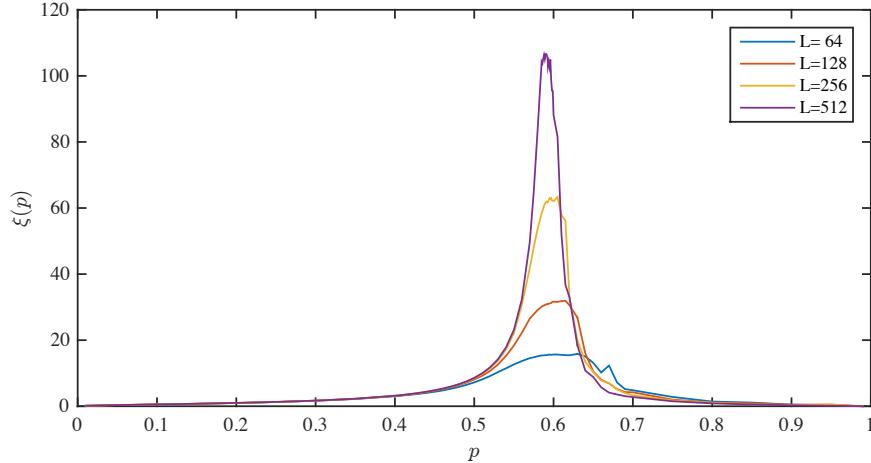


Fig. 5.3 A plot of ξ as a function of p for a $L = 64, 128, 256$ and 512 system as a function of p . We observe that ξ diverges when $p \rightarrow p_c$. We notice that the correlation length does not diverge, but crosses over as a result of the finite system size.

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

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

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

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.

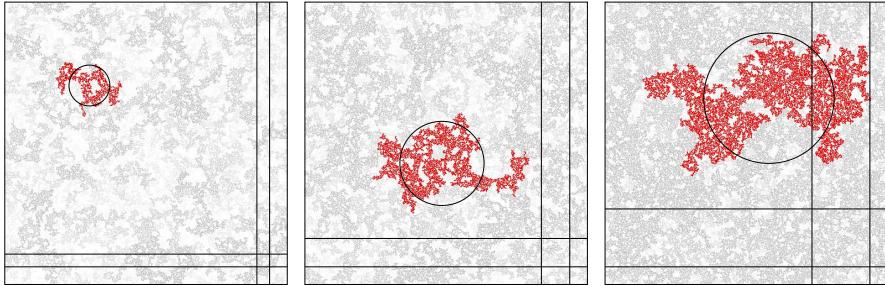


Fig. 5.4 Illustration of the largest cluster in 512×512 systems for $p = 0.55$, $p = 0.57$, and $p = 0.59$. The circles illustrate the radius of gyration of the largest cluster, and the boxes show the size of the average radius of gyration, $R = \langle R_s \rangle$. We observe that both lengths increase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

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

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

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:

$$S(p) = \left\langle \sum_j g(r_{ij}; p) \right\rangle = \sum_j g(r_{ij}; p) \rangle_i , \quad (5.25)$$

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

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

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

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

$$g(r) = r^0 f(r/\xi) , \quad (5.27)$$

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 = measurements.label(m)` command. We write a subroutine `perccorrfunc` to find the correlation function for a given lattice lw , and then we use this function to find the correlation function for several values of p and for several values of L :

```
from pylab import *
from scipy.ndimage import measurements
from numba import jit
@jit
def perccorrfunc(m,lw):
    nx = shape(lw)[0]
    ny = shape(lw)[1]
    L = max([nx,ny])
    r = arange(2*L) # Positions
    pr = zeros(2*L) # Correlation function
    npr = zeros(2*L) # Nr of elements
    for ix1 in range(nx):
        for iy1 in range(ny):
            lw1 = lw[ix1,iy1]
            if (lw1>0):
                for ix2 in range(nx):
```

```

for iy2 in range(ny):
    lw2 = lw[ix2,iy2]
    if (lw2>0):
        dx = (ix2-ix1)
        dy = (iy2-iy1)
        rr = hypot(dx,dy)
        nr = int(ceil(rr)+1) # Corresponding box
        pr[nr] = pr[nr] + (lw1==lw2)
        npr[nr] = npr[nr] + 1
pr = pr/npr
return r,pr
# Calculate correlation function
M = 10 # Nr of samples
L = 200 # System size
pp = [0.5,0.52,0.54,0.55,0.56] # p-value
lenpp = len(pp)
pr = zeros((2*L,lenpp),float)
rr = zeros((2*L,lenpp),float)
for i in range(M):
    print("i = ",i)
    z = rand(L,L)
    for ip in range(lenpp):
        p = pp[ip]
        m = z<p
        lw, num = measurements.label(m)
        r,g = perccorrfunc(m,lw)
        pr[:,ip] = pr[:,ip] + g
        rr[:,ip] = rr[:,ip] + r
pr = pr/M
r = r/M
# Plot data - linearly binned
for ip in range(lenpp):
    loglog(rr[:,ip],pr[:,ip],'.',label="p="+str(pp[ip]))
legend()

```

Fig. 5.5 shows the resulting plots of the correlation function $g(r; p)$ for various values of p for an $L = 200$ system. First, we notice that the scaling is rather poor. We will understand this as we develop a theory for $g(r; p)$ below. The plot shows that there is indeed a cross-over length ξ , beyond which the correlation function falls rapidly to zero. And there appears to be a scaling regime for $r > \xi$ where the correlation function is approximately a power-law, although it is unclear how wide that scaling regime is in this plot. The plot suggests the following functional form

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

where the cross-over function $f(u)$ falls rapidly 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 .

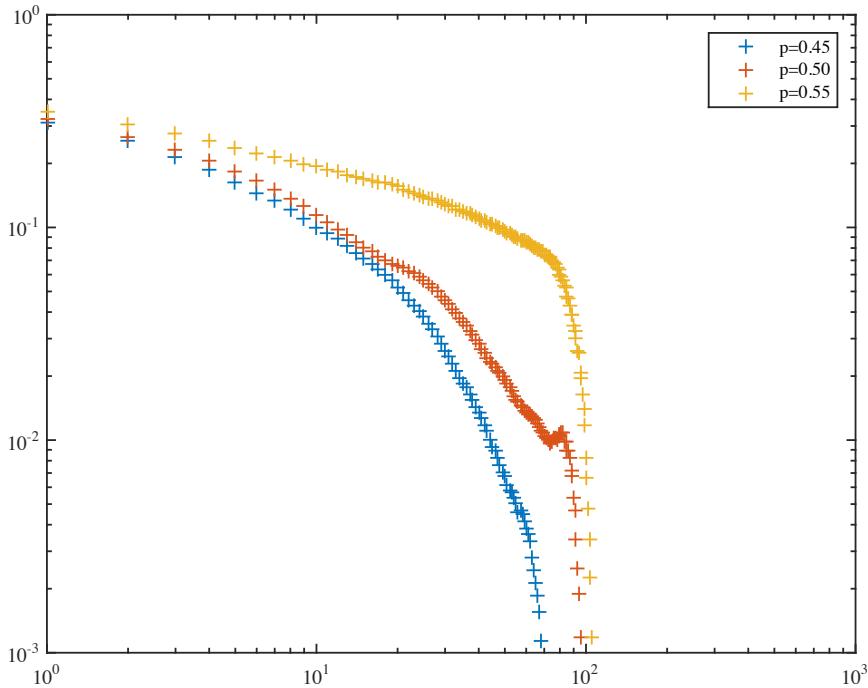


Fig. 5.5 A plot of $g(r; p)$ as a function of r for various values of p . The function approaches a power-law behavior $g(r) \propto r^x$ when p approaches p_c .

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

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

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 $p = p_c$:

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

where η is a positive number, ranging 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 $g(r, p)$:

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

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:

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

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

$$S = \sum_j g(r; p) = \int g(r; p) dr. \quad (5.33)$$

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 .

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

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

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

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

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

Consequently, we now know the behavior of ξ :

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

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 :

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

and we know that the average radius of gyration behaves as

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

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:

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

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:

$$\xi \propto R \propto R_{s_\xi} \propto |p - p_c|^{-\nu} . \quad (5.42)$$

We will typically only use the symbol ξ for this characteristic length of the system, and the exponent ν characterizes how ξ diverges as p approaches p_c :

Correlation length

The correlation length ξ scales as

$$\xi \propto |p - p_c|^{-\nu}, \quad (5.43)$$

when $p \rightarrow p_c$. The exponent $\nu = -1/(\sigma D) = -\gamma/(2 - \eta)$. For $d = 2$, $\nu = 4/3$.

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. 5.6 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

$$\xi_-(p_c - p)^{-\nu} = \xi \simeq L, \quad (5.44)$$

and therefore that

$$(p_c - p) \simeq (L/\xi_-)^{-(1/\nu)}, \quad (5.45)$$

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 :

$$|p - p_c| = cL^{-1/\nu}. \quad (5.46)$$

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 $p > p_c$.

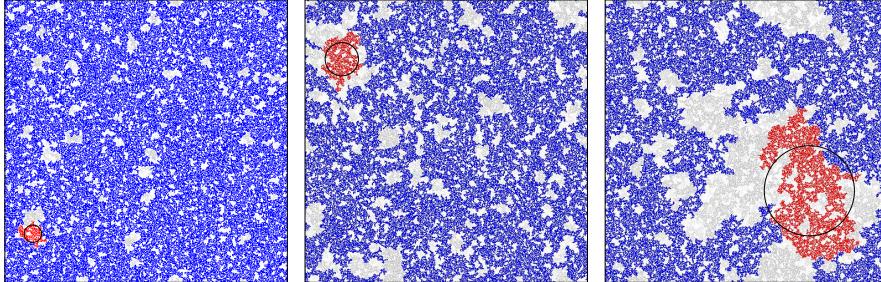


Fig. 5.6 Illustration of the largest cluster in 512×512 systems with $p > p_c$, for $p = 0.593$, $p = 0.596$, and $p = 0.610$. The circles illustrate the radius of gyration of the largest cluster. We observe that the radius of gyration increases as p approaches p_c .

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:

$$P(p, L) = \frac{M(p, L)}{L^d} \propto L^D / L^d \propto L^{D-d}. \quad (5.47)$$

Because we know that $P(p, L) \rightarrow 0$ when $L \rightarrow \infty$, 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

$$M \propto L^D , \quad (5.48)$$

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

$$\rho \propto \frac{M}{L^d} \propto L^{D-d} , \quad (5.49)$$

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

$$M = hL^2 , \quad (5.50)$$

which implies that the density of the sheet is

$$\rho = \frac{hL^2}{L^3} = hL^{-1} . \quad (5.51)$$

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 :

$$M(p, L) \propto L^D \text{ when } L \ll \xi . \quad (5.52)$$

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

$$M(p, L) \propto (\xi^D)(L/\xi)^d \propto \xi^{D-d} L^d . \quad (5.53)$$

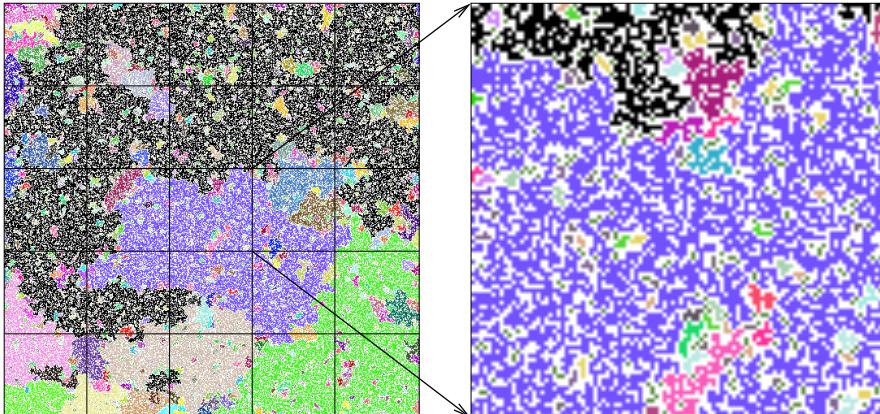


Fig. 5.7 Illustration of the spanning cluster in a 512×512 system at $p = 0.595 > p_c$. In this case, the correlation length is $\xi = 102$. The system is divided into regions of size ξ . Each such region has a mass $M(p, \xi) \propto \xi^D$, and there are $(L/\xi)^d \simeq 25$ such regions in the system.

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

$$M(p, L) \propto \begin{cases} L^D & L \ll \xi \\ \xi^{D-d} L^d & L \gg \xi \end{cases} . \quad (5.54)$$

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

$$M(p, L) = L^D Y\left(\frac{L}{\xi}\right), \quad (5.55)$$

where

$$Y(u) = \begin{cases} \text{constant} & u \ll 1 \\ u^{d-D} & u \gg 1 \end{cases} . \quad (5.56)$$

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

$$M(p', L') = (L')^D Y(L'/\xi') = (L/b)^D Y(L/\xi) = b^{-D} M(p, L), \quad (5.57)$$

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

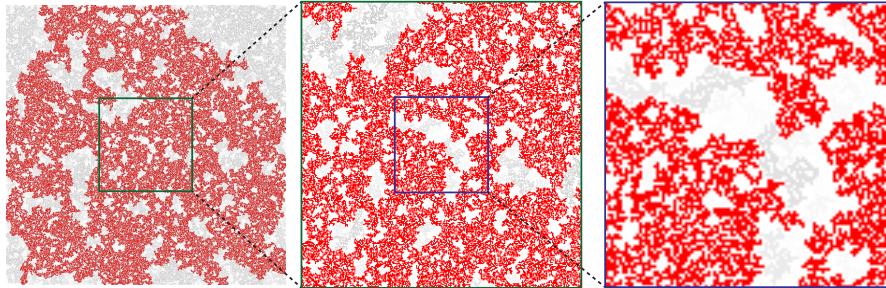


Fig. 5.8 Illustrations of the spanning cluster (shown in red), and the other clusters (shown in gray) at $p = p_c$ in a $L = 900$ site percolation system. **a** The 900×900 system. **b** The central 300×300 , and part. **c** The central 100×100 . Each step represents a rescaling by a factor 3. However, 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.

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:

$$M(L) = b^{-D} M(bL) , \quad (5.58)$$

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 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

$$L' = 2L . \quad (5.59)$$

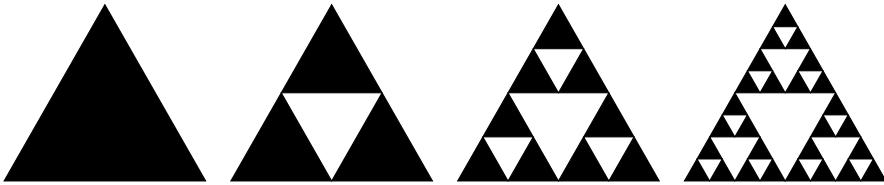


Fig. 5.9 Illustration of three generations of the Sierpinski gasket starting from an equilateral triangle.

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.

$$M' = 3M . \quad (5.60)$$

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

$$M(2L) = 3M(L) , \quad (5.61)$$

or, equivalently,

$$M(L) = 3^{-1} M(2L) . \quad (5.62)$$

If we compare this with the general relation,

$$M(L) = b^{-D} M(bL) , \quad (5.63)$$

we see that

$$2^{-D} = 3^{-1} , \quad (5.64)$$

giving

$$D = \frac{\ln 3}{\ln 2} . \quad (5.65)$$

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 directions to get back to the original cube, but the mass must be rescaled by a factor 8. We will therefore find the dimension from

$$D = \frac{\ln 8}{\ln 2} = 3 , \quad (5.66)$$

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

$$N(\delta) = \left(\frac{L}{\delta}\right)^d, \quad (5.67)$$

and for a fractal structure we expect

$$N(\delta) = \left(\frac{L}{\delta}\right)^D, \quad (5.68)$$

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

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 $L = 2^k$, $k = 4, \dots, 11$.
- 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 .