

Chapter 13

Percolation

©2001 by Harvey Gould, Jan Tobochnik, and Wolfgang Christian
24 April 2001

We introduce several concepts associated with critical phenomena in the context of percolation.

13.1 Introduction

Our discussion of percolation and geometrical phase transitions requires little background in physics, e.g., no classical or quantum mechanics and little statistical physics. All that is required is some understanding of geometry and probability. Much of the appeal of percolation models is their game-like aspects and their intuitive simplicity. Moreover, these models serve as an excellent introduction to discrete computer models and the importance of graphical analysis. On the other hand, if you have a background in physics, this chapter will be more meaningful and can serve as an introduction to phase transitions and to important ideas such as scaling relations, critical exponents, and the renormalization group.

Although the term “percolation” might be familiar to you in the context of the brewing of coffee, our use of the term has little to do with it. Instead, we consider another example from the kitchen and imagine a large metal sheet on which we randomly place drops of cookie dough. Assume that each drop of cookie dough can spread while the cookies are baking in an oven. If two cookies touch, they coalesce to form one cookie. If we are not careful, we might find a very large cookie that spans from one edge of the sheet to the opposite edge (see Fig. 13.1). If such a spanning cookie exists, we say that there has been a *percolation transition*. If such a cookie does not exist, the cookie sheet is below the percolation threshold. (There is no percolation transition for ground coffee, because the water dissolves some of the ground coffee beans and flows, regardless of the density of the ground coffee.)

Let us make the cookie example more abstract to make the concept of percolation more clear. We represent the cookie sheet by a lattice where each site can be in one of two states, occupied (by a cookie) or empty. Each site is occupied independently of its neighbors with probability p . This model of percolation is called *site* percolation. The occupied sites either are isolated or form

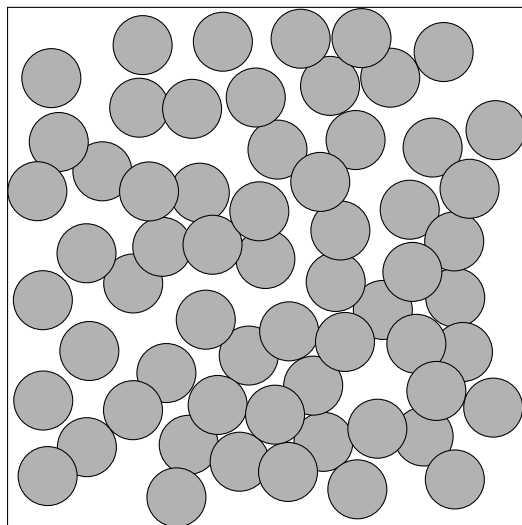


Figure 13.1: Cookies (circles) placed at random on a large sheet. Note that there is a path of overlapping circles that connects the bottom and top edges of the cookie sheet. If such a path exists, we say that the cookies “percolate” the lattice or that there is a “spanning path.” See Problem 13.4d for a discussion of the algorithm used to generate this configuration.

groups of nearest neighbors. We define a *cluster* as a group of occupied nearest neighbor lattice sites (see Fig. 13.2).

An easy way to study percolation uses the random number generator on a calculator. The procedure is to generate a random number r in the unit interval $0 < r \leq 1$ for each site in the lattice. A site is occupied if its random number satisfies the condition $r \leq p$. If p is small, we expect that only small isolated clusters will be present (see Fig. 13.3a). If p is near unity, we expect that most of the lattice will be occupied, and the occupied sites will form a large cluster that extends from one end of the lattice to the other (see Fig. 13.3c). Such a cluster is said to be a *spanning cluster*. Because there is no spanning cluster for small p and there is a spanning cluster for p near unity, there must be an intermediate value of p at which a spanning cluster first exists (see Fig. 13.3b). We shall see that in the limit of an infinite lattice, there exists a well defined threshold probability p_c such that:

For $p < p_c$, no spanning cluster exists and all clusters are finite.

For $p \geq p_c$, one spanning cluster exists.

We emphasize that the defining characteristic of percolation is *connectedness*. Because the connectedness exhibits a qualitative change at a well defined value of a continuous parameter, we shall see that the transition from a state with no spanning cluster to a state with one spanning cluster is a type of *phase transition*.

Our real interest is not in large cookies or in abstract models, but in the applications of percolation. An example of the application of percolation is to the electrical conductivity of

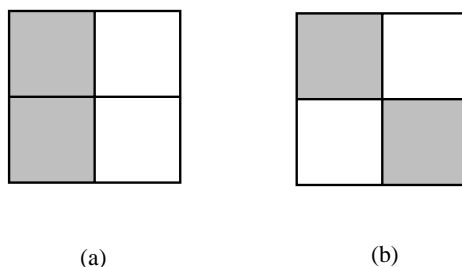


Figure 13.2: Example of a site percolation cluster on a square lattice of linear dimension $L = 2$. The two nearest neighbor occupied sites (shaded) in (a) are part of the same cluster; the two occupied sites in (b) are not nearest neighbor sites and do not belong to the same cluster.

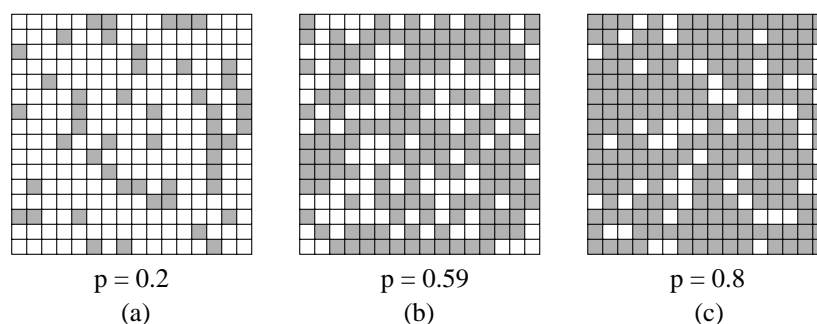


Figure 13.3: Examples of site percolation clusters on a square lattice of linear dimension $L = 16$ for $p = 0.2$, 0.59 , and 0.8 . On the average, the fraction of occupied sites (shaded squares) is equal to p . Note that in this example, there exists a cluster that “spans” the lattice horizontally and vertically for $p = 0.59$.

composite systems made of a mixture of metallic and insulating materials. An easy way to make such a system is to place a mixture of small plastic and metallic spheres of equal size into a container (see Fitzpatrick et al.). Care must be taken to pack the spheres at random. If the metallic domains constitute a small fraction of the volume of the system, electricity cannot be conducted and the composite system is an insulator. However, if the metallic domains comprise a sufficiently large fraction of the container, electricity can flow from one domain to another and the composite system is a conductor. The description of the conduction of electricity through composite materials can be made more precise by introducing the parameter ϕ , the volume fraction of the container that consists of metallic spheres. The transition between the two types of behavior (nonconducting and conducting) occurs abruptly as ϕ (the analog of p) is increased and is associated with the nonexistence or existence of a *connected path* of metallic spheres. More realistic composite systems are discussed in Zallen’s book.

Percolation phenomena also can be observed with a piece of chicken wire or wire mesh. Watson and Leath measured the electrical conductivity of a large piece of uniform steel-wire screen mesh

as a function of the fraction of the nodes that are present. The coordinates of the nodes to be removed were given by a random number generator. The measured electrical conductivity is a rapidly decreasing function of the fraction of nodes p still present and vanishes below a critical threshold. A related conductivity measurement on a sheet of conducting paper with random “holes” has been performed (see Mehr et al.).

The applications of percolation phenomena go beyond metal-insulator transitions and the conductivity of chicken wire, and include the spread of disease in a population, the behavior of magnets diluted by nonmagnetic impurities, the flow of oil through porous rock, and the characterization of gels. We concentrate on understanding several simple models of percolation that have an intuitive appeal of their own. Many of the applications of percolation phenomena are discussed in the references.

13.2 The Percolation Threshold

Because it is not convenient to generate many percolation configurations using a calculator, we develop a simple program to do so. Consider a square lattice of linear dimension L and unit lattice spacing, and associate a random number between zero and one with each site in the lattice. A site is occupied if its random number is less than p . **Program site**, listed below, generates site percolation configurations and shows the occupied sites as filled circles of diameter unity. The program uses the **FLOOD** statement in True BASIC so that if the user clicks on an occupied site, all the sites that are connected to it are changed to the same color and the clusters can be identified visually. A click of the mouse outside the lattice increases p by the desired amount. The array **rsite** stores the random number associated with each lattice site.

The percolation threshold p_c is defined as the probability p at which a spanning cluster first appears in an infinite lattice. However, for the finite lattices of linear dimension L that we can simulate on a computer, there is a nonzero probability of a spanning cluster connecting one side of the lattice to the opposite side for any value of p . For small p , this probability is order p^L (see Fig. 13.4). This probability goes to zero as L becomes large, and hence for small p and sufficiently large L , only finite clusters exist. For a finite lattice, the definition of spanning is arbitrary. For example, we can define a connected path as one that (i) spans the lattice either horizontally or vertically; (ii) spans the lattice in a fixed direction, e.g., vertically; or (iii) spans the lattice both horizontally and vertically. In addition, the criteria for defining $p_c(L)$ for a finite lattice are somewhat arbitrary. One possibility is to define $p_c(L)$ as the average value of p at which a spanning cluster first appears. Another possibility is to define $p_c(L)$ as the value of p for which half of the configurations generated at random span the lattice. These criteria should lead to the same extrapolated value for p_c in the limit $L \rightarrow \infty$. In Problem 13.1 we will find an estimated value for $p_c(L)$ that is accurate to about 10%. A more sophisticated analysis discussed in Project 13.1413.35 allows us to extrapolate our results for $p_c(L)$ to $L \rightarrow \infty$.

Problem 13.1. Site percolation on the square lattice

- a. Use **Program site** to generate random site configurations on a square lattice. Estimate $p_c(L)$ by finding the average value of p at which a spanning cluster is first attained. Choose $L = 4$ and begin at a value of p for which a spanning cluster is unlikely to be present. Then increase p in increments of `delta_p = 0.01` until you find a spanning cluster. Record the value of p

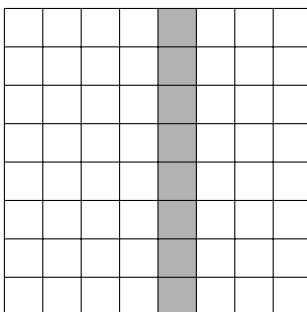


Figure 13.4: An example of a spanning cluster of probability p^L on a $L = 8$ lattice. How many other ways are there of realizing a spanning cluster of L sites?

at which spanning first occurs for each spanning criteria. Repeat this process for a total of ten configurations and find the average value of $p_c(L)$. (Remember that each configuration corresponds to a different set of random numbers.) Are your results for $p_c(L)$ using the three spanning criteria consistent with your expectations?

- b. Repeat part (a) for $L = 16$ and 32 . Is $p_c(L)$ better defined for larger L , that is, are the values of $p_c(L)$ spread over a smaller range of values? How quickly can you visually determine the existence of a spanning cluster? Describe your visual “algorithm” for determining if a spanning cluster exists.

The value of p_c depends on the symmetry of the lattice and on its dimension. In addition to the square lattice, the most common two-dimensional lattice is the triangular lattice. As discussed in Chapter 8, the essential difference between the square and triangular lattices is in the number of nearest neighbors.

**Problem 13.2.* Site percolation on the triangular lattice

Modify `Program site` to simulate random site percolation on a triangular lattice. Assume that a connected path connects the top and bottom sides of the lattice (see Fig. 13.5). Do you expect p_c for the triangular lattice to be smaller or larger than the value of p_c for the square lattice? Estimate $p_c(L)$ for $L = 4, 16$, and 32 . Are your results for p_c consistent with your expectations?

In *bond* percolation each lattice site is occupied, and only a fraction of the sites have connections or bonds between them and their nearest neighbor sites (see Fig. 13.6). Each bond either is occupied with probability p or not occupied with probability $1 - p$. A cluster is a group of sites connected by occupied bonds. The wire mesh described in Section 13.1 is an example of bond percolation if we imagine cutting the bonds between the nodes rather than removing the nodes themselves. An application of bond percolation to the description of gelation is discussed in Problem 13.3.

**Problem 13.3.* Bond percolation on a square lattice Suppose that all the lattice sites of a square lattice are occupied by monomers, each with functionality four, i.e., each monomer can react to form a maximum of four bonds. This model is equivalent to bond percolation on a square lattice. Also assume that the presence or absence of a bond between a given pair of monomers is random

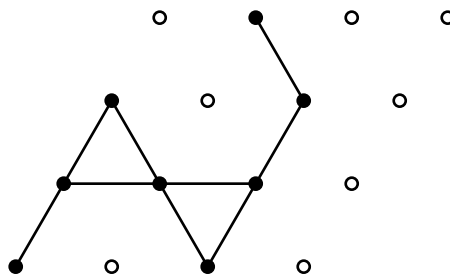


Figure 13.5: Example of a spanning cluster on a $L = 4$ triangular lattice. The bonds between the occupied sites are drawn to clarify the symmetry of the lattice.



Figure 13.6: Two examples of bond clusters. The occupied bonds are shown as bold lines.

and is characterized by a probability p . For small p , the system consists of only finite polymers (groups of monomers) and the system is in the *sol* phase. For some threshold value p_c , there will be a single polymer that is infinite in spatial extent. We say that for $p \geq p_c$, the system is in the *gel* phase. How does a bowl of jello, an example of a gel phase, differ from a bowl of broth? Write a program to simulate bond percolation on a square lattice and determine the bond percolation threshold. Are your results consistent with the exact result, $p_c = 1/2$?

We also can consider *continuum* percolation models. For example, we can place disks at random into a two-dimensional box. Two disks are in the same cluster if they touch or overlap. A typical continuum (off-lattice) percolation configuration is depicted in Fig. 13.7. One quantity of interest is the quantity ϕ , the fraction of the area (volume in three dimensions) in the system that is covered by disks. In the limit of an infinite size box, it can be shown that

$$\phi = 1 - e^{-\rho\pi r^2}, \quad (13.1)$$

where ρ is the number of disks per unit area, and r is the radius of a disk (see Xia and Thorpe). Equation (13.1) is significantly inaccurate for small boxes because disks located near the edge of the box might have a significant fraction of their area located outside of the box. `Program site` can be modified to simulate continuum percolation. Instead of placing the disks on regular lattice sites, place them at random within a square box of area L^2 . The relevant parameter is the density ρ , the number of disks per unit area, instead of the probability p . Because the disks overlap, it is convenient to replace the `BOX SHOW` statement in `Program site` with

```
BOX SHOW occup$ at x(i)-0.5,y(i)-0.5 using "or"
```

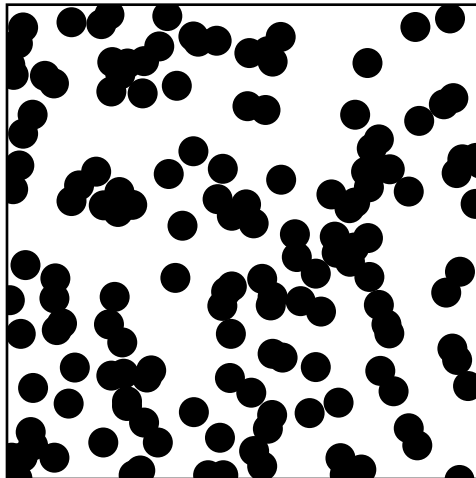


Figure 13.7: A model of continuum (off-lattice) percolation realized by placing disks of unit diameter at random into a square box of linear dimension L . If we concentrate on the voids between the disks rather than the disks, then this model of continuum percolation is known as the Swiss cheese model.

where the arrays $x(i)$ and $y(i)$ are used to store the disk positions of disk i . It also is a good idea to set the background color to red (not black or white).

Problem 13.4. Continuum percolation

- a. For site percolation, we can define ϕ as the area fraction covered by the disks that are placed on the sites as in `Program site`. Convince yourself that $\phi_c = (\pi/4)p_c$ (for disks of unit diameter and unit lattice spacing). It is easy to do a Monte Carlo calculation of the area covered by the disks to confirm this result. (Choose points at random in the box and calculate the fraction of points within any disk.)
- b. Modify `Program site` to simulate continuum percolation as discussed in the text. Estimate the value of the percolation threshold ρ_c . Given this value of ρ_c , use a Monte Carlo method to estimate the corresponding area fraction ϕ_c , and compare the value of ϕ_c for site and continuum percolation. Explain why you might expect ϕ_c to be bigger for continuum percolation than for site percolation. Compare your direct Monte Carlo estimate of ϕ_c with the indirect value of ϕ_c obtained from (13.1) using the value of ρ_c . Explain any discrepancy.
- c. Consider the simple model of the cookie problem discussed in Section 13.1. Write a program that places disks at random into a square box and chooses their diameter randomly between 0 and 1. Estimate the value of ρ_c at which a spanning cluster first appears. How is the value of

ρ_c changed from your estimate found in part (b)? Is your value for ϕ_c more or less than what was found in part (b)?

- d. A more realistic model of the cookie problem is to place disks with unit diameter at random into a square box with the constraint that the disks do not overlap. Continue to add disks until the probability of placing an additional disk becomes less than 1%, i.e., when one hundred successive attempts at adding a disk are not successful. Then increase the diameters of all the disks at a constant rate (in analogy to the baking of the cookies) until a spanning cluster is attained. How does ϕ_c for this model compare with ϕ_c found in part (c)?
- e. A continuum model that is applicable to random porous media is known as the *Swiss cheese* model. In this model the relevant quantity (the cheese) is the space between the disks. For the Swiss cheese model in two dimensions, the cheese area fraction at the percolation threshold, $\tilde{\phi}_c$, is given by $\tilde{\phi}_c = 1 - \phi_c$, where ϕ_c is the disk area fraction at the threshold of the disks. Do you think such a relation holds in three dimensions (see Project 13.15)? Imagine that the disks are conductors and that the cheese is an insulator and let $\sigma(\phi)$ denote the conductivity of this system. Alternatively, we can imagine that the cheese is a conductor and the disks are insulators and define a conductivity $\sigma(\tilde{\phi})$. Do you think that $\sigma(\phi) = \sigma(\tilde{\phi})$ when $\phi = \tilde{\phi}$? This question is investigated in Project 13.15.

Our discussion of percolation has emphasized the existence of the percolation threshold p_c and the appearance of a spanning path or cluster for $p \geq p_c$. Another quantity that characterizes percolation is $P_\infty(p)$, the probability that an occupied site belongs to the spanning cluster. P_∞ is defined as

$$P_\infty = \frac{\text{number of sites in the spanning cluster}}{\text{total number of occupied sites}}. \quad (13.2)$$

As an example, $P_\infty(p = 0.59) = 140/154$ for the single configuration shown in Fig. 13.3b. A realistic calculation of P_∞ involves an average over many configurations for a given value of p . For an infinite lattice, $P_\infty(p) = 0$ for $p < p_c$ and $P_\infty(p) = 1$ for $p = 1$. Between p_c and 1, $P_\infty(p)$ increases monotonically.

More information can be obtained from the *mean cluster size distribution* $n_s(p)$ defined by

$$n_s(p) = \frac{\text{average number of clusters of size } s}{\text{total number of lattice sites}}. \quad (13.3)$$

For $p \geq p_c$, the spanning cluster is excluded from n_s . (For historical reasons, the *size* of a cluster refers to the *number* of sites in the cluster rather than to its spatial extent.) As an example, we see from Fig. 13.3a that $n_s(1) = 20$, $n_s(2) = 4$, $n_s(3) = 5$, and $n_s(7) = 1$ for $p = 0.2$ and is zero otherwise. Because $N \sum_s s n_s$ is the total number of occupied sites (N is the total number of lattice sites), and $N s n_s$ is the number of occupied sites in clusters of size s , the quantity

$$w_s = \frac{s n_s}{\sum_s s n_s} \quad (13.4)$$

is the probability that an occupied site chosen at random is part of an s -site cluster. Hence, the *mean cluster size* S is given by

$$S = \sum_s s w_s = \frac{\sum_s s^2 n_s}{\sum_s s n_s}. \quad (13.5)$$

The sum in (13.5) is over the finite clusters only. As an example, the weights corresponding to the clusters in Fig. 13.3a are $w_s(1) = 20/50$, $w_s(2) = 8/50$, $w_s(3) = 15/50$, and $w_s(7) = 7/50$, and hence $S = 130/50$.

Problem 13.5. Qualitative behavior of $n_s(p)$, $S(p)$, and $P_\infty(p)$

- Visually determine the cluster size distribution $n_s(p)$ for a square lattice with $L = 16$ and $p = 0.4$, $p = p_c$, and $p = 0.8$. Take $p_c = 0.5927$. One way to help you identify the clusters is to modify `Program site` so that you can use the `FLOOD` statement to show different clusters in different colors. Consider at least five configurations for each value of p and average $n_s(p)$ over the configurations. Because the lattice is finite, more consistent results can be obtained by discarding those configurations that have a spanning cluster for $p < p_c$ and those that do not have a spanning cluster for $p \geq p_c$. For each value of p , plot n_s as a function of s and describe the observed s -dependence. Does n_s decrease more rapidly with s for $p = p_c$ or for $p \neq p_c$?
- Use the same configurations considered in part (a) to compute the mean cluster size S as a function of p . Remember that for $p > p_c$, the spanning cluster is excluded.
- Similarly, compute $P_\infty(p)$ for $L = 16$, and for various values of $p \geq p_c$. Plot $P(p)$ as a function of p and discuss its qualitative behavior.
- Verify that $\sum_s s n_s(p) = p$ for $p < p_c$ and explain this relation. How is this relation modified for $p \geq p_c$?

13.3 Cluster Labeling

Your visual algorithm for determining the existence of a connected path probably is very sophisticated. Although using the `FLOOD` command in True BASIC helps us to automate the process for a single configuration, we need to average over many configurations to obtain quantitative results. Hence, we need to develop an algorithm that finds the clusters. In the following, we will find that this task is not easy. The difficulty is that the assignment of a site to a cluster is a *global* rather than a *local* property of the site.

We consider the multiple cluster labeling method of Hoshen and Kopelman. The algorithm can best be described by an example. Consider the configuration shown in Fig. 13.8. We define an array `site` to store the occupancy of the sites; an occupied site initially is assigned the value -1 and an unoccupied site is assigned the value 0 . We assign cluster labels to sites beginning at the lower left corner and continue from left to right. Because `site(1,1)` is occupied, we assign to it cluster label 1. The next site is empty, and hence is not labeled. The next occupied site in the first row is `site(3,1)`. Because its left neighbor is unoccupied, we assign to it the next available cluster label, label 2. The assignment of cluster labels to the remainder of the row is straightforward, and we proceed to `site(1,2)` of the second row. Because this site is occupied and its nearest neighbor in the preceding row is labeled 1, we assign label 1 to `site(1,2)`. We continue from left to right along the second row checking the occupancy of each site. If a site is occupied, we check the occupancy of its nearest neighbors in the previous row and column. If neither neighbor is occupied, we assign the next available cluster label. If only one nearest neighbor site is occupied, the site is assigned the label of its occupied neighbor. For example, `site(2,2)` is assigned label 1 since its occupied neighbor, `site(1,2)` has label 1.

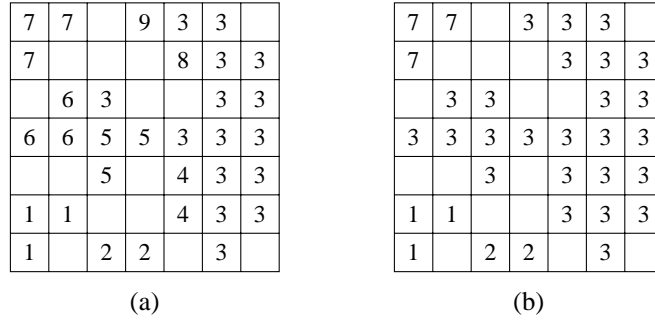


Figure 13.8: A percolation configuration on a square lattice with $L = 7$. Site coordinates are measured from the origin at the lower left corner (1,1). Part (a) shows the improper cluster labels initially assigned by `Program cluster`, a modified implementation of the Hoshen-Kopelman algorithm. Part (b) shows the proper cluster labels.

The difficulty arises when we come to an occupied site at which two clusters coalesce and cluster labels need to be reassigned. This case first occurs at `site(6,2)` — its two neighbors in the previous row and column have labels 3 and 4, respectively. We define the *proper* cluster label assignment at `site(6,2)` as the smaller of labels 3 and 4. Hence `site(6,2)` is assigned cluster label 3 and label 4 should be reassigned to label 3. It is inefficient to continually relabel the clusters, because there likely will be further reassignments. Hence, we delay the reassignment of cluster labels until the entire lattice is surveyed, and instead, keep track of the connections of the labels through a *label tree*. We introduce an array `np` that distinguishes proper and improper labels and provides their connections. Let us return to the configuration shown in Fig. 13.8 to explain the use of this array. Before we came to `site(6,2)`, labels 1 through 4 were proper labels and we set

$$np(1) = 1, \quad np(2) = 2, \quad np(3) = 3, \quad np(4) = 4. \quad (13.6)$$

At `site(6,2)` where labels 3 and 4 are linked, we set $np(4) = 3$. This reassignment of $np(4)$ tells us that label 4 is improper, and the numerical value of $np(4)$ tells us that label 4 is linked to label 3. Note that the argument `i` of `np(i)` always is greater than or equal to the value of `np(i)`.

This procedure is still not complete. What should we do when we come to a site with two previously labeled neighbors one or both of which are improper? For example, consider `site(5,4)` which has two occupied neighbors with labels 5 and 4. We might be tempted to assign `site(5,4)` the label 4 and set $np(5) = 4$. However, instead of assigning to a site the minimum label of its two neighbors, we should assign to it the minimum of the *proper* labels of the two neighboring sites. In addition, if the two neighboring sites have different proper labels, then we should set `np` of the maximum proper label equal to the minimum proper label. In this example, we have $np(5) = 3$.

The above version of the Hoshen-Kopelman cluster algorithm is implemented in the following subroutines. The arrays `site` and `np` are declared in a main program. After `SUB assign` is called, `site` contains the proper labels for each occupied site. The array `site` is given an extra empty column on the left ($x = 0$) and an extra empty row on the bottom ($y = 0$) so that the first column and row do not have to be treated differently. The following declaration in the main program for the arrays would be appropriate.

```
DIM site(0:64,0:64),np(0:1000)
```

A more efficient and different version of the Hoshen-Kopelman algorithm written in Fortran has been given by Stauffer and Aharony (see references). Although the Hoshen-Kopelman algorithm is the most efficient cluster identification method for two-dimensional systems, it is not obvious that this approach is the most efficient in higher dimensions. Can you think of another method for identifying the clusters?

Problem 13.6. Test of the cluster labeling algorithm

Incorporate `SUB assign` and its associated subroutines into `Program site`. You will need to add array `site` (in addition to array `rsite`) to the original subroutines of `Program site`. Use the following subroutine to show the improper and proper cluster labels and to help you check that the cluster labeling is being done correctly. Choose a particular configuration and explain how the Hoshen-Kopelman algorithm is implemented.

After the clusters are labeled, we can obtain a number of geometrical quantities of interest. Vertical spanning can be checked by seeing if there is a nonzero label in the bottom row equal to a nonzero label in the top row. The following subroutine checks for the existence of a vertical spanning cluster. Note that if `vspan <> 0`, there is a vertically spanning cluster with label `vspan`. Horizontal spanning can be determined by comparing the labels of the left and right columns.

The cluster distribution n_s can be found by computing the number of sites $m(i)$ in cluster i , counting the number of clusters of size s , and normalizing the results by dividing by L^2 , the number of sites. The probability of an occupied site belonging to the spanning cluster P_∞ , can be determined by finding the label of the spanning cluster, `vspan`, and taking the ratio of $m(\text{vspan})$ to the total number of occupied sites. The mean cluster size S of the nonspanning clusters can be computed from the relation

$$S = \frac{\sum_i m^2(i)}{\sum_i m(i)}. \quad (13.7)$$

Convince yourself that (13.7) is equivalent to the definition (13.5) of S . Note that the spanning cluster is not included in the sums in (13.7).

In Problem 13.7 we apply the Hoshen-Kopelman cluster algorithm to a more systematic study of site percolation. In Section 13.4 we use a finite size scaling analysis to estimate the critical exponents.

**Problem 13.7.* Applications of the cluster labeling algorithm

- Compute $F(p) dp$, the probability of *first* spanning an $L \times L$ square lattice for p in the range p to $p + dp$. Write a subroutine to find when the spanning first occurs as p is increased. Do a minimum of 100 configurations for each value of L and plot $F(p)$ as a function of p . Consider $L = 4, 16$, and 32 . How does the shape of $F(p)$ change with increasing L ? At what value of p is $F(p) dp \approx 0.5$ for the various spanning criteria and for each value of L ? Call this value $p_c(L)$. How strongly does $p_c(L)$ depend on L for a given spanning criterion? How strongly does $p_c(L)$ depend on the spanning criterion for fixed L ?
- Compute P_∞ for $p = p_c, p = 0.65, p = 0.75$, and $p = 0.9$ for $L = 4, 16$, and 32 . Do a minimum of 100 configurations for each value of p . Use either the estimated value of $p_c(L)$ determined

- in part (a) or the known value $p_c = 0.5927$ (to four decimal places). What is the qualitative p -dependence of P_∞ ? Is $P_\infty(p = p_c)$ an increasing or decreasing function of L ? (Discard those configurations that do not have a spanning cluster.)
- c. Write a subroutine to compute $n_s(p)$. Consider $p = p_c$ and $p = p_c \pm 0.1$ for $L = 4, 16$, and 32 and average over at least ten configurations. Why is n_s a decreasing function of s ? Does n_s decrease more quickly for $p = p_c$ or for $p \neq p_c$?
- d. Write a subroutine to compute the mean cluster size S for $p = p_c$ and $p = p_c \pm 0.1$ for $L = 4, 16$, and 32 . Average over at least ten configurations. What is the qualitative p -dependence of $S(p)$? How does $S(p = p_c)$ depend on L ? For $p < p_c$, discard the configurations that contain a spanning cluster and for $p > p_c$ discard the configurations that do not have a spanning cluster.

It is convenient to associate a characteristic linear dimension or *connectedness length* $\xi(p)$ with the clusters. One way to do so is to define the radius of gyration R_s of a single cluster of s particles as

$$R_s^2 = \frac{1}{s} \sum_{i=1}^s (\mathbf{r}_i - \bar{\mathbf{r}})^2, \quad (13.8)$$

where

$$\bar{\mathbf{r}} = \frac{1}{s} \sum_{i=1}^s \mathbf{r}_i, \quad (13.9)$$

and \mathbf{r}_i is the position of the i th site in the same cluster. The quantity $\bar{\mathbf{r}}$ is the familiar definition of the center of mass of the cluster. From (13.8), we see that R_s is the root mean square radius of the cluster measured from its center of mass. The connectedness length ξ can be defined as an average over the radii of gyration of all the finite clusters. The choice of the appropriate average is neither unique nor obvious. To find an expression for ξ , consider a site on a cluster of s sites. The site is connected to $s - 1$ other sites and the average square distance to these sites is R_s^2 (see Problem 13.8a). The probability that a site belongs to a cluster of size s is $w_s = sn_s$. These considerations suggest that one definition of ξ is

$$\xi^2 = \frac{\sum_s (s-1) w_s R_s^2}{\sum_s (s-1) w_s}. \quad (13.10)$$

To simplify the expression for ξ , we write s instead of $s - 1$ and let $w_s = sn_s$:

$$\xi^2 = \frac{\sum_s s^2 n_s R_s^2}{\sum_s s^2 n_s} \quad (13.11)$$

As before, the sum in (13.11) is over the nonspanning clusters only. The definition (13.11) and a simpler definition of $\xi(p)$ are explored in Problem 13.8.

**Problem 13.8.* The connectedness length

- a. An alternative way of defining the radius of gyration of a cluster of s sites is as follows:

$$R_s^2 = \frac{1}{2s^2} \sum_{i,j} (\mathbf{r}_i - \mathbf{r}_j)^2. \quad (13.12)$$

The sum (13.12) is over all pairs of particles in the cluster. What is the physical interpretation of (13.12)? Show that the form (13.12) is equivalent to (13.8). Which expression for R_s^2 is easier to compute?

- b. Write a subroutine to compute R_s for a nonspanning cluster of size s . Choose $L = 64$ and $p = 0.57$ and compute ξ using the definition (13.11). Average over at least five configurations. Does the largest nonspanning cluster make the dominant contribution to the sum?
- c. Compute the p -dependence of either $\xi(p)$ using (13.11) or associate ξ with the radius of gyration of the largest nonspanning cluster. Choose $L = 64$ and consider at least 50 configurations for each value of p . Consider values of p in steps of 0.01 in the interval $[p_c - 0.05, p_c - 0.01]$ and $[p_c + 0.01, p_c + 0.05]$ with $p_c = 0.5927$. For $p < p_c$ discard those configurations that contain a spanning cluster, and for $p > p_c$ discard those configurations that do not have a spanning cluster. Plot $\xi(p)$ and discuss its qualitative dependence on p . Is $\xi(p)$ a monotonically increasing or decreasing function of p for $p < p_c$ and $p > p_c$?

13.4 Critical Exponents and Finite Size Scaling

We are familiar with different phases of matter from our everyday experience. The most familiar example is water which can exist as a vapor, liquid, or solid. It is well known that water changes from one phase to another at a well defined temperature and pressure, e.g., the transition from ice to liquid water occurs at 0°C at atmospheric pressure. Such a change of phase is an example of a *thermodynamic phase transition*. Most substances also exhibit a *critical point*; that is, beyond a particular temperature and pressure, it is no longer possible to distinguish between the liquid and gaseous phases and the phase boundary terminates.

Another example of a critical point occurs in magnetic systems at the Curie temperature T_c and zero magnetic field. We know that at low temperatures some substances exhibit ferromagnetism, a spontaneous magnetization in the absence of an external magnetic field. If we raise the temperature of a ferromagnet, the spontaneous magnetization decreases and vanishes continuously at a critical temperature T_c . For $T > T_c$, the system is a paramagnet. In Chapter ?? we use Monte Carlo methods to investigate the behavior of a magnetic system near the magnetic critical point.

In the following, we will find that the properties of the *geometrical* phase transition in the percolation problem are qualitatively similar to the properties of thermodynamic phase transitions. Hence, a discussion of the percolation phase transition also can serve as an introduction to thermodynamic phase transitions. We will see that in the vicinity of a phase transition, the qualitative behavior of the system is governed by the appearance of long-range correlations.

We have seen that the essential physics near the percolation threshold is associated with the existence of large but finite clusters. For example, for $p \neq p_c$, we found in Problem 13.7c that n_s decays rapidly with s . However for $p = p_c$, the s -dependence of n_s is qualitatively different, and n_s decreases much more slowly. This different behavior of n_s at $p = p_c$ is due to the presence of clusters of all length scales, e.g., the “infinite” spanning cluster and the finite clusters of all sizes. We also found (see Problem 13.8) that $\xi(p)$ is finite, and an increasing function of p for $p < p_c$ and a decreasing function of p for $p > p_c$ (see Fig. 13.9). Moreover, we know that $\xi(p = p_c)$ is approximately equal to L and hence diverges as $L \rightarrow \infty$. This qualitative behavior of $\xi(p)$ is consistent with our physical picture of the clusters, that is, as p approaches p_c , the probability that

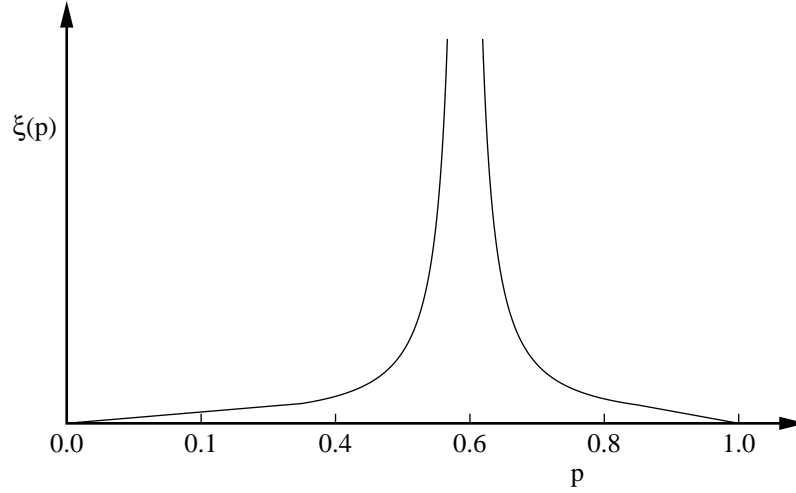


Figure 13.9: Qualitative p -dependence of the connectedness length $\xi(p)$. The divergent behavior of $\xi(p)$ in the critical region is characterized by the exponent ν (see (13.13)).

two occupied sites are in the same cluster increases. These qualitative considerations lead us to conjecture that in the limit $L \rightarrow \infty$, $\xi(p)$ grows rapidly in the *critical region*, $|p - p_c| \ll 1$. We can describe the divergence of $\xi(p)$ more quantitatively by introducing the *critical exponent* ν , defined by the relation

$$\xi(p) \sim |p - p_c|^{-\nu}. \quad (13.13)$$

Of course, there is no *a priori* reason why the divergence of $\xi(p)$ can be characterized by a simple power law. Note that ν is assumed to be the same above and below p_c .

How do the other quantities that we have considered behave in the critical region in the limit $L \rightarrow \infty$? According to the definition (13.2) of P_∞ , $P_\infty = 0$ for $p < p_c$ and is an increasing function of p for $p > p_c$. We conjecture that in the critical region, the increase of P_∞ with increasing p is characterized by an exponent β defined by the relation

$$P_\infty(p) \sim (p - p_c)^\beta. \quad (13.14)$$

Note that P_∞ is assumed to approach zero continuously from above p_c . We say that the percolation transition is a *continuous* phase transition. In the language of critical phenomena, P_∞ is an example of an *order parameter* of the system. An order parameter is a quantity that measures the “order” of a system, and is nonzero in the ordered phase and zero in the disordered phase. In the percolation context, we consider the phase with a spanning cluster to be ordered and the phase without a spanning cluster to be disordered.

The mean number of sites in the finite clusters, $S(p)$, also diverges in the critical region. Its critical behavior is written as

$$S(p) \sim |p - p_c|^{-\gamma}, \quad (13.15)$$

which defines the critical exponent γ . The common critical exponents for percolation are summarized in Table 13.1. For comparison, the analogous critical exponents of a magnetic critical point also are shown.

Quantity	Functional form	Exponent	$d = 2$	$d = 3$
Percolation				
order parameter	$P_\infty \sim (p - p_c)^\beta$	β	5/36	0.4
mean size of finite clusters	$S(p) \sim p - p_c ^{-\gamma}$	γ	43/18	1.8
connectedness length	$\xi(p) \sim p - p_c ^{-\nu}$	ν	4/3	0.9
cluster numbers	$n_s \sim s^{-\tau} \quad p = p_c$	τ	187/91	2.2
Ising model				
order parameter	$M(T) \sim (T_c - T)^\beta$	β	1/8	0.32
susceptibility	$\chi(T) \sim T - T_c ^{-\gamma}$	γ	7/4	1.24
correlation length	$\xi(T) \sim T - T_c ^{-\nu}$	ν	1	0.63

Table 13.1: Several of the critical exponents for the percolation and magnetism phase transitions in $d = 2$ and $d = 3$ dimensions. Ratios of integers correspond to known exact results. The critical exponents for the Ising model are discussed in Chapter ??.

Because we can simulate only finite lattices, a direct fit of the measured quantities ξ , P_∞ , and $S(p)$ to their assumed critical behavior for an infinite lattice would not yield good estimates for the corresponding exponents ν , β , and γ (see Problem 13.9b). The problem is that if p is close to p_c , the extent of the largest cluster becomes comparable to L , and the nature of the cluster distribution is affected by the finite size of the system. In contrast, for p far from p_c , $\xi(p)$ is small in comparison to L and the measured values of ξ , and hence the values of other physical quantities, are not appreciably affected by the finite size of the lattice. Hence for $p \ll p_c$ and $p \gg p_c$, the properties of the system are indistinguishable from the corresponding properties of a truly macroscopic system ($L \rightarrow \infty$). However, if p is close to p_c , $\xi(p)$ is comparable to L and the behavior of the system differs from that of an infinite system. In particular, a finite lattice cannot exhibit a true phase transition characterized by divergent physical quantities. Instead, ξ and S reach a finite maximum at $p = p_c(L)$.

The effects of the finite size of the system can be made more quantitative by the following argument. Consider for example, the critical behavior (13.14) of P_∞ . As long as ξ is much less than L , the power law behavior given by (13.14) is expected to hold. However, if ξ is comparable to L , ξ cannot change appreciably and (13.14) is no longer applicable. This qualitative change in the behavior of P_∞ and other physical quantities occurs for

$$\xi(p) \sim L \sim |p - p_c|^{-\nu}. \tag{13.16}$$

We invert (13.16) and write

$$|p - p_c| \sim L^{-1/\nu}. \tag{13.17}$$

The difference $|p - p_c|$ in (13.17) is the “distance” from the critical point at which “saturation” or finite size effects occur. Hence if ξ and L are approximately the same size, we can replace (13.14)

by the relation

$$P_\infty(p = p_c) \sim L^{-\beta/\nu} \quad (L \rightarrow \infty) \quad (13.18)$$

for the value of P_∞ at $p = p_c$ for a finite lattice. The relation (13.18) between P_∞ and L at $p = p_c$ is consistent with the fact that a phase transition, i.e., a singularity, is defined only for infinite systems.

One implication of (13.18) is that we can use it to determine the critical exponents. This method of analysis is known as *finite size scaling*. Suppose that we generate percolation configurations at $p = p_c$ for different values of L and analyze P_∞ as a function of L . If our values of L are sufficiently large, we can use the asymptotic relation (13.18) to estimate the ratio β/ν . A similar analysis can be used for $S(p)$ and other quantities of interest. We use this method in Problem 13.9.

Problem 13.9. Finite size scaling analysis of critical exponents

- Compute P_∞ at $p = p_c$ for at least 100 configurations. Consider $L = 10, 20, 40$, and 60 . Include in your average only those configurations that have a spanning cluster. Best results are obtained using the value of p_c for the infinite square lattice, $p_c \approx 0.5927$. Plot $\ln P_\infty$ versus $\ln L$, and estimate the ratio β/ν .
- Use finite size scaling arguments to determine the dependence of the mean cluster size S on L at $p = p_c$. Average S over the same configurations as considered in part (b). Remember that S is the mean number of sites in the nonspanning clusters.
- Analyze your data for the p -dependence of $S(p)$ obtained in Problem 13.5b for $L = 16$ and estimate the value of γ according to the assumed behavior given in (13.15). How does your estimate for γ compare with the answer that you obtained in part (b)?
- Find the mass (number of particles) M in the spanning cluster at $p = p_c$ as a function of L . Use the same configurations as in part (b). Determine an exponent from a plot of $\ln M$ versus $\ln L$. This exponent is called the fractal dimension of the cluster and is discussed in Chapter ??.

We found in Section 13.2 that the numerical value of the percolation threshold p_c depends on the symmetry and dimension of the lattice, e.g., $p_c \approx 0.5927$ for the square lattice and $p_c = 1/2$ for the triangular lattice. A remarkable feature of the power law dependencies summarized in Table 13.1 is that the values of the critical exponents do not depend on the symmetry of the lattice and are independent of the existence of the lattice itself, e.g., they are identical for the continuum percolation model discussed in Problem 13.4. Moreover, it is not necessary to distinguish between the exponents for site and bond percolation. In the vocabulary of critical phenomena, we say that site, bond, and continuum percolation all belong to the same *universality class* and that their critical exponents are identical.

Another important idea in critical phenomena is the existence of relations between the critical exponents. An example of such a *scaling law* is

$$2\beta + \gamma = \nu d, \quad (13.19)$$

where d is the spatial dimension of the lattice. The scaling law (13.19) indicates that the universality class depends on the spatial dimension. A more detailed discussion of finite size scaling and the scaling laws can be found in Chapter ?? and in the references.

13.5 The Renormalization Group

In Section 13.4, we studied the properties of various quantities on different length scales to determine the values of the critical exponents. The idea of examining physical quantities near the critical point on different length scales can be extended beyond finite size scaling and is the basis of the *renormalization group* method, probably the most important new method developed in theoretical physics during the past twenty-five years. Kenneth Wilson was honored in 1981 with the Nobel prize in physics for his contributions to the development of the renormalization group method. Although the method was first applied to thermodynamic phase transitions, it is simpler to introduce the method in the context of the percolation transition. We will find that the renormalization group method yields the critical exponents directly, and in combination with Monte Carlo methods, it is frequently more powerful than Monte Carlo methods alone.

To introduce the method, consider a photograph of a percolation configuration generated at $p = p_0 < p_c$. If we view the photograph (or screen) from further and further distances, what would we see? Convince yourself that when you are far away from the photograph, you cannot distinguish occupied sites that are adjacent to each other and you cannot observe single site clusters. In addition, branches emanating from larger clusters and narrow bridges connecting large “blobs” are lost in your distant view of the photograph. Hence for $p_0 < p_c$, the distant photograph looks like a percolation configuration generated at a value of $p = p_1$ less than p_0 . In addition, the connectedness length $\xi(p_1)$ of the remaining clusters is smaller than $\xi(p_0)$. If we move even further away from the photograph, the new clusters look even smaller with a value of $p = p_2$ less than p_1 . Eventually we will not be able to distinguish any clusters and the photograph will appear as if it were at the trivial *fixed point* $p = 0$.

What would we observe as we go away from the photograph for $p_0 > p_c$? We can use the same reasoning to deduce that we would see only small regions of unoccupied sites. As we move further away from the photograph, these spaces become less discernible and the configuration looks as though a larger percentage of the lattice were occupied. Hence, the photograph will look like a configuration generated at a value of $p = p_1$ greater than p_0 with $\xi(p_1) < \xi(p_0)$. As we move further and further away from the photograph, it will eventually appear to be at the other trivial fixed point $p = 1$.

What would we observe at $p_0 = p_c$? We know that at the percolation threshold, all length scales are present and it does not matter which length scale we use to observe the system. Hence, the photograph appears the same (although smaller overall) regardless of the distance at which we observe it. In this sense, p_c is a special, *nontrivial* fixed point.

We now consider a way of using a computer to change the configurations in a way that is similar to moving away from the photograph. Consider a square lattice that is partitioned into *cells* or *blocks* that cover the lattice (see Fig. 13.10). If we view the lattice from the perspective in which the sites in a cell merge to become a new supersite or renormalized site, then the new lattice has the same symmetry as the original lattice. However, the replacement of cells by the new sites has changed the length scale — all distances are now smaller by a factor of b , where b is the linear dimension of the cell. Hence, the effect of a “renormalization” is to replace each group of sites with a single renormalized site and to rescale the connectedness length for the renormalized lattice by a factor of b .

How can we decide whether the renormalized site is occupied or not? Because we want to

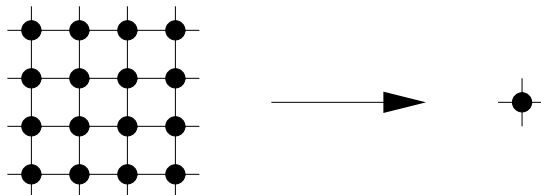


Figure 13.10: An example of a $b = 4$ cell used on the square lattice. The cell contains b^2 sites which are rescaled to a single supersite after a renormalization group transformation.

preserve the main features of the original lattice and hence its connectedness (and its symmetry), we assume that a renormalized site is occupied if the original group of sites spans the cell. We adopt the vertical spanning criterion for convenience. The effect of performing a renormalization transformation on typical percolation configurations for p above and below p_c is illustrated in Fig. 13.11 and Fig. 13.12 respectively. In both cases, the effect of the successive transformations is to move the system away from p_c . We see that for $p = 0.7$, the effect of the transformations is to drive the system toward $p = 1$. For $p = 0.5$, the trend is to drive the system toward $p = 0$. As we discuss in the following, we can associate p_c with an unstable fixed point of the renormalization transformation. Of course, because we began with a finite lattice, we cannot continue the renormalization transformation indefinitely.

Program rg implements a visual interpretation of the renormalization group. The program divides the screen into four windows with the original lattice in the first window and three renormalized lattices in windows 2 through 4. In **Program site** we represented an occupied site at lattice point \mathbf{x}, y as a filled circle of unit diameter centered about the point (x, y) . In contrast, **Program rg** represents an occupied site at \mathbf{x}, y as a filled box whose lower left corner is at $x - 1, y - 1$.

Problem 13.10. Visual renormalization group

Use **Program rg** with $L = 32$ to estimate the value of the percolation threshold. For example, confirm that for small p , e.g., $p = 0.4$, the renormalized lattice almost always renormalizes to a nonspanning cluster. What happens for $p = 0.8$? How can you use the properties of the renormalized lattices to estimate p_c ?

Although a visual implementation of the renormalization group allows us to estimate p_c , it does not allow us to estimate the critical exponents. In the following, we present a renormalization group method that allows us to obtain p_c and the critical exponent ν associated with the connectedness length. This analysis follows closely the method presented by Reynolds et al. (see references).

The implementation of a renormalization group method consists of two parts: (i) an average over the underlying variables together with a specification of the variables that determine the state of the renormalized configuration, and (ii) a parameterization of the renormalized configuration in terms of the original parameters and possibly others. We adopt the same average as before, i.e., we replace the b^d sites within a cell of linear dimension b by a single site that represents whether or not the original lattice sites span the cell. The second step is to determine which parameters specify the new configuration after the averaging. We make the simple approximation that each cell is independent of all the other cells and is characterized only by the probability p' that the cell

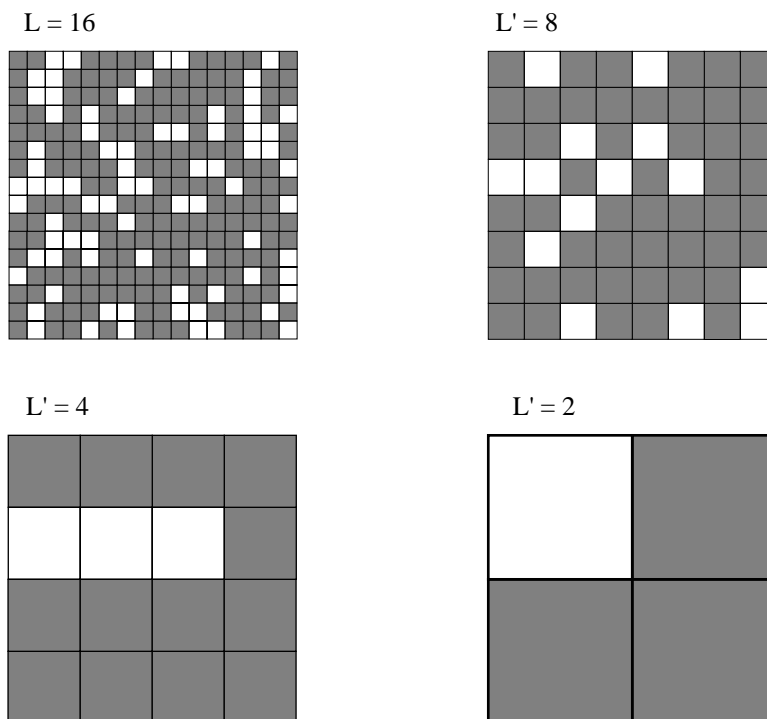


Figure 13.11: A percolation configuration generated at $p = 0.7$. The original configuration has been renormalized three times by transforming cells of four sites into one new supersite. What would be the effect of an additional transformation?

is occupied. The renormalization transformation between p and p' reflects the fact that the basic physics of percolation is connectedness, because we define a cell to be occupied only if it contains a set of sites that span the cell. If the sites are occupied with probability p , then the cells are occupied with probability p' , where p' is given by a *renormalization transformation* or a *recursion relation* of the form

$$p' = R(p). \tag{13.20}$$

The quantity $R(p)$ is the total probability that the sites form a spanning path.

An example will make the formal relation (13.20) more clear. In Fig. 13.13, we show the seven vertically spanning site configurations for a $b = 2$ cell. The probability p' that the renormalized site is occupied is given by the sum of the probabilities of all spanning configurations:

$$p' = R(p) = p^4 + 4p^3(1 - p) + 2p^2(1 - p)^2. \tag{13.21}$$

In general, the probability p' of the occupied renormalized sites is different than the occupation probability p of the original sites. For example, suppose that we begin with $p = p_0 = 0.5$. After a single renormalization transformation, the value of p' obtained from (13.21) is $p_1 = p' = R(p_0 =$

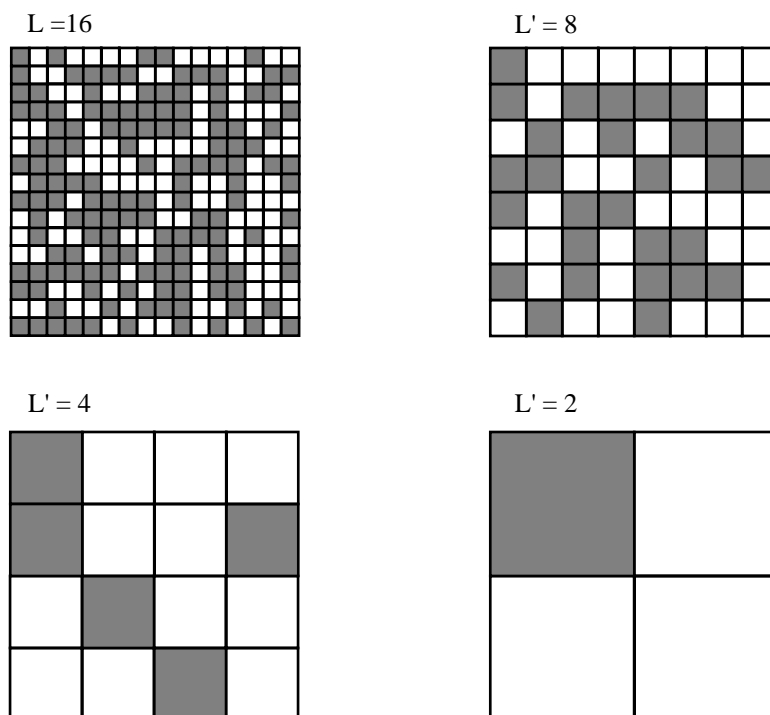


Figure 13.12: A percolation configuration generated at $p = 0.5$. The original configuration has been renormalized three times by transforming blocks of four sites into one new site. What would be the effect of an additional transformation?

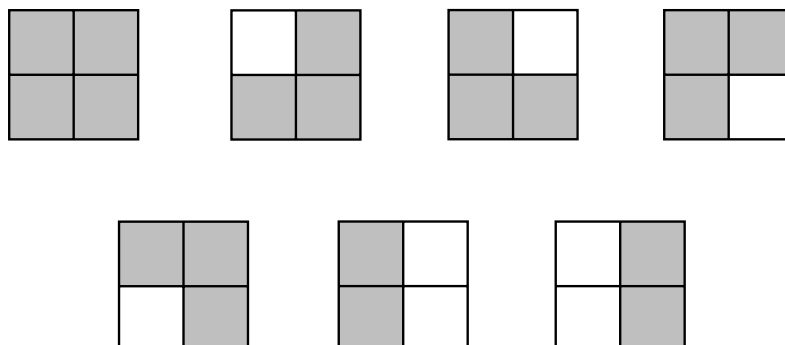
$0.5) = 0.44$. If we perform a second renormalization transformation, we have $p_2 = R(p_1) = 0.35$. It is easy to see that further transformations drive the system to the fixed point $p = 0$. Similarly, if we begin with $p = p_0 = 0.7$, we find that successive transformations drive the system to the fixed point $p = 1$. This behavior is qualitatively similar to what we observed in the visual renormalization group.

To find the nontrivial fixed point associated with the critical threshold p_c , we need to find the special value of p such that

$$p^* = R(p^*). \quad (13.22)$$

For the recursion relation (13.21), we find that the solution of the fourth degree equation for p^* yields the two trivial fixed points, $p^* = 0$ and $p^* = 1$, and the nontrivial fixed point $p^* = 0.61804$ which we associate with p_c . This calculated value of p^* for $b = 2$ should be compared with the estimate $p_c = 0.5927$.

To calculate the critical exponent ν , we recall that all lengths are reduced on the renormalized lattice by a factor of b in comparison to the lengths in the original system. Hence the connectedness

Figure 13.13: The seven (vertically) spanning configurations on a $b = 2$ cell.

length transforms as

$$\xi' = \xi/b. \quad (13.23)$$

Because $\xi(p) = \text{const}|p - p_c|^{-\nu}$ for $p \sim p_c$, we have

$$|p' - p^*|^{-\nu} = b^{-1}|p - p^*|^{-\nu}, \quad (13.24)$$

where we have identified p_c with p^* . To find the relation between p' and p near p_c , we expand the renormalization transformation (13.20) in a Taylor series about p^* and obtain to first order in $(p - p^*)$:

$$p' - p^* = R(p) - R(p^*) \approx \lambda(p - p^*), \quad (13.25)$$

where

$$\lambda = \frac{dR(p = p^*)}{dp}. \quad (13.26)$$

We need to do a little algebra to obtain an explicit expression for ν . We first raise both sides of (13.25) to the ν th power and write

$$|p' - p^*|^\nu = \lambda^\nu (p - p^*)^\nu. \quad (13.27)$$

We then compare (13.27) and (13.24) and obtain

$$b = \lambda^\nu. \quad (13.28)$$

Finally, we take the logarithm of both sides of (13.28) and obtain the desired relation for the critical exponent ν :

$$\nu = \frac{\log b}{\log \lambda}. \quad (13.29)$$

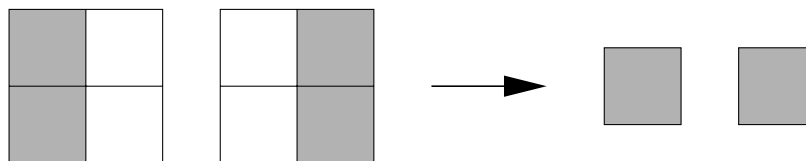


Figure 13.14: Example of the interface problem between cells. The two cells are not connected at the original site level but are connected at the cell level.

As an example, let us calculate λ for a square lattice with $b = 2$. We write (13.21) in the form $R(p) = -p^4 + 2p^2$. The derivative of $R(p)$ with respect to p yields $\lambda = 4p(1 - p^2) = 1.5279$ at $p = p^* = 0.61804$. We then use the relation (13.29) to obtain

$$\nu = \frac{\log 2}{\log 1.5279} = 1.635\dots \quad (13.30)$$

A comparison of (13.30) with the exact result $\nu = 4/3$ (see Table 13.1) in two dimensions shows remarkable agreement for such a simple calculation. (What would we be able to conclude if we were to measure $\xi(p)$ directly on a 2×2 lattice?) However, the accuracy of our calculation of ν is not known. What is the nature of our approximations? Our major assumption has been that the occupancy of each cell is independent of all other cells. This assumption is correct for the original sites, but after one renormalization, we lose some of the original connecting paths and gain connecting paths that are not present in the original lattice. An example of this interface problem is shown in Fig. 13.14. Because this surface effect becomes less probable with increasing cell size, one way to improve the renormalization group calculation is to consider larger cells. We consider a $b = 3$ calculation in Problem 13.11d. In Project 13.14 we combine the renormalization group method with a Monte Carlo approach to treat still larger cells.

Problem 13.11. Renormalization group method for small cells

- Enumerate the spanning configurations for a $b = 2$ cell assuming that a cell is occupied if a spanning path exists in either the vertical or the horizontal directions. Obtain the recursion relation and solve for the fixed point p^* . Although you could use a root finding algorithm to solve for p^* , it is easy to use trial and error to find the value of p such that $R(p) - p$ is zero. Or you can plot the function $R(p) - p$ versus p and find the value of p at which $R(p) - p$ crosses the horizontal axis. How do p^* and ν compare to their values using the vertical spanning criterion?
- Repeat the simple renormalization group calculation in (a) using the criterion that a cell is occupied only if a spanning path exists in both directions.
- The association of p_c with p^* is not the only possible one. Two alternatives involve the derivative $R'(p) = dR/dp$. For example, we could let $p_c = \bar{p} = \int_0^1 pR'(p) dp$. Alternatively, we could choose $p_c = p_{\max}$, where p_{\max} is the value of p at which $R'(p)$ has its maximum value. Compute p_c using these two alternative definitions and the various spanning criteria. In the limit of large cells, all three definitions should lead to the same values of p_c .
- Enumerate the possible spanning configurations of a $b = 3$ cell, assuming that a cell is occupied if a cluster spans the cell vertically. Determine the probability of each configuration, and verify

that the renormalization transformation $R(p) = p^9 + 9p^8q + 36p^7q^2 + 67p^6q^3 + 59p^5q^4 + 22p^4q^5 + 3p^3q^6$. It is possible to do the exact enumeration by hand. Solve the recursion relation (13.22) for p^* . Use this value of p^* to find the slope λ and the exponent ν . Then assume a cell is occupied if a cluster spans the cell both vertically and horizontally and obtain $R(p)$. Determine $p^*(b=3)$ and $\nu(b=3)$ for the two spanning criteria. Are your results for p^* and ν closer to their known values than for $b=2$ for the same spanning criteria?

Problem 13.12. Renormalization group method for triangular lattice

- There are some difficulties with the above renormalization group method in the infinite cell limit, if a cell is said to span when there is a path in one fixed direction (see Ziff). This problem is absent for the triangular lattice. For this symmetry a cell can be formed by grouping three sites that form a triangle into one renormalized site. The only reasonable spanning criterion is that the cell spans if any two sites are occupied. Verify that $R(p) = p^3 + 3p^2(1-p)$ and find $p_c = p^*$. How does p^* compare to the exact result $p_c = 1/2$?
- Calculate the critical exponent ν and compare its value with the exact result. Explain why b is given by $b^2 = 3$. Give a qualitative argument why the renormalization group argument might work better for small cells on a triangular lattice than on a square lattice.

It is possible to improve our renormalization group results for p_c and ν by enumerating the spanning clusters for larger b . However, because the 2^{b^2} possible configurations for a $b \times b$ cell increase rapidly with b , exact enumeration is not practical for $b > 7$, and we must use Monte Carlo methods if we wish to proceed further. Two Monte Carlo approaches are discussed in Project 13.14. The combination of methods, Monte Carlo and renormalization group (MCRG), provides a powerful tool for obtaining information on phase transitions and other properties of materials.

13.6 Projects

We have seen that the percolation problem illustrates many of the important ideas in critical phenomena. In later chapters we apply similar ideas and approaches to thermal systems. The following projects require larger systems and more computer resources than the problems in this chapter, but they are not much more difficult conceptually. More ideas for projects can be obtained from the references.

Project 13.13. Cell-to-cell renormalization group method

In Section 13.5 we discussed the cell-to-site renormalization group transformation for a system of cells of linear dimension b . An alternative transformation is to go from cells of linear dimension b_1 to cells of linear dimension b_2 . For such a “cell-to-cell” transformation, the rescaling length b_1/b_2 can be made close to unity. Many errors in a cell-to-cell renormalization group transformation cancel, resulting in a transformation that is more accurate in the limit in which the change in length scale is infinitesimal. We can use the fact that the connectedness lengths of the two systems are related by $\xi(p_2) = (b_1/b_2)^{-1}\xi(p_1)$ to derive the relation

$$\nu = \frac{\ln b_1/b_2}{\ln \lambda_1/\lambda_2}, \quad (13.31)$$

where $\lambda_i = dR(p^*, b_i)/dp$ is evaluated at the solution to the fixed point equation, $R(b_2, p^*) = R(b_1, p^*)$. Note that (13.31) reduces to (13.29) for $b_2 = 1$. Use the results you found in Problem 13.11d for one of the spanning criteria to estimate ν from a $b_1 = 3$ to $b_2 = 2$ transformation. Then consider larger values of b_2 and b_1 .

Project 13.14. Monte Carlo renormalization group

- a. One way to estimate $R(p)$, the total probability of all the spanning clusters, can be understood by writing $R(p)$ in the form

$$R(p) = \sum_{n=1}^N \binom{N}{n} p^n q^{(N-n)} S(n), \tag{13.32}$$

where $N = b^2$. The binomial coefficient $\binom{N}{n} = N!/((N-n)!n!)$ represents the number of possible configurations of n occupied sites and $N-n$ empty sites. The quantity $S(n)$ is the probability that a random configuration of n occupied sites spans the cell. A comparison of (13.21) and (13.32) shows that for $b = 2$ and the vertical spanning criterion, $S(1) = 0$, $S(2) = 2/6$, $S(3) = 1$, and $S(4) = 1$. What are the values of $S(n)$ for $b = 3$?

We can estimate the probability $S(n)$ by straightforward Monte Carlo methods. One way to sample $S(n)$ is to add a particle at random to an unoccupied site and check if a spanning path exists. If a spanning path does not exist, add another particle at random to a previously unoccupied site. If a spanning path exists after s particles are added, then let $S(n) = S(n) + 1$ for $n \geq s$ and generate a new configuration. After a reasonable number of configurations, the results for $S(n)$ can be normalized. Of course, this procedure can be made more efficient by checking for a spanning cluster only after the total number of particles added is near $s \sim p^*N$ and by checking for spanning after adding several particles.

Write a Monte Carlo program to sample $S(n)$. (Hint: store the location of the unoccupied sites in a separate array.) To check your program, first sample $S(n)$ for $b = 2$ and $b = 3$ and compare your results to the exact results for $S(n)$. Consider larger values of b and determine $S(n)$ for $b = 5, 8, 16$, and 32 . For $b \geq 16$, the total probability $R(p)$ can be found by using (13.32) and the Gaussian approximation for the probability of a configuration of n occupied sites:

$$P_N(n) = \binom{N}{n} p^n q^{(N-n)} \approx (2\pi Npq)^{-\frac{1}{2}} e^{-(n-pN)^2/2Npq}. \tag{13.33}$$

Because $P_N(n)$ is sharply peaked for large b , it is necessary to sample $S(n)$ only near $n = p^*N$. This method has been investigated by Hu (see references).

- b. In part (a) the number of particles rather than the occupation probability p was varied. Another Monte Carlo procedure is to vary p and sample $F(p) dp$, the probability of *first* spanning a $b \times b$ cell in the range p to $p + dp$. Because the renormalization group transformation defines p' as the *total* probability of spanning at p , p' can be interpreted as the *cumulative distribution function* and is related to $F(p)$ by

$$p' = R(p) = \int_0^p F(p) dp. \tag{13.34}$$

The sampling of $F(p)$ for finite width bins Δp implies that the integral in (13.34) reduces to a sum. Because $\lambda = dR(p = p^*)/dp$, we have $\lambda = F(p^*)$. The simplest way to estimate λ is by setting $\lambda = F(p_{\max})$, where p_{\max} is the value of p for which $F(p)$ is a maximum. Determine $p_c(b)$ and $\nu(b)$ for $b = 5, 8, 16$, and 32 . How do your results compare with those found in part (refproj:13/mcrgproba)? Which method yields smaller error estimates for p_c and ν ?

- c. It is possible to extrapolate the results for the successive estimates $p_c(b)$ and $\nu(b)$ to the limit $b \rightarrow \infty$. Finite size scaling arguments (cf. Stauffer and Aharony) suggest that

$$\nu(b) \approx \nu - c_1/\ln b \quad (13.35a)$$

and

$$p^*(b) \approx p_c - a_1 b^{-1/\nu} \quad (13.35b)$$

for b sufficiently large. The relation (13.35a) suggests that the sequence $\nu(b)$ should be plotted as a function of $1/\ln b$ and the extrapolated result should be a straight line with an intercept of ν . The quantities a_1 and c_1 in (13.35) are fitting parameters. The relation (13.35b) suggests that we should plot $p_c(b)$ versus $b^{-1/\nu}$ using the value of ν found from (13.35a). How sensitively does your result for p_c depend on the assumed value of ν ? It is necessary to consider cells on the order of $b = 500$, and to do a more sophisticated analysis of $\nu(b)$ and $p^*(b)$, to obtain extrapolated values that agree to four places with the exact value $\nu = 4/3$ and the estimate $p_c = 0.5927$.

Project 13.15. Percolation in three dimensions

- a. The value of p_c for site percolation on the simple cubic lattice is approximately 0.311. Write a program using the Hoshen-Kopelman cluster labeling method to verify this value. Compute ϕ_c , the volume fraction occupied at p_c , if a sphere with a diameter equal to the lattice spacing is placed on each occupied site.
- b. Consider continuum percolation in three dimensions where spheres of unit diameter are placed at random in a cubical box of linear dimension L . Two spheres that overlap are in the same cluster. As each sphere is added to the box, determine if the sphere overlaps with any other sphere in the box. If it does not, then the sphere constitutes a new cluster. If it does, then the sphere adopts the cluster label of the sphere with which it overlaps. If it overlaps with spheres of different cluster labels, then it is necessary to either relabel the clusters or to use the Hoshen-Kopelman algorithm to determine the proper label and generate a label tree. The volume fraction occupied by the spheres is given by

$$\phi = 1 - e^{-\rho 4\pi r^3/3}, \quad (13.36)$$

where ρ is the number density of the spheres, and r is their radius. Write a program to simulate continuum percolation in three dimensions and find the percolation threshold ρ_c . Use the Monte Carlo procedure discussed in Problem 13.4 to estimate ϕ_c and compare its value with the value obtained using (13.36). How does ϕ_c for continuum percolation compare with the value of ϕ_c found for site percolation in part (a)? Which do you expect to be larger and why?

- c. In the Swiss cheese model in three dimensions, we are concerned with the percolation of the space between the spheres. This model is appropriate for porous rock with the spheres representing

solid material and the space between the spheres representing the pores. Superimpose a regular grid with lattice spacing equal to $0.1r$ on the system, where r is the radius of the spheres. If a point on the grid is not within any sphere, it is “occupied.” The use of the grid allows us to determine the connectivity between different regions of the pore space. Use your cluster labeling routine from part (a) to label the clusters, and determine $\tilde{\phi}_c$, the volume fraction occupied by the pores at threshold. You might be surprised to find that $\tilde{\phi}_c$ is relatively small. If time permits, use a finer grid and repeat the calculation to improve the accuracy of your results.

- d. Use finite size scaling to estimate the critical percolation exponents for the three models presented in parts (a)–(c). Are they the same within the accuracy of your calculation?

Project 13.16. Conductivity in a random resistor network

- a. An important critical exponent for percolation is the conductivity exponent t defined by

$$\sigma \sim (p - p_c)^t, \quad (13.37)$$

where σ is the conductance (or inverse resistance) per unit length in two dimensions. Consider bond percolation on a square lattice where each occupied bond between two neighboring sites is a resistor of unit resistance. Unoccupied bonds have infinite resistance. Because the total current into any node must equal zero by Kirchhoff’s law, the voltage at any site (node) is equal to the average of the voltages of all nearest neighbor sites connected by resistors (occupied bonds). Since this relation for the voltage is the same as the algorithm for solving Laplace’s equation on a lattice, the voltage at each site can be computed using a relaxation method discussed in Chapter ???. To compute the conductivity for a given $L \times L$ resistor network, we fix the voltage $V = 0$ at sites for which $x = 0$ and fix $V = 1$ at sites for which $x = L + 1$. In the y direction we use periodic boundary conditions. We then compute the voltage at all sites using the relaxation method. The current through each resistor connected to a site at $x = 0$ is simply $I = \Delta V/R = (V - 0)/1 = V$. The conductivity is the sum of the currents through all the resistors connected to $x = 0$ divided by L . In a similar way, the conductivity can be computed from the resistors attached to the $x = L + 1$ boundary. Write a program to implement the relaxation method for the conductivity of a random resistor network on a square lattice. An indirect, but easier way of computing the conductivity, is considered in Problem ???.

- b. The bond percolation threshold on a square lattice is $p_c = 0.5$. Use your program to compute the conductivity for a $L = 30$ square lattice. Average over at least ten spanning configurations for $p = 0.51, 0.52$, and 0.53 . Note that you can eliminate all bonds that are not part of the spanning cluster and all occupied bonds connected to only one other occupied bond. Why? If possible, consider more values of p . Estimate the critical exponent t defined in (13.37).
- c. Fix p at $p = p_c = 1/2$ and use finite size scaling to estimate the conductivity exponent t .
- d. Use larger lattices and the multigrid method (see Project ???) to improve your results. If you have sufficient computing resources, compute t for a simple cubic lattice for which $p_c \approx 0.247$. (In two dimensions t is the same for lattice and continuum percolation. However, in three dimensions t can be different.)

References and Suggestions for Further Reading

- Joan Adler, “Series expansions,” *Computers in Physics* **8**, 287 (1994). The critical exponents and the value of p_c also can be determined by doing exact enumeration.
- I. Balberg, “Recent developments in continuum percolation,” *Phil. Mag.* **56**, 991 (1987). An earlier paper on continuum percolation is by Edward T. Gawlinski and H. Eugene Stanley “Continuum percolation in two dimensions: Monte Carlo tests of scaling and universality for non-interacting discs,” *J. Phys. A: Math. Gen.* **14**, L291 (1981). These workers divide the system into cells and use the Poisson distribution to place the appropriate number of disks in each cell.
- Armin Bunde and Shlomo Havlin, editors, *Fractals and Disordered Systems*, Springer-Verlag (1991). Chapter 2 by the editors is on percolation.
- C. Domb, E. Stoll, and T. Schneider, “Percolation clusters,” *Contemp. Phys.* **21**, 577 (1980). This review paper discusses the nature of the percolation transition using illustrations from a film of a Monte Carlo simulation of a percolation process.
- J. W. Essam, “Percolation theory,” *Reports on Progress in Physics* **53**, 833 (1980). A mathematically oriented review paper.
- Jens Feder, *Fractals*, Plenum Press (1988). See Chapter 7 on percolation. We discuss the fractal properties of the spanning cluster at the percolation threshold in Chapter ??.
- J. P. Fitzpatrick, R. B. Malt, and F. Spaepen, “Percolation theory of the conductivity of random close-packed mixtures of hard spheres,” *Phys. Lett. A* **47**, 207 (1974). The authors describe a demonstration experiment done in a first year physics course at Harvard.
- J. Hoshen and R. Kopelman, “Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm,” *Phys. Rev. B* **14**, 3438 (1976). The original paper on an efficient cluster labeling algorithm.
- Chin-Kun Hu, Chi-Ning Chen, and F. Y. Wu, “Histogram Monte Carlo position-space renormalization group: applications to site percolation,” preprint. The authors use a histogram Monte Carlo method which is similar to the method discussed in Project 13.14. A similar Monte Carlo method also was used by M. Ahsan Khan, Harvey Gould, and J. Chalupa, “Monte Carlo renormalization group study of bootstrap percolation,” *J. Phys. C* **18**, L223 (1985).
- Ramit Mehr, Tal Grossman, N. Kristianpoller, and Yuval Gefen, “Simple percolation experiment in two dimensions,” *Am. J. Phys.* **54**, 271 (1986). A simple experiment for an undergraduate physics laboratory is proposed.
- D. C. Rapaport, “Percolation on large lattices,” *Phil. Mag.* **56**, 1027 (1987). See also D. C. Rapaport, “Cluster size distribution at criticality,” *J. Stat. Phys.* **66**, 679 (1992). The author generates many independent sublattices and then combines them to study percolation on a $640\,000^2$ lattice.

- Peter J. Reynolds, H. Eugene Stanley, and W. Klein, “Large-cell Monte Carlo renormalization group for percolation,” *Phys. Rev.* **B21**, 1223 (1980). An especially clearly written research paper. Our discussion on the renormalization group in Section 13.5 is based upon this paper.
- Muhammad Sahimi, *Applications of Percolation Theory*, Taylor & Francis (1994). The author is a chemical engineer, and the emphasis is on modeling various phenomena in disordered media.
- Muhammad Sahimi and Hossein Rassamdana, “On position-space renormalization group approach to percolation,” *J. Stat. Phys.* **78**, 1157 (1995).
- Dietrich Stauffer and Amnon Aharony, *Introduction to Percolation Theory*, second edition, Taylor & Francis (1994). A delightful book by two of the leading workers in the field. An efficient Fortran implementation of the Hoshen-Kopelman algorithm is given in Appendix A.3.
- D. Stauffer, “Percolation clusters as teaching aid for Monte Carlo simulation and critical exponents,” *Am. J. Phys.* **45**, 1001 (1977); D. Stauffer, “Scaling theory of percolation clusters,” *Physics Reports* **54**, 1 (1979).
- B. P. Watson and P. L. Leath, “Conductivity in the two-dimensional-site percolation problem,” *Phys. Rev.* **B9**, 4893 (1974). A research paper on the conductivity of chicken wire.
- Kenneth G. Wilson, “Problems in physics with many scales of length,” *Sci. Am.* **241**, 158 (1979). An accessible article on the renormalization group method and its applications in particle and condensed matter physics. See also K. G. Wilson, “The renormalization group and critical phenomena,” *Rev. Mod. Phys.* **55**, 583 (1983). The latter article is the text of Wilson’s lecture on the occasion of the presentation of the 1982 Nobel Prize in Physics. In this lecture he claims that he “. . . found it very helpful to demand that a correctly formulated field theory be soluble by computer, the same way an ordinary differential equation can be solved on a computer . . .”
- W. Xia and M. F. Thorpe, “Percolation properties of random ellipses,” *Phys. Rev. A* **38**, 2650 (1988). The authors consider continuum percolation of elliptical shapes, and show that $\phi = e^{-A\rho}$, where A is the area of the object, and ρ is the number density.
- Richard Zallen, *The Physics of Amorphous Solids*, Wiley-Interscience (1983). Chapter 4 discusses many of the applications of percolation concepts to realistic systems.
- Robert M. Ziff, “Spanning probability in 2D percolation,” *Phys. Rev. Lett.* **69**, 2670 (1992). The author finds $p_c = 0.5927460 \pm 0.0000005$ for a square lattice.