(a) Modify your program for the Ising model on a square lattice so that single cluster flip dynamics (the Wolff algorithm) is used. Compute the mean energy and magnetization for $L = 16$ as a function of T for $T = 2.0$ to 2.7 in steps of 0.1. Compare your results to those obtained using the Metropolis algorithm. How many cluster flips do you need to obtain comparable accuracy at each temperature? Is the Wolff algorithm more efficient at every temperature near T_c?

(b) Fix T at the critical temperature of the infinite lattice ($T_c = 2/\ln(1+\sqrt{2})$) and use finite-size scaling to estimate the values of the various static critical exponents, for example, γ and α. Compare your results to those obtained using the Metropolis algorithm.

(c) Because we are generating site-bond percolation clusters, we can study their geometrical properties as we did for site percolation. For example, measure the distribution sn_s of cluster sizes at $p = p_c$ (see Problem 13.3). How does n_s depend on s for large s (see Project 13.15)? What is the fractal dimension of the clusters in the Ising model at $T = T_c$?

(d) The natural unit of time for single cluster flip dynamics is the number of cluster flips $t_{\rm cf}$. Measure $C_M(t_{\rm cf})$ and/or $C_E(t_{\rm cf})$ and estimate the corresponding correlation time $\tau_{\rm cf}$ for $T = 2.5$, 2.4, 2.3, and T_c for $L = 16$. As discussed in Problem 15.19, $\tau_{\rm cf}$ can be found from the relation $\tau_{\rm cf} = \sum_{t_{\rm cf}=1} C(t_{\rm cf})$. The sum is cut off at the first negative value of $C(t_{\rm cf})$. Estimate the value of $z_{\rm cf}$ from the relation $\tau_{\rm cf} = L^{z_{\rm cf}}$.

(e) To compare our results for the Wolff algorithm to our results for the Metropolis algorithm, we should use the same unit of time. Because only a fraction of the spins are updated at each cluster flip, the time $t_{\rm cf}$ is not equal to the usual unit of time, which corresponds to an update of the entire lattice or one Monte Carlo step per spin. We have that τ measured in Monte Carlo steps per spin is related to $\tau_{\rm cf}$ by $\tau = \tau_{\rm cf}\langle c\rangle/L^2$, where $\langle c\rangle$ is the mean number of spins in the single clusters, and L^2 is the number of spins in the entire lattice. Verify that the mean cluster size scales as $\langle c\rangle \sim L^{\gamma/\nu}$ with $\gamma = 7/4$ and $\nu = 1$. (The quantity $\langle c\rangle$ is the same quantity as the mean cluster size S defined in Chapter 12. The exponents characterizing the divergence of the various properties of the clusters are identical to the analogous thermal exponents.)

(f) To obtain the value of z that is directly comparable to the value found for the Metropolis algorithm, we need to rescale the time as in part (e). We have that $\tau \sim L^z \propto L^{z_{\rm cf}} L^{\gamma/\nu} L^{-d}$. Hence, z is related to the measured value of $z_{\rm cf}$ by $z = z_{\rm cf} - (d - \gamma/\nu)$. What is your estimated value of z? (It has been estimated that $z_{\rm cf} \approx 0.50$ for the $d = 2$ Ising model, which would imply that $z \approx 0.25$.)

(g) One of the limitations of the usual implementation of the Metropolis algorithm is that only one spin is flipped at a time. However, there is no reason why we could not choose f spins at random, compute the change in energy ΔE for flipping these f spins, and accepting or rejecting the trial move in the usual way according to the Boltzmann probability. Explain why this generalization of the Metropolis algorithm would be very inefficient, especially if $f \gg 1$. We conclude that the groups of spins to be flipped must be chosen with the physics of the system in mind and not simply at random. ■

Another cluster algorithm is to assign all bonds between parallel spins with probability p. As usual, no bonds are included between sites that have different spin orientations. From this configuration of bonds, we can form clusters of spins using one of the cluster identification algorithms we discussed in Chapter 12. The smallest cluster contains a single spin. After the clusters have been identified, all the spins in each cluster are flipped with probability $1/2$. This algorithm is known as the *Swendsen–Wang* algorithm and preceded the Wolff algorithm. Because the Wolff algorithm is easier to program and gives a smaller value of z than the Swendsen–Wang algorithm for the $d = 3$ and $d = 4$ Ising models, the Wolff algorithm is more commonly used.

Project 15.33 Invaded cluster algorithm

In Problem 13.7 we found that invasion percolation is an example of a self-organized critical phenomenon. In this cluster growth algorithm, random numbers are independently assigned to the bonds of a lattice. The growth starts from the seed sites of the left-most column. At each step the cluster grows by the occupation of the perimeter bond with the smallest random number. The growth continues until the cluster satisfies a stopping condition. We found that if we stop adding sites when the cluster is comparable in extent to the linear dimension L, then the fraction of bonds that are occupied approaches the percolation threshold p_c as $L \to \infty$. The invaded percolation algorithm automatically finds the percolation threshold!

Machta and co-workers have used this idea to find the critical temperature of a spin system without knowing its value in advance. For simplicity, we will discuss their algorithm in the context of the Ising model, although it can be easily generalized to the q-state Potts model (see the references). Consider a lattice on which there is a spin configuration $\{s_i\}$. The bonds of the lattice are assigned a random order. Bonds (i, j) are tested in this assigned order to see if s_i is parallel to s_j. If so, the bond is occupied and spins i and j are a part of the same cluster. Otherwise, the bond is not occupied and is not considered for the remainder of the current Monte Carlo step. The set of occupied bonds partitions the lattice into clusters of connected sites. The clusters can be found using the Newman–Ziff algorithm (see Section 12.3). The cluster structure evolves until a stopping condition is satisfied. Then a new spin configuration is obtained by flipping each cluster with probability $1/2$, thus completing one Monte Carlo step. The fraction f of bonds that were occupied during the growth process and the energy of the system are measured. The bonds are then randomly reordered and the process begins again. Note that the temperature is not an input parameter.

If open boundary conditions are used, the appropriate stopping rule is that a cluster spans the lattice (see Chapter 12, page 457). For periodic boundary conditions, the spanning rule discussed in Project 12.17 is appropriate.

Write a program to simulate the invaded cluster algorithm for the Ising model on the square lattice. Start with all spins up and determine how many Monte Carlo steps are needed for equilibration. How does this number compare to that required by the Metropolis algorithm at the critical temperature for the same value of L? An estimate for the critical temperature can be found from the relation (15.79) with f corresponding to p.

After you are satisfied that your program is working properly, determine the dependence of the critical temperature on the concentration c of nonmagnetic impurities. That is, randomly place nonmagnetic impurities on a fraction c of the sites. ■

Project 15.34 Physical test of random number generators

In Section 7.9 we discussed various statistical tests for the quality of random number generators. In this project we will find that the usual statistical tests might not be sufficient