PHYS 580 - Computational Physics

Computational Physics by *Nicholas J. Giordano, Hisao Nakanishi*

Student: **Ralph Razzouk**

Homework 5

Problem 7.2

Simulate a random walk in three dimensions allowing the walker to make steps of unit length in random directions; don't restrict the walker to sites on a discrete lattice. Show that the motion is diffusive, that is, $\langle r^2 \rangle \sim t$. Find the value of the proportionality constant.

Solution. In this problem, we simulate a random walk in three dimensions allowing the walker to make steps of unit length in random directions. The goal is to show that the motion is diffusive, i.e., $\langle r^2 \rangle \sim t$, and find the proportionality constant.

A key aspect when generating random points uniformly in any direction is to avoid biasing the distribution. If we were to use spherical coordinates and randomly generate r, θ, and ϕ, the resulting distribution would not be uniform due to the radial and angular dependence in the area element: $\mathrm{d}A = r^2 \sin\phi \, \mathrm{d}r \, \mathrm{d}\theta \, \mathrm{d}\phi$.

To achieve a uniform distribution, we use the parameterization:

$$x = \lambda^{1/3} \cos\theta \sqrt{1-u^2} y = \lambda^{1/3} \sin\theta \sqrt{1-u^2} z = -\lambda^{1/3} u,$$

where $\lambda \in [R_{\min}, R_{\max}]$, $u \in [-1, 1]$, and $\theta \in [0, 2\pi]$. For our case, we fix $r = 1$ for unit steps, so we only randomly generate u and θ.

Using this method, we implement the 3D random walk with steps of unit length in random directions. We start at the origin and record the position after each step. The statistics of interest are the mean squared displacement $\langle r^2 \rangle$ and its standard deviation σ_n, calculated over multiple walks.

After running the simulation with 1000 steps per walk and 100 different walks, we analyze the results by plotting both the mean squared displacement $\langle r^2 \rangle$ and its standard deviation σ_n as functions of the number of steps t in log-log scale. These are shown in Figures **??** and **??**.

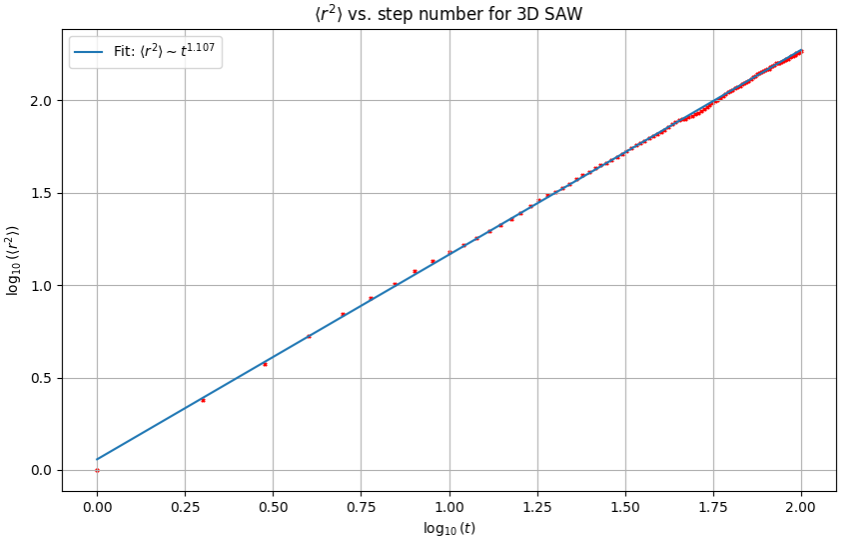

Figure 1: Mean squared displacement $\langle r_n^2 \rangle$ vs. number of steps for random walk in 3D.

If we fit the results with straight lines in log-log scale, we find:

$$\langle r_n^2 \rangle : m = 1.1073 \text{ and } b = 0.0577 \quad \sigma_n : m = 1.7781 \text{ and } b = -1.2170$$

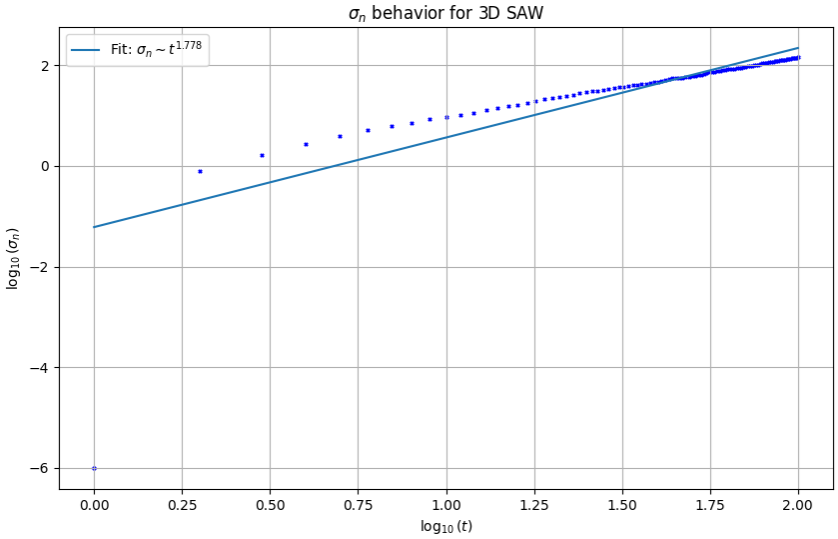

Figure 2: Standard deviation σ_n vs. number of steps for random walk in 3D.

From these results, we can conclude two important points. First, σ_n grows more rapidly than $\langle r_n^2 \rangle$, which indicates that we cannot determine the statistical properties by generating just a few random walksthe fluctuations become increasingly significant with time. Second, the mean squared displacement grows approximately linearly with the number of steps, confirming the diffusive nature of the motion.
From our fit, we calculate the Flory exponent $\nu = 0.5537 \pm 0.0012$, which is slightly less than the expected theoretical value of 0.588 for free diffusion. This slight discrepancy could be due to statistical fluctuations or finite-size effects in our simulation.
The fact that the slope in log-log scale is nearly 1 confirms that the motion is indeed diffusive, following the expected $\langle r^2 \rangle \sim t$ behavior, with a proportionality constant given by the exponentiated intercept. ■

Problem 7.6
Simulate SAWs in three dimensions. Determine the variation of $\langle r^2 \rangle$ with step number and find the value of v, where this parameter is defined through the relation $\sqrt{\langle r^2 \rangle} \sim At^{\nu}$. Compare your results with those in Figure 7.6. You should find that v decreases with successively higher dimensions. (It is 1 in one dimension and 3/4 in two dimensions.) Can you explain this trend qualitatively?
Also check whether you can reproduce the analytic result $v = 3/4$ for SAW on a 2D grid.

Solution. A Self-Avoiding Walk (SAW) is a random walk with the constraint that the walker cannot visit the same site twice. An important property of this type of walk is that each distinct SAW of the same number of steps must occur with equal probability. This property defines an ensemble. To satisfy this condition, we implement the simplest procedure described in the textbook: we grow a walk step by step, randomly selecting the next attempted step from all available directions, and if that attempt results in self-intersection, the entire walk is discarded and a new one is started from scratch.
We should note that this approach is computationally inefficient and significantly reduces the number of successful walks we can generate, especially for longer path lengths. This affects the statistical properties we can obtain. Nevertheless, the method correctly samples from the ensemble of SAWs.
For our simulation, we generate SAWs on a cubic lattice in 3D, allowing the walker to move in six possible directions at each step (x, y, z). We keep track of all visited sites and ensure that the walker never revisits a site. If the walker reaches a position where all neighboring sites have been visited, the walk is terminated and discarded.
After implementing this approach with multiple steps per walk and multiple attempted walks, we analyze the squared distance between the origin and the final point. We compute the mean squared displacement $\langle r^2 \rangle$ and its standard deviation σ_n across all successful walks. The results are shown in Figures **??** and **??**.

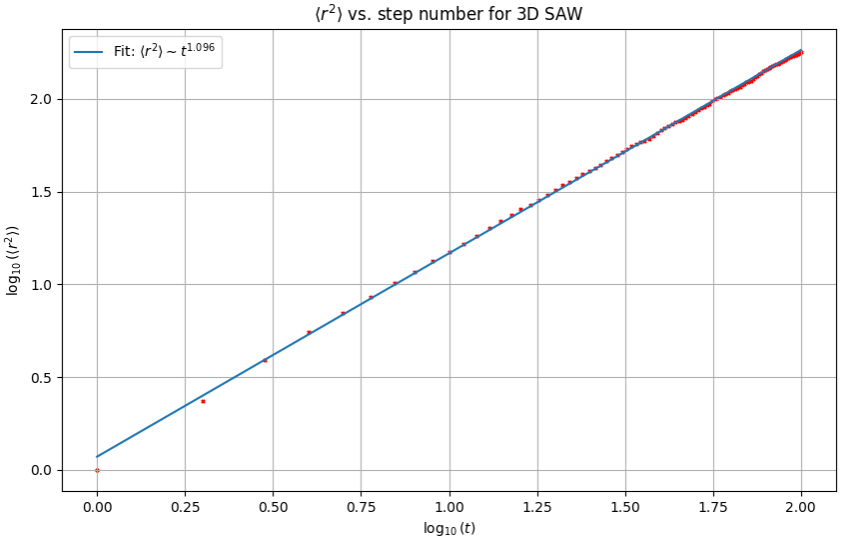

Figure 3: Mean squared displacement $\langle r_n^2 \rangle$ vs. number of steps for SAW in 3D.

In both cases (standard deviation and mean squared displacement), we observe a logarithmic linear dependence. Fitting these results with straight lines yields:

$$\langle r_n^2 \rangle : m = 1.0958 \text{ and } b = 0.0719 \sigma_n : m = 1.7663 \text{ and } b = -1.1884$$

From these fits, we calculate the Flory exponent $\nu = 0.5479 \pm 0.0013$. Comparing this with the exact result for 3D of $\nu = 0.588$, we see a difference of approximately 7%, which can be attributed to the computational limitations in our simulations and statistical fluctuations.

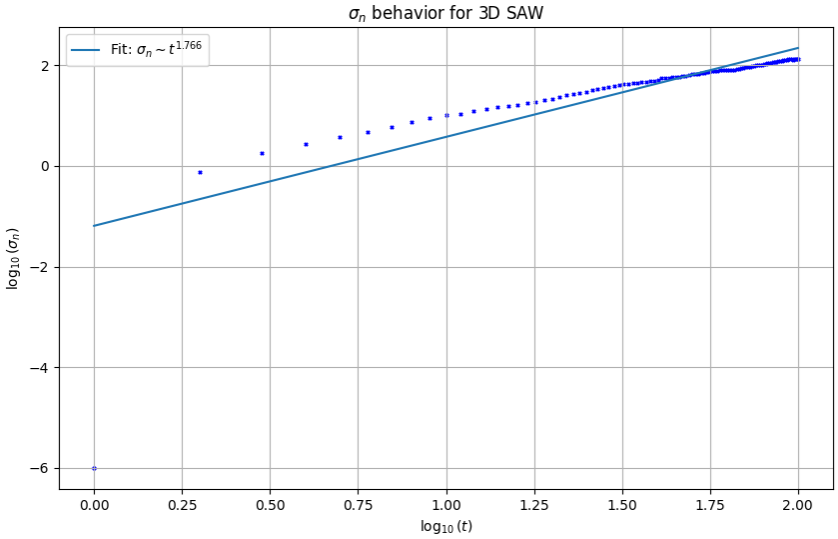

Figure 4: Standard deviation σ_n vs. number of steps for SAW in 3D.

It is known that the Flory exponent depends on the dimension d in the following way:

$$\nu \approx \begin{cases} \frac{3}{d+2} & \text{for } d < 4 \\ \frac{1}{2} & \text{for } d \geq 4 \end{cases}$$

This means that as we increase d, ν approaches $\frac{1}{2}$, and the Flory exponent decreases with increasing dimension.
The physical explanation for this trend is that in low dimensions, the walker has limited space to move, making self-avoidance a strong constraint. This forces the average separation to increase more rapidly with the number of steps as the walker must "go away" from the center to avoid previously visited sites. However, as we increase the number of dimensions, there are more possibilities to remain close to the initial point without self-intersecting, reducing the average separation relative to the number of steps taken.
In the extreme case of very high dimensions, the probability of random paths intersecting becomes so small that the SAW behaves almost like a regular random walk (with $\nu = 1/2$). The critical dimension at which this transition occurs is $d = 4$, where logarithmic corrections to the scaling behavior become important. ■

Problem 7.12
Calculate the entropy for the cream-in-your-coffee problem, and reproduce the results in Figure 7.16. If you have trouble with running time, then try first on coarser grids (e.g., 50×50 for the random walk, 4×4 for the entropy, and only 100 particles).

Solution. To simulate the entropy of cream particles in a cup of coffee, we follow the procedure described in the textbook. We treat the particles as points that can only move on a square lattice and can occupy the same point (unlike in the SAW case). We start with some initial distribution and allow the particles to move randomly on the lattice one step per unit of "time." This simulates the time evolution as a random walk. The entropy is calculated using the Shannon entropy formula:

$$S = -\sum_i P_i \ln P_i$$

where the sum is over all possible states of the system, and P_i is the probability of finding the system in state i. To apply this to our system, we divide it into bins and count the number of particles in each bin at a particular time to calculate the entropy.

In our implementation, we initialize $N = 400$ particles within a small square region of size $L_i = 20$ at the center of a larger $L \times L = 200 \times 200$ grid. At each time step, every particle moves randomly in one of four directions (up, down, left, right) with equal probability. We use periodic boundary conditions to keep particles within the grid.

To calculate the entropy, we divide the grid into an 8×8 array of cells, count the number of particles in each cell, and compute the entropy using the formula above. We repeat this process for 20,000 time steps to observe the full evolution of the system.

Figure **??** shows the entropy as a function of time steps. As expected, the entropy starts at a low value when the cream particles are concentrated in a small region, then increases rapidly as they begin to spread out, and finally approaches a plateau as the system reaches equilibrium with particles distributed approximately uniformly throughout the coffee.

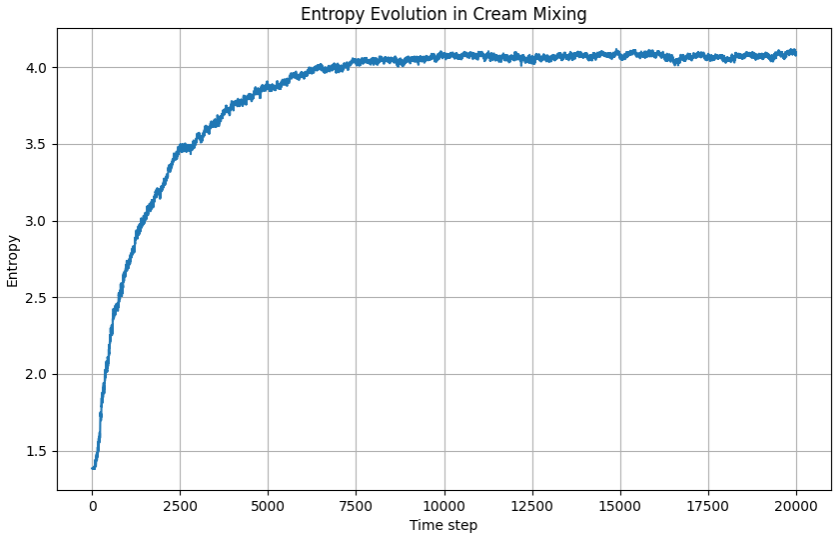

Figure 5: Entropy evolution in cream mixing as a function of time steps.

The shape of this curve illustrates the Second Law of Thermodynamics, which states that the entropy of an isolated system tends to increase over time. Initially, the cream particles are highly ordered (low entropy), but as they diffuse through random motion, they spread out to fill the available space, maximizing the entropy of the system.

The saturation of entropy at long times indicates that the system has reached equilibrium, where the particles are distributed uniformly throughout the container. At this point, the probability of finding a particle in any given cell is approximately the same for all cells, maximizing the entropy.

Our simulation shows that the entropy reaches a final value of approximately 4.0987, which is very close to the theoretical maximum entropy of 4.1589 for an 8×8 grid. This confirms that our system has indeed approached the maximum entropy state, as expected for a diffusive process.

This simulation provides a concrete example of how macroscopic irreversibility emerges from reversible microscopic dynamics. While each particle's random walk is locally reversible, the collective behavior of many particles leads to a systematic increase in entropy, making the reverse process (spontaneous concentration of particles) extremely unlikely. ■

Problem 7.15
Perform the random-walk simulation of spreading cream (Figures 7.13 and 7.14), and let one of the walls of the container possess a small hole so that if a cream particle enters the hole, it leaves the container. Calculate the number of particles in the container as a function of time. Show that this number, which is proportional to the partial pressure of the cream particle varies as $\exp(-t/\tau)$, where τ is the effective time constant for the escape.
Hint: Reasonable parameter choices are a 50×50 container lattice and hole 10 units in length along one of the edges.

Solution. In this problem, we use the same simulation approach as before for the evolution of cream particles, but instead of computing the entropy, we now consider a hole in one of the walls of the cup that allows particles to escape and not return. We are interested in how the number of particles inside the container changes with time, which is simply a matter of counting the number of particles remaining after each step in the random walk.
For our simulation, we use a 50×50 lattice with a hole of 10 units in length along the top edge. We initialize 400 particles in a small square region of size 10×10 at the center of the lattice. At each time step, particles move randomly in one of four directions. If a particle attempts to move outside the lattice boundaries, it is reflected back, except when it encounters the hole, in which case it escapes and is removed from the simulation.
We expect the number of particles to decay exponentially over time according to:

$$N(t) = N_0 \mathrm{e}^{-t/\tau}$$

where N_0 is the initial number of particles and τ is the effective time constant for the escape. Given this exponential form, we can apply the logarithm to both sides to obtain a linear relationship:

$$\ln N(t) = -\frac{1}{\tau} t + \ln N_0$$

If we plot the data in this form, the time constant will be given by the negative reciprocal of the slope.
Figures **??**, **??**, and **??** show the number of particles versus time in different formats. Figure **??** shows the raw count, Figure **??** shows the count on a semi-log scale, and Figure **??** shows the fitted exponential decay curve. After an initial period where particles diffuse through the container but have not yet reached the hole, we observe a linear decrease on the semi-log plot, confirming the exponential decay behavior.
After fitting the logarithm of particle count versus time, we obtain:

$$\text{Slope} = -0.000159$$

From this value, we derive the time constant and initial particle count:

$$\tau = 6274.99 \text{ time steps}, N_0 = 410.77$$

The slight discrepancy between our fitted N_0 (410.77) and the actual initial number of particles (400) can be attributed to statistical fluctuations in the fitting process.
The exponential decay of particle count confirms our hypothesis that the number of particles, which is proportional to the partial pressure, follows $N(t) \propto \mathrm{e}^{-t/\tau}$. This behavior is analogous to radioactive decay or the discharge of a capacitor. The time constant τ depends on the geometry of the container, the size of the hole, and the diffusion characteristics of the particles.
The physical process can be understood as follows: as particles diffuse throughout the container, they have some probability of encountering the hole and escaping. Since the particles are distributed approximately uniformly once they have had sufficient time to diffuse, the probability of any individual particle finding the hole in a given time interval is roughly constant. This constant probability of escape leads to the exponential decay in the number of remaining particles. ■

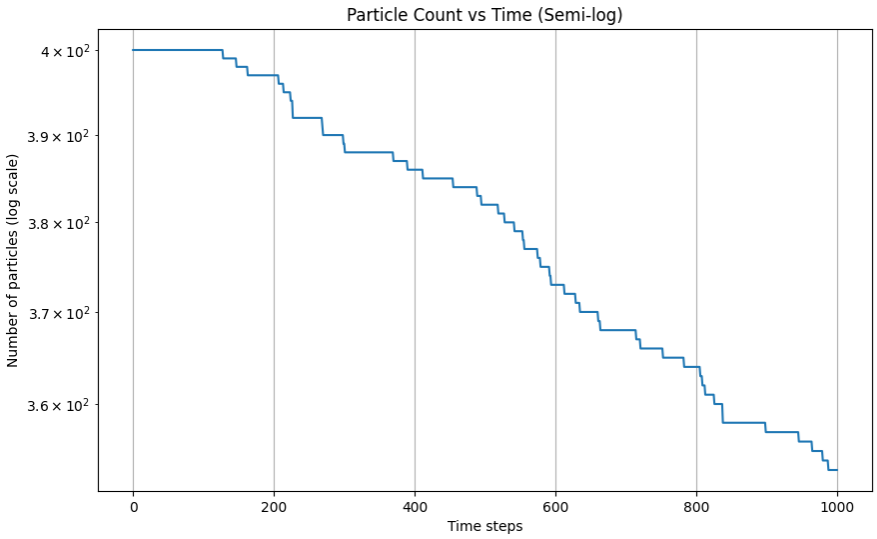

Figure 6: Number of particles in the container as a function of time (semi-log scale).

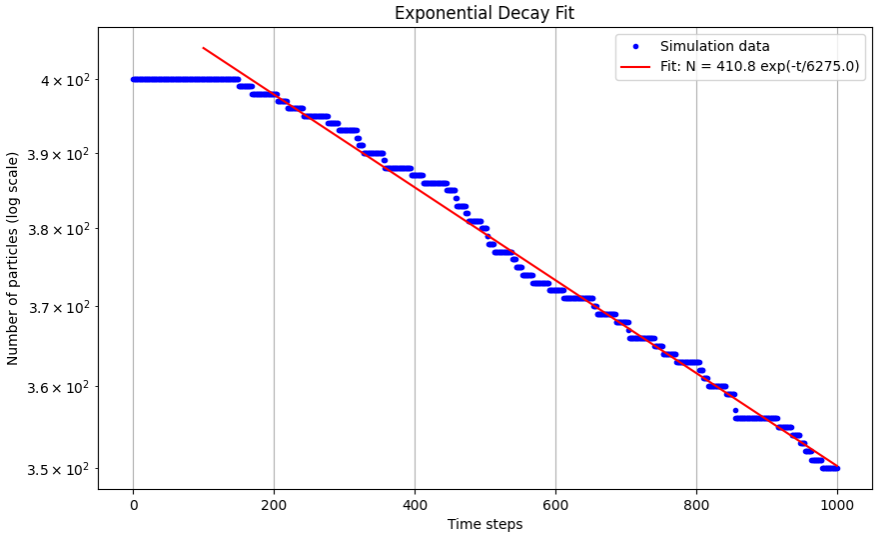

Figure 7: Exponential decay fit to the number of particles in the container.

Problem 7.30
Generate a spanning cluster for a two-dimensional square lattice at $p = p_c$ using any of the search methods discussed in connection with Figure 7.29. Estimate the fractal dimensionality of the cluster. You should find a value of *df*, which is slightly smaller than 2 (the expected value is $91/48 \approx 1.90$).

Solution. In this problem, we investigate the fractal properties of a spanning cluster at the percolation threshold on a 2D square lattice. A spanning cluster is a connected group of occupied sites that extends from one edge of the lattice to the opposite edge. At the critical percolation threshold ($p_c \approx 0.5927$ for site percolation on a 2D square lattice), these clusters exhibit fractal characteristics.
To find the spanning cluster, we use a connected-component labeling algorithm to identify all clusters on the lattice. We then select the cluster that connects the top and bottom boundaries. If multiple spanning clusters exist, we choose the largest one.
For estimating the fractal dimension, we employ a box-counting method to determine how the mass $m(r)$ scales with radius r, given the relation:

$$m(r) \sim r^{d_f}$$

where d_f is the fractal dimension. Taking the logarithm of both sides:

$$\log m(r) = d_f \log r + \text{constant}$$

The slope of this line on a log-log plot gives us the fractal dimension.
In our implementation, we use a lattice of size 200×200 with sites occupied with probability $p_c = 0.5927$. After identifying the spanning cluster, we calculate the mass (number of occupied sites) within circles of increasing radius centered at the cluster's center of mass.
Figure **??** shows the spanning cluster and the log-log plot used to determine the fractal dimension.

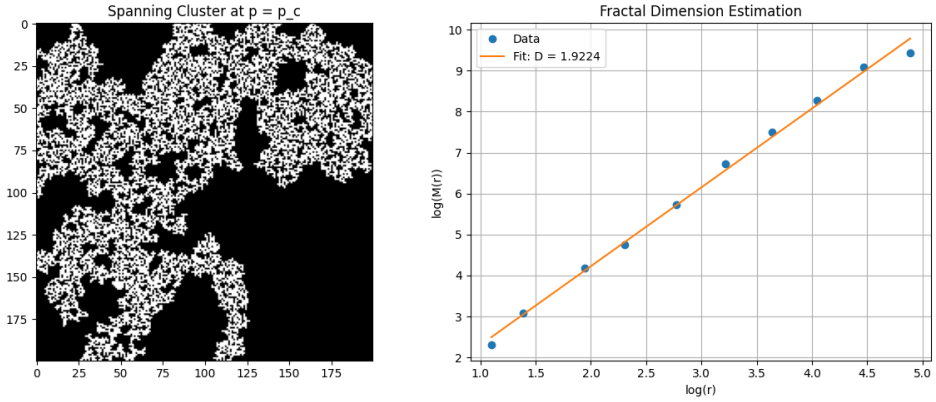

Figure 8: Left: Spanning cluster at the percolation threshold. Right: Log-log plot of mass vs. radius used to determine the fractal dimension.

From the linear fit to the log-log data, we find a fractal dimension of $d_f \approx 1.9224 \pm 0.0446$, which is very close to the expected theoretical value of $91/48 \approx 1.8958$.
The fact that $d_f < 2$ confirms that the spanning cluster at the percolation threshold is indeed a fractal. Unlike regular Euclidean objects, fractals exhibit complex, self-similar structures across different length scales. The spanning cluster has "holes" of all sizes, making it less "space-filling" than a solid object, hence its dimensionality is less than 2.
This fractal nature has important implications for physical systems like porous media, where the transport of fluids through the spanning cluster depends critically on its fractal properties. For example, the conductivity of a porous medium near the percolation threshold scales with $(p - p_c)^t$, where the exponent t is related to the fractal dimension.
The fractal dimension also relates to critical phenomena in statistical physics. At the percolation threshold, the system undergoes a phase transition, and quantities like the correlation length diverge, leading to scale-invariant properties. The fractal dimension is one such property that characterizes this critical state.

Our result confirms the universal nature of the fractal dimension at the percolation threshold, which is independent of the microscopic details of the system and depends only on the dimensionality of the space (in this case, 2D). ■