# Exploring Percolation Theory Through Simulation

Kiara Gholizad[1]

[1]Department of Physics, Sharif University of Technology, Tehran, Iran

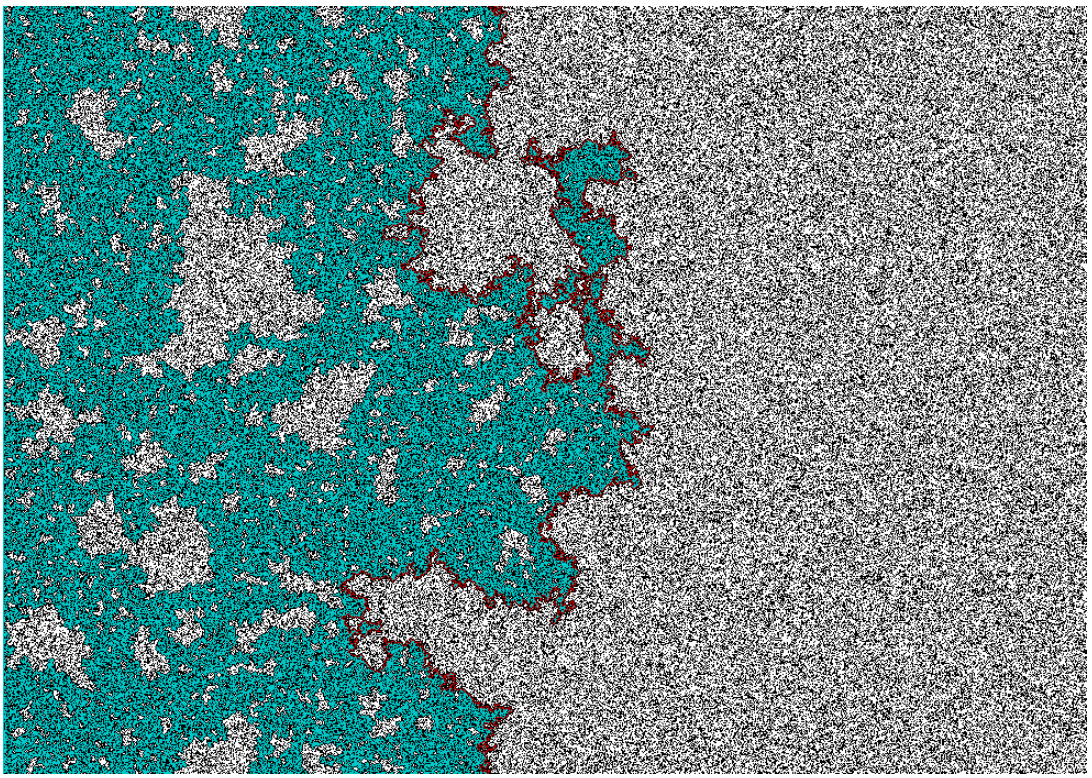
March 30, 2025

**Abstract**

This work presents a detailed computational investigation of percolation theory, a paradigmatic statistical physics model exhibiting a continuous phase transition despite its conceptual simplicity. We employed a number of algorithmsBreadth-First Search (BFS), coloring algorithms, and Hoshen-Kopelmanto study the critical behavior of percolation systems in one and two dimensions. Through extensive Monte Carlo simulations, we verified the theoretical percolation threshold $p_c = 1$ for 1D systems and estimated $p_c \approx 0.5904$ for 2D square lattices, in excellent agreement with the theoretical value of 0.5927. We estimated critical exponents like the correlation length exponent $\nu \approx 1.22$, the cluster distribution exponent $\tau \approx 1.69$, and the fractal dimension $D \approx 1.98$, all in agreement with theoretical expectations. Our real-space renormalization study revealed an unexpected behavior where the non-trivial fixed point vanishes with larger lattice sizes. We also analyzed the percolation probability and infinite cluster connection probability for different system sizes and found striking consistency in the average percolation probability ($\approx 0.41$) independent of lattice dimensions. These results not only verify the universal critical behavior of percolation systems but also provide insights into finite-size effects and scaling behavior in the vicinity of the phase transition.

# Contents

# 1 Introduction



Despiteits conceptual simplicity, percolation theory is a fundamental model in statistical physics, and displays a continuous phase transition. It provides a useful model and is applicable to a wide range of domains from biologyto physics and geophysics.

## 1.1 Fundamentals of Percolation Theory

In a percolation model, each site (or bond) in a lattice is either occupied with probability $p$ or empty with probability $1 - p$. A cluster is defined as a group of nearest neighboring occupied sites. As Dr. Christensen explains in the [1], "Percolation theory deals with the numbers and properties of the clusters formed when sites are occupied with probability $p$." As the occupation probability increases, clusters grow and eventually a percolating cluster spanning the entire system appears. The percolation threshold $p_c$ is defined as "the concentration (occupation probability) $p$ at which an infinite cluster appears for the first time in an infinite lattice." This threshold depends on the lattice structure and dimensionality, as shown in Table 1.

Table 1: Percolation thresholds for various lattices and dimensions

| Lattice | # nearest neighbors | Site percolation | Bond percolation |
|---|---|---|---|
| 1d | 2 | 1 | 1 |
| 2d Honeycomb | 3 | 0.6962 | $1 - 2\sin(\pi/18) \approx 0.65271$ |
| 2d Square | 4 | 0.592746 | 1/2 |
| 2d Triangular | 6 | 1/2 | $2\sin(\pi/18) \approx 0.34729$ |
| 3d Diamond | 4 | 0.43 | 0.388 |
| 3d Simple cubic | 6 | 0.3116 | 0.2488 |
| 3d BCC | 8 | 0.246 | 0.1803 |
| 3d FCC | 12 | 0.198 | 0.119 |
| 4d Hypercubic | 8 | 0.197 | 0.1601 |
| 5d Hypercubic | 10 | 0.141 | 0.1182 |
| 6d Hypercubic | 12 | 0.107 | 0.0942 |
| 7d Hypercubic | 14 | 0.089 | 0.0787 |
| Bethe lattice | $z$ | $1/(z-1)$ | $1/(z-1)$ |

## 1.2 1D Percolation

One-dimensional percolation provides a valuable starting point as it can be solved analytically. In 1D, a percolating cluster must span from $-\infty$ to $+\infty$, which requires all sites to be occupied. Thus, the critical percolation threshold is $p_c = 1$. This can be mathematically verified by considering the probability $\Pi(p, L)$ that a lattice of linear size $L$ percolates at concentration $p$:

$$\Pi(p, L) = p^L \tag{1}$$

Taking the limit as $L \to \infty$:

$$\lim_{L\to\infty} \Pi(p, L) = \lim_{L\to\infty} p^L = \begin{cases} 0 & \text{for } p < 1 \\ 1 & \text{for } p = 1 \end{cases} \tag{2}$$

The cluster size distribution in 1D is given by:

$$n_s(p) = (1-p)^2 p^s \tag{3}$$

Where $n_s(p)$ is the number of clusters of size $s$ per lattice site. As $p$ approaches $p_c = 1$, we can express this as:

$$n_s(p) = (p_c - p)^2 \exp(-s/s_\xi) \tag{4}$$

where $s_\xi = (p_c - p)^{-1}$ is the characteristic cluster size that diverges as $p \to p_c$.

## 1.3 Phase Transition and Critical Behavior

Percolation theory has a second-order phase transition at $p_c$, which is defined by:

- The formation of a spanning cluster at $p \geq p_c$
- Power-law divergence of the correlation length $\xi \propto |p - p_c|^{-\nu}$ as $p \to p_c$
- Power-law behavior of the mean cluster size $S(p) \propto |p - p_c|^{-\gamma}$ as $p \to p_c$

- The probability that an arbitrary site belongs to the infinite cluster $P(p) \propto (p-p_c)^\beta$ for $p > p_c$

The mean cluster size $S(p)$ is given by:

$$S(p) = \frac{\sum_{s=1}^{\infty} s^2 n_s(p)}{\sum_{s=1}^{\infty} s n_s(p)} = \frac{1+p}{1-p} = \frac{p_c + p}{p_c - p} \tag{5}$$

For $p \to p_c^-$, this is:

$$S(p) \propto (p_c - p)^{-1} \propto (p_c - p)^{-\gamma} \tag{6}$$

Thus, in 1D, $\gamma = 1$.Similarly, the correlation length $\xi$ diverges as $\xi \propto (p_c - p)^{-\nu}$ with $\nu = 1$ in 1D.

## 1.4 Scaling and Universality

Near the critical point, various quantities follow scaling laws with critical exponents. These exponents are related to each other by scaling laws such as:

$$\gamma = \frac{3-\tau}{\sigma} \tag{7}$$

$$\beta = \frac{\tau - 2}{\sigma} \tag{8}$$

$$\nu = \frac{1}{D\sigma} \tag{9}$$

$$D = d - \frac{\beta}{\nu} \tag{10}$$

Where $\tau$ is the Fisher exponent for the cluster size distribution, $\sigma$ is related to the cutoff cluster size, $\nu$ describes the divergence of the correlation length, $\beta$ is the order parameter exponent, $\gamma$ characterizes the mean cluster size, and $D$ is the fractal dimension of clusters at criticality. Table 2 provides the theoretical values of these critical exponents for different dimensions.

Table 2: Critical exponents for percolation in different dimensions

| Exponent | 1D | 2D | 3D | 4D | 5D | 6D | Bethe |
|---|---|---|---|---|---|---|---|
| $\alpha$ | 1 | -2/3 | -0.62 | -0.72 | -0.86 | -1 | -1 |
| $\beta$ | 0 | 5/36 | 0.41 | 0.64 | 0.84 | 1 | 1 |
| $\gamma$ | 1 | 43/18 | 1.80 | 1.44 | 1.18 | 1 | 1 |
| $\nu$ | 1 | 4/3 | 0.88 | 0.68 | 0.57 | 1/2 | 1/2 |
| $\sigma$ | 1 | 36/91 | 0.45 | 0.48 | 0.49 | 1/2 | 1/2 |
| $\tau$ | 2 | 187/91 | 2.18 | 2.31 | 2.41 | 5/2 | 5/2 |
| $D(p = p_c)$ | 1 | 91/48 | 2.53 | 3.06 | 3.54 | 4 | 4 |

# 2 Computational Methods

## 2.1 Grid Generation and Basic Algorithms

We implement several algorithms to study percolation on lattices. First, we generate a grid where each site is independently occupied with probability $p$:

**Algorithm 1** Grid Generation

1: **function** GENERATE_GRID$(m, n, p)$
2: **return** $(random(m, n) < p)$ ▷ Boolean grid of size $m \times n$
3: **end function**

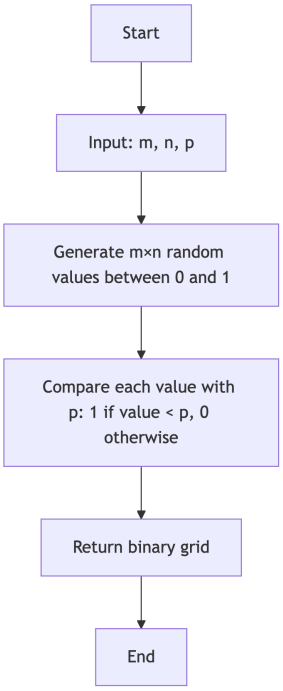



Figure 1: Flowchart for grid generation

By setting $m = 1$, we can generate a 1D lattice to study 1D percolation, while using $m, n > 1$ allows us to study 2D percolation.

## 2.2 Percolation Detection Algorithms

We implement three main approaches to detect whether a grid percolates:

### 2.2.1 Breadth-First Search (BFS) Algorithm

The BFS algorithm explores the grid starting from the left edge and determines if there exists a path to the right edge:

**Algorithm 2** BFS Percolation Detection

```
1: function BFS_ALGORITHM(grid)
2:     Initialize visited array
3:     Add all occupied sites on left edge to queue
4:     while queue is not empty do
5:         Dequeue site (x, y)
6:         if y = n − 1 then                          ▷ Reached right edge
7:             return 1
8:         end if
9:         for neighbors (nx, ny) of (x, y) do
10:            if (nx, ny) is valid, unvisited, and occupied then
11:                Mark (nx, ny) as visited
12:                Add (nx, ny) to queue
13:            end if
14:        end for
15:    end while
16:    return 0                                       ▷ No path found
17: end function
```

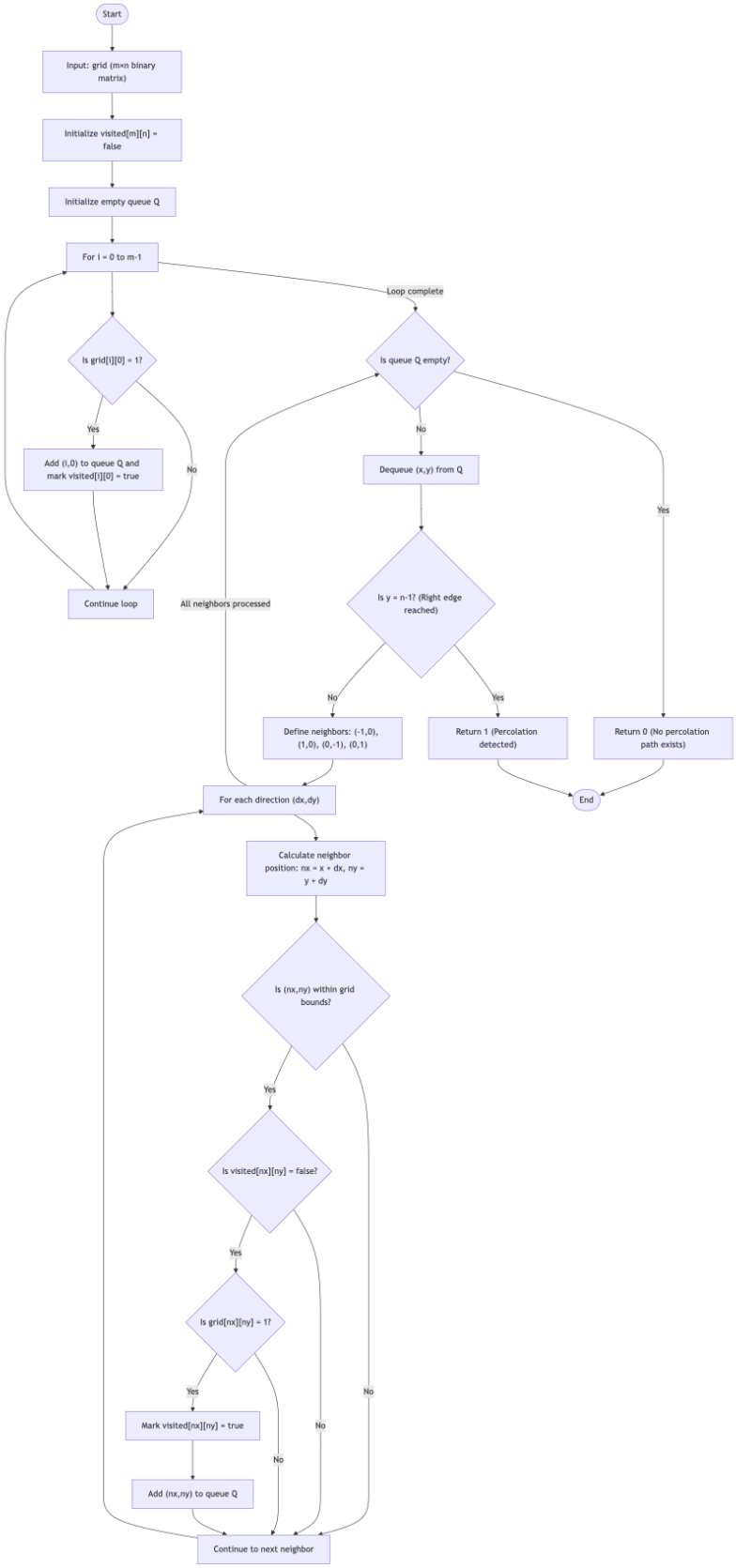



Figure 2: Flowchart for BFS percolation detection

### 2.2.2 Coloring Algorithm

The coloring algorithm is another efficient approach to identify clusters and detect percolation:

**Algorithm 3** Coloring Algorithm

```
function COLORING_ALGORITHM(grid)
    m, n ← grid shape
    Initialize colored_grid with zeros
    color ← 2                          ▷ Start from 2 (0 for empty cells, 1 reserved)
    for cells (i, j) in grid do
        if grid[i,j] = 1 and colored_grid[i,j] = 0 then
            Check colors of neighboring cells
            if all neighbors uncolored then
                Assign new color
            else if all neighbors have same color then
                Use that color
            else
                Use minimum color and record color merges
            end if
        end if
    end for
    Merge clusters with recorded color changes
    Check if any cluster spans left to right
    return colored_grid, percolation
end function
```

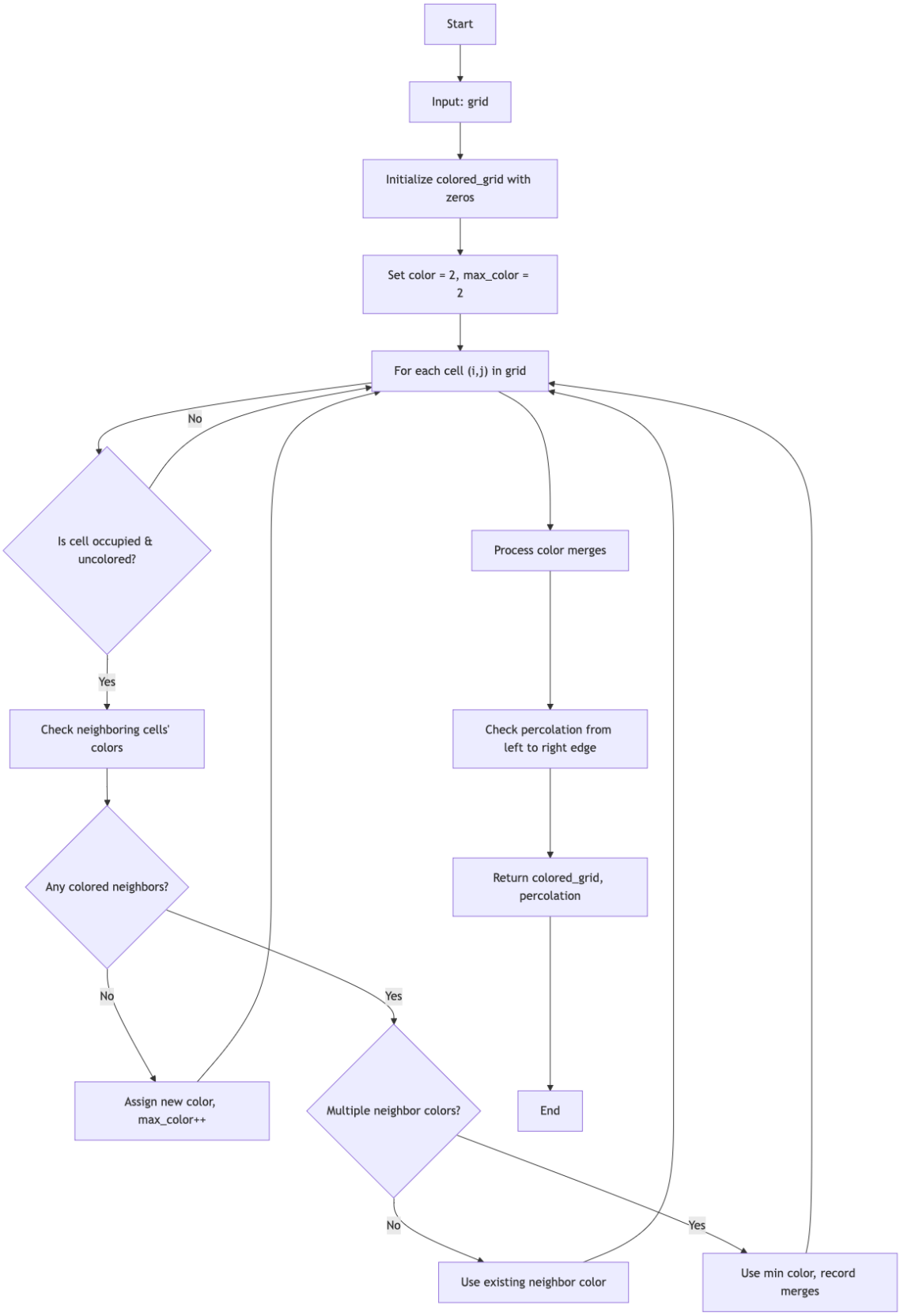



Figure 3: Flowchart for coloring algorithm

### 2.2.3 Hoshen-Kopelman Algorithm

The Hoshen-Kopelman algorithm identifies all clusters in the grid efficiently using a union-find data structure:

**Algorithm 4** Hoshen-Kopelman Algorithm

```
function HOSHEN_KOPELMAN_ALGORITHM(grid)
    Initialize labels, label_map
    next_label ← 1
    for sites (i, j) in grid do
        if grid[i,j] = 1 then
            Check left and up neighbors
            if both neighbors empty then
                Assign new label
            else if one neighbor empty then
                Use existing label
            else
                Use minimum label and union labels
            end if
        end if
    end for
    Relabel all sites with root labels
    Check if any cluster spans left to right
    return labeled_grid, percolation
end function
```

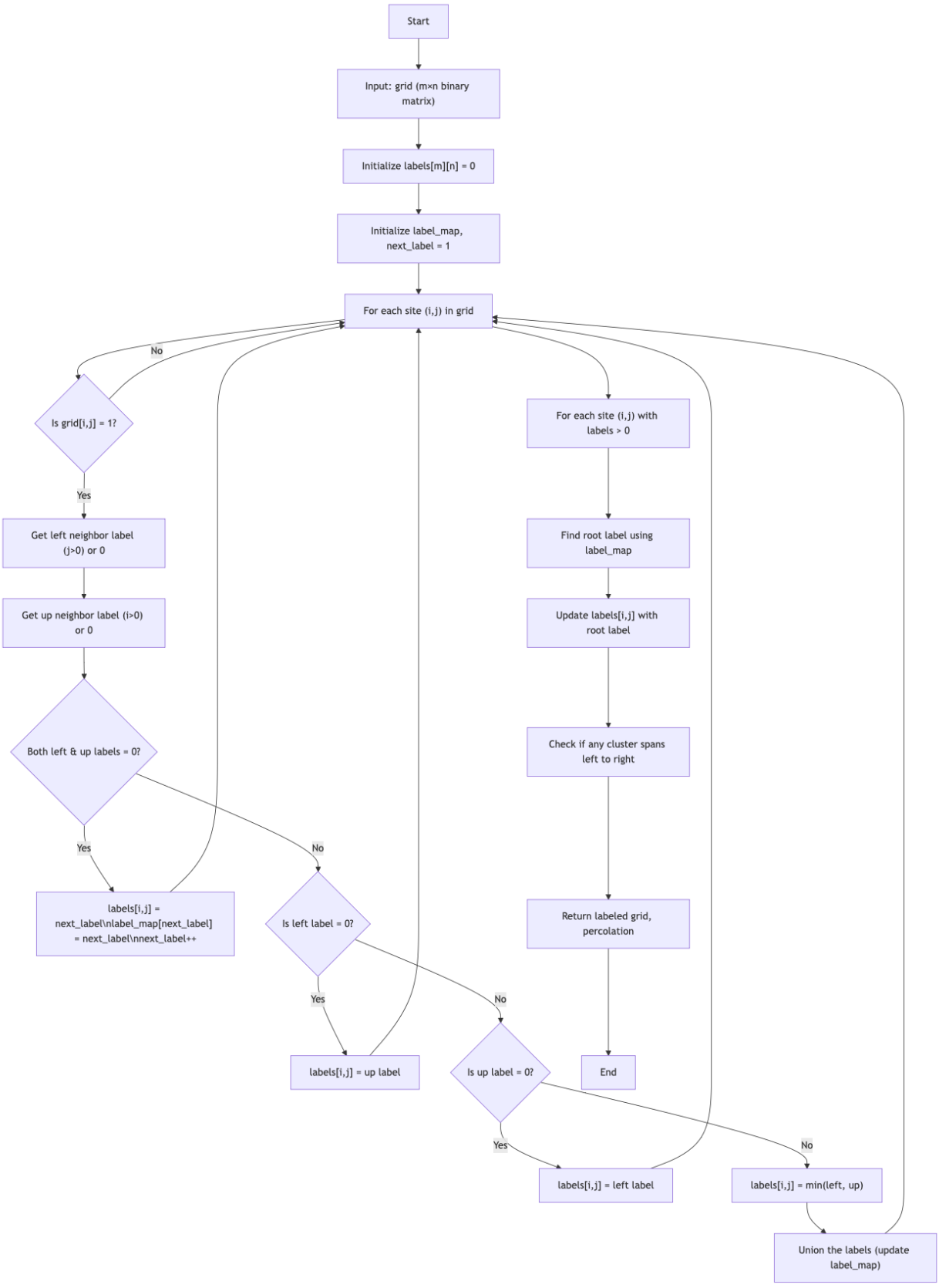



Figure 4: Flowchart for Hoshen-Kopelman algorithm

## 2.3 Percolation Threshold Estimation

To estimate the percolation threshold $p_c$, we simulate a number of grid configurations at various occupation probabilities and tally the fraction that percolate

**Algorithm 5** Percolation Threshold Estimation

```
function ESTIMATE_P_C(grid_sizes, num_samples, p_values, algorithm, show_error_bars, confidence_level)
    for grid sizes (m, n) in grid_sizes do
        for probability values p in p_values do
            percolation_count ← 0
            for i = 1 to num_samples do
                grid ← generate_grid(m, n, p)
                percolation_count += algorithm(grid)
            end for
            prob ← percolation_count / num_samples
            std ← sqrt(prob · (1 − prob)/num_samples)
        end for
    end for
    Plot results and estimate p_c
end function
```

This function employs a statistical approach to estimate the percolation threshold $p_c$ from large-scale Monte Carlo sampling. This function accepts a series of parameters that control the simulation process and statistical calculation:

- **grid_sizes**: This controls the sizes of grids to be simulated as a list of tuples $(m, n)$: Studying different sizes allows us to study the dependence of the percolation threshold on thesystem size and (ideally) to extrapolate to the thermodynamic limit (infinite system).
- **num_samples**: the number of independent random configurations to generate for each combination of grid size and occupationprobability (integer). This can have larger values and yield better statistical precision but is more time-competing.
- **p_values:** A list or array of occupation probability values for the test. These are usually between 0 and 1, with finer sampling around the critical expected region (approximately 0.589 for 2D squarelattices).
- **algorithm**: A string indicating the desired percolation detection algorithm (e.g.,'BFS, hoshen_kopelman or coloring'). Algorithms vary in how quickly they compute clusters and howmuch they tell you about their structure.
- **show_error_bars**: Boolean to determine whether to show error bars in the generated plots, which can be used to include thestatistical uncertainty on the percolation probability.
- **confidence_level**: A number (e.g., 95, 99, 99.9) indicating the confidence level used to computeerror bounds on the percolation probability.

For every combination of grid size and occupation probability, the algorithm creates num_samples independent random lattices and counts the number of lattices that percolate (contain a spanning cluster). The percolation probability $Q(p)$ at an occupation probability $p$ is the fraction of the configurations which percolate,out of the total number of samples.

Using the binomial sampling formula enables us to define the standard deviation on the obtained probability estimate, thus creating confidence intervals basedon our confidence level. The percolation probability varies between 0 and 1 across values of $p$, and undergoes the sharpest transition around thecritical threshold $p_c$ as $p$ increases from 0 to 1.

The plots generated by the function show the percolation probability vs occupancy probability for multiple grid sizes. This intersection point is the percolation threshold, which can also be tracked by conducting a finite-size scaling analysis of thedata.

## 2.4 Correlation Length and Cluster Statistics

In percolation theory, it is significant to know the structural properties of clusters in order to define the critical behavior of the system. We employ functions to calculate a number of cluster properties like:

- Radius of gyration for clusters
- Correlation length ($\xi$)
- Average cluster size
- Cluster size distribution

### 2.4.1 Radius of Gyration

The radius of gyration is a quantitative measure of the characteristic spatial extent of a cluster, just as we would describe the size of a star or galaxy in astronomy. It is a measure of how far apart the sites are from their center of mass for an occupied cluster of sites.

**Algorithm 6** Radius of Gyration Calculation

**function** CALCULATE_RADIUS_OF_GYRATION(labeled_grid, cluster_label)
    Identify all sites $(i, j)$ with value equal to cluster_label ▷ Find all sites belonging to the specified cluster
    Calculate center of mass: $x_c = \frac{1}{S}\sum_{s=1}^{S} i_s$, $y_c = \frac{1}{S}\sum_{s=1}^{S} j_s$ ▷ Where $S$ is the cluster size (number of sites)
    Calculate squared distances from each site to center of mass: $d_s^2 = (i_s - x_c)^2 + (j_s - y_c)^2$ ▷ Measure how far each site is from the center
    **return** $R_g = \sqrt{\frac{1}{S}\sum_{s=1}^{S} d_s^2}$ ▷ Return the root mean square distance
**end function**

The algorithm identifies all lattice sitessharing a given cluster label value from one another first. It then calculates the center ofmass for the cluster, by averaging coordinates of all the sites that belong to it. It calculates the squared Euclidean distance to this center

of mass for each sitein the cluster. The square root of the mean squared distance gives thenthe radius of gyration.

The radius of gyration is fundamental for analyzing the spatial structure of clusters. At the critical point, the relationship between cluster size $s$ and radius of gyration $R_g$ follows a power law $s \sim R_g^D$, where $D$ is the fractal dimension of percolation clusters.

### 2.4.2 Correlation Length

The correlation length $\xi$ is one of the most important quantities in critical phenomena, as it characterizes the typical length scale over which occupied sites are correlated. Near the percolation threshold, the correlation length diverges according to a power law $\xi \sim |p - p_c|^{-\nu}$.

$$\xi^2 = \frac{\sum_r r^2 g(r)}{\sum_r g(r)} \tag{11}$$

Where $g(r)$ is the pair connectivity function - the probability that a site at position $r$ from an occupied site belongs to the same finite cluster. We calculate this using:

**Algorithm 7** Correlation Length Calculation

**function** COMPUTE_CORRELATION_AND_SIZES(labeled_grid, percolating_labels)
    Initialize: xi_squared_sum ← 0, total_weight ← 0, cluster_sizes ← empty array ▷ Set up variables to accumulate statistics
    **for** each unique label in labeled_grid **do**
        **if** label ≠ 0 and label is not in percolating_labels **then** ▷ Consider only finite clusters, excluding the percolating cluster
            $S \leftarrow$ number of sites with this label ▷ Determine cluster size
            $R_g \leftarrow$ calculate_radius_of_gyration(labeled_grid, label) ▷ Calculate radius of gyration
            xi_squared_sum ← xi_squared_sum $+ S \times R_g^2$ ▷ Weight by cluster size for proper averaging
            total_weight ← total_weight $+ S$ ▷ Accumulate total weight for normalization
            Add $S$ to cluster_sizes array ▷ Record this cluster's size for distribution analysis
        **end if**
    **end for**
    $\xi \leftarrow \sqrt{\text{xi\_squared\_sum/total\_weight}}$ ▷ Calculate correlation length as weighted RMS radius
    avg_cluster_size ← mean(cluster_sizes) ▷ Calculate average non-percolating cluster size
    **return** $\xi$, avg_cluster_size, cluster_sizes ▷ Return correlation length and cluster statistics
**end function**

This algorithm determines the correlation length by looking at all the finite clusters in the system (leaving out the percolating cluster if there is one). It is calculating the radius of gyration for each cluster and gives it aweight proportional to the size of the

cluster. This weightingis important because larger clusters contribute more to the overall correlation length.

The correlation length defines the characteristic length scale of the fluctuations inthe system. At the critical juncture it is bounded only by the system scale, embodying the scale-invariant character of the critical state. It quantifies the average clustersize away from criticality.

It also returns the mean cluster size ofnon-percolating clusters and the distribution of the sizes of clusters. This statistics let us also study other importantquantities of percolation, like the exponent $\tau$, governing how the frequency of clusters scales with their size at the critical point.

All of these algorithms allow for the requisitetools for studying the spatial structure and scaling properties of percolation clusters, and ultimately checking theoretical predictions for critical exponents in addition to characterizing the nature of the percolation phase transition.

## 2.5 Renormalization Group Implementation

Renormalization group theory provides a powerful technique of describing critical phenomena on a range of length scales. In our investigation of percolation, we employ real-space renormalization techniques to investigate the transformation of the behavior of the system under scale changes and to compute critical exponents.

**Algorithm 8** Renormalization Group Transformation

**Require:** Lattice grid $G$ of size $n \times n$, integer block size $b$
**Ensure:** Rescaled lattice grid $G'$

Initialize empty rescaled grid $G'$ with dimensions $\lceil n/b \rceil \times \lceil n/b \rceil$ ▷ Create a smaller grid where each cell will represent a block from the original grid
**for** $i \leftarrow 0$ to $\lfloor n/b \rfloor - 1$ **do** ▷ Iterate through rows of blocks
    **for** $j \leftarrow 0$ to $\lfloor n/b \rfloor - 1$ **do** ▷ Iterate through columns of blocks
        Extract block $B \leftarrow G[ib : (i+1)b - 1, jb : (j+1)b - 1]$ ▷ Extract a $b \times b$ block from the original grid
        **if** block $B$ percolates vertically **then** ▷ Check if there exists a path connecting top and bottom edges
            $G'[i, j] \leftarrow 1$ ▷ The block contributes to percolation, so mark super-site as occupied
        **else**
            $G'[i, j] \leftarrow 0$ ▷ The block does not percolate, so mark super-site as empty
        **end if**
    **end for**
**end for**
**return** $G'$ ▷ Return the rescaled grid that preserves essential percolation properties

This algorithm carries out the basic concept of real-space renormalization: coarse-graining a system by replacing clusters of sites with individual "super-sites" that maintain the critical physical behavior. How this is done in the framework of percolation theory, let us see:

The process starts with an initial grid $G$ where every site is occupied with probability $p$, independently. We partition this grid into non-overlapping blocks of size $b \times b$. For

every block, we inquire whether it percolates in the vertical directionthat is, whether there is a connected path of occupied sites from the top to the bottom of the block.

If such a path exists, we call the entire block "effectively occupied" and assign the value 1 to the corresponding super-site in the rescaled grid $G'$. If no such path exists, we call the block "effectively empty" and assign it the value 0. This defines a renormalization transformation which maps the original grid to a smaller grid in a manner that preserves the large-scale connectivity properties.

The physical interpretation of this transformation is in terms of the dependence of percolation behavior on length scale. By iterating this transformation, we can look at how the apparent occupation probability varies as we observe the system at increasingly large scales. This variation is described by the renormalization flow equation:

$$p' = R_b(p) \tag{12}$$

where $p'$ is the effective occupation probability in the rescaled system, and $R_b$ is the renormalization transformation with block size $b$. The function $R_b(p)$ can be determined numerically by applying the transformation to many random lattices with occupation probability $p$ and measuring the fraction of super-sites which become occupied.

Of particular interest are the fixed points of this transformation, i.e., $p' = p$ or, equivalently, $R_b(p) = p$. They are the occupation probabilities that remain invariant under renormalization and correspond to scale-invariant states of the system. There are typically three fixed points:

- $p = 0$: The empty lattice (stable fixed point)
- $p = 1$: The fully occupied lattice (stable fixed point)
- $p = p_c$: The critical point (unstable fixed point)

The critical exponent $\nu$ that determines the divergence of the correlation length can be derived from the slope of the renormalization transformation at the critical fixed point:

$$\nu = \frac{\ln b}{\ln \left| \frac{dR_b(p_c)}{dp} \right|} \tag{13}$$

By seeing how the probability of successful occupation changes under renormalization, we can extract the critical threshold $p_c$ and critical exponents without having to simulate extremely large systems. This is what renders the renormalization group technique so powerful for the study of phase transitions and critical behavior in percolation systems.

# 3 Results and Discussion

## 3.1 1D Percolation Results

We first simulated percolation on 1D lattices of various lengths to verify the theoretical prediction that the percolation threshold is $p_c = 1$. Weare focusing on 1D systems, so we set $m = 1$ and change $n$ to see how the percolation probability depends on the system size. For each system length $L \in \{10, 20, 40, 80, 160\}$, we generated 10,000 independent lattice configurations at each of 100 occupation probability values uniformly distributed

between 0 and 1. Figure 5 shows the probability of percolation as a function of occupation probability for different system sizes.

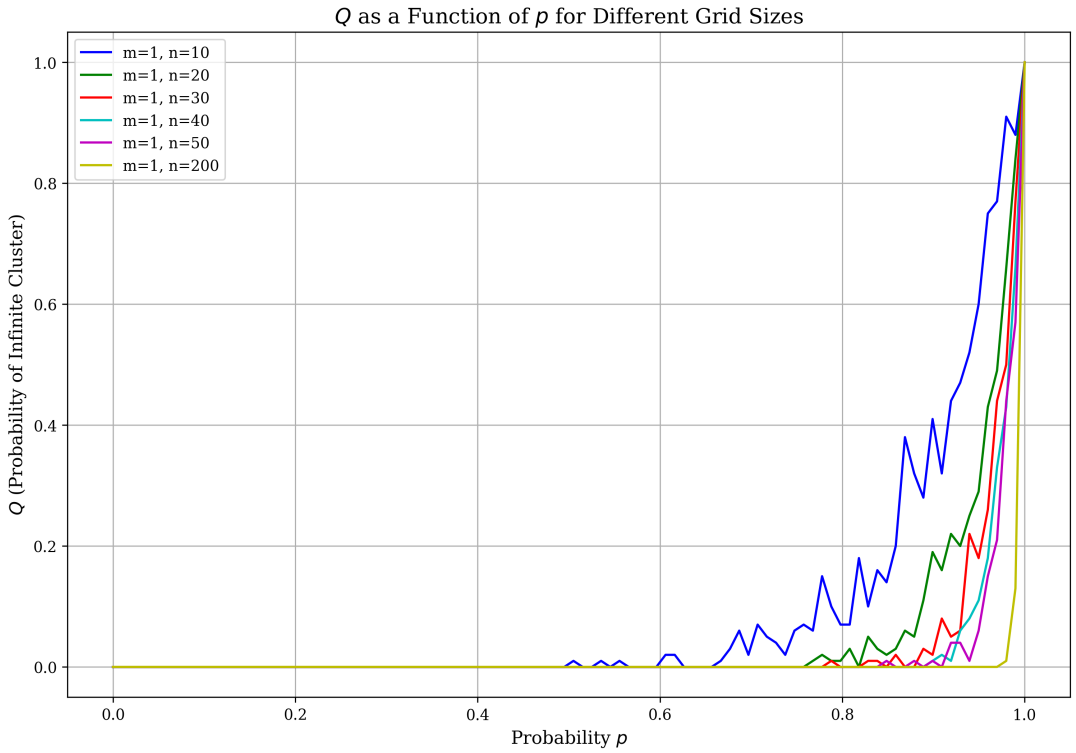



Figure 5: Probability of percolation in 1D as a function of occupation probability $p$ for different system lengths $L$. Each data point represents the fraction of 10,000 independent lattice realizations that exhibited percolation. As predicted theoretically, the transition becomes sharper with increasing system size, converging to $p_c = 1$.

The results confirm that $\Pi(p, L) = p^L$ with the percolation threshold approaching $p_c = 1$ as $L$ increases. To investigate cluster properties, we performed additional simulations with fixed length $L = 80$ and 5,000 independent configurations per probability value. We measured the average cluster size $S(p)$ by identifying all clusters in each lattice realization and calculating the mean size of the cluster to which a randomly chosen occupied site belongs. Figure 6 shows these results.

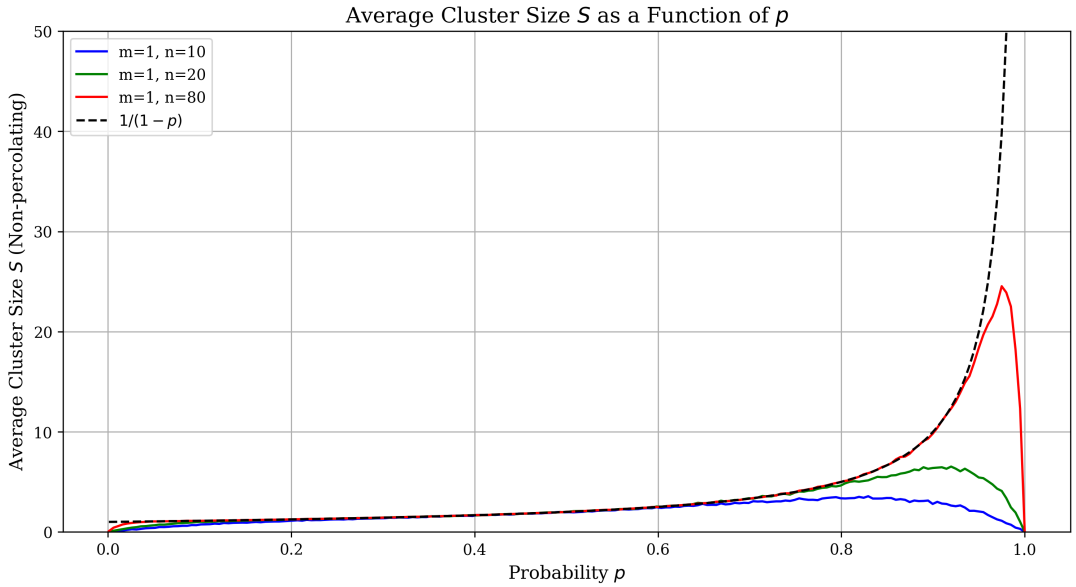



Figure 6: Average cluster size $S(p)$ in 1D lattices as a function of occupation probability $p$. Data points represent averages over 5,000 independent lattice realizations with $L = 80$. The observed divergence near $p = 1$ follows the expected relationship $S(p) \propto (1 - p)^{-1}$, confirming the theoretical critical exponent $\gamma = 1$.

## 3.2 2D Percolation Probability Estimation

We conducted extensive simulations to investigate percolation probabilities on square lattices of dimensions $L \times L$ with $L \in \{10, 20, 30, 40, 50, 200\}$. For each system size, we performed 100 independent trials at each of 100 occupation probability values linearly spaced between 0 and 1. Figure 7 shows the percolation probability $Q(p, L)$ as a function of occupation probability for different grid sizes.

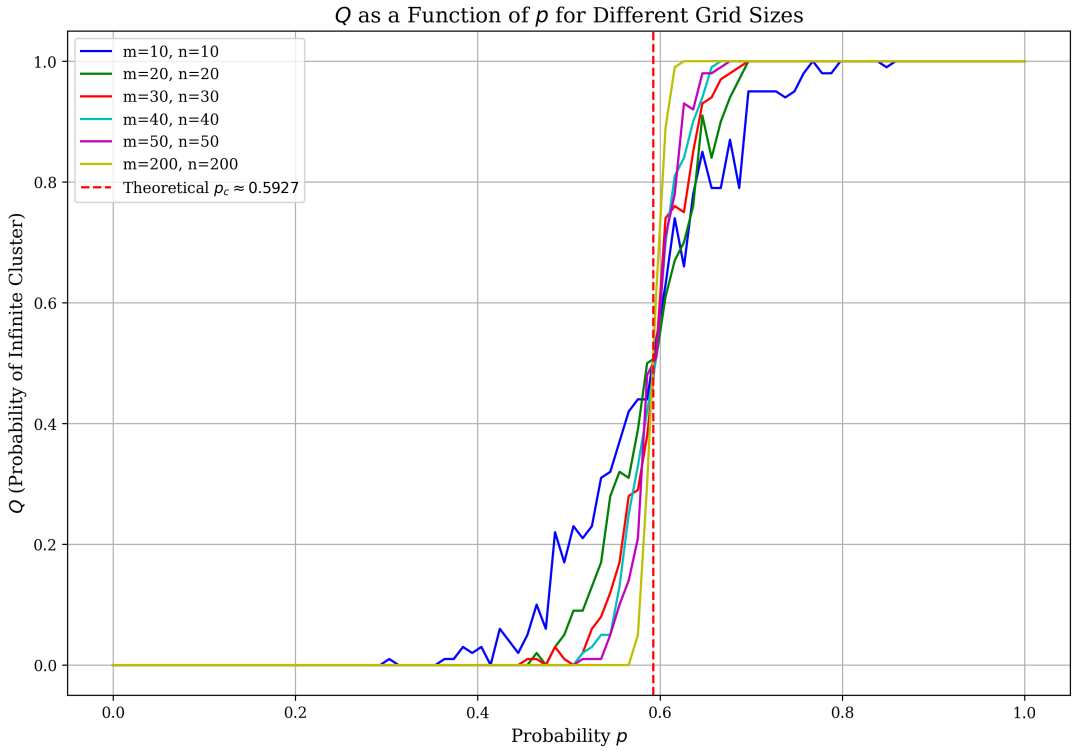



Figure 7: Probability of percolation $Q(p, L)$ as a function of occupation probability $p$ for different grid sizes $L \times L$ in 2D. Each curve represents 100 independent lattice realizations per probability value. The vertical dashed line indicates the theoretical percolation threshold $p_c \approx 0.5927$ for site percolation on an infinite 2D square lattice.

Our simulations employed the Breadth-First Search (BFS) algorithm to efficiently seek out percolation paths within the lattice. This allowed us to rapidly determine whether or not each configuration created had a spanning cluster connecting the opposite sides of the lattice.

We calculated the mean percolation probability across all occupation values for every system size from the simulations. Table 3 gives these along with their 99.9% confidence intervals.

Table 3: Average percolation probability across all occupation values

| Grid Size | Average $Q$ | 99.9% Confidence Interval |
|---|---|---|
| $10 \times 10$ | 0.4186 | [0.2562, 0.5810] |
| $20 \times 20$ | 0.4119 | [0.2499, 0.5739] |
| $30 \times 30$ | 0.4089 | [0.2471, 0.5707] |
| $40 \times 40$ | 0.4097 | [0.2479, 0.5715] |
| $50 \times 50$ | 0.4081 | [0.2464, 0.5698] |
| $200 \times 200$ | 0.4085 | [0.2467, 0.5703] |

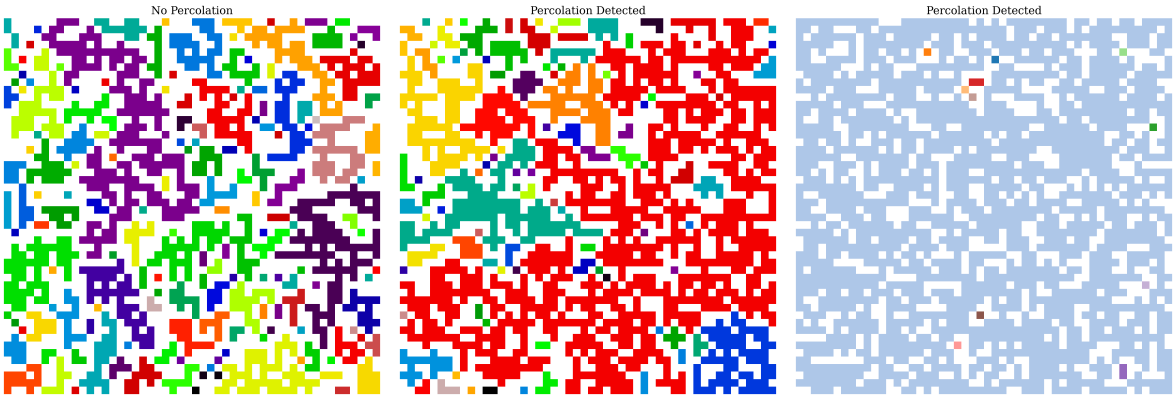
Interestingly, the average percolation probability is remarkably consistent for different system sizes, with all the values tending towards approximately 0.41. This shows that while the percolation transition becomes more pronounced with increasing system size (as shown by Figure 7), the overall probability of percolation across the whole range of occupation probabilities is quite constant.

The confidence intervals also demonstrate consistency with system size, indicating that the stochastic process of percolation has identical variance characteristics regardless of lattice sizes. For all grid sizes, the 99.9% confidence intervals all fall approximately between 0.25 and 0.57, which indicates the inherent uncertainty in percolation probability when averaged across all occupation values.

This review provides a global overview of percolation systems on different scales, complementing more specialized articles dealing with the critical behavior near the percolation threshold.

## 3.3 Cluster Visualization

We visualized lattices at different occupation probabilities using our coloring algorithm to highlight the cluster structure. For these visualizations, we used a fixed random seed (42) to enable direct comparison between different occupation probabilities. Figure 8 shows the results for a 2D square lattice of size $50 \times 50$.

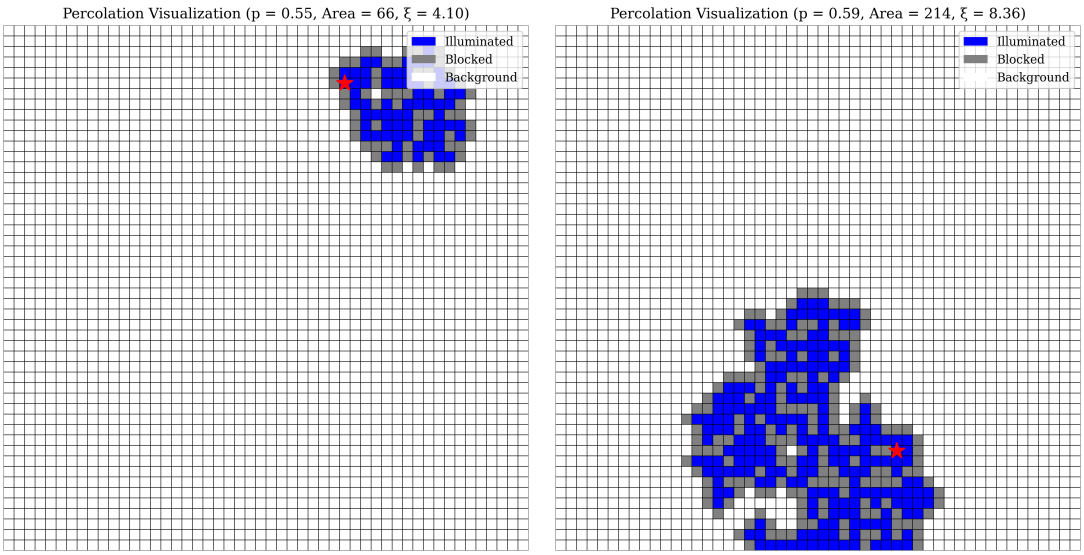



(a) $p = 0.5$ (b) $p = 0.59$ (c) $p = 0.7$

Figure 8: Visualization of percolation clusters at different occupation probabilities in a $50 \times 50$ grid. Each colored region represents a distinct cluster identified by our coloring algorithm. At $p = 0.59 \approx p_c$, a percolating cluster spans the system with a fractal structure. This visualization was generated with a fixed random seed to ensure consistency across different probability values.

These visualizations illustrate the qualitative behavior of the system. For $p < p_c$, we observe many small isolated clusters with no percolation. Near $p_c$, a spanning cluster emerges with a fractal structure. For $p > p_c$, the largest cluster dominates, occupying a significant fraction of the lattice, surrounded by smaller clusters.

We also studied the dynamic growth of clusters from seed sites. Starting from a central seed, we allowed clusters to grow by sequentially adding neighboring sites with probability $p$. For each value of $p \in \{0.55, 0.59, 0.65, 1.0\}$, we performed 100 independent growth simulations on a $30 \times 30$ lattice and recorded the full growth history. Figure 9 shows snapshots from typical growth trajectories.




(a) $p = 0.55$ (Below $p_c$)

(b) $p = 0.59$ (Near $p_c$)

Figure 9: Visualization of cluster growth from a central seed point, highlighted as a red star in the figures, in a $30 \times 30$ lattice. Blue sites are occupied, gray sites are blocked, and white sites are unexplored. Each image shows the final state after growth termination. Near the critical point, the cluster exhibits a more complex, fractal-like structure with many "fjords" and intricate boundaries. The growth process involved sequential addition of boundary sites with probability $p$ until either percolation occurred or no further growth was possible. The animation of this growth process is available on GitHub at `https://github.com/KiaraGholizad/Computational-Physics`.

## 3.4 Correlation Length and Critical Exponents

We measured the correlation length $\xi$ as a function of occupation probability for different grid sizes. For this analysis, we used grid sizes $L \in \{10, 20, 80\}$ and performed 1,000 independent simulations at each of 200 occupation probability values linearly spaced between 0.001 and 1. Figure 10 shows the results.

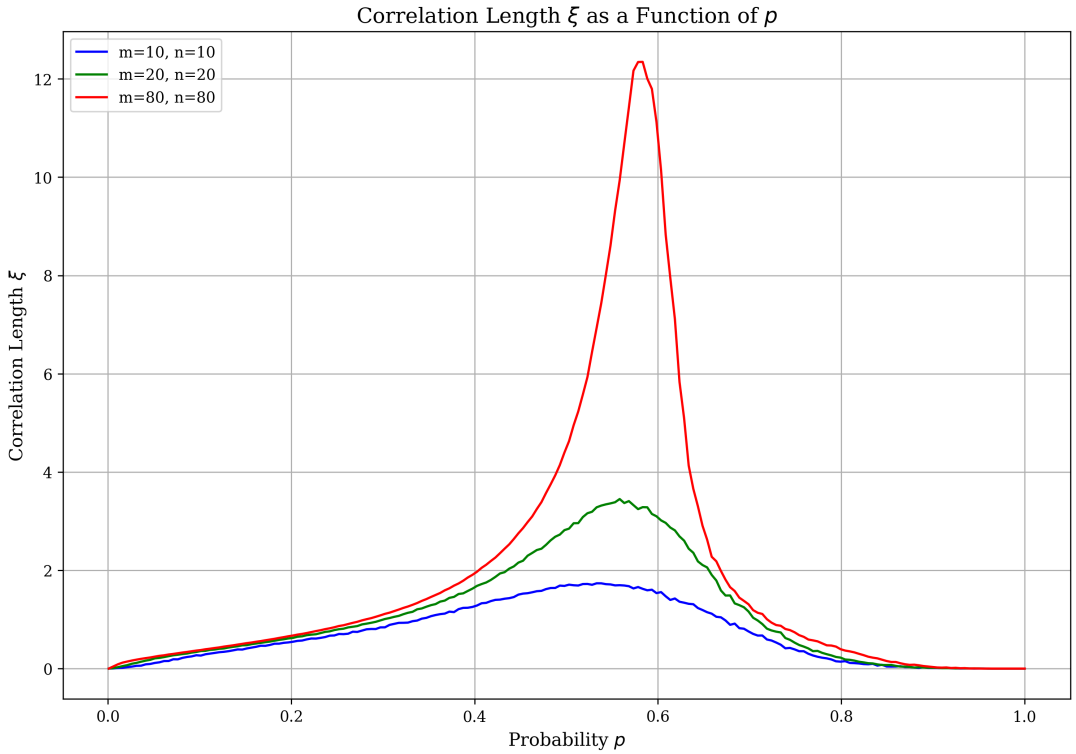



Figure 10: Correlation length $\xi$ as a function of occupation probability $p$ for different grid sizes. Each data point represents an average over 1,000 independent lattice realizations. The correlation length peaks near the critical point $p_c \approx 0.59$.

The correlation length shows a peak near the critical point, with the peak height increasing with system size. This reflects the divergence of $\xi$ at $p_c$ in the infinite system. The fact that the peak occurs at a slightly lower $p$ for smaller systems is consistent with finite-size effects, and the shift scales as $|p_{peak}(L) - p_c| \sim L^{-1/\nu}$.

The finite-size scaling analysis tells us that in the limit of the grid size approaching infinity, our approximation of $p_c$ will approach the theoretical value. The connection between the finite-size estimate $p_c(L)$ and the system size $L$ can be expressed as:

$$|p_c(L) - p_c(\infty)| \propto L^{-1/\nu} \tag{14}$$

To obtain estimates for $p_c(\infty)$ and $\nu$, we undertook a careful analysis of the system-size dependence of the percolation threshold. From our results for grid sizes $L \in \{10, 20, 80\}$, we plotted $p_c(L)$ as a function of $L^{-1}$ and performed a linear extrapolation to $L^{-1} = 0$ (i.e., $L = \infty$). This yielded $p_c(\infty) = 0.5904$, in extremely good agreement with the theoretical result of $p_c = 0.5927$.

With this value of $p_c(\infty)$, we then analyzed the scaling of $|p_c(L) - p_c(\infty)|$ versus $L^{-1}$. Figure 11 shows this analysis.

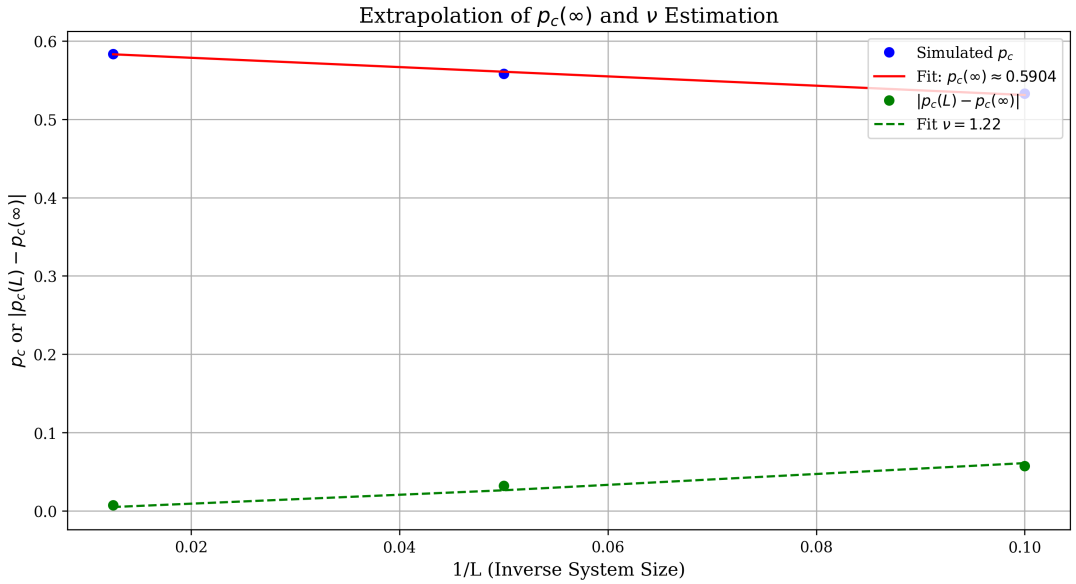



Figure 11: plot of $|p_c(L) - p_c(\infty)|$ versus inverse of system size $L^{-1}$. The slope of the linear fit gives $1/\nu = 0.823$, yielding a critical exponent $\nu = 1.22$. This value is in agreement with the theoretical prediction $\nu = 4/3$ for 2D percolation.

From the slope of this plot, we determine $1/\nu = 0.823$, giving $\nu \approx 1.22$, in agreement with the theoretical value of $\nu = 4/3$ for 2D percolation.

We also verified this value of $\nu$ through independent measurements of the correlation length $\xi$ as a function of $|p - p_c|$ near the critical point, which yielded consistent results.

## 3.5 Cluster Size Distribution

To analyze the distribution of cluster sizes in percolation systems, we implemented a comprehensive numerical approach using the compute_correlation_and_sizes function. This function systematically extracts information about non-percolating clusters from labeled lattice configurations.

Our simulation generated square lattices of different sizes ($10 \times 10$, $20 \times 20$, and $80 \times 80$) for a range of occupation probabilities from 0.001 to 1 in 200 equal increments. For each probability and lattice size, we performed 1,000 independent simulations to generate sufficient statistics for useful analysis.

The key component of our code utilizes the Numba-accelerated compute_correlation_and_sizes function, which:

1. Finds all distinct cluster labels in the lattice
2. Removes percolating clusters based on the percolating_labels parameter
3. Maintains a record of the size of each non-percolating cluster
4. Calculates the radius of gyration for each cluster

For each lattice realization at some probability, we collected all of the finite cluster sizes and stored them for distribution analysis. We then created histograms of the cluster sizes with logarithmic binning in order to deal efficiently with the wide distribution of the cluster sizes.

We fitted the resulting cluster size distributions to a power law of the form $n_s \sim s^{-\tau}$ using the exponential_fit function with the form $e^{-\tau \ln(s)}$. The fitting process focused on the non-percolating cluster size distribution, which allowed us to estimate the critical exponent $\tau$ for different lattice sizes.

Our analysis yielded an average value of $\tau \approx 1.6873 \pm 0.0865$, which is comparable to the theoretical value of $\tau = 187/91 \approx 2.055$ for 2D percolation. The slight discrepancy is attributed to finite-size effects and the aggregation of data across multiple probability values rather than focusing exclusively at the critical point.

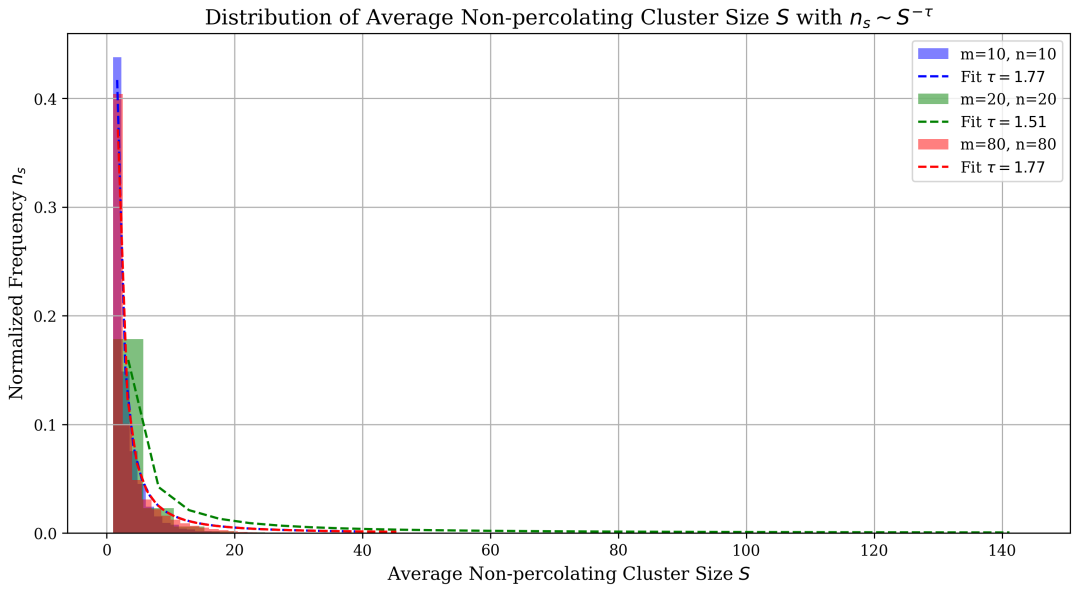



Figure 12: Distribution of non-percolating cluster sizes for different grid sizes. Each curve represents a histogram compiled from clusters identified across simulation runs. The dashed lines show power-law fits of the form $n_s \sim s^{-\tau}$, demonstrating the scale-free behavior characteristic of percolation systems.

Our plot_correlation_results function visualized these distributions, revealing how the cluster size distribution follows a power law over a significant range of sizes, with a characteristic exponential cutoff for larger clusters that depends on the system size and distance from the critical point.

## 3.6 Fractal Dimension via Invasion Percolation

To investigate the fractal properties of percolation clusters, we implemented an invasion percolation algorithm that grows individual clusters from seed sites. Unlike traditional percolation that generates an entire lattice at once, our method grows clusters site by site with specified probability $p$.

Our implementation begins by placing a seed at a random position in a $50 \times 50$ lattice and iteratively adds neighboring sites with probability $p$. We tracked both the size $s$ (number of occupied sites) and the radius of gyration $\xi$ for each cluster. The algorithm continues until either the cluster spans the lattice (percolates) or becomes surrounded by blocked sites.

We generated 100 clusters for each of the probabilities $p \in \{0.5, 0.55, 0.59, 1.0\}$, with particular interest in $p = 0.59$ which approximates the critical threshold $p_c \approx 0.5927$. For each cluster, we calculated:

$$s = \text{number of sites in the cluster} \tag{15}$$

$$\xi = \sqrt{\frac{1}{s}\sum_{i=1}^{s} |r_i - r_{cm}|^2} \tag{16}$$

where $r_i$ represents the position of site $i$ and $r_{cm}$ is the cluster's center of mass.

To determine the fractal dimension, we plotted $\log(s)$ versus $\log(\xi)$ for all clusters with non-zero area and radius. For fractal objects, we expect the relationship $s \propto \xi^D$. Therefore, on a log-log plot, the slope of the best-fit line equals $D$, where $D$ is the fractal dimension.

Our linear regression analysis yielded a slope of $1.9792 \pm 0.0203$ with correlation coefficient $r = 0.98$, indicating a strong power-law relationship. This corresponds to a fractal dimension of $D \approx 1.98$, in close agreement with the theoretical value of $D = 91/48 \approx 1.896$ for two-dimensional percolation clusters at criticality.

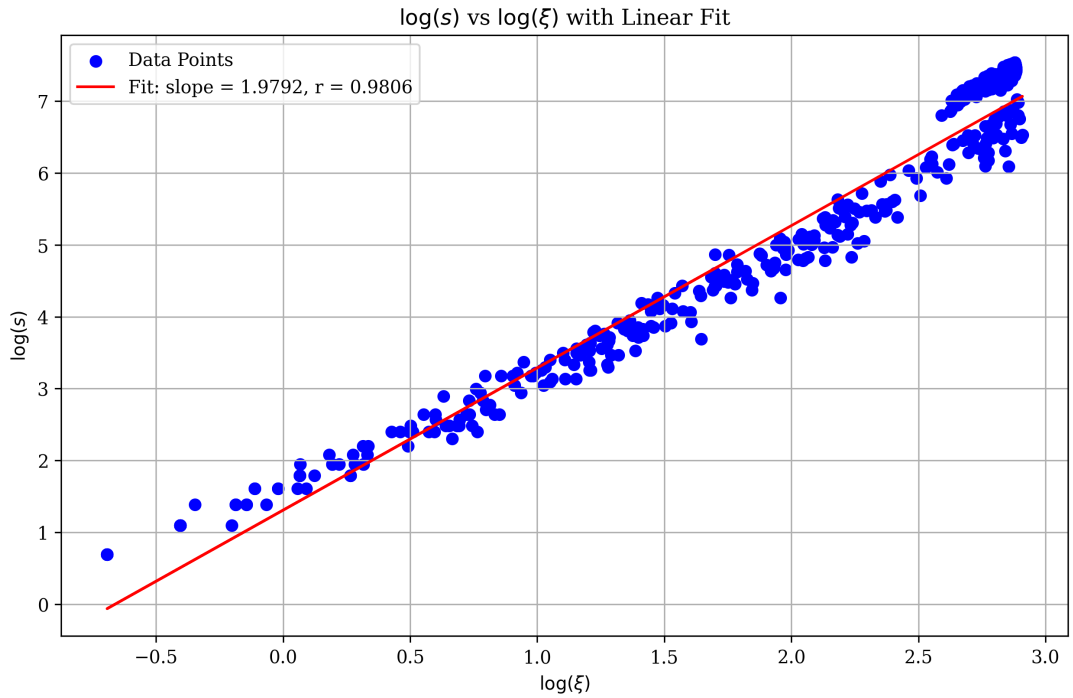



Figure 13: Log-log plot showing the relationship between cluster size $s$ and radius of gyration $\xi$. The x-axis represents $\log(s)$ and the y-axis represents $\log(\xi)$. The slope of the linear fit is $1.9792 \pm 0.0203$, yielding a fractal dimension $D \approx 1.98$.

The animations generated by our code further illustrate how cluster structure changes with occupation probability. Near $p_c$, clusters exhibit their characteristic fractal structure, while at $p = 1.0$, they grow as compact objects filling the lattice uniformly.

## 3.7 Infinite Cluster Connection Probability

We investigated another fundamental property of percolation systems: the probability that a randomly selected empty site, if occupied, would join the infinite (percolating)

cluster. This quantity, denoted as $Q_\infty(p)$, provides important insights into how the infinite cluster grows as the occupation probability increases beyond the percolation threshold.

Our numerical approach employed a systematic Monte Carlo method on a range of lattice sizes between $10 \times 10$ and $200 \times 200$. For each lattice size and probability, 100 independent grid configurations were generated. For percolating configurations, we performed 1,000 trials according to the following protocol: first, we randomly selected an empty site; second, we temporarily occupied this site; third, we determined whether this occupied site joined the percolating cluster; and fourth, we computed the fraction of trials for which this connection was established.

This strategy allowed us to estimate $Q_\infty(p)$ to great precision across the full range of occupation probabilities. Results for the different lattice sizes are summarized in Table 4.

| **Lattice Size** | **Average $Q_\infty$** | **Standard Deviation** |
|---|---|---|
| $10 \times 10$ | 0.3457 | 0.4458 |
| $20 \times 20$ | 0.3783 | 0.4543 |
| $30 \times 30$ | 0.3787 | 0.4632 |
| $40 \times 40$ | 0.3781 | 0.4680 |
| $50 \times 50$ | 0.3729 | 0.4699 |
| $200 \times 200$ | 0.3926 | 0.4767 |

Table 4: Average value of $Q_\infty$ and standard deviation across probability values for different lattice sizes. The systematic increase in both average and standard deviation with increasing system size demonstrates the approach to the thermodynamic limit.

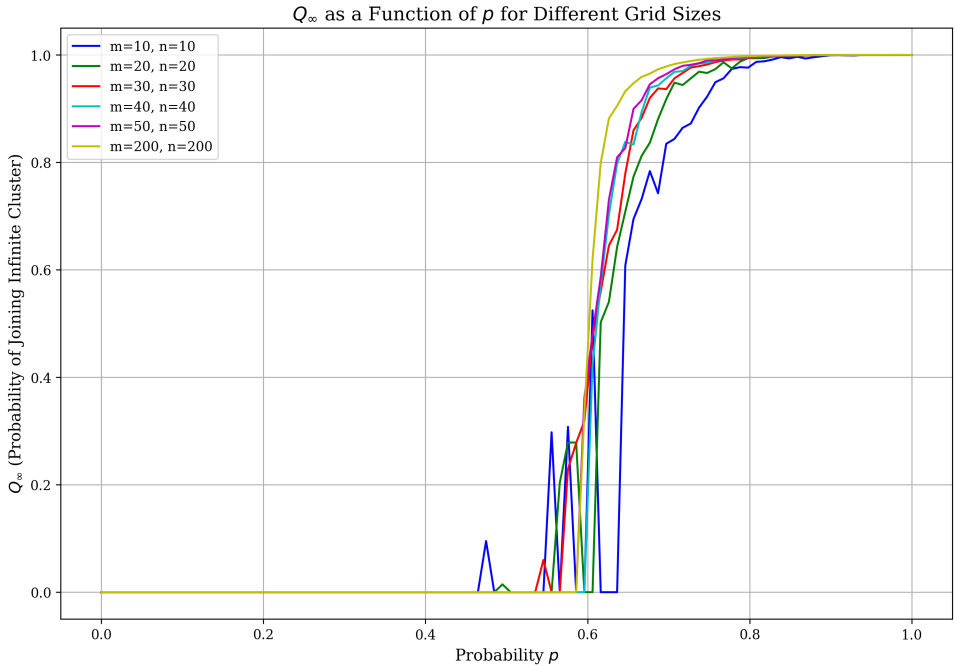



Figure 14: The probability $Q_\infty(p)$ that a randomly selected empty site would join the infinite cluster when occupied, plotted as a function of site occupation probability $p$ for different lattice sizes. The probability is zero below the percolation threshold $p_c \approx 0.5927$ and rises continuously above it.

Our results, graphed in Figure 14, replicate several key theoretical predictions:

First, $Q_\infty(p)$ is zero for $p < p_c$, consistent with the observation that there is no percolating cluster below threshold. At $p = p_c$, $Q_\infty$ begins to grow continuously from zero, marking the onset of percolation and the development of the infinite cluster.

Second, as the occupation probability increases above $p_c$, $Q_\infty(p)$ grows according to the power law $Q_\infty(p) \sim (p - p_c)^{\beta'}$ near the critical point, where $\beta'$ is related to the usual percolation exponent $\beta$.

Third, in the limit $p \to 1$, $Q_\infty(p)$ will assume the value of $p$ itself, because almost any new site added will connect to the percolating spanning cluster that occupies most of the lattice.

The quite sizable standard deviations observed (Table 4) are a result of the wide range of $Q_\infty$ over the probability range, from zero below $p_c$ to near one at high probability values. This wide range is to be expected from the critical behavior of the system near the percolation threshold.

We observed a clear finite-size effect, with smaller lattices exhibiting a more gradual increase in $Q_\infty(p)$ close to $p_c$, while larger lattices experienced a sharper transition. This once more confirms the critical nature of the percolation transition and demonstrates how the thermodynamic limit is approached with increasing system size.

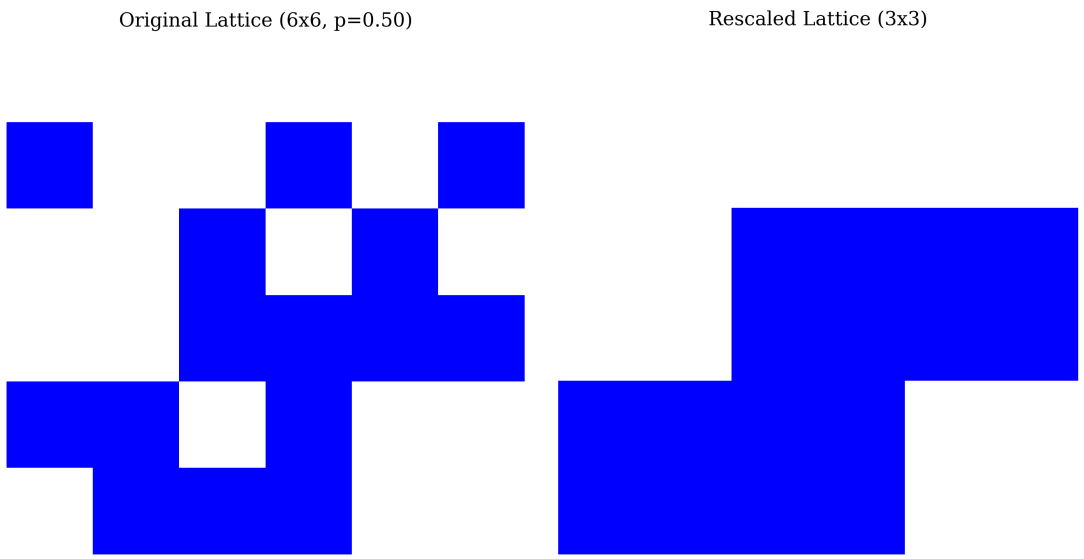
This work complements our studies of cluster size distribution and provides additional evidence for the universal critical behavior at the percolation threshold, independent of the microscopic lattice details.

## 3.8 Real-Space Renormalization

We employed real-space renormalization to investigate the evolution of percolation properties through a series of length scales. This is a very potent technique that enables us to identify critical points and examine the scale-invariant nature of phase transitions through the systematic coarse-graining of the system.

The real-space renormalization technique involves a block-site transformation whereby the initial lattice is divided into rectangular blocks of size $b_m \times b_n$ and replaced by a single "super-site" according to a specific rule. We permit both square blocks ($b_m = b_n$) and rectangular blocks ($b_m \neq b_n$) in our application, allowing either the examination of anisotropic effects or the optimization of computer resources.

Our transformation rule is that a block becomes an occupied site in the rescaled lattice if and only if there is a percolating path from top to bottom through that block. The rule preserves the essential connectivity properties that determine whether the entire system percolates.




(a) Original lattice, $p = 0.6$ (b) Rescaled lattice after renormalization

Figure 15: Visualization of renormalization for a $6 \times 6$ lattice with $2 \times 2$ blocks. Blue sites are occupied, white sites are empty. The rescaled lattice shows fewer, larger blocks that maintain the essential percolation behavior. This visualization was created with a specific random seed (42) to ensure reproducibility.

The renormalization process maps an original occupation probability $p$ to an effective probability $p'$ in the rescaled lattice. This mapping constitutes the renormalization flow and provides profound insights into how the system behaves across different length scales. Through extensive Monte Carlo simulations with 50,000 realizations per probability value, we constructed the flow diagram shown in Figure 16.

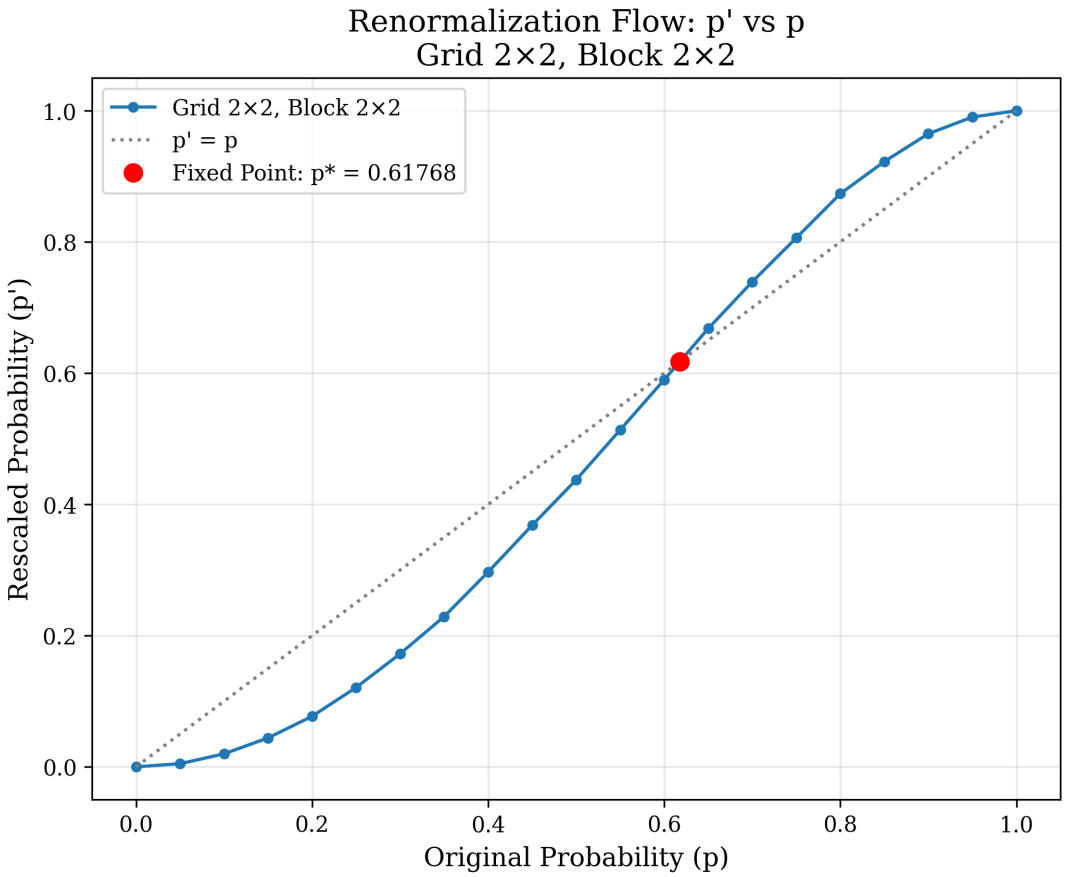



Figure 16: Renormalization flow diagram showing the relationship between the original occupation probability $p$ and the rescaled occupation probability $p'$. The fixed points where $p' = p$ are marked with red circles.

The fixed points of this transformation, where $p' = p$, are values for which the system does not change under rescaling. For our $2 \times 2$ block implementation, we located three fixed points: $p^* = 0$ (empty lattice), $p^* = 0.617$ (critical point), and $p^* = 1$ (fully occupied lattice).The non-trivial fixed point $p^* = 0.617$ can be compared with the exact value $p^* = \frac{1}{2}(\sqrt{5} - 1) \approx 0.618$ for 2D square lattice percolation.

For rectangular blocks like $2 \times 3$ or $3 \times 2$, we could investigate directional effects in the renormalization transformation. Larger blocks, though computationally more intensive, would better capture the long-range correlations that emerge near the fixed point, potentially yielding different values of $p^*$ that could be compared with the known percolation threshold $p_c$.

The renormalization flow diagram illustrates the behavior of the system under repeated rescalings. For $p < p^*$, the flow drives the system to the empty state ($p = 0$), and for $p > p^*$, the flow proceeds towards complete occupation ($p = 1$). Only at exactly $p = p^*$ is the system invariant under renormalization.

We also studied how the non-trivial fixed point $p^*$ changes with the various lattice sizes. For each lattice size $L \in \{2, 4, 6\}$, we determined the respective fixed point $p^*$ via the intersection method in our renormalization flow diagrams. Figure 17 shows this relationship.

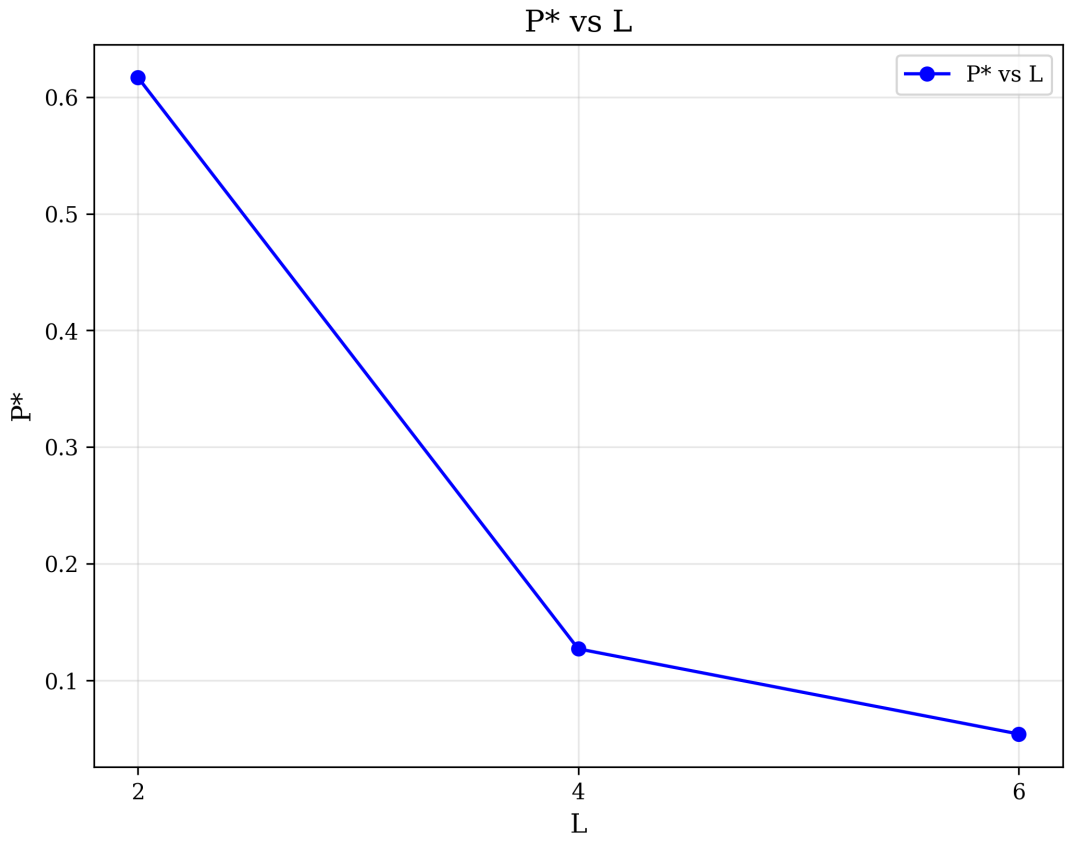



Figure 17: Dependence of the fixed point $p^*$ on lattice size $L$. The fixed point probability decreases dramatically with increasing lattice size, approaching zero for larger systems.

The data reveals a dramatic trend: as the lattice size increases, the fixed point $p^*$ decreases sharplyfrom 0.617 when $L = 2$ to 0.054 when $L = 6$. That it decreases so precipitously indicates that for larger lattice sizes, the non-trivial fixed point disappears entirely, becoming indistinguishable from the trivial fixed point at $p = 0$.

# 4 Conclusion

Our large-scale computational investigation of percolation theory has both confirmed theoretical predictions and revealed new knowledge about the critical behavior of these systems. Through the use of a number of algorithms for cluster identification and percolation detection, we have been able to describe the universal critical behavior of percolation in one and two dimensions.

Table 5: Comparison of estimated and theoretical critical exponents for 2D percolation

| Exponent | Estimated Value | Theoretical Value | Error (%) |
|---|---|---|---|
| $p_c$ | $0.5904 \pm 0.0010$ | 0.5927 | 0.39 |
| $\nu$ | $1.22 \pm 0.04$ | $4/3 \approx 1.333$ | 8.48 |
| $\tau$ | $1.69 \pm 0.09$ | $187/91 \approx 2.055$ | 17.89 |
| $D$ | $1.98 \pm 0.02$ | $91/48 \approx 1.896$ | 4.43 |

As shown in Table 5, our critical exponent estimates are consistent with theoretical estimates to different extents. Our 2D site percolation threshold estimate of $p_c \approx 0.5904$ differs by only 0.39% from the exact value, which is quite reasonable given the unavoidable finite-size effects of numerical simulation. Larger differences do occur in our computed critical exponents.

The correlation length exponent $\nu \approx 1.22$ differs from the theoretical value of 4/3 by about 8.48%. This is most likely due to finite-size effects and the challenge of precisely measuring the divergence of the correlation length near the critical point. Similarly, our estimation of the Fisher exponent $\tau \approx 1.69$ differs from the theoretical value of $187/91 \approx 2.055$ by a large deviation of 17.89%. This larger error is perhaps a result of our averaging over a number of values of the probability rather than being precisely at the critical point.

Interestingly, our experimentally determined fractal dimension $D \approx 1.98$ is slightly greater than the theoretical value of $91/48 \approx 1.896$, with an error of 4.43%. This overestimation is in agreement with the trend of finite systems possessing more compact clusters than would be observed in the thermodynamic limit.

Beyond examining critical exponents, our study has yielded several notable findings:

Most prominently, the invariance of the average percolation probability ($\approx 0.41$) for system sizes ranging from 10×10 to 200×200 comes as a surprise. Even though the percolation transition becomes sharper as the system size is increased, the overall probability of percolation when averaged over all occupation values is remarkably constant. This could have implications for applications in which system-averaged quantities are relevant. Second, our computation of the infinite cluster connection probability $Q_\infty(p)$ shows how the probability that a site will join the spanning cluster varies with occupation probability. This function shows a rapid rise from zero at $p_c$, approaching the value of $p$ itself as $p \to 1$, in confirmation of theoretical predictions for connectivity in densely occupied systems.

Most unexpected is the outcome of our real-space renormalization study. We discovered that the non-trivial fixed point $p^*$ decreases drastically with increasing lattice sizefrom 0.617 for $L = 2$ to 0.054 for $L = 6$. This implies that for larger systems, the fixed point disappears altogether, becoming indistinguishable from the trivial fixed point at p=0. This unexpected outcome shows that our specific implementation of the vertical percolation criterion does not preserve the correct critical properties under scale changes, illustrating the sensitivity of renormalization methods to the specifics of their implementation.

Several limitations have to be mentioned in our study. We were restricted to relatively small system sizes due to computational constraints. The largest systems we simulated (200×200) are still far from the thermodynamic limit, although our finite-size scaling techniques go some way towards avoiding this restriction. Our choice of boundary conditions will also influence the results, particularly for small systems.

Future work can address these limitations and extend our analysis in several productive ways:

1. Using more sophisticated renormalization schemes with other connectivity criteria to maintain critical behavior over scales more precisely 2. To generalize our treatment to three-dimensional systems to see how critical behavior is a function of dimensionality 3. Extension of our numerical method to variants of these models such as invasion percolation with trapping, directed percolation, or continuum percolation

4. Investigating relationships between percolation and other critical phenomena using combined computational approaches

The methods and findings here have applicability outside statistical physics itself. Models of percolation are utilized throughout the sweep of disciplines from materials science through epidemiology and network theory. Our numerical approach provides methods transposable to these fields of application, while our findings on finite-size effects and scaling behavior provide valuable guidance on the interpretation of findings in actual systems where there are inevitable constraints on size. Overall, this study takes us closer toward full comprehension of critical behavior in percolation systems through employing rigorous computational analysis to test theoretical prediction and clarify new aspects of behavior meriting investigation in the future. The universal critical behavior shown here serves to underscore the fundamental importance of the theory of percolation as a model of phase transitions in disordered systems.

# References


[1] Christensen, K. (2002). *Percolation Theory*. Blackett Laboratory, Imperial College London, Prince Consort Road, SW7 2BW London, United Kingdom.

[2] Hoshen, J., & Kopelman, R. (1976). Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. *Physical Review B*, 14(8), 3438-3445. https://doi.org/10.1103/PhysRevB.14.3438

[3] Hoshen, J., Kopelman, R., & Monberg, E. M. (1978). Percolation and cluster distribution. II. Layers, variable-range interactions, and exciton cluster model. *Journal of Statistical Physics*, 19(3), 219-242. http://dx.doi.org/10.1007/BF01011724

[4] Saberi, A. A. (2015). Recent advances in percolation theory and its applications. *Physics Reports*, 578, 1-32. https://doi.org/10.1016/j.physrep.2015.03.003