COUNTING CLUSTERS: INVESTIGATING UNIVERSALITY IN CRITICAL PERCOLATION

A Senior Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Arts with Honors in the Integrated Science Program

by

Sam Frank

Advisor: István Kovács

Northwestern University

May 2023

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my collaborators and mentors Professor István Kovács and Dr. Helen Ansell. It is difficult to imagine so much progress being made on the cluster tomography project without their sincere dedication and guidance, and I would especially like to acknowledge Helen's extensive feedback on various aspects of this thesis.

I must also emphasize István's role in my academic career. As a newly hired assistant professor, he took a chance on a first-year student who had yet to even print, "Hello world." More than three years later, I would not be half of the physicist, researcher, or problem solver I am today without the mentorship he has provided throughout my entire undergraduate career, and for that I am truly grateful.

I also wish to acknowledge Professor Jens Koch whose mentorship and kindness have been hugely influential to my growth as a student, as well as Professor Michelle Driscoll for taking the time to read and evaluate this thesis.

Next, it would be a huge disservice not to mention my peers in ISP: Jay, Andrew, Ali, Lila, Kelly, Akhil, Nikhil, Chris, and many more. Their support and encouragement have motivated me to persevere through tough challenges, and their hard-work and intellect have inspired me to become a better student. I would also like to thank Ian, Tyler, Donnie, Nathan, Emma, Iman, and countless others, including my parents and siblings, who have provided me support and friendship outside of the academic realm. I am truly touched by how proud they are of my achievements.

Lastly, I want to give my sincerest thanks to Kayla Jacobson who has been a constant pillar of support, patience, and understanding. Her presence has been a constant source of comfort and motivation, and without her partnership, I am certain I would not have achieved even a fraction of what I accomplished during my time here.

ABSTRACT

In percolation theory, clusters emerge when lattice sites or bonds occupied with probability p form nearest neighbor links with other occupied sites or bonds. Of particular interest are systems where the occupation probability reaches its critical point p_c as this is where a phase transition occurs. The clusters in these critical systems exhibit several features that make them useful in the study of complex systems, such as self-similarity.

One measurement that can be made on a percolation system is that of *cluster tomography*: counting the number of clusters encountered by a line. Previous results from conformal field theory demonstrate that the cluster count at criticality has a non-leading logarithmic correction with universal prefactor b. Here we identify four geometries on the square lattice, each with its own value of b: a line segment in the bulk, a line segment along a free surface, a line segment that begins on the surface and ends in the bulk, and a line that fully traverses the system (starting on one surface and ending on another). We present several numerical and, if applicable, analytic, results for each of these geometries. We end by demonstrating a specific application of partial cluster tomography to computational neuroscience.

TABLE OF CONTENTS

Chapter 4: Application to Computational Neuroscience

References

CHAPTER 1

MATHEMATICAL AND NUMERICAL BACKGROUND

1.1 Percolation Theory

We start by considering a square lattice. In *site percolation*, each site of the lattice is independently occupied with probability p, and in *bond percolation*, the edges of the lattice itself are independently occupied with probability p [1]. Without loss of generality and unless stated otherwise, we will focus on site percolation in the following discussion, noting that any claim made about site percolation also applies to bond percolation.

We define a *cluster* as a group of nearest neighbor occupied sites (not counting diagonal sites) [2]. As depicted in Figure 1.1, when the percolation probability is low, there are typically more small to medium-sized clusters present. However, as the probability increases, a single large cluster dominates the system with only a few smaller clusters present. This difference in behavior implies the existence of a *critical point* or critical probability at which the first *system-spanning cluster* emerges [1].

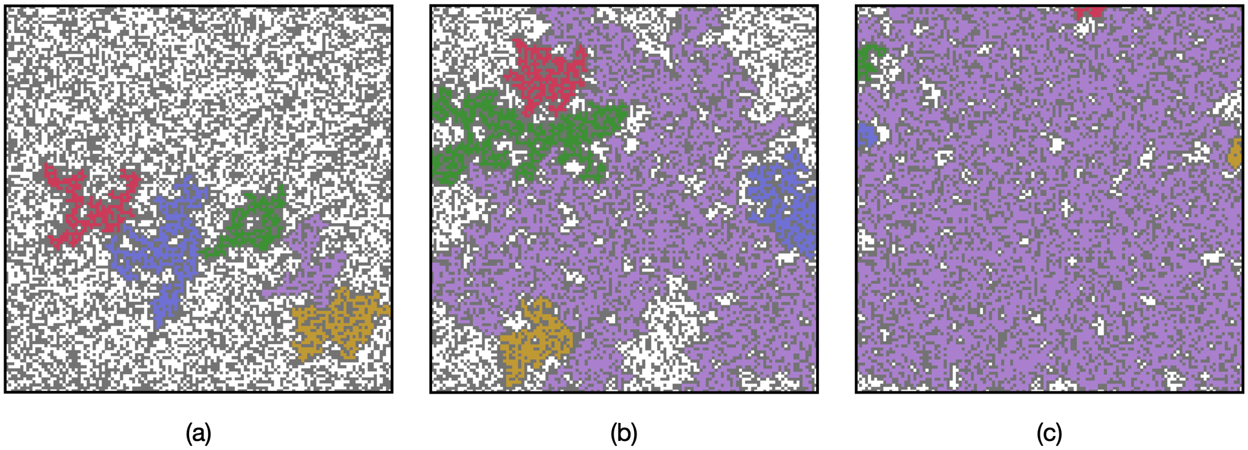


Figure 1.1: 2d site percolation system of linear size $L = 256$ at different occupation probabilities p. In all cases we've colored in only the five largest clusters. a). a subcritical system ($p = 0.55$). b). a critical system ($p = 0.59275$). c). a supercritical system ($p = 0.65$).

The critical probability depends on dimension and differs between site and bond percolation. In two dimensions, we have $p_{site}^{(2d)} \approx 0.59275$ and $p_{bond}^{(2d)} = 0.5$ [3]. It is important to note that percolation lattices are stochastic in nature, so these critical probabilities are defined in the infinite size limit [2].

Critical percolation systems exhibit several features that are not observed outside of criticality, including fractality and self-similarity of the spanning cluster, as illustrated in Figure 1.2 [2].

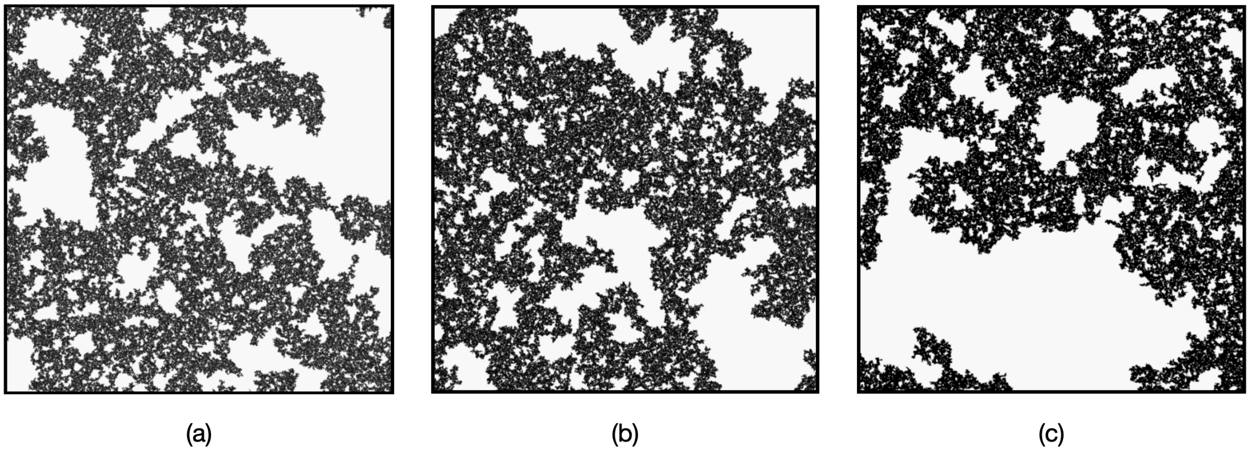


Figure 1.2: Self-similarity of the spanning cluster at different length scales in a critical bond percolation system ($p = 0.5$). Here we show snapshots of the cluster taken at successively zoomed-in portions [4].

A central theme of this thesis is that critical systems can generate universality of certain measurements. In this context, *universality* means that the behavior of the system is determined by numerical values that are independent of the microscopic details of the system, such as the type of percolation (e.g., site or bond) or lattice geometry (e.g., square or honeycomb) [1]. For example, the correlation length ξ near criticality scales as $\xi \propto |p - p_c|^{-\nu}$, where the numerical value of ν in a given dimension d is the same for all types of percolation and lattice geometries [2]. Universality is a key aspect of critical phenomena, and it is what allows us to use simple mathematical models, such as percolation, in order to study more complex systems such as the brain (see Chapter 4).

We will also distinguish between two types of boundary conditions: periodic and free. Let $\mathcal{S}$ be a square percolation lattice of length L. $\mathcal{S}$ is said to have periodic boundary conditions (PBC) in, for instance, the x direction if $\mathcal{S}(x, y) = \mathcal{S}(x \pm L, y)$. On the other hand, $\mathcal{S}$ is said to have free boundary conditions (FBC) if the boundary is freely summed over in the partition function [5]. In practice, this means that we can treat the free boundary as the *surface* of the system. We will see that different physics emerge in a system depending on whether it has PBC, FBC, or some combination of the two.

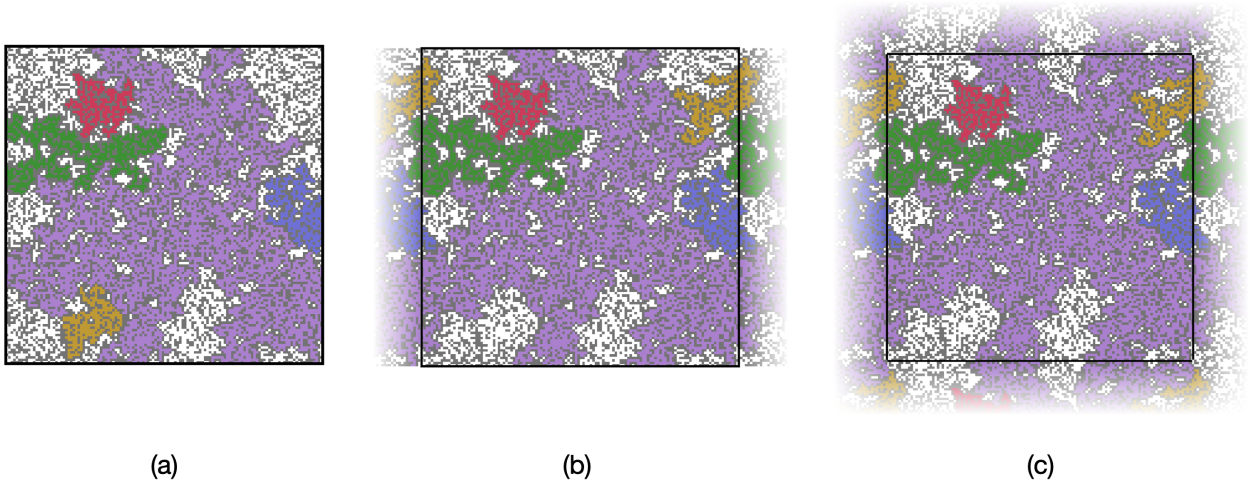


Figure 1.3: Demonstrating different boundary conditions for a site percolation system at criticality. a). FBC on all sides. b). PBC on the left and right only. c). PBC on all sides. In all cases, only the five largest clusters are colored.

1.2 The Hoschen-Kopelman Algorithm

In order to simulate percolation clusters numerically, we employ the Hoshen-Kopelman (HK) Algorithm. It was first designed specifically to study percolation and is a simple extension of the well-known union-find algorithm [6]. As an overview, the algorithm employs a raster scan to label distinct clusters on the lattice and then uses a global variable `labels` and the functions `union` and `find` in order to merge the labels of sites that belong to the same cluster. What follows is a more detailed discussion of its implementation, including how to incorporate different boundary conditions.

1.2.1 Simulating Site Percolation

To generate a 2d site percolation system, we start by randomly filling a square $L \times L$ matrix called perc_matrix. Each cell is either occupied and contains a 1 with probability $p_c \approx 0.59275$ or is unoccupied and contains a 0 with probability $1 - p_c$. We then iterate through the matrix, checking adjacent cells to the left and above the current cell. For each cell in the matrix, let left be the label of the cell's left neighbor and above be the label of the cell's above neighbor. If the cell is occupied, we perform one of four actions: (1) merge that cell with left if left is nonzero and above is zero, (2) merge that cell with above if above is nonzero and left is zero, (3) merge that cell with left and above if both labels are nonzero, or (4) create a new cluster label if both left and above are zero [7]. A Python implementation of this initial scan is given below:

```python
import numpy as np

occ = 0.59275
size = 8
labels = np.array([0])  # Global array that keeps track of all cluster labels
perc_matrix = np.random.choice([0, 1], size=(size, size), p=[1 - occ, occ])
largest_label = 0

for y in range(len(perc_matrix)):
    for x in range(len(perc_matrix[0])):
        if x == 0:
            left = 0
        elif x > 0:
            left = perc_matrix[y][x - 1]
        if y == 0:
            above = 0
        elif y > 0:
            above = perc_matrix[y - 1][x]

        if perc_matrix[y][x] == 1:
            if left != 0 and above == 0:
                perc_matrix[y][x] = find(left)
            elif left == 0 and above != 0:
                perc_matrix[y][x] = find(above)
            elif left == 0 and above == 0:
                largest_label += 1
                perc_matrix[y][x] = largest_label
                labels = np.append(labels, largest_label)
            elif left != 0 and above != 0:
                union(left, above)
                perc_matrix[y][x] = find(left)
```

Listing 1.1: Initial scan of the 2d site percolation HK algorithm on an array of linear size 8.

Notice that during this first scan, we choose not to handle the boundary conditions: we deal with those in Section 1.2.3. Also, we make frequent use of the `find` and `union` functions which we now describe.

A Python implementation of `find` is as follows [7]:

```
def find(x):
    global labels
    y = x  # Assume x is an int
    while (labels[y] != y):
        y = labels[y]
    while (labels[x] != x):
        z = labels[x]
        labels[x] = y
        x = z
    return y
```

Listing 1.2: Implementation of the find function

The `find` function searches for the cluster root; i.e., the representative member of the equivalence class for that cluster label. The cluster root is the entry in `labels` that equates its own index. We need to find the root because during our scan, we may identify several different numerical labels that actually belong to the same cluster, so we must assign each of these “dummy labels” to be equal to the cluster root. Notably, we do not yet change previous entries in `perc_matrix`; see below.

A Python implementation of `union` is as follows [7]:

```python
def union(a, b):
    global labels
    labels[find(a)] = find(b)
```

Listing 1.3: Implementation of the union function

The `union` function sets two cluster roots equal to each other. In our HK algorithm implementation, these correspond to the roots of the cluster labels `left` and `above` of our current cell when both are nonzero. This action merges these cluster labels, allowing us to assign the current cell to the value of `find(left)` and join all three cells into the same cluster.

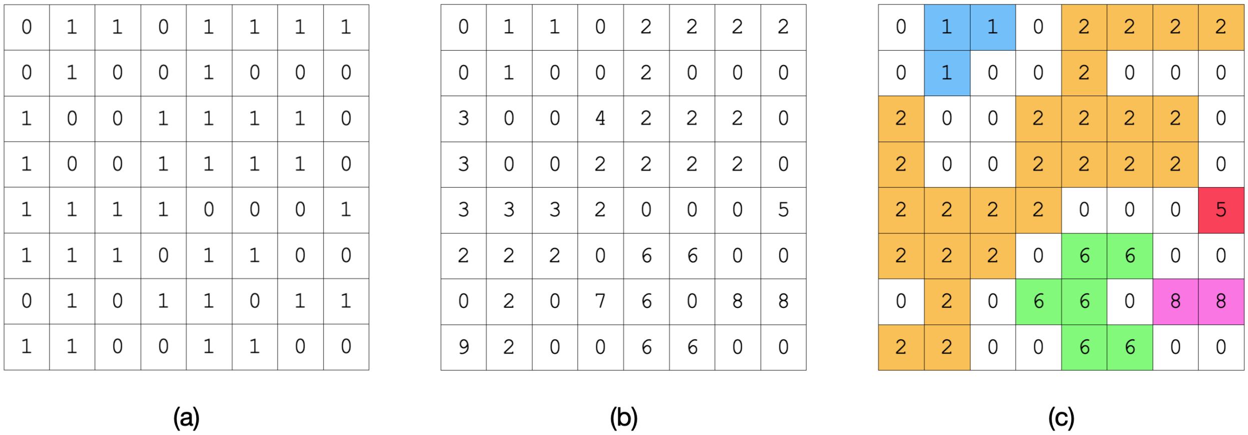
After our initial scan, we then need to iterate through the matrix one more time and assign each cluster label the label of its root. This is necessary because the initial label we assign to a cell may differ from its correct label. In Python:

```python
for y in range(len(perc_matrix)):
    for x in range(len(perc_matrix)):
        if perc_matrix[y][x] != 0:
            perc_matrix[y][x] = find(perc_matrix[y][x])
```

Listing 1.4: The second scan through the matrix of the HK Algorithm

Figure 1.4 shows `perc_matrix` after progressive steps in the HK Algorithm: the initialization of 1’s and 0’s, the first raster scan, and the final iteration through the matrix.

For simulating 3d percolation, we simply adapt the HK algorithm in a straightforward way to scan through three dimensions instead of only two.



0	1	1	0	1	1	1	1
0	1	0	0	1	0	0	0
1	0	0	1	1	1	1	0
1	0	0	1	1	1	1	0
1	1	1	1	0	0	0	1
1	1	1	0	1	1	0	0
0	1	0	1	1	0	1	1
1	1	0	0	1	1	0	0

(a)

0	1	1	0	2	2	2	2
0	1	0	0	2	0	0	0
3	0	0	4	2	2	2	0
3	0	0	2	2	2	2	0
3	3	3	2	0	0	0	5
2	2	2	0	6	6	0	0
0	2	0	7	6	0	8	8
9	2	0	0	6	6	0	0

(b)

0	1	1	0	2	2	2	2
0	1	0	0	2	0	0	0
2	0	0	2	2	2	2	0
2	0	0	2	2	2	2	0
2	2	2	2	0	0	0	5
2	2	2	0	6	6	0	0
0	2	0	6	6	0	8	8
2	2	0	0	6	6	0	0

(c)

Figure 1.4: Output of the HK Algorithm after: a). The initialization of 1's and 0's; b). The initial raster scan; c). The final iteration through the matrix. In this final output, we've colored the resultant clusters. Note that in going from (b) to (c), cluster labels 3, 4, and 9 merge with 2, and label 7 merges with 6.

1.2.2 Simulating Bond Percolation

The implementation of the HK Algorithm for bond percolation is similar to that for site percolation but with a few key differences. Bond percolation describes the random connectivity of a lattice with each site occupied. Thus, our final output of the algorithm should have no cells with the 0 label. Additionally, for each site in the matrix, there can independently be a link either to the left or above the site, so to model this behavior, we need two arrays, as opposed to only one, that are randomly filled with 1's with probability p. Apart from these differences, the implementation for bond percolation carries out much the same as for site percolation.

1.2.3 Different Boundary Conditions

The HK Algorithm as described in Sections 1.2.1 and 1.2.2 generates a percolation system with FBC around all sides. If we wanted to implement PBC on, say, the left and right sides of the system, then after our initial raster scan but before the second iteration through the matrix, we simply use `union` to merge the cluster roots of the left- and right-most columns of the system. We provide here a sample implementation in Python and note that PBC along the top and bottom can be achieved by merging the top- and bottom-most rows of the system.

```
for y in range(len(perc_matrix)):
    left = perc_matrix[y][-1]
    if perc_matrix[y][0] != 0 and left != 0:
        union(left, perc_matrix[y][0])
```

Listing 1.5: Implementing PBC on the left and right sides of the system

1.3 Aim of Thesis

With the foundational mathematical and numerical concepts established, we can now delve into a specific area of percolation theory: cluster tomography (introduced in Chapter 2). We will examine both known and novel findings in 2d and 3d cluster tomography (Chapters 2-3) and present a practical application of cluster tomography in computational neuroscience (Chapter 4).

CHAPTER 2

KNOWN RESULTS IN PARTIAL CLUSTER TOMOGRAPHY

2.1 Introducing Cluster Tomography

Clusters emerge naturally in a variety of complex systems (Figure 2.1). Within such systems, an intriguing question that arises is that of *cluster tomography*: how many clusters can be identified along a cross section? Such a measurement can clearly provide insight into the density of clusters in a certain region of the system, a question that may be interesting in its own right, but we will see that it also has the surprising ability to identify the universality class of the system. Therefore, given the ease with which such a measurement can be performed, we claim that cluster tomography can be a valuable tool for studying complex and critical systems.

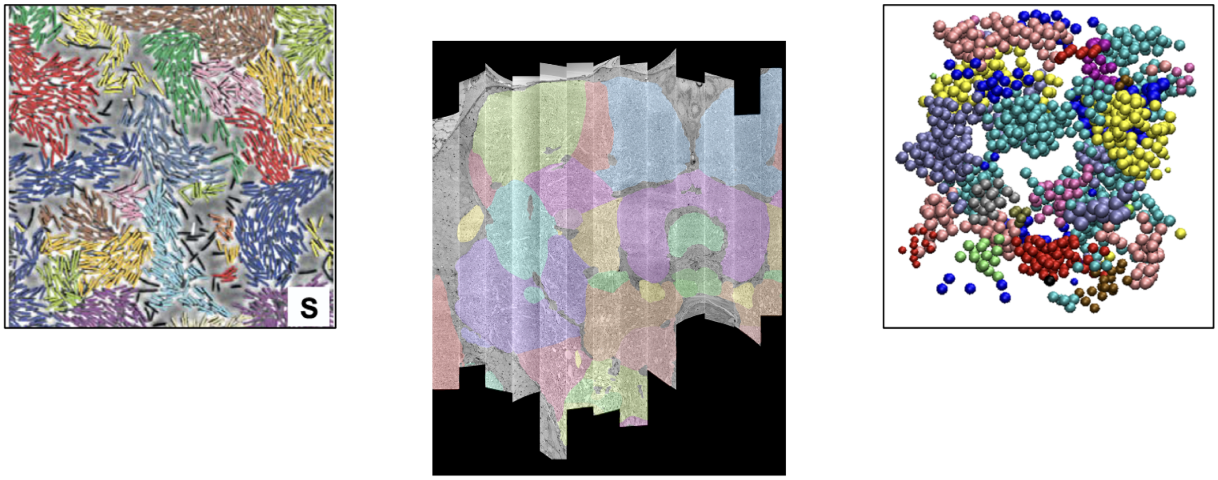


Figure 2.1: Examples of clustering in nature. From left to right: bacterial cell swarming [8], different brain regions [9], active Brownian clustering [10].

Here we distinguish between two types of cluster tomography: partial and full. *Partial tomography* describes the action of counting clusters along a line segment traversing only a part of the system, whereas *full tomography* refers to a line traversing the entire system. In this chapter, we focus on the known results in partial tomography: line segments along a free surface in 2d and line segments in the bulk in 2d and 3d. The next chapter will focus on the unknown results: full tomography, so-called "mixed" tomography, and the surface line segment in 3d. See Figure 2.2.

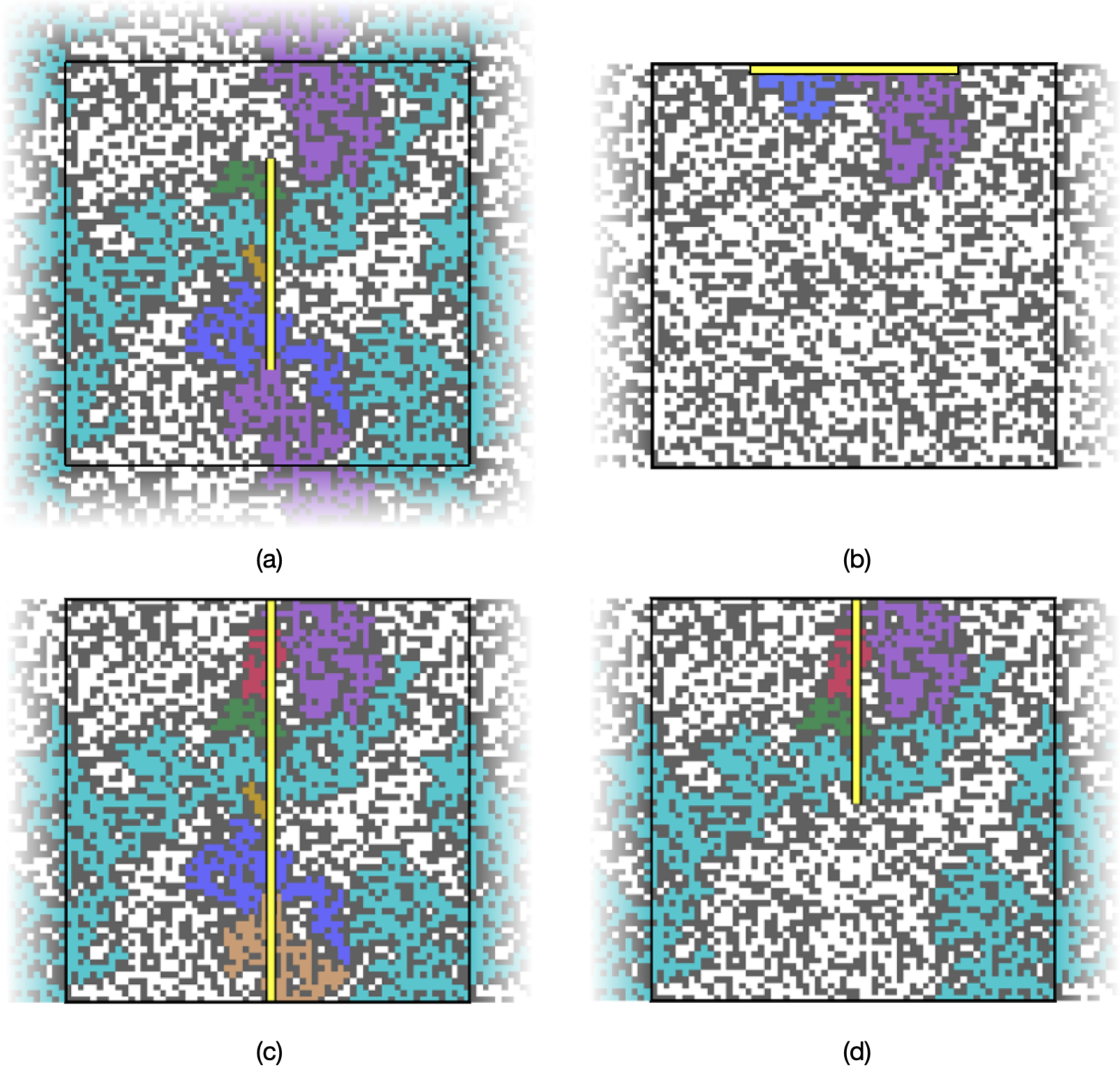


Figure 2.2: Cluster Tomography. Only the clusters encountered by the cross section are colored, and BCs are identified as in Figure 1.3. (a) Partial tomography in the bulk. (b) Partial tomography along a free surface. (c) Full tomography from free surface to free surface. (d) Partial mixed tomography. We discuss cases (a) and (b) in this chapter, saving cases (c) and (d) for Chapter 3.

2.2 Clusters Crossed by Line Segments

2.2.1 Line Segment in the Bulk

Consider a line segment of length ℓ in the bulk of a 2d percolation system (site or bond) of linear length L. This case is illustrated in Figure 2.2a. Here "bulk" means that the line segment is far from any boundaries present in the system. The cluster tomography question asks how many clusters N are encountered by the line segment. Perhaps one would naively expect the cluster count $N(\ell)$ to scale linearly with the length of the line segment: $N(\ell) = a\ell$, where a is a non-universal constant in general and therefore depends on microscopic details of the system. The reasoning behind this assumption is sound: the longer the line, the more clusters it proportionately encounters. This expectation is correct for systems outside of criticality. However at criticality, we see a nonleading logarithmic correction term with a **universal** prefactor b [11]:

$$N(\ell) = a\ell + b \log \ell \tag{2.1}$$

Therefore all line segments in the bulk of critical 2d (and 3d) percolation systems exhibit this behavior, where b takes on the same numerical value for all systems in the same dimensionality [12].

What is so special about clusters at criticality that could lead to such nonlinear behavior? To answer this question, compare the cluster counts between two half-line segments of length $\ell = L/2$ with a single whole line of length L. At first it may seem that the number of clusters crossed in both cases is the same; however, closer inspection reveals that certain clusters, such as the purple cluster in Figure 2.3, are "double counted" for the half-line segments. Because two half-line segments and one whole line have the same length, it must be the case that any difference comes from the endpoints that are present in the line segments but not in the line [13]. These endpoints are referred to as "corners", and indeed, the logarithmic term of Equation (2.1) is often called the *corner contribution*. Thus, if we assume that the cluster count for each half-line segment (with corners) takes on the form of Equation (2.1) and that the cluster count for the whole line (without corners) takes on the form $N_{\text{whole}}(L) = N_{\text{whole}}(2\ell) = a(2\ell)$, then we can write:

$$\begin{aligned} \text{corner contribution} &= b \log \ell \qquad (2.2) \\ &= \frac{1}{2}\big[2(a\ell + b\log\ell) - 2a\ell\big] \\ &= \frac{1}{2}\big[2N_{\text{half}}(\ell) - N_{\text{whole}}(2\ell)\big] \qquad (2.3) \end{aligned}$$

Thus, Equation (2.3) gives a prescription to determine the corner contribution at a certain length scale ℓ: Take half of the difference between the cluster counts of two half-line segments and one whole line. Equivalently, the corner contribution is one-half of the number of clusters crossed by both half-line segments; therefore, only clusters encountered by both half-line segments will add to to the corner contribution.

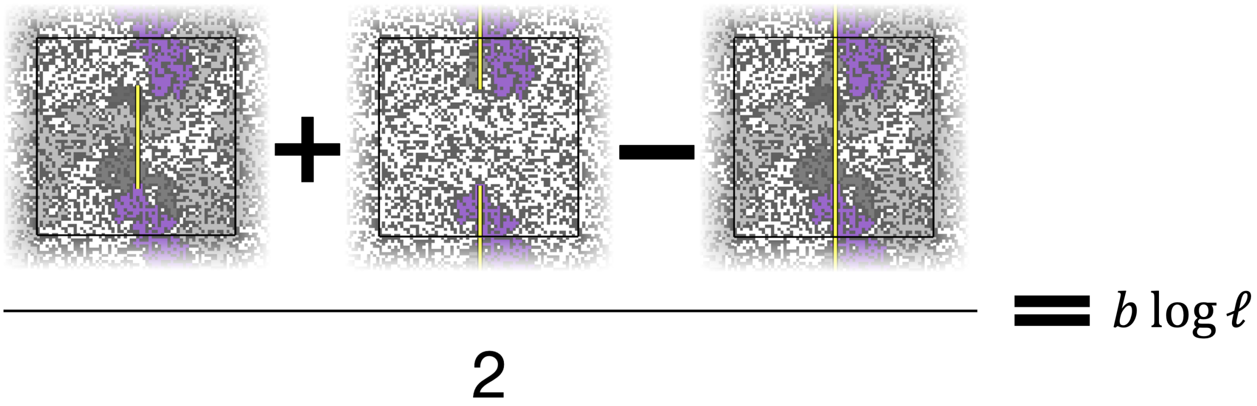


Figure 2.3: Pictorial schematic of how to determine the corner contribution for a line segment in the bulk. Notice that we can obtain the corner contribution of $1/2$ in two ways: either taking the difference in the cluster counts (where $(6 + 2 - 7)/2 = 1/2$) or by noting that only the purple cluster is encountered by both half-line segments.

It is worth noting that two types of clusters can cross both half-line segments and therefore give a corner contribution. The first is if the cluster directly encounters the corner: it is clear that this cluster must touch both half-line segments. The second is if the cluster is concave, as demonstrated in Figure 2.4b. These "rainbow-shaped" clusters are believed to be the main sources of the overall corner contribution. We therefore see why the corner contribution is a strictly critical phenomenon: such "rainbow-shaped" clusters only appear at criticality [14]. In subcritical systems, the clusters are too small to cross both half-line segments and contribute, and in supercritical systems, the large, dominating cluster is the *only* one that contributes.

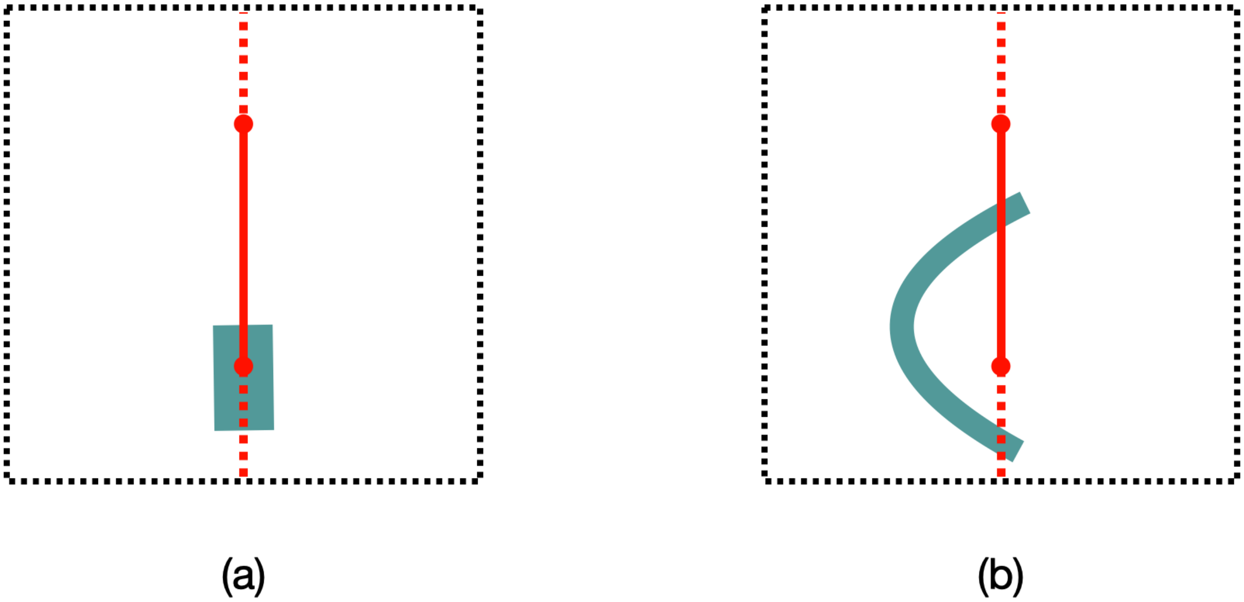


Figure 2.4: Two different clusters can give a corner contribution. a). A cluster directly touching a corner is encountered by both line segments. b). A "rainbow-shaped" cluster also contributes since it encounters both half-line segments even without touching the corner. Such clusters are only expected to occur at criticality.

Analytically Deriving $b_b^{(2d)}$

How do we know that the bulk line segment's b is universal? We can use a technique discovered by Fortuin and Kastelyn that maps percolation into more conventional critical spin systems and then use conformal field theory to analyze the cluster count [5]. This calculation is quickly detailed below.

We will consider 2d bond percolation with n number of sites and m number of nearest neighbor bonds (since our end result for b is universal, it wouldn't matter if we instead considered site percolation). Percolation is the $Q \to 1$ limit of the Potts model, so by using the aforementioned Fortuin-Kastelyn clusters, we can write the general Q-state Potts model's partition function up to an irrelevant prefactor as [13]

$$\begin{aligned} Z(Q) &\sim \sum_F Q^{N_{tot}(F)} p^{M(F)} (1-p)^{m-M(F)} \\ &= \langle Q^{N_{tot}} \rangle \end{aligned} \tag{2.4}$$

where F is the set of total clusters, and each cluster contains $N_{tot}(F) \leq n$ connected components and $M(F) \leq m$ open bonds. The average number of clusters N_{tot} is given by

$$\langle N_{tot} \rangle = \left. \frac{\partial \log Z(Q)}{\partial Q} \right|_{Q=1} \tag{2.5}$$

where we set $Q = 1$ in accordance with the percolation limit.

Now consider a line segment Γ of length ℓ. This can be modeled by fixing all the states along the line segment (say in state 1) while not changing the coupling [13]. By a similar argument given in (2.4), we can write

$$Z_\Gamma(Q) \sim \langle Q^{N_{tot} - N_\Gamma} \rangle \tag{2.6}$$

and therefore:

$$\langle N_{tot} - N_\Gamma \rangle = \left. \frac{\partial \log Z_\Gamma(Q)}{\partial Q} \right|_{Q=1} \tag{2.7}$$

At $p = p_c$ we write [13]:

$$\log Z(Q) \sim A f_b(Q) \tag{2.8}$$

$$\log Z_\Gamma(Q) \sim A f_b(Q) + f_s(Q)\ell + C_\Gamma(Q) \log \ell \tag{2.9}$$

where $A \propto n$ is the total area of the system, and f_s and f_b are the surface and bulk free-energy densities, respectively, which are non-universal. Thus, combining equations (2.5), (2.7), (2.8), and (2.9), we see that

$$\langle N_\Gamma \rangle = -f_s'(1)\ell - C_\Gamma'(1) \log \ell \tag{2.10}$$

where we identify the last term as the corner contribution. $C_\Gamma(Q)$ is given by the *Cardy-Peschel Formula* [15]:

$$C_\Gamma(Q) = \frac{c(Q)}{24}\sum_k \left[\frac{\pi}{\gamma_k} - \frac{\gamma_k}{\pi} + \frac{\pi}{2\pi - \gamma_k} - \frac{2\pi - \gamma_k}{\pi}\right] \tag{2.11}$$

where $c(Q)$ is the central charge of the Q-state Potts model, and γ_k is the angle of the corner. Note that this formula accounts for the fact that a general subsystem could have interior angle γ_k and exterior angle $2\pi - \gamma_k$. We do not consider subsystems beyond the line or line segment in this thesis and therefore only consider the angle $\gamma_k = 2\pi$[1]. Thus our Cardy-Peschel formula looks like:

$$C_\Gamma(Q) = \frac{c(Q)}{24}\cdot 2\left(\frac{\pi}{\gamma} - \frac{\gamma}{\pi}\right)\Bigg|_{\gamma=2\pi} \tag{2.12}$$

$$= -\frac{c(Q)}{8} \tag{2.13}$$

where the 2 comes from the fact we have two corners in our line segment. The central charge is related to the Q-state of the Potts model via [16]:

$$c(Q) = 1 - \frac{6}{k(k+1)} \tag{2.14}$$

$$\sqrt{Q} = 2\cos\left(\frac{\pi}{k+1}\right) \tag{2.15}$$

where k is a parameter equal to 2 for percolation. We therefore find that $c'(1) = 5\sqrt{3}/(4\pi)$, leading to the following expression:

$$b_b^{(2d)} = -C'_\Gamma(1) = \frac{c'(1)}{8} = \frac{5\sqrt{3}}{32\pi} \approx 0.08615 \tag{2.16}$$

where we write $b_b^{(2d)}$ in order to distinguish it from other b's that show up in different dimensions and/or tomography types. Because $b_b^{(2d)}$ depends only on Q and the corner angle, it is universal.

[1] The $\gamma = 0$ angle can be ignored since there is no volume inside the line segment for that angle. However, in the scaling limit in which the angle γ is lowered by keeping the internal volume fixed, one sees the singularity [13].

Although no analogous analytical derivation yet exists for $b_b^{(3d)}$, it has been computationally determined to be 0.130(3) [12].

2.2.2 Line Segment Along a Free Surface in 2d

We now turn to the second partial tomography case: the line segment along a free surface in 2d (Figure 2.2b). The exact same story of corner contributions and concave "rainbow-shaped" clusters at criticality applies here as in the bulk case, so here we offer an analytical calculation of $b_s^{(2d)}$. A mathematical proof of $b_s^{(2d)}$ on the triangular lattice is provided in [17].

We begin by once again writing the partition function of the Potts model, this time in terms of clusters' boundary loops [18]:

$$Z = \sqrt{Q}^n \sum_{loops} \sqrt{Q}^{N_{loops}} \tag{2.17}$$

where n is the total number of spins. As before, consider a line segment of length ℓ, this time along the system's free surface. Let ξ_1 and ξ_2 be the endpoints of the line segment. Then we can express the two-point correlation function between the boundary condition changing operators (BCCOs) $\phi(\xi_1)$ and $\phi(\xi_2)$ as [19]

$$\langle \phi(\xi_1)\phi(\xi_2) \rangle = \frac{\sqrt{Q}^n}{Z} \sum_{loops} \sqrt{Q}^{N_{loops}} \left(\frac{y}{\sqrt{Q}}\right)^{N_\Gamma} \tag{2.18}$$

where changing the boundary is necessary in order to give a different weight y to every cluster that touches the line segment. If the total number of clusters encountered by the line segment is N_Γ, then

$$\left. \frac{\partial \langle \phi(\xi_1)\phi(\xi_2) \rangle}{\partial y} \right|_{y=\sqrt{Q}} = \frac{1}{\sqrt{Q}} \langle N_\Gamma \rangle \tag{2.19}$$

For large ℓ, the correlation function is hypothesized to be expressed as

$$\langle \phi(\xi_1)\phi(\xi_2) \rangle \sim e^{-f(y)\ell} \ell^{-2h(y)} \tag{2.20}$$

where $f(y)$ is the boundary free energy, and $h(y)$ is the scaling dimension of the BCCOs [19].

Carrying out the derivative in (2.19), and noting that $h(\sqrt{Q}) = f(\sqrt{Q}) = 0$ since there is no change on the boundary for $y = \sqrt{Q}$, we have:

$$\langle N_\Gamma \rangle = -\sqrt{Q} f'(\sqrt{Q})\ell - 2\sqrt{Q} h'(\sqrt{Q}) \log \ell \tag{2.21}$$

where the last term is identified as the corner contribution. In order to calculate $h'(\sqrt{Q}) = \frac{\partial h}{\partial y}\Big|_{y=\sqrt{Q}}$, we note that $h(y)$ is equivalent to the $h_{r,r}$ weight in the Kac table, with [20]

$$h_{r,r} = \frac{r^2 - 1}{4k(k+1)} \tag{2.22}$$

where r is related to y via

$$y(r) = \frac{\sin\left(\frac{(r+1)\pi}{k+1}\right)}{\sin\left(\frac{r\pi}{k+1}\right)} \tag{2.23}$$

Applying the chain rule in order to calculate the derivative of (2.22) with respect to y gives:

$$h'(y = \sqrt{Q}) = -\frac{\sqrt{4-Q}}{4\pi k} \tag{2.24}$$

after we identify that $y = \sqrt{Q}$ is equivalent to $r = 1$. Thus we see that

$$\begin{aligned} b_s^{(2d)} &= -2\sqrt{Q} h'(\sqrt{Q})\Big|_{Q=1} \\ &= \frac{\sqrt{3}}{4\pi} \approx 0.13783 \end{aligned} \tag{2.25}$$

where we've once again identified that $Q = 1$ and $k = 2$ in the percolation limit.

At the time of writing, there is no analogous derivation or even numerical approximation of $b_s^{(3d)}$. We therefore turn to this tomography case in Chapter 3.

CHAPTER 3

INVESTIGATING UNKNOWN PROBLEMS IN FULL AND PARTIAL TOMOGRAPHY

3.1 Motivating the Full and Mixed Tomographies

In the last chapter, we introduced cluster tomography as a means of identifying the universality class of the system; i.e., if a tomography measurement yields a logarithmic correction, then the system must be critical. However, to this point in time, the only known results in cluster tomography are for the bulk and surface line segments, and it is unrealistic that a researcher would not interact with both the surface *and* the bulk of the system when making a measurement. At the very least, it is unreasonable to demand that a researcher restrict their measurement to one or the other when it is likely easiest (and perhaps most useful) to make a measurement across the system starting at a boundary. Thus we introduce the need for *full tomography* and *mixed tomography* as illustrated in Figures 2.2c and 2.2d, respectively. Full tomography describes the cluster count of a line starting on one free boundary and traversing the entire system to the other free boundary. Mixed tomography describes the cluster count of a line segment starting on one free boundary and ending in the bulk of the system. Because previous results in cluster tomography relied on the stark distinction between the bulk of the system and the system surface, it is incorrect to assume that the known values of b_s and b_b would apply to b_f (b full) or b_m (b mixed) since these involve the *interplay* of the bulk and surface. Therefore, the majority of this chapter is devoted to determining the values of b_f and b_m numerically. At the end of the chapter, we also discuss the surface line segment in 3d because its value has not been determined yet, even numerically.

3.2 Results in Full and Mixed Tomography

3.2.1 Canceling the Linearity

In Section 2.2.1, we detailed a way of finding the corner contribution for bulk and surface line segments that compared two half-line segments to one whole line. This approach offers the advantage of canceling the linear $a\ell$ term in Equation (2.1) by comparing systems of equal length, allowing us to isolate the corner contribution $b \log \ell$. While this exact geometric technique does not apply to full or mixed tomography, we are able to devise alternative methods to cancel the linear law and therefore isolate the universal logarithmic term if one exists. We describe these methods now.

Full Tomography

In the case of full tomography, the geometric trick of Section 2.2.1 is insufficient because the subsystem of interest is itself a whole line whose length L is equal to the system's length. Instead, we compare two different $systems$: one with FBC where the line interacts with the surface, and one with the FBC replaced by PBC. Taking the difference in this cluster count gives the corner contribution, where the corners are the angles that the line makes with the free boundary:

$$\begin{aligned}\text{corner contribution} &= b \log L \\ &= (aL + b \log L) - aL \\ &= N_{\text{FBC}}(L) - N_{\text{PBC}}(L) \qquad (3.1)\end{aligned}$$

Figure 3.1 demonstrates this boundary condition changing approach, and we note that different angles γ are possible. In order to use angles that are compatible with a square lattice, we introduce a parameter $n \in \{1, 2, \dots, 20\}$ and define $\gamma = \arctan(1/n)$[1]. We also include the angle $\gamma = \pi/2$ which corresponds to $n \to 0$. Assuming some dependence on the Cardy-Peschel Formula (2.11),

[1] For lattice sizes $L \in \mathbb{Z}$ that are less than 20, we only use angles $\arctan(1/n)$ for $n \in \{1, 2, \dots, L-1\}$.

each corner angle would correspond to a different value of b_f, so we will henceforth write b_f in terms of an explicit dependence on the corner angle: $b_f(\gamma)$.

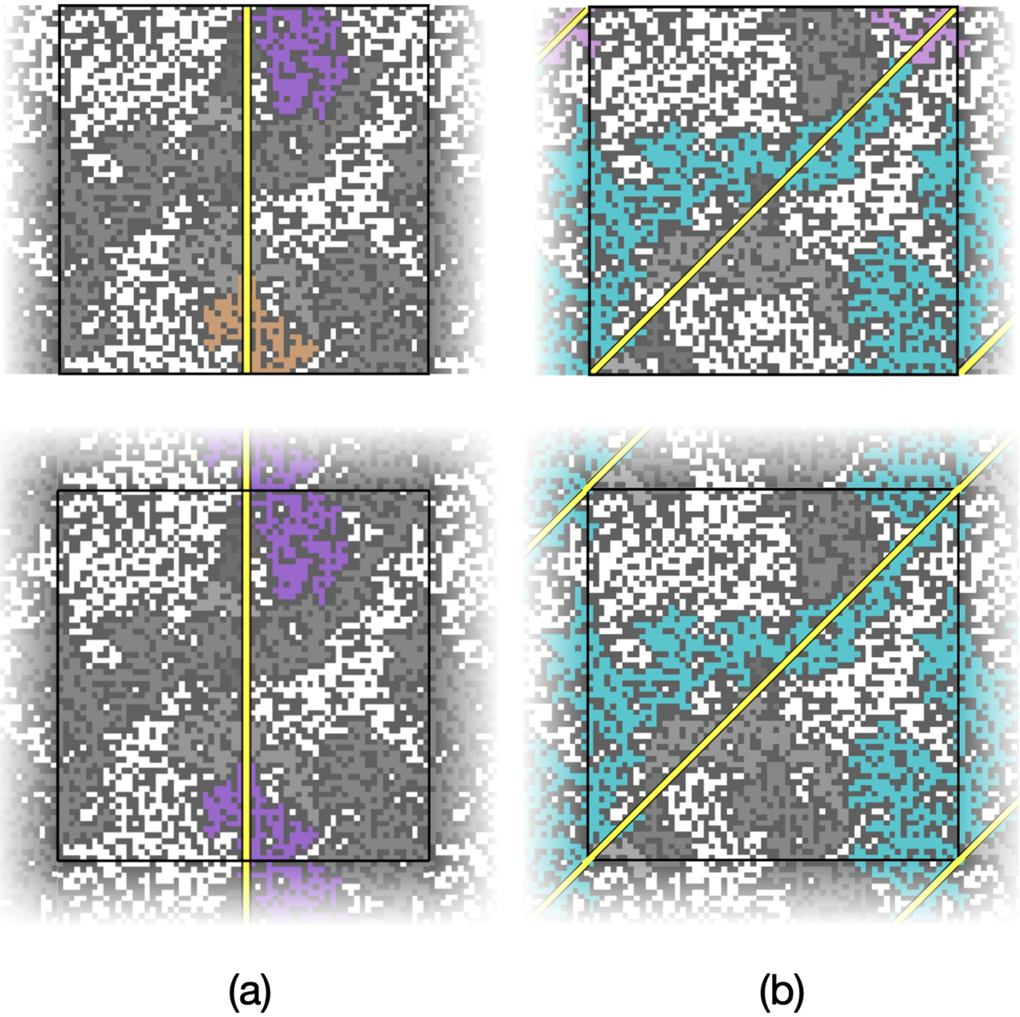


Figure 3.1: The boundary condition changing approach to isolating the corner contribution in full tomography for (a) $\gamma = \pi/2$ and (b) $\gamma = \pi/4$. The top row has free boundaries on the top and bottom and is the full tomography case we are interested in, and the bottom row replaces the free boundaries with periodic boundaries. Only the clusters that result in a corner contribution are colored in.

Mixed Tomography

In the case of mixed tomography, neither the geometric nor the boundary condition changing approach are sufficient enough to isolate the corner contribution: The former doesn't account for the corner on the free surface, and the latter doesn't account for the corner in the bulk. Thus, an approach that combines these two methods is required. Assuming a line segment of length $\ell = L/2$:

$$\begin{aligned} \text{corner contribution} &= b \log \ell \\ &= \frac{1}{2}\Big[2(a\ell + b\log\ell) - 2a\ell\Big] \\ &= \frac{1}{2}\Big[2N_{\text{mixed, FBC}}(\ell) - N_{\text{whole, PBC}}(2\ell)\Big] \end{aligned} \tag{3.2}$$

where $N_{\text{whole, PBC}}(2\ell)$ is exactly the same as $N_{\text{PBC}}(L)$ in (3.1).

Furthermore, based on the geometry of this tomography case, we can hypothesize that the bulk and full tomography b's will contribute equally:

$$b_m(\gamma) = \frac{1}{2}\Big[b_b + b_f(\gamma)\Big] \tag{3.3}$$

3.2.2 Numerical Results

Let $\kappa_\gamma(L) = b_f(\gamma, L)\log L$ be the corner contribution for a full tomography line[2] of length L, where $\lim_{L\to\infty} b_f(\gamma, L) = b_f(\gamma)$. Then we see that the two-point slope on a logarithmic scale between corner contributions in systems of length L and $2L$ is

$$m_{2pt} = \frac{\kappa_\gamma(2L) - \kappa_\gamma(L)}{\log 2} = b_f(\gamma, L) \tag{3.4}$$

For finite L, the value of m_{2pt} is merely an approximation for $b_f(\gamma)$, but extrapolating this result to $L \to \infty$ yields its exact value. In order to extrapolate to infinite size, we plot $b_f(\gamma, L)$ values on the y-axis versus $1/L$ on the x-axis and determine the y-intercept of the linear fit. Using this method, we were able to reproduce the results for $b_b^{(2d)}$ and $b_s^{(2d)}$ that were given in (2.16) and (2.25), respectively, and this is also how we calculate $b_f^{(2d)}(\gamma)$, $b_f^{(3d)}(\gamma)$, $b_m^{(2d)}(\gamma)$, $b_m^{(3d)}(\gamma)$, and $b_s^{(3d)}$.

[2] without loss of generality we could also consider b_m or any other b instead of b_f. In tomography cases without an angle-dependence γ, we simply omit it as an argument.

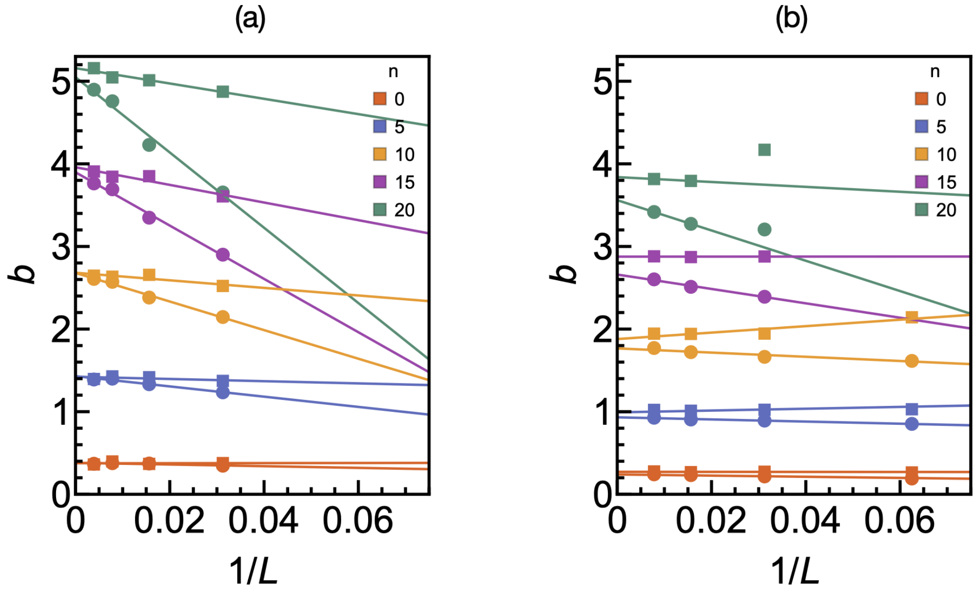


Figure 3.2: Extrapolating the value of $b_f(\gamma)$ for $\gamma = \arctan(1/n)$ in (a) 2d and (b) 3d. Squares are for bond percolation, and circles are for site percolation. Figures and 3d results used with permission of Helen Ansell.

In 2d, we use sizes up to $L = 512$, and in 3d we use sizes up to $L = 256$. In all cases, the minimum number of samples was $10,000$.

Figure 3.2 gives the b vs. $1/L$ fits for the full tomography case in 2d and 3d. Note that each color represents a different value of n and therefore a different angle. For each system, the corner contribution was found by averaging the cluster counts across lines placed in all possible starting positions (i.e., a system of length L averages over L positions). Each data point in Figure 3.2 is then the average of the corner contributions across at least $10,000$ randomly-generated systems. We see that the b values in site and bond percolation are quite different for finite sizes, but as we extrapolate to infinite size, the values converge to a single limit on the y-intercept. This limit is what we claim to be the value of $b_f(\gamma)$, where, for instance, $b_f(\pi/2) = 0.378(6)$[3]. We were the first to hypothesize that the extrapolations of site and bond percolation would exhibit strong agreement for full tomography. Our work demonstrates this agreement and *therefore implies that $b_f(\gamma)$ is universal*.

There is a slight caveat in the way we define the clusters that are crossed by a line or line

[3] Our error-reporting notation here is such that $0.378(6) = 0.378 \pm 0.006$.

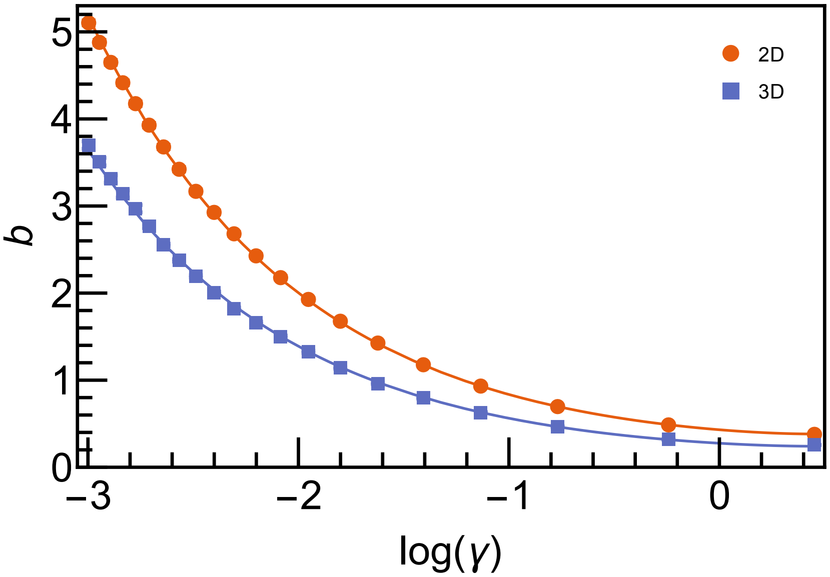


Figure 3.3: Plot of extrapolated $b_f(\gamma)$ values for $\gamma = \pi/2$ and $\gamma = \arctan(1/n)$, $n = 1, \ldots, 20$. Fit is given by (3.5). Figure and 3d results used with permission of Helen Ansell.

segment. We can either count the "visited" clusters, those that touch the line at any point along its path, or count the "crossed" clusters, those that have sites both inside and outside of the line. The set of crossed clusters is a subset of the visited clusters. For example, a cluster with all of its sites contained inside the line is said to be visited but not crossed. We determined corner contributions according to both definitions of cluster counting and obtained the same b values. Therefore, we need not be precise in our cluster counting definition, but we note that all of the reported data is for visited clusters.

Figure 3.3 illustrates the angle dependence of $b_f(\gamma)$ in 2d and 3d. Motivated by the Cardy-Peschel formula, the fits are given by

$$b_f(\gamma) = c_1 \left(\frac{1}{\gamma} + \frac{1}{\pi - \gamma} \right) + c_2 \tag{3.5}$$

where $c_1(2d) = 0.2510(6)$, $c_2(2d) = 0.062(4)$, $c_1(3d) = 0.1758(5)$, and $c_2(3d) = 0.026(2)$.

By following a similar procedure for $b_m(\gamma)$, we find that it is also universal and has the numerical values $b_m^{(2d)}(\pi/2) = 0.237(3)$ and $b_m^{(3d)}(\pi/2) = 0.195(1)$.

It is important to note that while we have numerical evidence for the existence and universality of the prefactors $b_f(\gamma)$ and $b_m(\gamma)$, there are no analytical derivations of these values at this time. The reason that we cannot simply mimic the argument of Section 2.2.1 is that it requires conformal invariance in order to use the Cardy-Peschel formula, but it is still unclear how conformal invariance can be applied here since the full line and mixed line segment exist in different states from the surface with which they interact. The argument of Section 2.2.2 is also not directly helpful since it requires that the line or line segment lies entirely along a free surface.

3.3 The Surface Line Segment in 3d

To this point, every tomography case we've investigated has resulted in a universal b prefactor to the logarithm in (2.1). However, the surface line segment in 3d seems to be the first case we've examined that has a vanishing prefactor: $b_s^{(3d)} = 0$. Although not proven analytically, our initial extrapolations of $b_s^{(3d)}$ analogous to those in Figure 3.2 were approximately 0.014 and 0.018 for site and bond percolation, respectively. These values are an order of magnitude smaller than almost every other b value in 2d and 3d. Furthermore, these initial linear fits had a positive slope for both site and bond percolation, contradicting the negative slopes we've come to expect from Figure 3.2, especially for site percolation. Both of these discrepancies prompted us to plot an additional data point at $L = 512$, revealing that the data seems not to follow a linear relationship but rather slopes downward nonlinearly toward 0. Therefore, we conclude here that there is no logarithmic correction to the surface cluster count in 3d, but we comment that more analysis is necessary in order to identify what kind of nonleading relationship exists and whether that relationship is universal.

But what is so special about the surface in 3d? Here we can only postulate, but one possible explanation has to do with the *concavity* of contributing clusters. We've established in Section 2.2.1 that it is the "rainbow-shaped" clusters that give a corner contribution to the 2d bulk and surface line segments. Similar arguments apply to b_f and b_m as well: in order to contribute to b_f, a contiguous cluster in the PBC system must encounter the full line on either side of the free

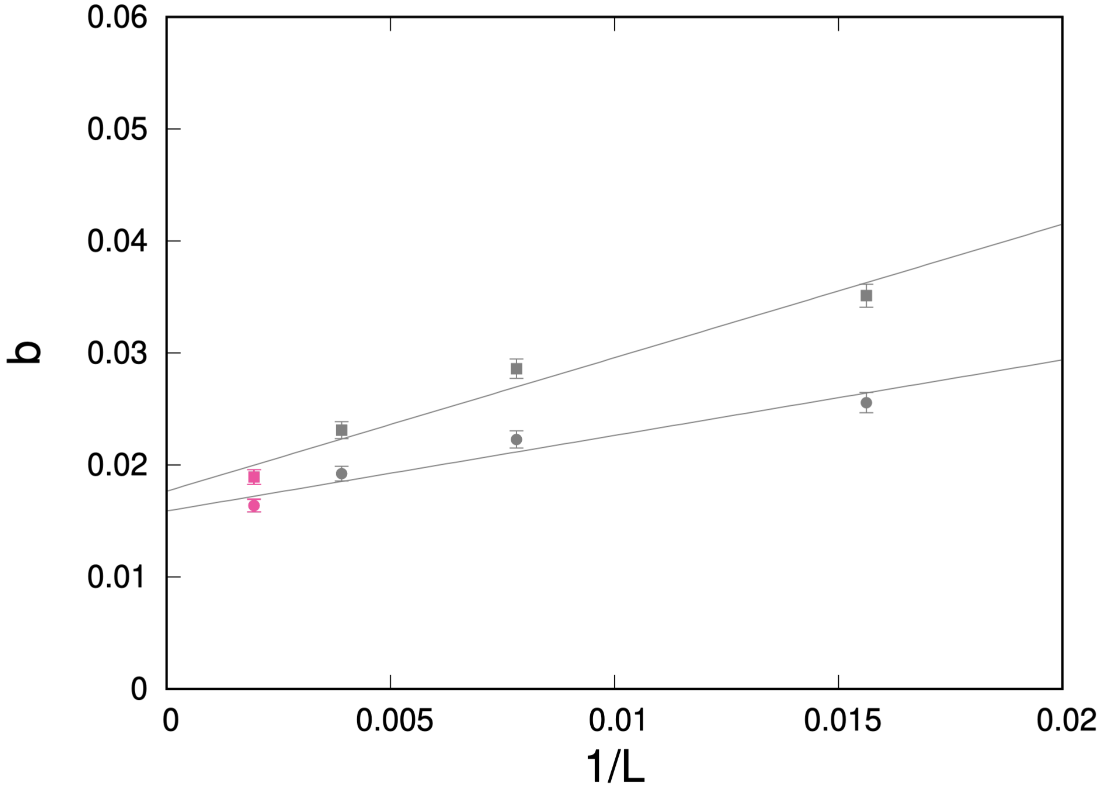


Figure 3.4: 3d surface tomography results. Squares are for bond percolation, and circles are for site percolation. The last data point colored in pink corresponds to $L = 512$ and demonstrates that the data curves down to 0. This fact along with the low value of the extrapolated y-intercept imply that there is no logarithmic correction to the cluster count for a surface line segment in 3d; hence, we claim that $b_s^{(3d)} = 0$. Final data point used with permission of Helen Ansell.

boundary, and in order to contribute to b_m, a cluster must encounter both half mixed tomography line segments. Both of these criteria are fulfilled by “rainbow” clusters.

What about the surface line segment? The concave “rainbow” clusters are also known to contribute to $b_s^{(2d)}$. Notice, however, that in 2d, if a cluster is on one side of the free boundary, then it must cross the line segment in order to get to the other side. This is not necessarily true in 3d since we can have a cluster move from one side of the boundary to the other without ever touching the line segment (e.g., by going “over” the boundary) . Upon applying FBC, these looping clusters fracture into multiple pieces, and finding a contiguous piece that encounters both half line-segments may be difficult. Thus, it is not that “rainbow” clusters *can’t* contribute in 3d, but rather it is possible that these so-called “looping” clusters are more common in 3d and result in a much lower proportion of concave “rainbow” clusters compared to the other tomography cases.

Although the reason remains unclear, it seems improbable that a line segment along a free surface in 3d has a corner contribution. This is in contrast to all other cluster tomography measurements that are examined in this thesis. Table 3.1 therefore shows our best understanding of each of the four tomography cases in 2d and 3d:

	b_b	b_s	$b_f(\gamma = \frac{\pi}{2})$	$b_m(\gamma = \frac{\pi}{2})$
2d	$5\sqrt{3}/(32\pi)$ ≈ 0.08615 [13]	$\sqrt{3}/(4\pi)$ ≈ 0.13783 [13]	0.378(6)	0.237(3)
3d	0.130(3) [12]	0	0.256(2)	0.195(1)

Table 3.1: Summary of universal b values for different cluster tomography configurations.

CHAPTER 4

APPLICATION TO COMPUTATIONAL NEUROSCIENCE

4.1 Evidence of Structural Criticality in the Brain from Tomography Measurements

In 2020, the Janelia Research Campus released a groundbreaking connectome of the hemibrain of an adult fruit fly (*Drosophila melanogaster*) [9]. A connectome refers to a comprehensive spatial map of neurons and neuroglia, while "hemibrain" means that only half of the brain was reconstructed. This achievement is significant because it represents the first spatial mapping of physical connections among neural cells in an animal as complex as the fruit fly, on a scale as large as the hemibrain [9].

The brain segmentation data was acquired using transmission electron microscope images of nanoscale slices of the fly brain. These slices were then stacked to form 3D volumes, where flood-filling networks traced individual neurons. Finally, the researchers developed an online platform called Neuroglancer that allows users to search through the petabyte-scale dataset online [9].

From Figure 4.1, we see that the neural cells form clusters that are qualitatively similar to what we may see in percolation. Thus, if we perform partial bulk tomography measurements on the slices of hemibrain data similar to those described in Section 2.2.1 and find a value for b_b, then we should be able to determine whether the brain is in a state of *structural criticality*, a question that has long intrigued researchers for its possible clinical applications [21].

Using the Python package CloudVolume, a library specifically designed to analyze Neuroglancer data, we extract segmentation data for the fruit fly and determine the corner contribution $\kappa(\ell) = b \log \ell$ for a bulk line segment of length $\ell = L/2$ according to (2.3). If the plot of $\kappa(\ell)$ vs. $\log \ell$ has 0 slope, then this indicates that there is no corner contribution.

The fly hemibrain dataset contains approximately $35,000$ units in all three directions, and working with a square lattice of the same size can be computationally expensive. To address this issue,

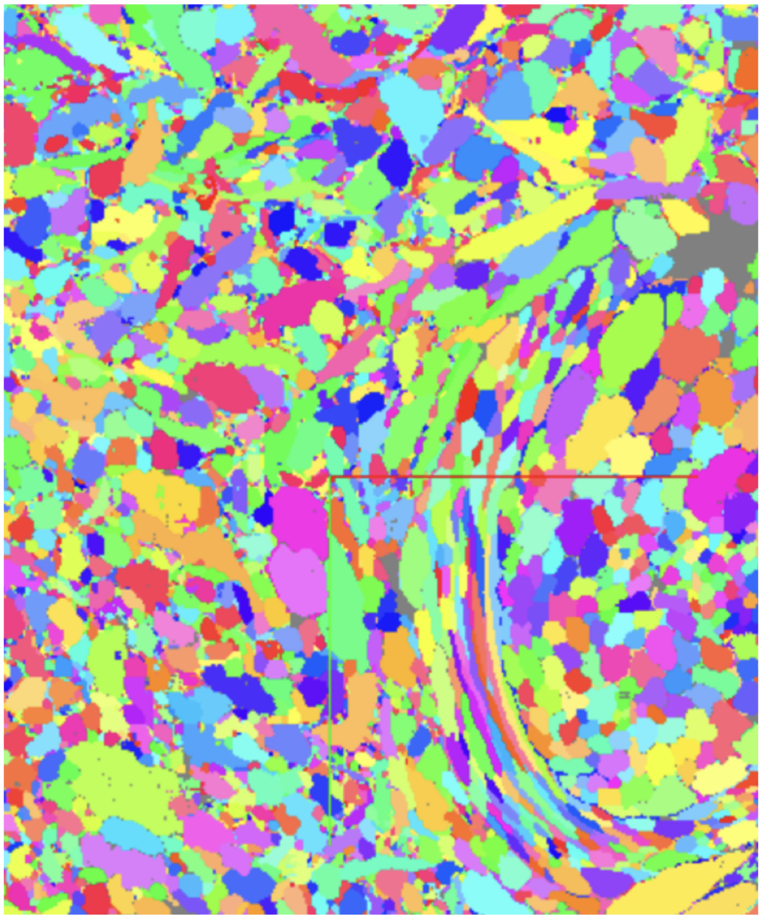


Figure 4.1: The layers of the fly hemibrain dataset exhibit clustering that is qualitatively similar to 2d percolation. Thus, we can perform cluster tomography measurements on the dataset, treating the colored neural cells as distinct clusters.

we employed downsampling by using the “mip” parameter, an nonnegative integer that indicates the resolution of the segmentation data. Data collected with a mip value m is 2^m times less detailed than data collected at mip $= 0$ (Figure 4.2).

To determine the most appropriate mip value for our analysis, we plotted fits of $\kappa(\ell)$ vs. $\log \ell$ for mip values ranging from 0 to 5. We found that the smallest errors in the fit were obtained for mip $= 4$, which was therefore the value we used to sample the segmentation data. Using this higher mip value allowed us to sample the fly hemibrain dataset across all length scales while still maintaining reasonable computational efficiency (Figure 4.3).

As demonstrated in Figure 4.4, we see that the corner contribution κ varies greatly with ℓ. The fit of the first five points (corresponding to linear sizes $L = 64$, 128, 256, 512, and 1024) is given

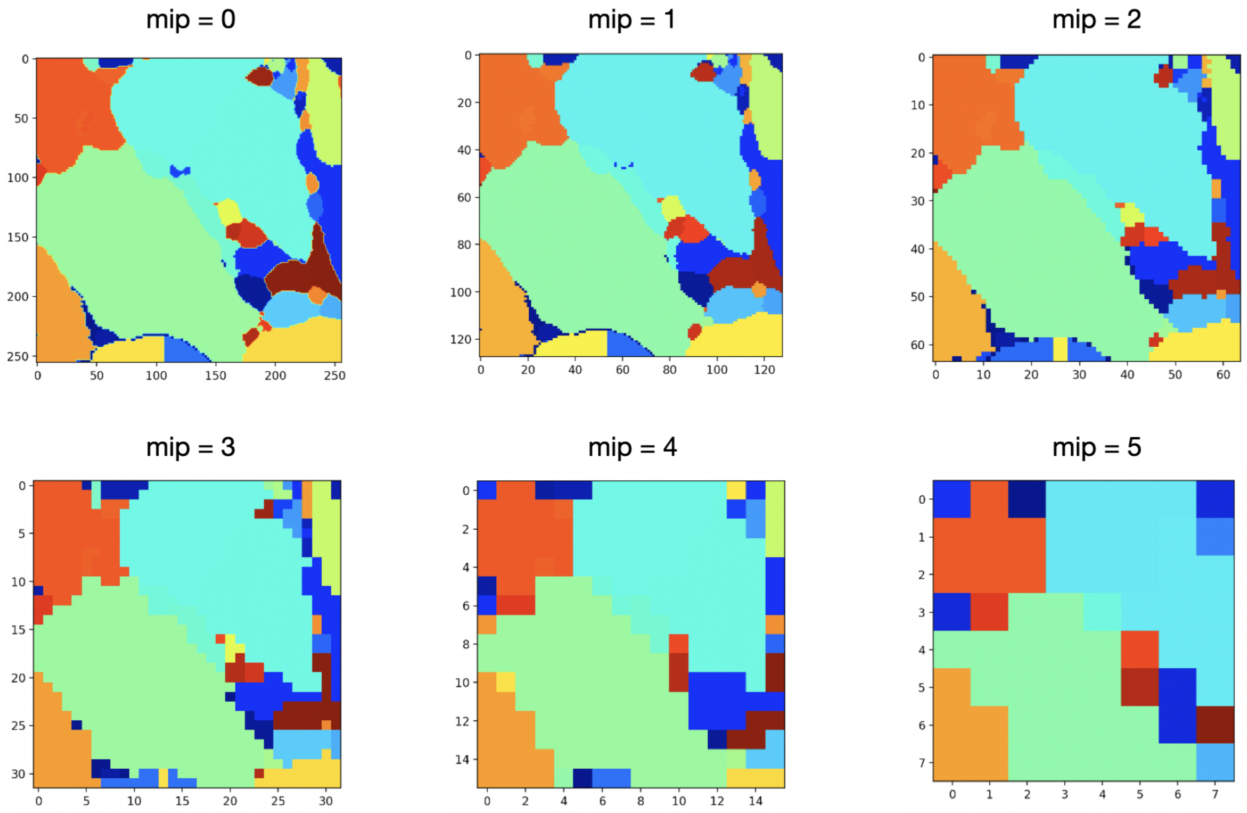


Figure 4.2: Fly segmentation images taken at different mip values. A mip value of 4 was found to correspond to the least resolute image without losing any necessary information.

by

$$\kappa(L) = 0.25(2)\log L - 0.1(4) \tag{4.1}$$

Because in percolation we only see a nonzero corner contribution at criticality, these results imply that the brain exists in a state of structural criticality. Similar methods applied to the human and mouse brain datasets (which were also released on Neuroglancer) also indicate non-constant corner contributions:

$$\kappa_{\text{human}}(L) = 0.19(7)\log L + 0.3(7)$$

$$\kappa_{\text{mouse}}(L) = 0.33(1)\log L + 0.01(9) \tag{4.2}$$

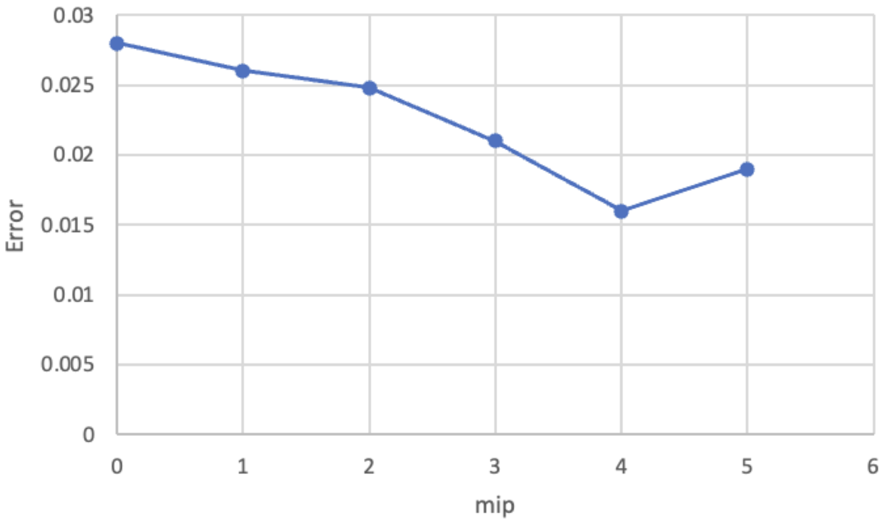


Figure 4.3: Plot of corner contribution fit error vs. mip value. We see that mip $= 4$ corresponds to the lowest error

Of course, these results do not amount to a *proof* of criticality in the brains of these organisms. Additionally, these results treat the brain as homogeneous, when we know that it is not (certain regions of the brain are denser in neurons than others, for instance). These results are, however, a nice indication that more tests of criticality in the brain could prove successful. Indeed, Kovács and Ansell have recently used other tests of universality, such as diverging correlation length and fractal dimension, in order to to demonstrate that the cellular anatomy of the brain appears at or close to a state of structural criticality [22].

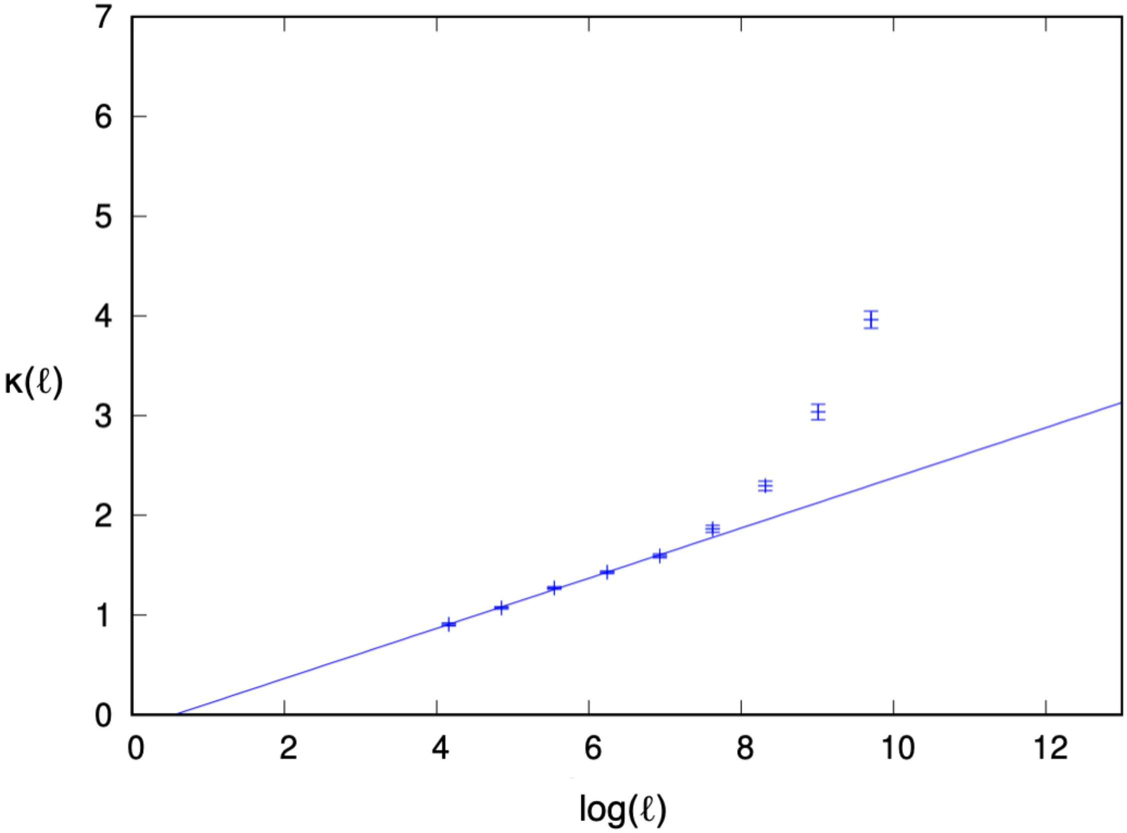


Figure 4.4: A plot of the corner contribution $\kappa(\ell) = b \log \ell$ vs. $\log \ell$ for the fruit fly brain. A nonzero slope indicates a nonzero b, implying that the brain could be in a state of structural criticality. Equation (4.1) gives the fit of the first five data points.

REFERENCES

[1] T. Gowers, J. Barrow-Green, and I. Leader, Eds., *The Princeton Companion to Mathematics*. Princeton: Princeton University Press, 2008, ISBN: 9781400830398 1400830397.

[2] K. Christensen, "Percolation theory," *Lecture Notes*, Oct. 2002.

[3] D. Stauffer and A. Aharony, *Introduction to Percolation Theory*, 2nd. London: Taylor & Francis, 1992.

[4] T. Dugnolle. "Percolation zoom.gif." (2020), (visited on 04/22/2023).

[5] J. L. Cardy, "Lectures on conformal invariance and percolation," *arXiv:0103018*, 2001.

[6] J. Hoshen and R. Kopelman, "Percolation and cluster distribution. I. cluster multiple labeling technique and critical concentration algorithm," *Physical Review B*, vol. 14, no. 8, pp. 3438–3445, 1976.

[7] T. Fricke, *The H-K algorithm*, https://www.ocf.berkeley.edu/ fricke/projects/hoshenkopelman/hoshenkopelman.html, Accessed on April 22, 2023, 2004.

[8] A. Be'er, B. Ilkanaiv, R. Gross, and et al., "A phase diagram for bacterial swarming," *Communications Physics*, vol. 3, no. 1, p. 66, 2020.

[9] C. S. Xu *et al.*, "A connectome of the adult drosophila central brain," *bioRxiv*, 2020.

[10] B. M. Mognetti, A. Šari ć, S. Angioletti-Uberti, A. Cacciuto, C. Valeriani, and D. Frenkel, "Living clusters and crystals from low-density suspensions of active colloids," *Phys. Rev. Lett.*, vol. 111, p. 245 702, 24 Dec. 2013.

[11] R. Yu, H. Saleur, and S. Haas, "Entanglement entropy in the two-dimensional random transverse field ising model," *Physical Review B*, vol. 77, no. 14, p. 140 402, 2008.

[12] I. A. Kovács and F. Iglói, "Corner contribution to percolation cluster numbers in three dimensions," *Physical Review B*, vol. 89, no. 17, p. 174 202, 2014.

[13] I. A. Kovács, F. Iglói, and J. Cardy, "Corner contribution to percolation cluster numbers," *Physical Review B*, vol. 86, no. 21, p. 214 203, 2012.

[14] I. A. Kovács, "Infinitely disordered critical behavior in higher dimensional quantum systems," Ph.D. dissertation, Eötvös Lorand University, 2012.

[15] J. L. Cardy and I. Peschel, "Finite-size dependence of the free energy in two-dimensional critical systems," *Nuclear Physics B*, vol. 300, pp. 377–392, 1988.

[16] J. Cardy, "SLE for theoretical physicists," *Annals of Physics*, vol. 318, no. 1, pp. 81–130, 2005.

[17] J. van den Berg and R. P. Conijn, "The expected number of critical percolation clusters intersecting a line segment," *Electronic Communications in Probability*, vol. 21, no. none, pp. 1–10, 2016.

[18] J. L. Jacobsen and H. Saleur, "Conformal boundary loop models," *Nuclear Physics B*, vol. 788, pp. 137–166, 2008.

[19] R. Yu, H. Saleur, and S. Haas, "Entanglement entropy in the two-dimensional random transverse field Ising model," *Phys. Rev. B*, vol. 77, p. 140 402, 14 Apr. 2008.

[20] J. L. Jacobsen and H. Saleur, "Conformal boundary loop models," *Nuclear Physics B*, vol. 788, pp. 137–166, 2008.

[21] V. Zimmern, "Why brain criticality is clinically relevant: A scoping review," *Frontiers in Neural Circuits*, vol. 14, p. 54, 2020.

[22] H. Ansell and I. A. Kovács, "Signatures of criticality in the physical structure of the brain," *Bulletin of the American Physical Society*, 2023.