

Modelling in Biology Coursework II

Harsh Agrawal (CID: 02320622)

March 24, 2025

Question 1

a. Draw a graphical representation of the Markov Process describing the formation of a single channel, and give the associated rate matrix.

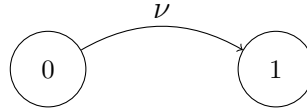


Figure 1: Markov Process for single channel formation from state 0 to 1 with transition rate ν .

The Rate Matrix of a Continuous Time Discrete State Markov Process is given by:

$$\mathbf{Q}_{ij} = \begin{pmatrix} K_{00} & K_{01} \\ K_{10} & K_{11} \end{pmatrix}$$

We know that $K_{01} = 0$ as state 1 is an absorbing state (the channel has formed) and $K_{10} = \nu$ (rate of formation of channel). We can use these values to fill the diagonal as the negative sum of the respective column.

$$\boxed{\mathbf{Q}_{ij} = \begin{pmatrix} -\nu & 0 \\ \nu & 0 \end{pmatrix}} \quad (1)$$

b. Derive the probability $p_0(t)$ that the channel has not formed at time t

To find the probability that the channel hasn't formed at time t , we first start with the evolution equation for the probability vector $\vec{p}(t)$.

$$\frac{d}{dt} \vec{p}(t) = \mathbf{Q} \cdot \vec{p}(t), \quad \text{where } \vec{p}(t) = \begin{pmatrix} p_0(t) \\ p_1(t) \end{pmatrix}$$

We can further expand this to:

$$\frac{d}{dt} \begin{pmatrix} p_0(t) \\ p_1(t) \end{pmatrix} = \begin{pmatrix} -\nu & 0 \\ \nu & 0 \end{pmatrix} \cdot \begin{pmatrix} p_0(t) \\ p_1(t) \end{pmatrix} = \begin{pmatrix} -\nu \cdot p_0(t) \\ \nu \cdot p_1(t) \end{pmatrix}$$

Since we're only interested in $p_0(t)$, we obtain:

$$\frac{d}{dt}p_0(t) = -\nu \cdot p_0(t)$$

We can solve this using the method of separation of variables.

$$\frac{dp_0(t)}{p_0(t)} \cdot \frac{1}{dt} = -\nu$$

Integrating both sides, we get:

$$\int \frac{dp_0(t)}{p_0(t)} \cdot \frac{dt}{dt} = \int -\nu dt \implies \ln p_0(t) = -\nu t + C$$

Exponentiating both sides, we get:

$$p_0(t) = e^{-\nu t + C}$$

We have a known boundary condition at $t = 0$, $p_0(0) = 1$ (the probability of the channel not being formed at time 0 is 1), so we can substitute this into the equation to find C :

$$p_0(0) = e^{-\nu \cdot 0 + C} = e^C = 1 \implies C = 0$$

This yields the final solution:

$$\boxed{p_0(t) = e^{-\nu t}} \tag{2}$$

c. What is the most likely time at which the channel forms?

Since we already know that the time to form a channel is exponentially distributed with the parameter ν , we can use this to find the time where the Probability Density Function is maximized.

$$f_T(t) = \nu e^{-\nu t}$$

We can find the maximum by taking the derivative and setting it to 0. But we can logically also see that at $t = 0$, $f_T(t) = \nu$. As t increases, the probability decreases as the function follows an exponential decay. Therefore, without calculation, we can say that the maximum probability is at $t = 0$ and thus the most likely time at which the channel forms is at:

$$\boxed{t = 0} \tag{3}$$

Question 2

a. Draw a graphical representation of the stochastic process $\mathbf{N}(t)$

We have another Continuous Time Discrete State Markov Process with states $n = 0, 1, 2, \dots, M$. This is a pure birth process (i.e, no transitions can occur from $n \rightarrow n - 1$). Moreover, every transition occurs successively as $n \rightarrow n + 1$ (i.e, no transitions can occur directly from $n \rightarrow n + 2$).

- At state $n = 0$, M channels can be formed independently with rate ν . Thus *the total rate of transition from $n = 0$ to $n = 1$ is $\nu \cdot M$. This makes intuitive sense as at $n = 0$, transitions occur the fastest as any of the M channels can form.*
- At state $n = 1$, $M - 1$ channels are unformed and thus the total rate of transition from $n = 1$ to $n = 2$ is $\nu \cdot (M - 1)$. At state $n = M$, all M channels are formed and thus this is an absorbing state.
- In general, at state k for $k < M$, the rate of transition from $k \rightarrow k + 1$ is $\nu \cdot (M - k)$

We can use this reasoning to draw the stochastic process as follows:

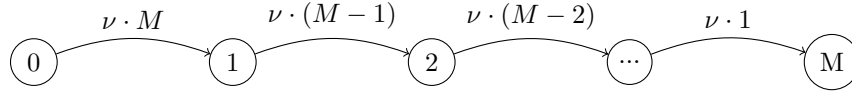


Figure 2: Markov diagram for the stochastic process $\mathbf{N}(t)$.

b. Explain why, for $0 \leq n \leq M$, we may write the following ODE for $p_n(t)$, the probability that $N = n$ at time t . Note that $p_n = 0$ for $n < 0$ or $n > M$ by definition.

$$\frac{dp_n}{dt} = -\nu(M - n)p_n + \nu(M - (n - 1))p_{n-1}$$

We start with the master equation for a generic Continuous Time Discrete State Markov Process which states that the rate of change of the probability of being in state n is given by the sum of the incoming transitions to that state minus the outgoing transitions from that state.

$$\frac{dp_n}{dt} = \sum_{m \neq n} p(n, t + dt | m, t) \cdot p(m, t) - \sum_{n \neq m} p(m, t + dt | n, t) \cdot p(n, t)$$

For our condition, we know that there is a single incoming transition (from $n - 1$ to n) and a single outgoing transition (from n to $n + 1$). We can thus simplify the incoming and outgoing transitions to:

$$\sum_{m \neq n} p(n, t + dt | m, t) \cdot p(m, t) = p(n, t | n - 1, t) \cdot p(n - 1, t) \quad (\text{Incoming Transitions})$$

$$\sum_{n \neq m} p(m, t + dt | n, t) \cdot p(n, t) = p(n + 1, t | n, t) \cdot p(n, t) \quad (\text{Outgoing Transitions})$$

Since we know the transition rates for going from $k \rightarrow k + 1$ as $\nu \cdot (M - k)$, we can substitute these into the master equation to get:

$$p(n, t | n-1, t) \cdot p(n-1, t) = \nu(M - (n-1)) \cdot p_{n-1} \quad (k = n-1)$$

$$p(n+1, t | n, t) \cdot p(n, t) = \nu(M - n) \cdot p_n \quad (k = n)$$

We can now substitute these into the master equation to get:

$$\boxed{\frac{dp_n}{dt} = -\nu(M - n)p_n + \nu(M - (n-1))p_{n-1}} \quad (4)$$

c. What is the stationary distribution π_n of the process?

The stationary distribution π_n is the long-term probability of being in state n as $t \rightarrow \infty$. In a pure birth process, once all M channels form, the system stays in state M forever as it's an absorbing state.

As $t \rightarrow \infty$, the probability of being in state $n = M \rightarrow 1$ (as all channels are formed). Since the probabilities in $\vec{\pi}$ must sum to 1, we can say all other probabilities for $n \neq M \rightarrow 0$ at $t \rightarrow \infty$.

Therefore, the stationary distribution is:

$$\pi_n = \begin{cases} 1 & \text{if } n = M \\ 0 & \text{otherwise} \end{cases}$$

d. Using the ‘multiply and sum’ technique, show that

$$\frac{d}{dt}\langle N(t) \rangle = \nu(M - \langle N(t) \rangle)$$

We know that the expectation of a random variable $N(t)$ and its derivative is given by:

$$\langle N(t) \rangle = \sum_{n=0}^M n \cdot p_n \implies \frac{d}{dt}\langle N(t) \rangle = \sum_{n=0}^M n \cdot \frac{dp_n}{dt}$$

We already evaluated $\frac{dp_n}{dt}$ in part b, so we can substitute it in to get:

$$\begin{aligned} \frac{d}{dt}\langle N(t) \rangle &= \sum_{n=0}^M n \cdot [-\nu(M - n)p_n + \nu(M - (n-1))p_{n-1}] \\ &= \sum_{n=0}^M [-n \cdot \nu(M - n)p_n + n \cdot \nu(M - (n-1))p_{n-1}] \\ &= \sum_{n=0}^M [-n \cdot \nu(M - n)p_n] + \sum_{n=0}^M [n \cdot \nu(M - (n-1))p_{n-1}] \end{aligned}$$

We'll first evaluate the first term:

$$\begin{aligned}
\sum_{n=0}^M [-n \cdot \nu(M-n)p_n] &= \sum_{n=0}^M [-n \cdot \nu \cdot M \cdot p_n + n \cdot \nu \cdot n \cdot p_n] \\
&= -\nu M \sum_{n=0}^M n p_n + \nu \sum_{n=0}^M n^2 p_n = -\nu M \langle N(t) \rangle + \nu \langle N^2(t) \rangle
\end{aligned}$$

To evaluate the second term, we can use an arbitrary variable $k = n - 1$ to get:

$$\sum_{n=0}^M [n \cdot \nu(M - (n-1))p_{n-1}] = \nu \sum_{k=-1}^{M-1} (k+1)(M-k)p_k$$

At $k = -1$, $(k+1)(M-k) = (0)(M+1) = 0$ And at $k = M$, $(k+1)(M-k) = (M+1)(0) = 0$

Thus, we can extend the limits of the sum from $\sum_{k=-1}^{M-1}$ to $\sum_{k=0}^M$ and switch the variable k back to n and then expand the product to get:

$$\begin{aligned}
\nu \sum_{k=0}^M (k+1)(M-k)p_k &= \nu \sum_{n=0}^M (n+1)(M-n)p_n \\
&= \nu \sum_{n=0}^M (nM + M - n^2 - n)p_n \\
&= \nu \cdot M \sum_{n=0}^M n \cdot p_n + \nu \cdot M \sum_{n=0}^M p_n - \nu \sum_{n=0}^M n^2 \cdot p_n - \nu \sum_{n=0}^M n \cdot p_n \\
&= \nu \cdot M \langle N(t) \rangle + \nu \cdot M \cdot 1 - \nu \langle N^2(t) \rangle - \nu \langle N(t) \rangle
\end{aligned}$$

Adding both the sums together, we get:

$$\begin{aligned}
\frac{d}{dt} \langle N(t) \rangle &= -\nu M \langle N(t) \rangle + \nu \langle N^2(t) \rangle + \nu \cdot M \langle N(t) \rangle + \nu \cdot M - \nu \langle N^2(t) \rangle - \nu \langle N(t) \rangle \\
&= \nu \cdot M - \nu \cdot \langle N(t) \rangle
\end{aligned}$$

This gives us the final equation:

$$\boxed{\frac{d}{dt} \langle N(t) \rangle = \nu(M - \langle N(t) \rangle)} \tag{5}$$

e. Solve for $\langle N(t) \rangle$ and sketch (do not use plotting software) the function you obtain. Justify the form of your sketch.

We can solve this ODE by first rearranging the equation above to get:

$$\frac{d}{dt} \langle N(t) \rangle + \nu \cdot \langle N(t) \rangle = \nu \cdot M$$

Using the integrating factor method, we can multiply both sides by $e^{\nu t}$ to get:

$$e^{\nu t} \frac{d}{dt} \langle N(t) \rangle + e^{\nu t} \nu \cdot \langle N(t) \rangle = e^{\nu t} \nu \cdot M$$

$$\frac{d}{dt} (e^{\nu t} \langle N(t) \rangle) = e^{\nu t} \nu \cdot M$$

Integrating both sides, we get:

$$\int \frac{d}{dt} (e^{\nu t} \langle N(t) \rangle) dt = \int e^{\nu t} \nu \cdot M dt$$

$$e^{\nu t} \langle N(t) \rangle = \nu \cdot M \int e^{\nu t} dt$$

$$e^{\nu t} \langle N(t) \rangle = M \cdot e^{\nu t} + C$$

$$\langle N(t) \rangle = M + C \cdot e^{-\nu t}$$

We can now solve for C by using the initial condition $\langle N(0) \rangle = 0$:

$$\langle N(0) \rangle = M + C \cdot e^{-\nu \cdot 0} = 0 \implies C = -M \implies \boxed{\langle N(t) \rangle = M(1 - e^{-\nu t})}$$

The graph can be sketched as follows:

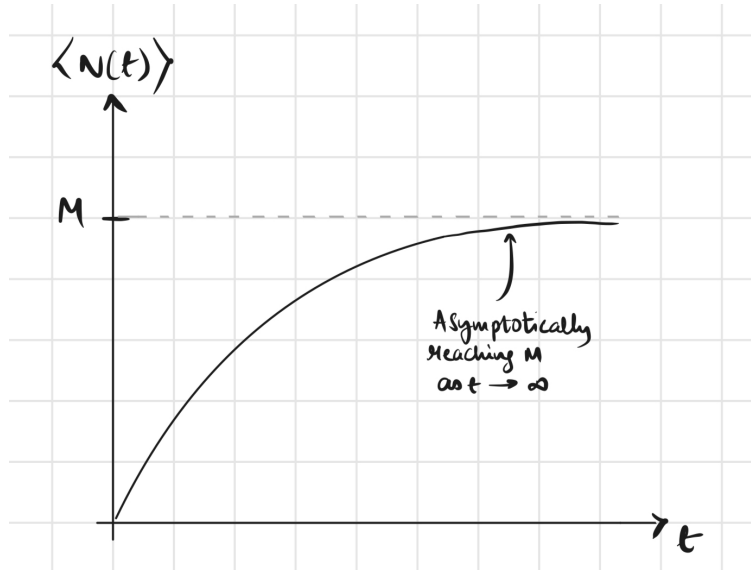


Figure 3: Graph of $\langle N(t) \rangle = M(1 - e^{-\nu t})$ showing asymptotic behavior approaching M as $t \rightarrow \infty$. The curve starts at origin and increases monotonically with decreasing slope.

Question 3: The tissue as a network (/37)

a. What is the degree distribution of the cubic network when $t = 1/\nu$?

We know from Q2b, that the probability of channel not formed at time t is given by $p_0(t) = e^{-\nu t}$. As the total probability must sum to 1, we can say that the probability of a channel being formed at time t is given by $p_1(t) = 1 - p_0(t) = 1 - e^{-\nu t}$. Thus, at $t = 1/\nu$, we have:

$$p_1(1/\nu) = 1 - e^{-1} \approx 0.632$$

The number of channels connected to a node (its degree) here ranges from 0 to 6. Since every event is independent, this follows a binomial distribution where probability of success is $1 - e^{-1}$ and probability of failure is e^{-1} and the number of trials is 6.

$$P(X = k, t = 1/\nu) = \binom{6}{k} (1 - e^{-1})^k (e^{-1})^{6-k} \quad (6)$$

$$P(0) = (e^{-1})^6 \approx 0.002479$$

$$P(1) = 6(1 - e^{-1})(e^{-1})^5 \approx 0.025534$$

$$P(2) = 15(1 - e^{-1})^2(e^{-1})^4 \approx 0.109714$$

$$P(3) = 20(1 - e^{-1})^3(e^{-1})^3 \approx 0.251452$$

$$P(4) = 15(1 - e^{-1})^4(e^{-1})^2 \approx 0.323747$$

$$P(5) = 6(1 - e^{-1})^5(e^{-1}) \approx 0.222409$$

$$P(6) = (1 - e^{-1})^6 \approx 0.063665$$

b. Matlab Question 1

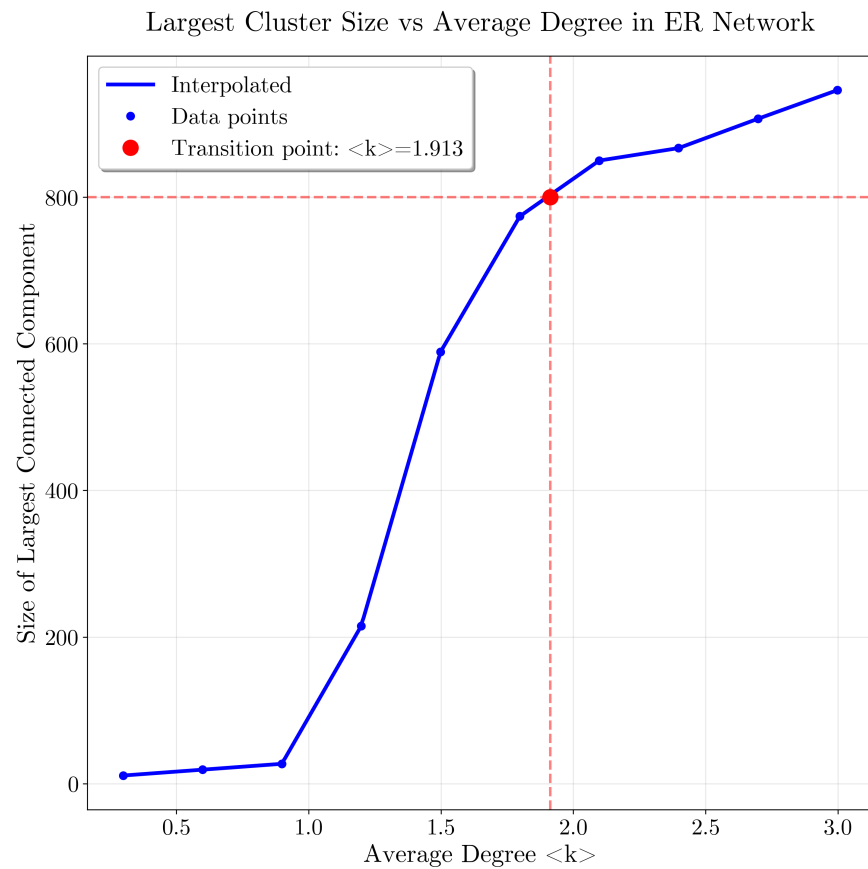


Figure 4: Size of largest connected component vs average node degree in an Erdős-Rényi network. A phase transition occurs near the critical degree, showing emergence of a giant component.

The following Python code was used to generate the plot in Figure 4:

```
1 import numpy as np
2 # networkx library is required for this code
3 import networkx as nx
4 import matplotlib.pyplot as plt
5 from scipy.interpolate import interp1d
6
7 # Set font family and sizes for better quality
8 plt.rcParams.update({
9     {
10         "font.family": "CMU Serif",
11         "font.size": 18,
12         "axes.labelsize": 20,
13         "axes.titlesize": 22,
14         "xtick.labelsize": 18,
```



```

15         "ytick.labelsize": 18,
16         "legend.fontsize": 18,
17     }
18 )
19
20 # Parameters
21 N = 1000
22 p_values = np.linspace(0.0003, 0.003, 10)
23 avg_degree = p_values * (N - 1)
24 largest_cluster_sizes = []
25
26 # Simulate ER graph for each p
27 for p in p_values:
28     A = np.triu(np.random.rand(N, N) < p, 1)
29     A = A + A.T
30     G = nx.from_numpy_array(A)
31     largest_cluster_sizes.append(max(len(c) for c in nx.connected_components(G)))
32
33 # Create interpolation function
34 f = interp1d(avg_degree, largest_cluster_sizes, kind='linear')
35
36 # Create finer x points for smoother curve
37 x_smooth = np.linspace(min(avg_degree), max(avg_degree), 200)
38 y_smooth = f(x_smooth)
39
40 # Find transition point at 80% threshold using interpolated values
41 threshold = 0.8 * N
42 k_transition = x_smooth[next(i for i, size in enumerate(y_smooth) if size >= threshold)]
43
44 # Create high quality figure
45 plt.figure(figsize=(10, 10), dpi=300)
46 plt.plot(x_smooth, y_smooth, 'b-', linewidth=3, label='Interpolated')
47 plt.plot(avg_degree, largest_cluster_sizes, 'b.', markersize=12, label='Data points')
48 plt.axvline(x=k_transition, color="r", linestyle="--", alpha=0.5, linewidth=2)
49 plt.axhline(y=threshold, color="r", linestyle="--", alpha=0.5, linewidth=2)
50 plt.plot(
51     k_transition,
52     threshold,
53     "ro",
54     markersize=12,
55     label=f"Transition point: <k>={k_transition:.3f}",
56 )
57
58 plt.xlabel("Average Degree <k>", fontweight="bold")
59 plt.ylabel("Size of Largest Connected Component", fontweight="bold")
60 plt.title("Largest Cluster Size vs Average Degree in ER Network", pad=20)
61 plt.grid(True, alpha=0.3)
62 plt.legend(frameon=True, fancybox=True, shadow=True)
63
64 # Add padding to prevent label cutoff
65 plt.tight_layout()
66 plt.show()

```

Listing 1: Erdős-Rényi Network Analysis

The code simulates an Erdős-Rényi network with 1000 nodes across various probability values. It calculates the size of the largest connected component for each probability and plots this against the average degree. The transition point where the largest component reaches 80% of the network size is identified.

c. Matlab Question 2

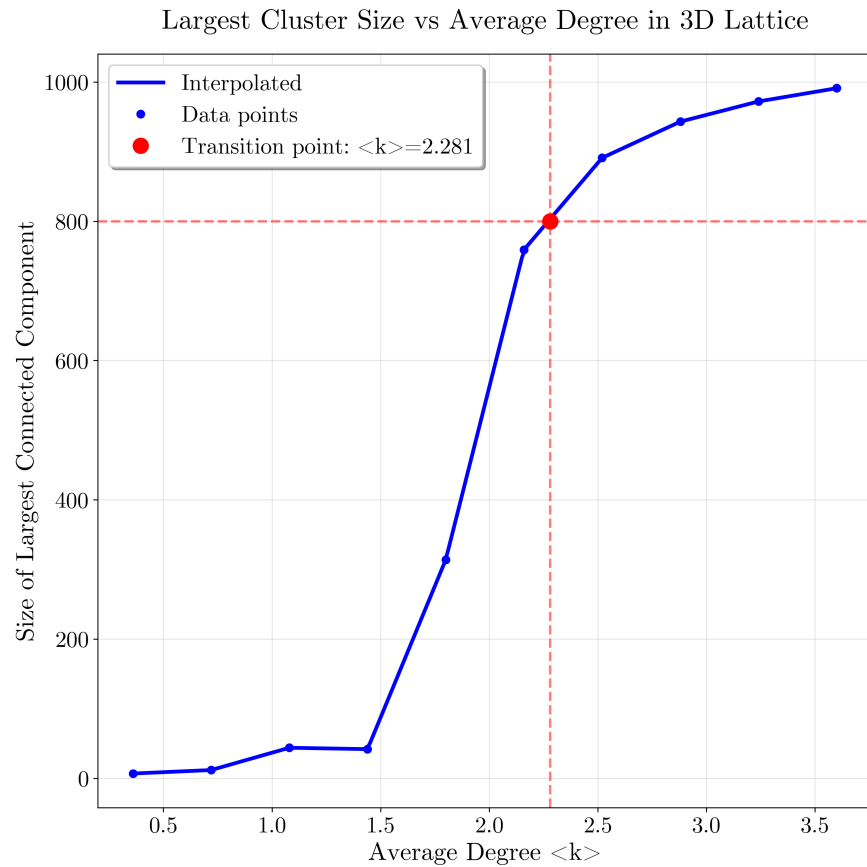


Figure 5: Percolation in a $10 \times 10 \times 10$ Cubic Lattice: Largest Cluster Size vs. Average Degree with Nearest-Neighbor Connections

The following Python code was used to generate the plot in Figure 5:

```

1 # CODE INDEPENDENT OF 3.B for Reproducibility
2 import numpy as np
3 import networkx as nx
4 import matplotlib.pyplot as plt
5 from scipy.interpolate import interp1d
6
7 # Set font sizes and DPI for high-quality output
8 plt.rcParams.update(
9     {
10         "font.family": "CMU Serif",
11         "font.size": 18,
12         "axes.labelsize": 20,
13         "axes.titlesize": 22,
14         "xtick.labelsize": 18,

```

```

15         "ytick.labelsize": 18,
16         "legend.fontsize": 18,
17     }
18 )
19
20 # Parameters
21 N = 1000 # 10x10x10 lattice = 1000 nodes
22 dims = (10, 10, 10)
23 p_values = np.arange(0.06, 0.61, 0.06) # 0.06 to 0.6, step 0.06
24 avg_degree = 6 * p_values # Max 6 neighbors in 3D lattice
25 largest_cluster_sizes = []
26
27
28 # Function to convert 3D coordinates to node index
29 def xyz_to_index(x, y, z):
30     return 100 * x + 10 * y + z
31
32
33 # Simulate cubic lattice for each p
34 for p in p_values:
35     G = nx.grid_graph(dim=dims) # 10x10x10 lattice
36     # Remove edges randomly: keep with probability p
37     edges_to_remove = [(u, v) for u, v in G.edges() if np.random.rand() > p]
38     G.remove_edges_from(edges_to_remove)
39     largest_cluster_sizes.append(max(len(c) for c in nx.connected_components(G)))
40
41 # Interpolation for smoother curve
42 f = interp1d(avg_degree, largest_cluster_sizes, kind="linear")
43 x_smooth = np.linspace(min(avg_degree), max(avg_degree), 200)
44 y_smooth = f(x_smooth)
45
46 # Find <k> at 80% threshold (800 nodes)
47 threshold = 0.8 * N
48 k_transition = x_smooth[next(i for i, size in enumerate(y_smooth) if size >= threshold)]
49
50 # Plot
51 plt.figure(figsize=(10, 10), dpi=300)
52 plt.plot(x_smooth, y_smooth, "b-", linewidth=3, label="Interpolated")
53 plt.plot(avg_degree, largest_cluster_sizes, "b.", markersize=12, label="Data points")
54 plt.axvline(x=k_transition, color="r", linestyle="--", alpha=0.5, linewidth=2)
55 plt.axhline(y=threshold, color="r", linestyle="--", alpha=0.5, linewidth=2)
56 plt.plot(
57     k_transition,
58     threshold,
59     "ro",
60     markersize=12,
61     label=f"Transition point: <k>={k_transition:.3f}",
62 )
63
64 plt.xlabel("Average Degree <k>", fontweight="bold")
65 plt.ylabel("Size of Largest Connected Component", fontweight="bold")
66 plt.title("Largest Cluster Size vs Average Degree in 3D Lattice", pad=20)
67 plt.grid(True, alpha=0.3)
68 plt.legend(frameon=True, fancybox=True, shadow=True)
69 plt.tight_layout()
70 plt.show()

```

Listing 2: Cubic Lattice Percolation Analysis