Uniwersytet Wrocławski
Wydział Fizyki i Astronomii

Analiza perkolacji przekrywających się hipersfer w wymiarach D = 3, 4, 5

Piotr Brzeski

Rozprawa doktorska
Promotor: dr hab. Grzegorz Kondrat

Wrocław, 2023

Spis treści

1 Cel rozprawy **5**

2 Perkolacja **7**

2.1 Wstęp 7

2.2 Perkolacja węzłów 8

2.3 Perkolacja na sieci wiązań 11

2.4 Warunki brzegowe 12

2.5 Wielkości opisujące perkolację 12

2.6 Uniwersalne wykładniki krytyczne 14

2.6.1 Wartości wykładników krytycznych 15

2.6.2 Związki pomiędzy wykładnikami krytycznymi 16

2.7 Inne modele perkolacji 16

3 Opis wybranych metod numerycznych **19**

3.1 Poprawki do skalowania skończonego rozmiaru 19

3.2 Wielki układ kanoniczny 20

3.3 Wydajny algorytm symulacji 21

3.4 Wyłączona objętość 23

3.5 Rozszerzone sąsiedztwo 23

4 Perkolacja hipersfer **31**

4.1 Model 31

4.1.1 Wyłączona objętość hipersfery 33

4.2 Algorytm i optymalizacje 35

4.2.1 Odwzorowanie zawartości hipersfery 35

4.2.2 Odwzorowanie połączeń 37

4.2.3 Znajdowanie klastrów 37
4.2.4 Generator liczb losowych 38
4.3 Obliczenia 39
4.3.1 Oszacowanie błędów 41
4.4 Wyniki 41
4.4.1 Perkolacja węzłów dla $k = 1$ 41
4.4.2 Perkolacja hipersfer 43

5 Podsumowanie **47**

Bibliografia **49**

Rozdział 1

Cel rozprawy

Celem niniejszej rozprawy jest analiza perkolacji hipersfer w modelach wielowymiarowych dla wymiarów od 3 do 5.

Punktem wyjścia przeprowadzonej przeze mnie analizy są dane pochodzące z symulacji komputerowych rozpatrywanego modelu zdyskretyzowanych hipersfer na sieciach hipersześciennych. Aby umożliwić przeprowadzenie takich symulacji dla odpowiednio dużych systemów, musiałem zaprojektować wydajne struktury danych zawierające informacje o badanych hipersferach i połączeniach między nimi. Konieczne było zoptymalizowanie tych struktur pod względem ilości zajmowanej pamięci oraz szybkości dostępu do przechowywanych w nich danych. Ponadto, ze względu na bardzo duże rozmiary badanych hipersfer, opracowałem wydajne algorytmy do wypełnienia tych struktur danymi.

Wykorzystując dane z przeprowadzonych symulacji Monte–Carlo wyznaczyłem wartości wykładnika krytycznego długości korelacji ν dla rozpatrywanych wymiarów oraz progów perkolacji dla perkolacji hipersfer o różnych średnicach.

Następnie wykorzystałem otrzymane wartości krytyczne dla skończonych hipersfer do wyznaczenia wartości progów perkolacji nachodzących na siebie hipersfer w przypadku ciągłym w wymiarach 3, 4, i 5, w granicy średnicy hipersfery $k \to \infty$.

Rozdział 2

Perkolacja

2.1 Wstęp

Perkolacje są procesami, które pokazują jak zestaw prostych reguł może prowadzić do powstawania skomplikowanych systemów. W dalszej części tego rozdziału wyjaśniłem czym są procesy perkolacji, oraz, na prostym przykładzie, pokazałem sposób w jaki można przeprowadzać symulację takiego procesu na dwuwymiarowej sieci kwadratowej. W kolejnych podrozdziałach zamieściłem przykłady wielkości, których używa się do opisywania systemów, w których bada się perkolacje. Następnie opisałem ich zachowanie w momencie powstawania struktury perkolującej. Wtedy to właśnie można zaobserwować najciekawszą właściwość omawianych systemów, czyli przejście fazowe, w pobliżu którego są one opisywane prawami potęgowymi z uniwersalnymi wartościami wykładników krytycznych.

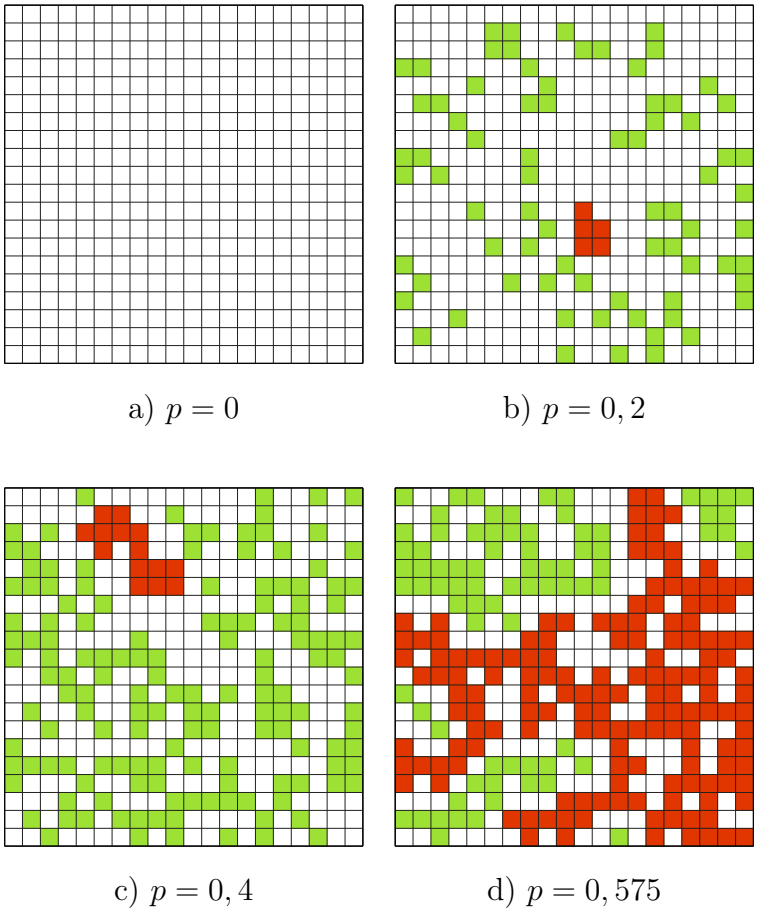
Rozdział 3 zawiera opis metod stosowanych w analizie procesów perkolacji, z których skorzystałem podczas badania perkolacji hipersfer. Ponieważ do badania perkolacji przy użyciu metod Monte–Carlo konieczne jest wykonanie symulacji jak największych układów w możliwie krótkim czasie, w jednym z podrozdziałów opisałem wydajny algorytm, który to umożliwia.

W rozdziale 4 zaprezentowałem analizę perkolacji hipersfer w 3, 4 i 5 wymiarach. Składa się ona z opisu algorytmu, wraz z zastosowanymi w nim optymalizacjami, dzięki któremu byłem w stanie przeprowadzić wiele symulacji powstawania struktur perkolujących w skończonych układach zdyskretyzowanych hipersfer. Następnie za-

prezentowałem sposób w jaki, na podstawie wyników przeprowadzonych przeze mnie symulacji, wyznaczyłem, między innymi, wartości progów perkolacji dla przypadku ciągłego.

2.2 Perkolacja węzłów

Jednym z najprostszych modeli perkolacji jest model oparty na sieci kwadratowej, takiej jak widoczna na **Rysunku 2.1a**.



a) $p = 0$ b) $p = 0,2$

c) $p = 0,4$ d) $p = 0,575$

Rysunek 2.1: Przykłady wypełnienia sieci kwadratowej o rozmiarze 20×20 dla różnych wartości prawdopodobieństwa wypełnienia pola p. Największy klaster jest zaznaczony kolorem czerwonym.

Każde z pól początkowo pustej sieci wypełniamy z pewnym, ustalonym prawdopodobieństwem p. W wyniku takiej procedury mogą pojawić się grupy wypełnionych

pól przylegających do siebie bokami. Takie połączone pola tworzą klastry. Dla małych wartości prawdopodobieństwa wypełnienia pola p powstanie kilka małych klastrów. Jednak gdy będziemy powtarzać naszą procedurę dla coraz większych wartości p, będziemy dostawiać układy zawierające coraz więcej klastrów. Wraz z dalszym wzrostem prawdopodobieństwa zapełnienia pola klastry będą większe, ale będzie ich mniej. W końcu, przy odpowiednio dużym prawdopodobieństwie p powstanie klaster, który połączy ze sobą wszystkie brzegi rozpatrywanej sieci. Taki klaster, nazywany klastrem perkolującym, został pokazany na **Rysunku 2.1d**.

Teraz możemy zadać sobie pytanie:
„Jaka jest minimalna wartość prawdopodobieństwa wypełnienia pola sieci, przy której może powstać klaster perkolujący?”.

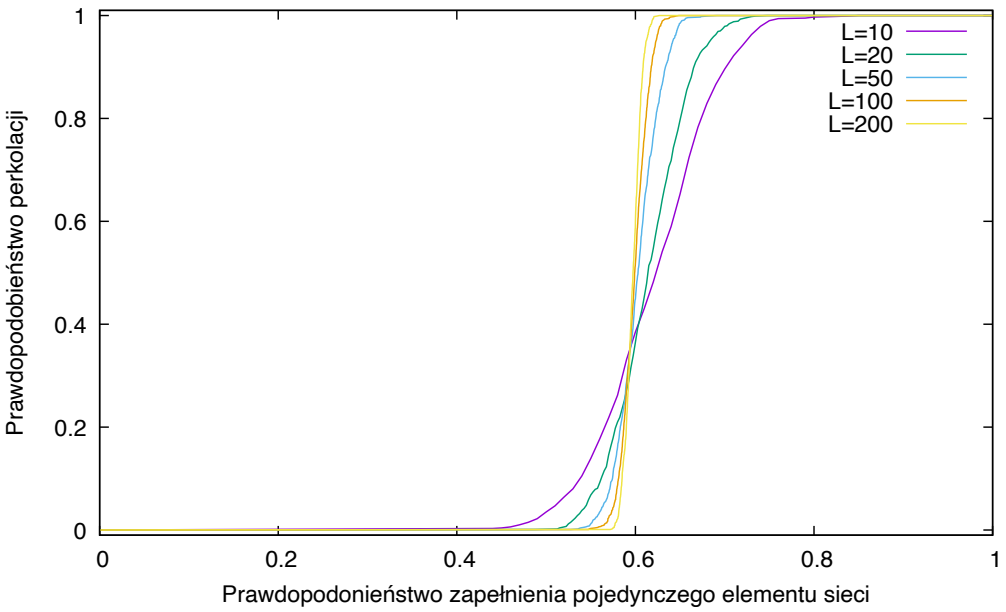
Szukane prawdopodobieństwo nazywamy *prawdopodobieństwem krytycznym* lub *progiem perkolacji* i oznaczamy jako p_c. W celu jego znalezienia, musimy generować układy z coraz większymi wartościami prawdopodobieństwa p i dla każdego z nich sprawdzać czy nie powstał klaster perkolujący.

W związku z tym, że sieć jest wypełniana w sposób losowy, każde powtórzenie opisanego procesu może dać inny wynik. Z tego powodu musimy wykonać wiele symulacji (im więcej, tym wyniki będą bardziej dokładne), w wyniku których otrzymamy nie konkretną wartość szukanego progu perkolacji, a prawdopodobieństwo wystąpienia perkolacji dla różnych wartości prawdopodobieństwa wypełnienia elementu sieci.

Wykresy prezentujące zależność prawdopodobieństwa powstania klastra perkolującego od prawdopodobieństwa wypełnienia elementu sieci, dla różnych rozmiarów sieci, pokazane są na **Rysunku 2.2**. Widać na nim, że zależność ta ma postać funkcji sigmoidalnej, która staje się coraz bardziej stroma wraz ze wzrostem wielkości układu. Można oczekiwać, że w granicy termodynamicznej, czyli w granicy nieskończonego układu, pojawi się przejście fazowe — nieciągłość w punkcie p_c. Z tego powodu najbardziej interesujące byłyby wyniki symulacji na sieci nieskończonej, której jednak nie jesteśmy w stanie przeprowadzić. Na szczęście, dzięki zastosowaniu szybkich komputerów i zoptymalizowanych algorytmów, możemy prowadzić symulacje coraz większych systemów, co daje nam coraz dokładniejsze wyniki.

Obecnie znana wartość progu perkolacji dla sieci kwadratowej wynosi [1]:

$$p_c = 0,59274605079210(2). \tag{2.1}$$



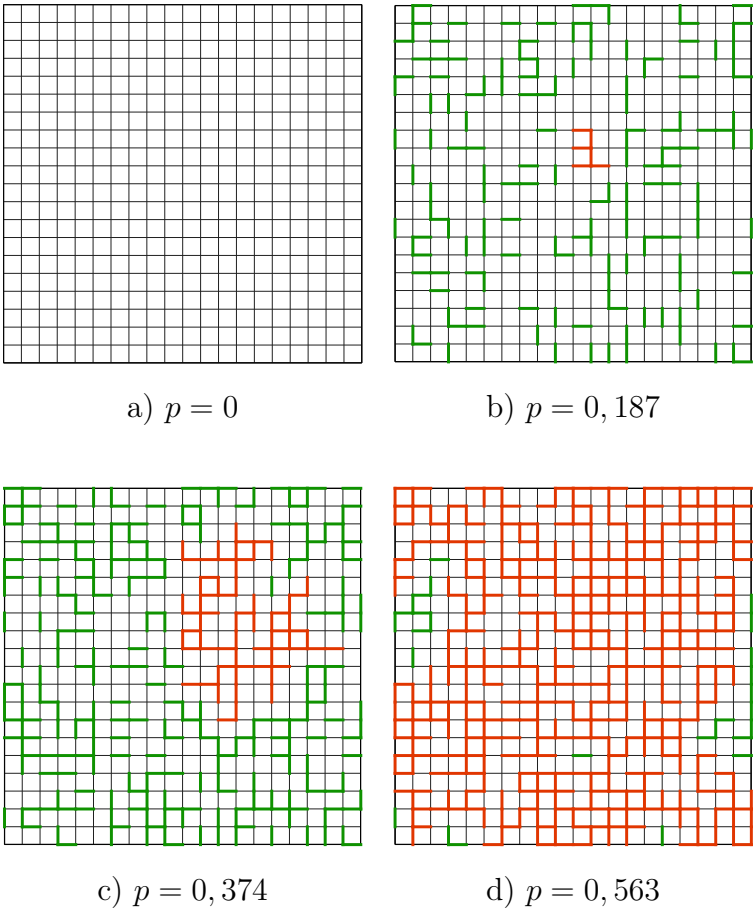
Rysunek 2.2: Zależność prawdopodobieństwa powstania klastra perkolującego od prawdopodobieństwa zapełnienia pojedynczego elementu sieci dla sieci kwadratowych $L \times L$ o różnych rozmiarach. Dane z 1000 symulacji dla każdej zaprezentowanej wartości L.

Przedstawiony powyżej model perkolacji na sieci kwadratowej nazywany jest zwykle *perkolacją węzłów* (ang. site percolation). Nazwa ta odnosi się do innej reprezentacji opisanego układu. Zamiast układu połączonych brzegami kwadratowych pól, możemy wyobrazić sobie sieć składającą się z przecinających się ze sobą pionowych i poziomych linii. Punkty ich przecięć, czyli węzły sieci, które mogą być wypełnione lub puste, odpowiadają kwadratowym polom, a odcinki łączące węzły są odpowiednikami wspólnych krawędzi łączących te pola. Obie te reprezentacje są topologicznie równoważne. W dalszych częściach będę używał wymiennie pojęć *węzeł sieci* lub *element sieci*, prezentując rysunki wybranych układów w postaci przylegających do siebie kwadratowych pól, ze względu na ich większą czytelność.

Perkolacja węzłów, poza zaprezentowanym powyżej przykładem sieci kwadratowej, może być badana także na sieciach o innych topologiach — na przykład sieciach trójkątnych czy sześciokątnych. Można także rozpatrywać sieci wielowymiarowe.

2.3 Perkolacja na sieci wiązań

Innym dwuwymiarowym modelem, często stosowanym do analizy perkolacji, jest model sieci wiązań (ang. bond percolation). W modelu tym rozpatrujemy krawędzie łączące węzły rozpatrywanej sieci. Każda z krawędzi reprezentuje połączenie między dwoma punktami, które może być otwarte z prawdopodobieństwem p lub zamknięte z prawdopodobieństwem $1 - p$. Połączone ze sobą otwarte połączenia tworzą klastry. Wraz ze wzrostem wartości prawdopodobieństwa p, klastry są coraz większe, a przy wartości krytycznej p_c powstaje klaster perkolujący, który łączy przeciwległe brzegi układu. Przykłady perkolacji wiązań, dla różnych wartości prawdopodobieństwa p, na sieci kwadratowej przedstawia **Rysunek 2.3**.



Rysunek 2.3: Przykłady perkolacji wiązań na sieci kwadratowej o rozmiarze 20×20 dla różnych wartości prawdopodobieństwa wypełnienia wiązania p. Na kolorowo oznaczone są otwarte wiązania. Zaznaczone na czerwono tworzą największy klaster.

Tak zdefiniowana perkolacja na sieci wiązań, jest mniej ogólna od perkolacji węzłów. Wynika to z tego, że każdy model perkolacji wiązań może być przedstawiony za pomocą równoważnego mu modelu perkolacji węzłów na innej sieci. Natomiast nie każdemu modelowi sieci węzłów odpowiada model sieci wiązań [2].

2.4 Warunki brzegowe

Tak jak wspomniałem wcześniej, celem przeprowadzania symulacji na coraz większych systemach jest uzyskanie zachowania jak najbardziej zbliżonego do przypadku nieskończonego. Jednak nawet bardzo duże modele skończone różnią się od układu nieskończenie dużego. Jedną z głównych różnic pomiędzy nimi jest występowanie brzegów. W sposób oczywisty zachowanie się układu na brzegach różni się od jego zachowania we wnętrzu. Aby zminimalizować wpływ brzegów na wyniki uzyskane z symulacji, zwykle stosuje się modele z cyklicznymi warunkami brzegowymi.

W takim modelu zakłada się, że dwa przeciwległe brzegi są ze sobą połączone. W przypadku dwuwymiarowym, po połączeniu brzegów w obydwu kierunkach, rozpatrywany układ staje się torusem na powierzchni którego nie występują brzegi. Rozpatrując taki układ musimy przyjąć nową definicję klastra perkolującego, ponieważ nie możemy już wymagać aby łączył on nieistniejące brzegi.

Można na przykład przyjąć, że za klaster perkolujący uznajemy klaster opasujący układ (ang. wrapping cluster). Jest to klaster, wewnątrz którego istnieje droga, po której można przechodzić wokół układu.

2.5 Wielkości opisujące perkolację

W celu scharakteryzowania stanu badanego układu stosuje się różne wielkości. Poniżej podałem kilka z nich, wraz z opisem.

- $P_\infty(p)$ – Siła klastra perkolującego (ang. strength of the percolating cluster)

  Prawdopodobieństwo tego, że wybrana losowo komórka należy do nieskończonego klastra perkolującego, dla prawdopodobieństwa zapełnienia pojedynczej komórki sieci wynoszącego p.

- $\xi(p)$ – Długość korelacji (ang. correlation length).

  Poniżej progu perkolacji, czyli dla $p < p_c$ jest to średnica typowego klastra w systemie, a powyżej p_c typowa wielkość obszaru złożonego z sąsiadujących ze sobą, niezapełnionych elementów sieci. Dla $p > p_c$, ξ jest więc skalą wielkości, powyżej której system jest makroskopowo jednolity.

- $n_s(p)$ – Znormalizowana średnia liczba klastrów wielkości s.

  Dla układu składającego się z M elementów sieci, i danego prawdopodobieństwa zapełnienia komórki sieci p, możemy policzyć średnią liczbę klastrów składających się z s komórek. Po podzieleniu otrzymanej wartości przez wielkość sieci M otrzymujemy znormalizowaną średnią liczbę klastrów wielkości s przypadającą na jedną komórkę sieci.

- S – Średnia wielkość klastra (ang. mean cluster size).

  Jest to średnia liczba elementów sieci należących do klastra zawierającego losowo wybraną komórkę:

$$S = \frac{\sum_s s^2 n_s(p)}{\sum_s s n_s(p)}. \tag{2.2}$$

- φ – Stopień zapełnienia (ang. volume fraction).

  Poziom zapełnienia sieci zdefiniowany jako:

$$\varphi = nv/V, \tag{2.3}$$

  gdzie n jest liczbą zapełnionych węzłów sieci, v jest objętością pojedynczego węzła sieci, a V objętością całej sieci. Średnią wartość φ, przy której powstaje klaster perkolujący, oznaczamy jako φ_c i nazywamy krytycznym stopniem zapełnienia (ang. critical volume fraction) lub po prostu progiem perkolacji.

- η – Gęstość liczbowa (ang. reduced number density).

  W modelach perkolacji obiektów o objętości V_o (nakładających się lub nie), poziom wypełnienia systemu możemy opisać przy pomocy gęstości liczbowej zdefiniowanej jako:

$$\eta = NV_o/V, \tag{2.4}$$

  gdzie N jest liczbą obiektów dodanych do systemu, a V całkowitą objętością systemu.

Związek pomiędzy stopniem zapełnienia φ i gęstością liczbową η, obowiązujący w granicy termodynamicznej, opisuje poniższe równanie [3]:

$$\varphi = 1 - \exp(-\eta). \tag{2.5}$$

2.6 Uniwersalne wykładniki krytyczne

Jedną z najbardziej interesujących i jednocześnie najważniejszych właściwości procesów perkolacji jest ich zachowanie w pobliżu progu perkolacji, gdzie wartości wielu z charakteryzujących perkolację wielkości można opisać za pomocą prawa potęgowego. Co jest najważniejsze, wykładniki krytyczne charakteryzujące te prawa, mają wartości uniwersalne – zależne jedynie od liczby wymiarów systemu. Oznacza to, że zachowanie się systemu w pobliżu progu perkolacji nie zależy od jego struktury, ani od tego czy rozważamy np. perkolację na sieci wiązań czy perkolację węzłów.

Poniżej przedstawiam przykłady, obowiązujących w pobliżu progu perkolacji p_c, praw potęgowych z uniwersalnymi wykładnikami krytycznymi:

- Wykładnik Fishera τ

  Rozkład znormalizowanej, średniej liczby klastrów $n_s(p)$ w progu perkolacji p_c opisany jest prawem potęgowym z wykładnikiem τ:

  $$n_s(p_c) \sim s^{-\tau}. \tag{2.6}$$

- Wykładnik Ω

  Wykładnik krytyczny Ω związany jest z poprawkami do skalowania skończonego rozmiaru, które są opisane w rozdziale 3.1. Występuje on, na przykład, w rozwinięciu rozkładu $n_s(p_c)$:

  $$n_s(p_c) \sim s^{-\tau}(1 + As^{-\Omega}), \tag{2.7}$$

  gdzie A jest stałą, a $s \to \infty$.

- Wykładnik σ

  Poniżej progu perkolacji $p < p_c$ nie istnieje nieskończony klaster perkolujący. W związku z tym można wyznaczyć maksymalną wielkość klastra s_{max}. Wielkość ta jest opisana za pomocą poniższego prawa potęgowego:

$$s_{max}(p) \sim |p - p_c|^{-1/\sigma}. \tag{2.8}$$

- Wykładnik β

  Siła klastra perkolującego P_∞ powyżej progu perkolacji opisana jest prawem potęgowym z wykładnikiem β:

$$P_\infty(p) \sim (p - p_c)^{\beta}. \tag{2.9}$$

- Wykładnik ν

  Wykładnik krytyczny ν jest zdefiniowany poprzez zachowanie długości korelacji w pobliżu punktu krytycznego:

$$\xi(p) \sim |p - p_c|^{-\nu}. \tag{2.10}$$

- Wykładnik γ

  Wartość średniej wielkości klastra $S(p)$ w pobliżu progu perkolacji p_c opisana jest za pomocą prawa potęgowego zawierającego wykładnik γ:

$$S(p) \sim |p - p_c|^{-\gamma}. \tag{2.11}$$

2.6.1 Wartości wykładników krytycznych

W **Tabeli 2.1** przedstawiłem wartości wymienionych powyżej uniwersalnych wykładników krytycznych dla wymiarów od 1 do 5. Znane dokładnie wartości wykładników dla wymiarów $D = 1$ [4] i $D = 2$ [5] wyróżniłem pogrubioną czcionką. Wartości pozostałych wykładników zostały określone na podstawie analizy wyników symulacji Monte–Carlo.

D	1	2	3	4	5
τ	**2**	**187/91**	2,18909(5) [6]	2,3142(5) [7]	2,419(1) [7]
Ω	–	**72/91**	0,77(3) [7]	0,4008 [8]	0,210(2) [9]
σ	**1**	**36/91**	0,4524(6) [6]	0,4789(14) [9]	0,49396(13) [9]
β	**0**	**5/36**	0,429(4) [9]	0,658(1) [9]	0,8454(2) [9]
ν	**1**	**4/3**	0,8774(13) [10]	0,6852(28) [10]	0,5723(18) [10]
γ	**1**	**43/18**	1,78(3) [9]	1,430(6) [9]	1,1792(7) [9]

Tabela 2.1: Wartości wykładników krytycznych dla wymiarów od 1 do 5.

2.6.2 Związki pomiędzy wykładnikami krytycznymi

Opisane powyżej uniwersalne wykładniki krytyczne nie są od siebie niezależne. Związki pomiędzy nimi, które zostały opisane na przykład w [5], są następujące:

$$\beta = \frac{\tau - 2}{\sigma} \tag{2.12}$$

$$\gamma = \frac{3 - \tau}{\sigma} \tag{2.13}$$

$$\beta + \gamma = \frac{1}{\sigma} \tag{2.14}$$

$$\nu = \frac{2\beta + \gamma}{D} = \frac{\tau - 1}{D\sigma} \tag{2.15}$$

gdzie D oznacza liczbę wymiarów.

2.7 Inne modele perkolacji

Opisane powyżej przykłady perkolacji na sieciach węzłów czy wiązań są jednymi z prostszych. Zagadnienie perkolacji jest dużo bardziej złożone i obejmuje badanie wielu różnych modeli, na przykład zawierających obiekty o różnych kształtach i orientacjach oraz w różnych liczbach wymiarów. Rozpatrywane są także modele na sieciach o różnych topologiach oraz ciągłe, nieoparte na sieciach.

Jednak, pomimo wielu lat badań i rozwoju teorii perkolacji, bardzo mało mamy dokładnych, analitycznych wyników. W przypadku progu perkolacji takie wartości są znane tylko dla małej ilości najprostszych modeli. Należą do nich: perkolacja w jednym wymiarze, w przypadku której w sposób oczywisty widać, że $p_c = 1$ oraz modele perkolacji na sieciach trójkątnych, gdzie $p_c = 1/2$ w przypadku sieci węzłów i $p_c = 2\sin(\pi/18)$ dla sieci wiązań [5]. W przypadku wartości wykładników krytycznych, które przedstawiłem w **Tabeli 2.1**, jedynie wyniki dla liczby wymiarów $D = 1$ oraz $D = 2$ są znane dokładnie.

Zdecydowana większość znanych wartości progów perkolacji czy wykładników krytycznych została wyznaczona za pomocą symulacji Monte–Carlo i jest znana z pewną ograniczoną dokładnością.

Z powyższego wynikają dwa ważne wnioski:

- Wciąż jest miejsce na rozwój nowych metod numerycznych i algorytmów.

  Pozwalają one, między innymi, poprawiać dokładność dotychczas znanych wielkości oraz badać systemy coraz bardziej złożone, coraz większe lub modelować je w większej liczbie wymiarów.

- Każdy nowy rezultat analityczny ma dużą wartość.

  W szczególności może on potwierdzić lub nie, wnioski z symulacji Monte–Carlo. Przykładem takiego wyniku jest stwierdzenie o możliwości powstania klastra perkolującego w modelu długich igieł na dwuwymiarowej sieci. Wyniki z symulacji coraz większych systemów sugerowały, że po przekroczeniu pewnej długości igły, zawsze dojdzie do zablokowania (nie będzie już miejsca na dodanie do systemu kolejnej igły) zanim powstanie klaster perkolujący. Dopiero dowód analityczny, który zaprezentowałem wspólnie z dr hab. Grzegorzem Kondratem i prof. Zbigniewem Kozą w artykule [11] pokazał, że w takim modelu, niezależnie od rozmiaru igły, zawsze może powstać klaster perkolujący.

W symulacjach komputerowych często stosowane są modele losowej, sekwencyjnej, adsorpcji obiektów (ang. Random Sequential Adsorption), w których zamiast określania prawdopodobieństwa zapełnienia węzła, zapełnia się układ dodając do niego kolejne obiekty w losowych miejscach. Modele takie wykazują analogiczne przejście fazowe i charakteryzują się (w granicy dużych układów) takimi samymi wartościami

progów i wykładników krytycznych jak modele z prawdopodobieństwem zapełnienia. Jednocześnie, co opisałem w rozdziale 3.3, pozwalają one na przeprowadzanie wydajniejszych symulacji. Taki właśnie model zastosowałem w przypadku badania perkolacji hipersfer.

Rozdział 3

Opis wybranych metod numerycznych

Poniżej zamieściłem opis kilku ważniejszych metod numerycznych, które są stosowane w badaniu procesów perkolacji. Są to metody, których użyłem w analizie perkolacji hipersfer, której dokładny opis zawarłem w rozdziale 4.

3.1 Poprawki do skalowania skończonego rozmiaru

W granicy termodynamicznej, prawdopodobieństwo istnienia klastra perkolującego wynosi 0 dla prawdopodobieństwa zapełnienia pojedynczego elementu sieci p poniżej progu perkolacji p_c i 1 powyżej p_c. W przypadku układów skończonych, przebieg funkcji zależności prawdopodobieństwa perkolacji od wartości p jest różny dla różnych wielkości układu, co pokazałem, dla różnych sieci kwadratowych, na **Rysunku 2.2**. Z widocznych na nim wykresów widać, że jeśli wybierzemy pewną wartość prawdopodobieństwa perkolacji, wynoszącą τ z zakresu $0 < \tau < 1$, to wartości prawdopodobieństwa p odpowiadające τ, zbliżają się do wartości krytycznej p_c wraz ze wzrostem wielkości układu L. Jeżeli prawdopodobieństwo zapełnienia pojedynczego elementu sieci dla sieci o boku L, odpowiadające wartości τ oznaczymy jako $p_L(\tau)$, to możemy to zapisać jako:

$$p_L(\tau) - p_c \to 0, L \to \infty \tag{3.1}$$

Znajomość zależności $p_L(\tau)$ od L pozwoliłaby na wyznaczenie progu perkolacji p_c, na podstawie wyników symulacji przeprowadzonych dla różnych rozmiarów układu L.

Autorzy artykułu [12] wyprowadzili zależność w postaci:

$$p_L(\tau) - p_c \sim L^{-(2+1/\nu)} \tag{3.2}$$

dla przypadku klastra opasującego dwuwymiarową sieć kwadratową i szczególnej wartości τ wynoszącej $\tau^* = 0,521058290$. Korzystając z takiej zależności byli oni w stanie wyznaczyć bardzo dokładną wartość progu perkolacji dla badanego systemu, wynoszącą $p_c = 0,59274621(13)$. Jest to bardzo dokładny wynik, jednak zależność 3.2 ma zastosowanie jedynie dla opisanego systemu i znanej w jego przypadku dokładnej wartości τ^*.

Bardziej ogólna postać zależności $p_L(\tau)$ od L, obowiązująca dla dowolnej wartości $0 < \tau < 1$, została zaproponowana przez autorów artykułu [13], i jest ona opisana równaniem:

$$p_L(\tau) = p_c + L^{-1/\nu}[A_0(\tau) + A_1(\tau)L^{-1} + A_2(\tau)L^{-2} + \ldots + A_M(\tau)L^{-M}] \tag{3.3}$$

gdzie M jest parametrem odcięcia. Zależność ta została zweryfikowana dla opisanego powyżej przypadku dwuwymiarowej sieci kwadratowej z klastrem opasującym sieć. Otrzymana przy jej zastosowaniu wartość progu perkolacji $p_c = 0,59274675(88)$ (dla $\tau = 0,9$), jest zgodna z wartością wyznaczoną w [12].

3.2 Wielki układ kanoniczny

Z symulacji komputerowych, wykonywanych na sieciach o skończonych rozmiarach, składających się z M komórek, dostajemy wartości różnych interesujących nas wielkości, dla konkretnych poziomów zapełnienia sieci, opisywanych przez liczbę zapełnionych komórek sieci N. Wielkościami tymi mogą być np. długość korelacji ξ_N, czy prawdopodobieństwo wystąpienia perkolacji P_N.

Jeżeli oznaczymy interesującą nas wielkość jako Q, to w wyniku przeprowadzonych symulacji komputerowych, dostaniemy zbiór dyskretnych wartości $\{Q_N\}$, czyli *układ mikrokanoniczny.*

Często jednak chcemy znać zależność interesującej nas wielkości od prawdopodobieństwa p, w postaci ciągłej funkcji $Q(p)$. Taki opis systemu nazywamy *wielkim układem kanonicznym.*

Opisany poniżej sposób na to jak można, mając układ mikrokanoniczny danej wielkości, wyznaczyć jej opis w wielkim układzie kanonicznym został zaprezentowany w artykule [14]. Warto zwrócić tutaj uwagę na to, że jego autorzy opisany powyżej wielki układ kanoniczny nazywają po prostu *układem kanonicznym*. Stosowana przeze mnie nazwa — wielki układ kanoniczny — pokazuje podobieństwo rozpatrywanego modelu do układu cząstek rozpatrywanego w fizyce statystycznej. Możemy założyć, że każda zapełniona komórka sieci (lub wiązanie w przypadku perkolacji na sieci wiązań) w modelu perkolacji opisuje cząstkę, a N całkowitą energię układu, gdzie stosuje się właśnie takie nazewnictwo.

W przypadku perkolacji węzłów, prawdopodobieństwo tego, że na sieci o wielkości M jest dokładnie N wypełnionych komórek, opisane jest przez rozkład dwumianowy:

$$B(M, N, p) = \binom{M}{N} p^N (1-p)^{M-N} \tag{3.4}$$

Znając układ mikrokanoniczny danej wielkości Q, będący zbiorem jej wartości dla różnych wartości N: $\{Q_N\}$, możemy wyznaczyć jej wartość w wielkim układzie kanonicznym jako:

$$Q(p) = \sum_{N=0}^{M} B(M, N, p) Q_N = \sum_{N=0}^{M} \binom{M}{N} p^N (1-p)^{M-N} Q_N. \tag{3.5}$$

3.3 Wydajny algorytm symulacji

Standardowo, tak jak to opisałem w rozdziałach 2.2 i 2.3, trzeba przygotować osobne układy z różnymi wartościami prawdopodobieństwa zapełnienia elementu sieci p. Z każdego z nich dostaniemy wartość interesujących nas wielkości Q dla pewnej liczby wypełnionych komórek N. Taka procedura wymaga bardzo dużo czasu i zasobów obliczeniowych.

We wspomnianym powyżej artykule [14] został opisany wydajny algorytm, który pozwala na przyspieszenie symulacji wykorzystywanych podczas badania procesów perkolacji. Zaproponowane w nim podejście, które możemy zastosować żeby otrzymać układy z N z zakresu od 0 do pewnej maksymalnej wartości, polega na zapełnianiu kolejnych, losowo wybranych węzłów układu zaczynając do pustej sieci. Po każdym

kroku takiej symulacji dostaniemy interesującą nas wartość dla danego N, co wcześniej wymagało stworzenia i przeanalizowania całego układu.

Ponadto, ze względu na niewielkie różnice pomiędzy stanem badanego systemu pomiędzy kolejnymi krokami algorytmu, możemy w wydajny sposób znajdować powstające klastry.

Po zapełnieniu nowego węzła sieci sprawdzamy węzły z nim sąsiadujące. W zależności od wyniku tego sprawdzenia, mamy jedną z trzech możliwości:

- Jeśli żaden z nich nie jest zapełniony, to nowo zapełniony węzeł tworzy zaczątek nowego klastra.
- W przypadku, gdy wszystkie z zapełnionych sąsiednich węzłów należą do tego samego klastra, to dołączamy do niego także nowo wypełniony węzeł.
- Gdy zapełnione sąsiednie węzły należą do wielu klastrów, to wybieramy jeden z nich, przyłączamy do niego nowo wypełniony węzeł oraz całe klastry, do których należą pozostali sąsiedzi.

Jest to algorytm typu *Union/Find*, czyli zapewniający wydajny sposób sprawdzenia, czy dwa elementy należą do tego samego zbioru oraz operację łączenia dwóch zbiorów w jeden.

W przypadku gdy nowo wypełniony węzeł zostanie przyłączony do wcześniej istniejącego klastra, musimy też sprawdzić czy nie powstał klaster perkolujący. Sposób w jaki należy to zrobić zależy, między innymi, od sposobu w jaki zdefiniujemy perkolujący klaster i od warunków brzegowych.

Opisana powyżej metoda symulacji perkolacji ma złożoność obliczeniową $O(N)$, podczas gdy w przypadku tradycyjnego podejścia z powtarzaniem analizy systemów dla poszczególnych wartości p mamy do czynienia ze złożonością rzędu $O(N^2)$. Jak widać, zastosowanie opisanego algorytmu pozwala na znaczne przyspieszenie obliczeń.

Rezultaty zastosowania takiego algorytmu pokazałem na **Rysunkach 2.1** oraz **2.3**, które pokazują przykłady różnych wartości prawdopodobieństwa wypełnienia elementów sieci dla perkolacji węzłów oraz wiązań. Wygenerowałem je dodając kolejne elementy do początkowo pustej sieci.

Do przeprowadzenia symulacji nakładających się hipersfer użyłem opisanego powyżej algorytmu, wzbogacając go o szereg optymalizacji, które opisałem dokładnie w rozdziale 4.2.

3.4 Wyłączona objętość

Zgodnie z teorią wyłączonej objętości (ang. excluded volume theory) [15] [16], próg perkolacji dla układu nakładających się obiektów może być, w przybliżeniu, określony przy użyciu średniej liczby połączeń z sąsiednimi obiektami B, oraz wyłączonej objętości V_{ex}:

$$\eta \approx BV/V_{ex}. \tag{3.6}$$

W przypadku dyskretnym wyłączona objętość dla obiektu jest liczbą elementów zbioru możliwych położeń innych obiektów, takich, że byłyby one połączone z wybranym. Wartość B powinna być, w przybliżeniu, niezmienna dla pewnych systemów w momencie perkolacji [17]. W pracach [10] [18] zostało pośrednio dowiedzione, że w przypadku perkolacji hipersześcianów na sieci wartość ta nie zależy od ich wielkości. Ponadto autorzy ustalili prosty związek pomiędzy progiem perkolacji a wyłączoną objętością, dla obiektów o różnej wielkości:

$$\frac{V_{ex,k}}{V_k} \ln(1 - \varphi_c^k) \approx const, \tag{3.7}$$

gdzie φ_c^k jest krytycznym stopniem zapełnienia hipersześcianów o krawędzi k w granicy dużych systemów $L \to \infty$, V_k jest objętością hipersześcianu o krawędzi k, a $V_{ex,k}$ wyłączoną objętością takiego hipersześcianu. Dla hipersześcianów o krawędziach długości k mamy:

$$V_k = k^D, \tag{3.8}$$

$$V_{ex,k} = (2k-1)^{D-1}(2k + 2D - 1). \tag{3.9}$$

Autorzy prac [10] [18] zaobserwowali, że względny błąd przybliżenia 3.7 dąży do zera wraz z $k \to \infty$ jak $k^{-\theta}$ z $\theta = 3/2$ dla wszystkich badanych wymiarów $D = 2, ..., 7$. Zastosowanie tej właściwości pozwoliło im uzyskać bardzo dokładne wartości progów perkolacji dla ciągłego przypadku zorientowanych (równoległych do siebie) hipersześcianów.

3.5 Rozszerzone sąsiedztwo

W tym rozdziale zaprezentowałem metodę rozszerzonego sąsiedztwa, która została opisana w artykule [19]. Tutaj ograniczyłem się do przedstawienia jej na przykładzie sieci dwuwymiarowej – rozszerzenie go na inną liczbę jest oczywiste.

Zwykle, rozpatrując perkolację na dwuwymiarowej sieci wiązań, dwie komórki sieci uważamy za bezpośrednio ze sobą połączone jeśli mają one wspólną krawędź. Można jednak zastanowić się jak będzie wyglądał proces perkolacji w przypadku, gdy definicję połączenia komórek rozszerzymy na dalszych sąsiadów. W tym celu możemy rozpatrywać różne poziomy najbliższego sąsiedztwa, zależne od odległości między komórkami sieci. Komórki oddalone od siebie o 1 jednostkę sieci (j.s.) w linii prostej, czyli mające wspólne boki są najbliższymi sąsiadami stopnia 1. Te oddalone o 2 j.s. najbliższymi sąsiadami stopnia 2, itd. Rozłożenie sąsiadów kolejnych stopni na dwuwymiarowej, kwadratowej sieci zostało pokazane na **Rysunku 3.1**.

			10	9	10			
		8	7	6	7	8		
	8	5	4	3	4	5	8	
10	7	4	2	1	2	4	7	10
9	6	3	1	**0**	1	3	6	9
10	7	4	2	1	2	4	7	10
	8	5	4	3	4	5	8	
		8	7	6	7	8		
			10	9	10			

Rysunek 3.1: Położenie najbliższych sąsiadów komórki **0** od stopnia 1 do 10.

Jeśli założymy, że każda komórka ma połączenie ze swoimi najbliższymi sąsiadami stopni od 1 do n, to liczbę komórek, z którymi jest ona połączona nazywamy *liczbą koordynacyjną* (ang. coordination number) i oznaczamy jako z. Liczbę najbliższych sąsiadów kolejnych stopni oraz wartości całkowitej liczby koordynacyjnej zamieściłem w **Tabeli 3.1**.

Za autorami publikacji [19], w której została zaprezentowana opisywana tutaj metoda oraz wyniki, oznaczyłem system oparty na kwadratowej sieci z sąsiedztwem rozszerzonym na najbliższych sąsiadów stopni 1 i 2 na jako *SQ-1,2*, stopni 1, 2, 3 jako *SQ-1,2,3*, stopni 1, 2, . . . , n jako *SQ-1,2,. . . ,n*.

W celu wyznaczenia progów perkolacji zaczynamy od przeprowadzenia serii symulacji Monte–Carlo w interesującym nas modelu z rozszerzonym sąsiedztwem. Symula-

Stopień sąsiedztwa	Liczba sąsiadów	Całkowita liczba koordynacyjna
1	4	4
2	4	8
3	4	12
4	8	20
5	4	24
6	4	28
7	8	36
8	8	44
9	4	48
10	8	56

Tabela 3.1: Liczba sąsiadów i całkowita liczba koordynacyjna z dla najbliższych sąsiadów stopnia od 1 do 10.

cje te dostarczą nam następujących wartości:

- $P_{\geq s}$ – prawdopodobieństwo powstania klastra zawierającego s lub więcej elementów sieci,
- $n_s(p)$ – rozkład wielkości klastrów, czyli znormalizowana liczba klastrów o rozmiarze s.

W granicy, dla dużych wartości s i małych $(p - p_c)$, gdzie $(p - p_c)s^\sigma$ jest stałe, zachowanie $n_s(p)$ opisane jest przez:

$$n_s(p) \approx A_0 s^{-\tau} f(B_0(p - p_c)s^\sigma), \tag{3.10}$$

gdzie τ i σ są wykładnikami uniwersalnymi, $f(x)$ uniwersalną funkcją o właściwości $f(0) = 1$, a wartości A_0 i B_0 są współczynnikami charakterystycznymi dla konkretnej sieci. Rozwinięcie równania 3.10 w szereg wokół $p = p_c$ daje nam:

$$n_s(p) \approx n_s(p_c) + (p - p_c)\frac{dn_s(p)}{dp}\bigg|_{p_c} + \dots \tag{3.11}$$

Rozkład wielkości klastrów w punkcie krytycznym przyjmuje wartość:

$$n_s(p_c) \approx A_0 s^{-\tau} \tag{3.12}$$

Dla skończonych wartości s w puncie perkolacji p_c, musimy rozpatrzyć poprawki do $n_s(p_c)$:

$$n_s(p_c) \approx A_0 s^{-\tau}(1 + C_0 s^{-\Omega} + \ldots) \tag{3.13}$$

Uwzględniając 3.13, równanie 3.11 możemy zapisać jako:

$$n_s(p) \approx A_0 s^{-\tau}(1 + C_0 s^{-\Omega} + \ldots) + (p - p_c) A_0 s^{-\tau} f'(0) B_0 s^{\sigma} + \ldots \tag{3.14}$$

Prawdopodobieństwo tego, że dana komórka sieci należy do klastra o rozmiarze równym lub większym od s opisuje równanie:

$$P_{\geq s} = \sum_{s'=s}^{\infty} s' n_{s'} \tag{3.15}$$

Po uwzględnieniu 3.14 dostaniemy:

$$P_{\geq s} \approx \sum_{s'=s}^{\infty} [A_0 (s')^{1-\tau} + A_0 C_0 (s')^{1-\Omega-\tau} + (p - p_c) A_0 f'(0) B_0 (s')^{1+\sigma-\tau}] \tag{3.16}$$

Jeśli przybliżymy sumę całką:

$$\sum_{s'=s}^{\infty} (s')^{-\alpha} \approx \int_s^{\infty} x^{-\alpha} dx = \frac{x^{1-\alpha}}{1-\alpha}\bigg|_s^{\infty} = \frac{s^{1-\alpha}}{\alpha - 1}, \tag{3.17}$$

to równanie 3.16 możemy zapisać jako:

$$P_{\geq s} \approx \frac{s^{2-\tau}}{\tau - 2} + A_0 C_0 \frac{s^{2-\Omega-\tau}}{\Omega - \tau - 2} + A_0 B_0 f'(0)(p - p_c) \frac{s^{2+\sigma-\tau}}{\tau - \sigma - 2} \tag{3.18}$$

Mnożąc równanie 3.18 obustronnie przez $s^{\tau-2}$ otrzymamy:

$$s^{\tau-2} P_{\geq s} \approx A_1 [1 + C_1 s^{-\Omega} + B_1 (p - p_c) s^{\sigma}], \tag{3.19}$$

gdzie τ, σ i Ω są uniwersalnymi wykładnikami, a A_1, B_1 i C_1 nieuniwersalnymi stałymi.

Z równania tego wynika, że dla dużych s w punkcie perkolacji p_c, $s^{\tau-2} P_{\geq s}$ będzie dążyć do wartości stałej. A wraz z oddalaniem się p od p_c będzie się od niej oddalać w sposób liniowy.

Liniowa część krzywej $s^{\tau-2} P_{\geq s}$ staje się pozioma wraz ze zbliżaniem się p do p_c. Rozpatrując równanie 3.19 jako zależność $s^{\tau-2} P_{\geq s}$ od s^{σ}, możemy wykorzystać tę właściwość do wyznaczenia wartości progu perkolacji p_c:

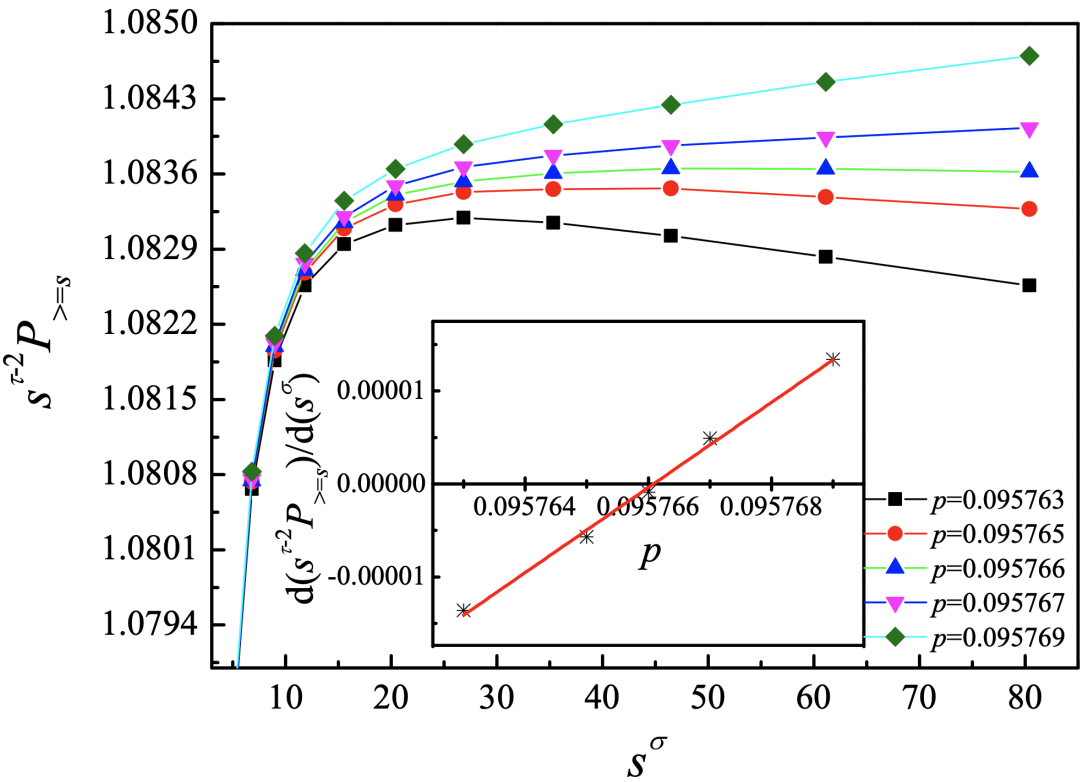
$$\frac{d(s^{\tau-2} P_{\geq s})}{d(s^{\sigma})} \approx B_1 (p - p_c) \tag{3.20}$$

W przypadku dwuwymiarowym potrzebne wartości wykładników krytycznych, które można znaleźć w **Tabeli 2.1**, są znane dokładnie i wynoszą $\tau = 187/91$ oraz

$\sigma = 36/91$. Możemy więc je wykorzystać do wyznaczenia zależności $s^{\tau-2}P_{\geq s}$ od s^{σ}. Wykresy takich zależności, dla sieci dwuwymiarowej i rozszerzonego sąsiedztwa stopni od 1 do 8 przedstawione zostały na **Rysunku 3.2**. Zamieszczony na nim wewnętrzny wykres przedstawia nachylenie liniowych części krzywych z głównej części dla różnych wartości prawdopodobieństwa p. Można z niego otrzymać szukaną wartość progu perkolacji. W zaprezentowanym przypadku wynosi ona:

$$p_c = 0,0957661(9). \tag{3.21}$$

Widoczny na **Rysunku 3.2** szybki wzrost wartości $s^{\tau-2}P_{\geq s}$ dla małych klastrów wynika z czynnika $s^{-\Omega}$ występującego w równaniu 3.19.



Rysunek 3.2: Wykresy zależności $s^{\tau-2}P_{\geq s}$ od s^{σ} dla *SQ-1,2,...,8* dla różnych wartości p. Wewnętrzny wykres przedstawia nachylenie liniowych części krzywych z głównej części. Rysunek pochodzi z artykułu [19].

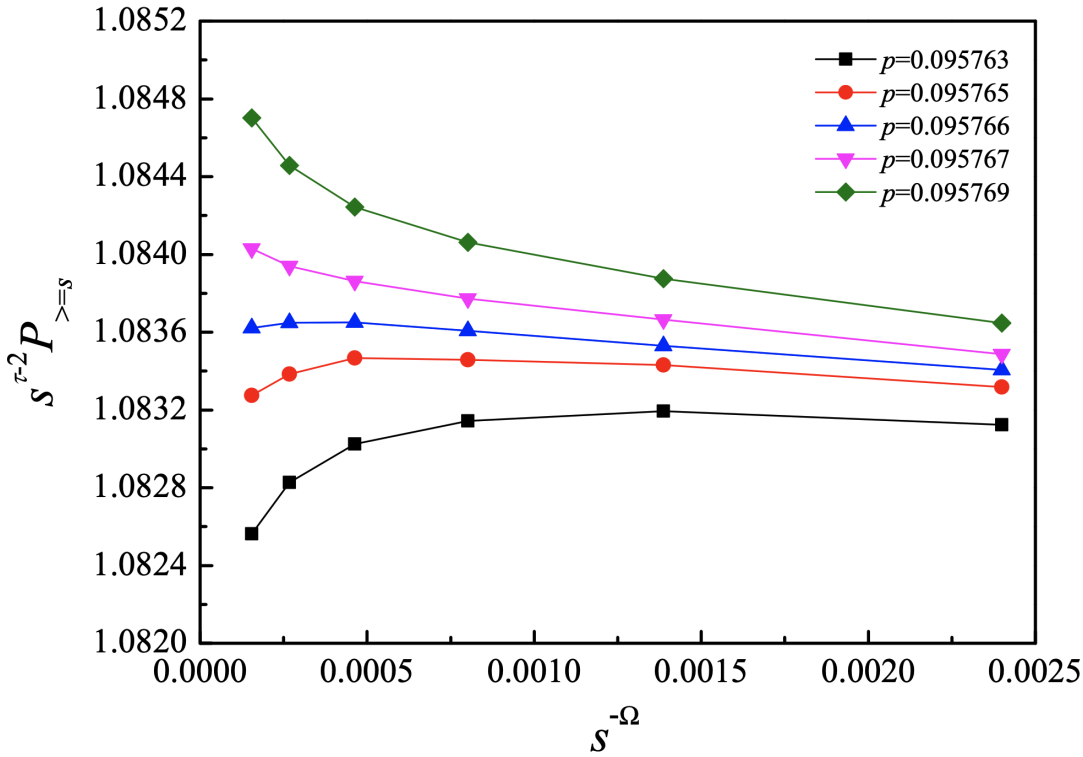
Wartość progu perkolacji p_c można także otrzymać rozpatrując zależność $s^{\tau-2}P_{\geq s}$ od $s^{-\Omega}$ dla dużych wartości s. Z równania 3.19 wynika, że dla $p = p_c$ zależność ta

jest liniowa, a dla $p \neq p_c$ nieliniowa. Podobnie jak w poprzednim przypadku, dokładne wartości wykładników krytycznych $\tau = 187/91$ i $\Omega = 72/91$ są znane i można ich użyć do wyznaczenia opisanej zależności.

Wykresy, które przedstawiają te zależności dla różnych wartości p dla sieci dwuwymiarowej i rozszerzonego sąsiedztwa stopni od 1 do 8 zostały pokazane na **Rysunku 3.3**. Można z nich wyznaczyć zakres, w którym powinien się znajdować próg perkolacji:

$$0,095766 < p_c < 0,095767 \tag{3.22}$$

Otrzymany wynik jest zgodny z wartością krytyczną uzyskaną poprzednio, wynoszącą $p_c = 0,0957661(9)$.



Rysunek 3.3: Wykresy zależności $s^{\tau-2}P_{\geq s}$ od $s^{-\Omega}$ dla *SQ-1,2,...,8* dla różnych wartości p. Rysunek pochodzi z artykułu [19].

Modele perkolacji z rozszerzonym sąsiedztwem są równoważne odpowiednim modelom perkolacji nakładających się obiektów składających się z wielu komórek sieci. Przykładami takich modeli są modele perkolacji nakładających się obiektów, takich

jak hipersfery, które opisałem szczegółowo w rozdziale 4, albo także kwadraty czy hipersześciany opisane w [18]. Autorzy tej pracy wyznaczyli wartości progów perkolacji dla badanych modeli w postaci krytycznego stopnia zapełnienia φ_c. W przypadku modeli kwadratów i sześcianów o bokach długości k jednostek sieci, prawdopodobieństwo zapełnienia komórki sieci spełnia równanie

$$\varphi_c^k = 1 - (1 - p_c)^{k^D}. \tag{3.23}$$

Korzystając z tego wzoru możemy z progów perkolacji φ_c^k policzyć odpowiadające im prawdopodobieństwa krytyczne p_c jako

$$p_c = 1 - (1 - \varphi_c^k)^{1/k^D} \tag{3.24}$$

i porównać wyniki otrzymane za pomocą obydwu metod. Przykłady kilku takich porównań przedstawiłem w **Tabeli 3.2**.

Nakładające się obiekty			Rozszerzone sąsiedztwo	
Obiekt	φ_c	p_c	Sąsiedztwo	p_c
2 x 2	0,58365(2)	0,196724(10)	*SQ-1,2,3,4*	0,1967293(7)
3 x 3	0,59586(2)	0,095765(10)	*SQ-1,2,...,8*	0,0957661(9)
2 x 2 x 2	0,23987(2)	0,033702(10)	*SC-1,2,...,6*	0,0337049(9)

Tabela 3.2: Porównanie progów perkolacji p_c wyliczonych na podstawie φ_c [18] dla modeli nakładających się kwadratów i sześcianów z progami perkolacji wyznaczonymi z modeli rozszerzonego sąsiedztwa [19].

Rozdział 4

Perkolacja hipersfer

W niniejszym rozdziale zaprezentowałem szczegóły oraz wyniki przeprowadzonych przeze mnie badań perkolacji nakładających się hipersfer w wymiarach od 3 do 5.

Bezpośrednią inspiracją do przeprowadzenia takich badań były prace [10] oraz [18], w których autorzy wyznaczyli wartości progów perkolacji oraz wykładniki krytyczne dla przypadku ciągłego perkolacji hipersześcianów, obliczając granicę wartości otrzymanych z coraz większych systemów dyskretnych. Inne podejście do badania perkolacji nakładających się obiektów, z zastosowaniem rozszerzonego sąsiedztwa, opisałem w rozdziale 3.5. Zawiera on także porównanie wyników otrzymanych dla perkolacji sześcianów, z użyciem obu metod.

Kolejną motywacją dla przeprowadzenia niniejszych badań była, opisana w pracy [10] obserwacja, iż wartość wykładnika θ, opisującego zbieżność progu perkolacji modeli dyskretnych hipersześcianów, jest uniwersalna — niezależna od liczby wymiarów badanego systemu. W ramach przeprowadzonej przeze mnie analizy perkolacji hipersfer także podjąłem próbę potwierdzenia uniwersalności wartości wykładnika θ w badanym przeze mnie przypadku.

4.1 Model

W badanym modelu wszystkie symulacje przeprowadzałem na D-wymiarowej sieci mającej kształt hipersześcianu o boku długości L jednostek sieci ($D = 3, 4, 5$). Początkowo pustą sieć stopniowo wypełniałem zdyskretyzowanymi hipersferami o średnicy

równej k stałych sieci.

Warto tutaj zaznaczyć, że istnieje wiele metod na dyskretyzację hipersfery. Zastosowany przeze mnie sposób, opisałem dokładnie w rozdziale 4.2.1.

Hipersfery umieszczane były w losowych pozycjach, tak że mogły one na siebie nachodzić. Jednak żadne dwie nie mogły zajmować dokładnie tego samego miejsca. Aby zredukować wpływ skończonej wielkości badanego systemu na wyniki, założyłem cykliczne warunki brzegowe — oznacza to, że rozpatrywana sieć jest powierzchnią $(D+1)$–wymiarowego hipertorusa.

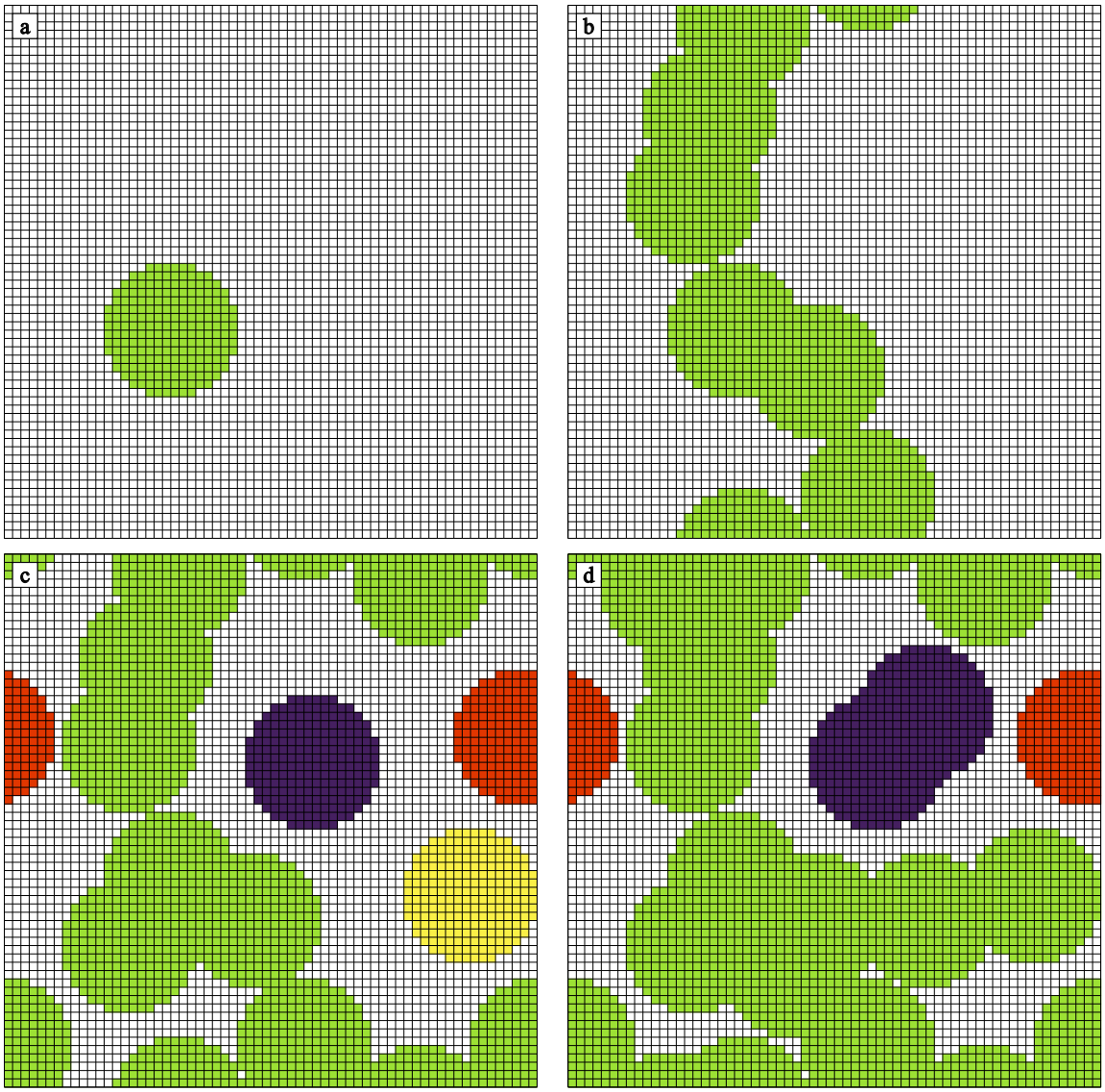
Hipersfery, które są dodawane do układu, w pewnym momencie zaczną się łączyć tworząc klastry, przy czym dwie hipersfery uważamy za połączone, gdy dowolny element sieci należący do jednej z nich jest bezpośrednio połączony z dowolnym elementem sieci należącym do drugiej. Przy pewnym poziomie zapełnienia sieci powstanie klaster perkolujący opasujący sieć. Przykład tworzenia się klastra perkolującego w opisanym modelu dla przypadku $D = 2$ został pokazany na **Rysunku 4.1**.

W przypadku $k = 1$ opisany model sprowadza się do zwykłej perkolacji węzłów, takiej jak opisana w rozdziale 2.2, ale w większej liczbie wymiarów. W granicy dużych średnic hipersfer ($k \to \infty$) model opisuje perkolację ciągłych hipersfer w $\mathbf{R}^D$.

Dla systemu o skończonej wielkości, zawierającego N obiektów rozmieszczonych w losowych miejscach, wartość stopnia zapełnienia φ można policzyć przy użyciu poniższego wzoru:

$$\varphi = 1 - \left(1 - \frac{N}{L^D}\right)^V . \tag{4.1}$$

Aby uzasadnić jego poprawność, wybierzmy dowolny węzeł sieci i rozważmy zbiór wszystkich możliwych położeń obiektów, które zawierają w sobie ten węzeł (takich położeń jest dokładnie V). Żeby wybrany węzeł sieci pozostał pusty, położenie żadnego z N obiektów znajdujących się na sieci, nie może należeć do tego zbioru (tutaj istotne jest to, że żadne dwa obiekty nie mogą mieć takich samych położeń). Prawdopodobieństwo tego, że pojedyncze pole sieci nie będzie należało do zbioru położeń N obiektów znajdujących się na sieci, wynosi $(1 - N/L^D)$. Iloczyn V takich prawdopodobieństw daje dopełnienie stopnia zapełnienia $1 - \varphi$.

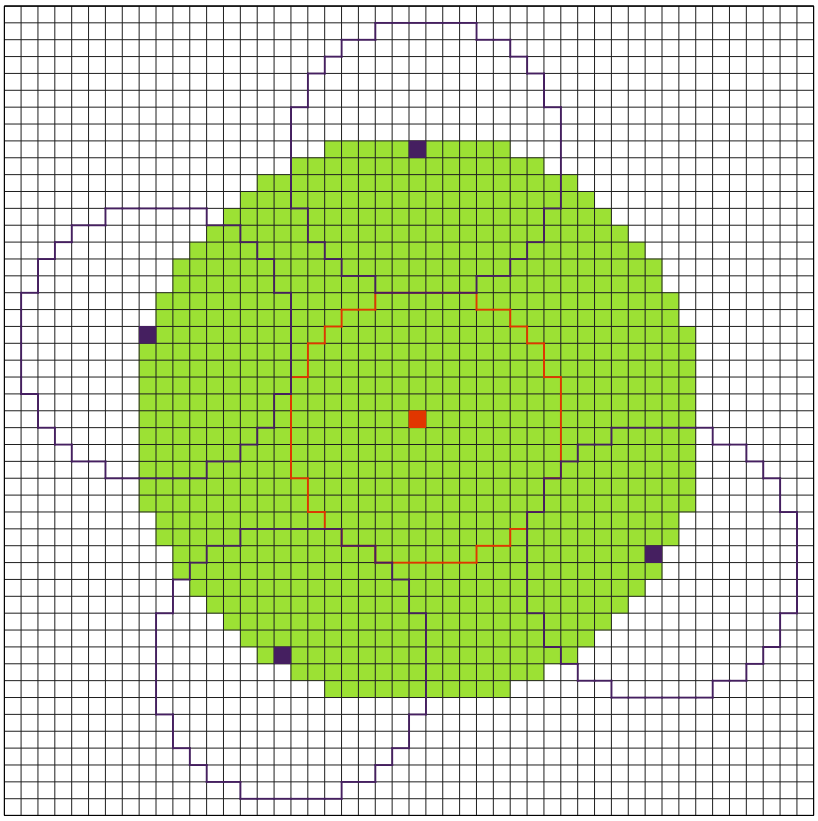


Rysunek 4.1: Przykładowe kroki formowania się klastra perkolującego.
Przypadek $D = 2, L = 64, k = 16$. Poszczególne klastry zostały oznaczone różnymi kolorami. a) 1 sfera, 1 klaster, brak perkolacji, b) 6 sfer, 1 klaster, perkolacja wzdłuż osi y, c) 12 sfer, 4 klastry, perkolacja wzdłuż osi y, d) 16 sfer, 3 klastry, perkolacja wzdłuż obu osi.

4.1.1 Wyłączona objętość hipersfery

Obszar wyłączonej objętości dla hipersfery o promieniu k ma kształt zbliżony do hipersfery o promieniu $2k + 1$, co jest pokazane na **Rysunku 4.2**.

W niektórych przypadkach (na przykład dla $k = 2, 5, 10$ w dwóch wymiarach) obszar ten pokrywa się z hipersferą o odpowiedniej średnicy, w innych (na przykład



Rysunek 4.2: Kształt obszaru wyłączonej objętości (wypełnione węzły sieci). Dla zdyskretyzowanej sfery o średnicy $k = 16$ (czerwony kontur), wyłączona objętość jest obszarem o średnicy $2k + 1 = 33$ (wypełnione pola) wokół położenia sfery. Na rysunku są pokazane cztery sfery położone na krawędzi obszaru wyłączonej objętości.

dla $k = 1, 3, 4, 6, 7, 8, 9$ w dwóch wymiarach) nie. Na rysunku zaznaczone pola sieci oznaczają położenia hipersfer.

Ponieważ, aby stwierdzić czy dwie hipersfery są ze sobą bezpośrednio połączone, wystarczy znać ich względne położenia, położenia hipersfer nie są z żaden sposób związane z ich środkami. Na **Rysunku 4.2** położenia sfer są zaznaczone w pobliżu ich środków jedynie w celu zwiększenia czytelności.

W związku z tym, że nie są znane wzory pozwalające na obliczenie objętości zdyskretyzowanej hipersfery o zadanej średnicy, ani wyłączonej objętości dla takiej hipersfery, odpowiednie wartości musiałem wyznaczyć numerycznie.

4.2 Algorytm i optymalizacje

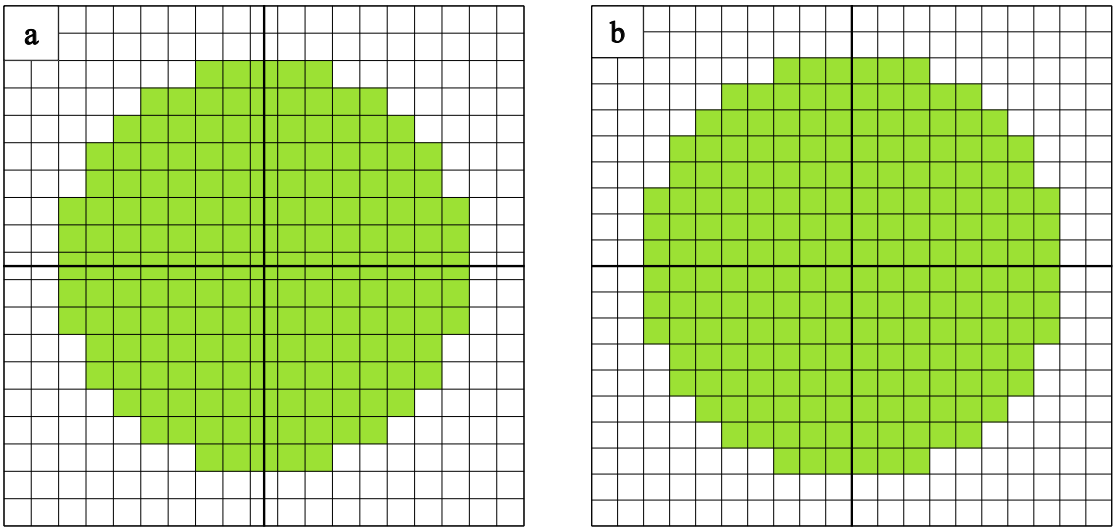
Dzięki zastosowaniu odpowiedniego algorytmu, bazującego na opisanym w rozdziale 3.3, oraz wielu optymalizacji byłem w stanie przeprowadzić symulacje dla dużych systemów i hipersfer o różnych rozmiarach. Maksymalna średnica hipersfer wynosiła $k = 1000$ dla $D = 3$, $k = 900$ dla $D = 4$ i $k = 200$ dla $D = 5$. Wielkości sieci to odpowiednio $L \leq 148k$ $(D = 3)$, $L \leq 60k$ $(D = 4)$ oraz $L \leq 32k$ $(D = 5)$. W bieżącym rozdziale opisałem, między innymi, zastosowany przeze mnie algorytm oraz optymalizacje, które opracowałem w celu umożliwienia przeprowadzenia symulacji tak dużych układów.

4.2.1 Odwzorowanie zawartości hipersfery

Umieszczanie na sieci obiektów, których kształt i orientacja pokrywają się ze strukturą sieci, na przykład hiperprostopadłościanów o całkowitych długościach boków (wyrażonych w stałych sieci), jest proste. Inaczej wygląda sytuacja z kształtami, których nie można złożyć z całych, hipersześciennych węzłów sieci, takimi jak na przykład hipersfery. W ich przypadku potrzebujemy zdyskretyzowanej reprezentacji obiektów, którą możemy umieścić na sieci.

W celu dyskretyzacji hipersfery używamy następującej procedury. Umieśćmy w środku hipersfery początek układu współrzędnych o osiach równoległych do boków sieci i jednostce równej stałej sieci. Ponieważ rozważamy tylko hipersfery o całkowitych średnicach k, mamy dwa możliwe przypadki. Dla nieparzystych wartości k środek układu współrzędnych będzie umieszczony w węźle sieci, a współrzędne węzłów sieci będą całkowite. W przypadku parzystych wartości k początek układu współrzędnych będzie znajdował się w narożniku pomiędzy węzłami sieci, a węzły te będą miały współrzędne połówkowe. Przykłady obydwu przypadków zostały zaprezentowane na **Rysunku 4.3**. W przeprowadzonych symulacjach, poza przypadkiem $k = 1$, wszystkie rozpatrywane przeze mnie hipersfery miały średnice parzyste. Następnie musimy określić, które węzły sieci będą należały do zdyskretyzowanej reprezentacji hipersfery. W przypadku hipersfery o danej średnicy k, będą to te węzły sieci, których środki są oddalone od środka układu współrzędnych o nie więcej niż $k/2$.

Do zmniejszenia liczby kroków potrzebnych do stworzenia obrazu zdyskretyzowa-



Rysunek 4.3: Dyskretyzacja sfer z parzystymi i nieparzystymi średnicami. Przykład dwuwymiarowy: a) $k = 15$, b) $k = 16$. Węzły sieci są reprezentowane jako kwadraty, a osie symetrii sfery zaznaczone pogrubionymi liniami.

nej hipersfery wykorzystałem fakt, że jest ona obiektem symetrycznym. Po pierwsze, pozwoliło to ograniczyć rozpatrywane węzły sieci jedynie do tych, które mają dodatnie wartości współrzędnych we wszystkich kierunkach. Ponadto, mogłem przeprowadzić obliczenia dla tylko jednej permutacji dla każdego zbioru rozpatrywanych współrzędnych. Otrzymane dane o kształcie zdyskretyzowanej hipersfery zapisałem w postaci struktury danych, która pozwala odwzorować współrzędne węzła na informację o tym, czy dany węzeł należy do obrazu hipersfery. W celu zredukowania ilości pamięci potrzebnej do przechowania takiego odwzorowania także zastosowałem pewne optymalizacje. Po pierwsze przechowywane są jedynie informacje dla współrzędnych, które zostały wyliczone. Dodatkowo, zamiast przechowywać wartość logiczną dla każdego interesującego zestawu N współrzędnych, odwzorowanie zawiera $(N-1)$–wymiarowe klucze i odpowiadające im wartości graniczne ostatniej współrzędnej.

Stworzone w opisany powyżej sposób struktury danych, nazwane przeze mnie *odwzorowaniami zawartości hipersfery*, wykorzystałem następnie do wyznaczenia wyłączonych objętości zdyskretyzowanych hipersfer oraz do przygotowania *odwzorowań połączeń* między hipersferami.

4.2.2 Odwzorowanie połączeń

W tej części opisałem sposób, który wykorzystałem do sprawdzania czy dwie zdyskretyzowane hipersfery są ze sobą połączone. Ze względu na ich nieregularny kształt, w przeciwieństwie do idealnych sfer, nie wystarczy porównanie odległości między obiektami z ich średnicą k. Szukanie połączenia między obiektami o nieregularnych kształtach może być operacją bardzo kosztowną obliczeniowo i długotrwałą. Co istotne, jest to bardzo często powtarzany krok w przeprowadzanych symulacjach komputerowych. W związku z tym, jego optymalizacja była kluczowa dla możliwości wykonania wielu symulacji. Do wydajnego stwierdzania czy zdyskretyzowane hipersfery są ze sobą bezpośrednio połączone przygotowałem *odwzorowania połączeń*. Pozwalają one na odczytanie informacji o tym, czy dwa obiekty są połączone na podstawie D–elementowego wektora ich względnego położenia.

Proces przygotowania takiego odwzorowania rozpocząłem od umieszczenia jednej z hipersfer w środku układu współrzędnych. Następnie sprawdzałem czy jest ona połączona z inną hipersferą umieszczaną w obszarze odległym, w przybliżeniu, o średnicę k. W celu określenia czy istnieje bezpośrednie połączenie między dwoma obiektami użyłem *odwzorowania zawartości hipersfery* do odczytania czy którykolwiek węzeł sieci, sąsiadujący z węzłami należącymi do brzegu pierwszej hipersfery, należy do drugiej z nich.

Wykorzystując fakt, że układ dwóch hipersfer jest symetryczny, przy wyliczaniu *odwzorowań połączeń* oraz do przechowania ich reprezentacji w pamięci mogłem wykorzystać optymalizacje podobne do użytych w przypadku *odwzorowań zawartości hipersfer*.

4.2.3 Znajdowanie klastrów

Podczas wykonywania symulacji, za każdym razem po dodaniu do systemu nowej hipersfery, trzeba sprawdzić czy nie powstał klaster perkolujący. Aby zrobić to w sposób wydajny, użyłem wersji algorytmu typu *Union/Find* opisanego w rozdziale 3.3. Cały system został podzielony na hipersześcienne fragmenty o długości boku równej średnicy hipersfery k. Dla każdego takiego fragmentu przechowywana jest lista hipersfer, które zostały w nim umieszczone. Taki podział całej sieci na fragmenty umożliwił

ograniczenie liczby sprawdzeń połączeń między obiektami po dodaniu nowej hipersfery do systemu. Sprawdzane były jedynie połączenia nowej hipersfery z innymi, które znajdują się w tym samym fragmencie lub jego sąsiadach.

Połączone hipersfery tworzą klastry. Dla każdego z klastrów jedna ze składających się na niego hipersfer oznaczona jest jako początkowa. Ponadto, dla każdej hipersfery tworzącej klaster przechowywane jest jej położenie względem początkowej hipersfery klastra, do którego należy. Gdy nowo dodana hipersfera łączy się bezpośrednio z dwiema innymi, należącymi do tego samego klastra, różnica odległości od początkowej hipersfery klastra nowo dodanej hipersfery i jednej z tych, z którymi się połączyła jest używana do określenia czy powstał nowy klaster opasujący sieć. Jeżeli wartość tej różnicy, w którymkolwiek z kierunków, jest większa od średnicy hipersfery k, to oznacza, że powstał klaster perkolujący w tym kierunku.

4.2.4 Generator liczb losowych

Pseudolosowe wartości, potrzebne do przeprowadzenia symulacji, zostały wygenerowane przy użyciu generatora 32-bitowych liczb pseudolosowych *Mersenne Twister*, opartego na liczbie pierwszej Mersenne'a wynoszącej $2^{19937}-1$ [20]. Użyłem implementacji tego algorytmu dostępnej w bibliotece standardowej języka C++ — std::mt19937.

W celu sprawdzenia czy wyniki, które otrzymałem, nie zależą od zastosowanego generatora, powtórzyłem część symulacji z użyciem innego generatora, a mianowicie *RANLUX24* [21]. Wyniki otrzymane na ich podstawie, w zasadzie, nie różnią się od tych z generatora *Mersenne Twister*. Mówiąc dokładnie, różnice pomiędzy wartościami krytycznymi progów perkolacji $\varphi_c^k(L)$, φ_c^k oraz pomiędzy wartościami uniwersalnego wykładnika krytycznego ν, otrzymane na podstawie wyników symulacji przeprowadzonych z użyciem obu generatorów są mniejsze do sumy ich odchyleń standardowych. Same wartości odchyleń standardowych dla obu przypadków różniły się od siebie nie więcej niż dwukrotnie — $0,5 < \sigma_{MT}/\sigma_{RANLUX} < 2$.

4.3 Obliczenia

W rozpatrywanym modelu sieć ma cykliczne warunki brzegowe, a za perkolujące uważamy klastry opasujące sieć. W takim układzie możliwe są trzy różne definicje perkolacji. Możemy uznać, że perkolacja występuje gdy istnieje klaster perkolujący opasujący sieć w jednym, wybranym spośród D możliwych kierunków (przypadek A), w dowolnym kierunku (przypadek B) albo we wszystkich D kierunkach (przypadek C).

Dla każdej z wymienionych definicji perkolacji powinniśmy, w granicy termodynamicznej, otrzymać jednakowe krytyczne stopnie zapełnienia [14].

Ze wszystkich przeprowadzonych symulacji zebrałem wyniki dla każdej z trzech, przedstawionych powyżej, definicji perkolacji i, dla każdej z nich przedstawię wyniki obliczeń.

Dla każdego zestawu parametrów D, L oraz k przeprowadziłem n niezależnych symulacji, gdzie n wynosi od $4 \cdot 10^3$ dla największych systemów do $2 \cdot 10^6$ dla najmniejszych. Związek pomiędzy wartością stopnia zapełnienia φ, a otrzymaną bezpośrednio z symulacji liczbą dodanych do układu hipersfer N opisany jest wzorem 4.1.

Zgodnie z teorią skalowania skończonego rozmiaru [22] prawdopodobieństwo wystąpienia perkolacji powinno zależeć od L oraz odchylenia $\varphi - \varphi_c^k$ od krytycznego stopnia zapełnienia zgodnie z równaniem:

$$P_{L,k}(\varphi) = f_k\left((\varphi - \varphi_c^k)L^{1/\nu}\right) \tag{4.2}$$

dla dużych L, gdzie $f_k(\cdot)$ jest funkcją skalującą o kształcie sigmoidalnym, taką jak na przykład $(1 + \mathrm{tgh}(\cdot))/2$ lub $(1 + \mathrm{erf}(\cdot))/2$, a ν jest uniwersalnym wykładnikiem krytycznym długości korelacji.

Oznaczmy prawdopodobieństwo wystąpienia perkolacji dla układu zawierającego N obiektów jako P_N. Ponieważ N jest liczbą całkowitą, to wartości prawdopodobieństwa P_N mamy określone jedynie dla dyskretnych punktów. Możemy skorzystać z, opisanego w rozdziale 3.2, równania 3.5 opisującego przejście do wielkiego układu kanonicznego:

$$P(p) = \sum_{N=0}^{L^D} \binom{L^D}{N} p^N (1-p)^{L^D - N} P_N, \tag{4.3}$$

gdzie związek pomiędzy prawdopodobieństwem p znalezienia obiektu na danej pozycji $(0 \leq p \leq 1)$, a stopniem zapełnienia φ jest następujący (porównaj z 4.1):

$$\varphi = 1 - (1-p)^V. \tag{4.4}$$

Łącząc równania 4.3 i 4.4 dostajemy funkcję $P(\varphi)$ o ciągłej dziedzinie $0 \leq \varphi \leq 1$.

Dla danych k oraz L możemy numerycznie wyznaczyć krytyczny stopień zapełnienia jako rozwiązanie równania:

$$P(\varphi_c^k(L)) = \tau \tag{4.5}$$

z ustaloną wartością parametru $0 < \tau < 1$ (ja przyjąłem $\tau = 0,5$).

Mając zbiór wartości $\varphi_c^k(L)$ dla różnych L, możemy wyznaczyć wartość φ_c^k w granicy termodynamicznej oraz wykładnik krytyczny ν korzystając z zależności skalującej 3.3, którą opisałem w rozdziale 3.1. W rozważanym tutaj przypadku możemy ją zapisać w postaci:

$$\varphi_c^k(L) - \varphi_c^k = L^{-1/\nu}(A_0 + A_1 L^{-1} + \ldots + A_M L^{-M}), \quad L \gg 1, \tag{4.6}$$

gdzie $A_0, \ldots, A_M$ mają wartości stałe dla danych k i D, a M jest parametrem odcięcia.

Z drugiej strony, różniczkując równanie 4.2 względem φ dostaniemy:

$$\left.\frac{\partial P_{L,k}(\varphi)}{\partial \varphi}\right|_{\varphi=\varphi_c^k} = L^{1/\nu} f_k'(0), \quad L \gg 1, \tag{4.7}$$

gdzie $f_k'(x) \equiv df_k(x)/dx$. Po rozwinięciu równania 4.7 w szereg dostaniemy:

$$\frac{\partial P_{L,k}}{\partial \varphi}(\varphi_c^k) = L^{1/\nu}\left(B_0 + B_1 L^{-1} + \ldots + B_M L^{-M}\right), \tag{4.8}$$

gdzie $B_0, \ldots, B_M$ są pewnymi stałymi, a M jest parametrem odcięcia.

Z dopasowania danych otrzymanych z symulacji do tego równania możemy otrzymać wartość wykładnika krytycznego ν niezależnie od 4.6. Wartość tę można następnie wykorzystać w równaniu 4.6, zmniejszając tym samym liczbę parametrów do znalezienia i jednocześnie poprawiając dokładność otrzymanych wyników. Ponieważ prawdopodobieństwo $P_{L,k}(\varphi)$ rośnie bardzo szybko w pobliżu φ_c^k, przeprowadzając obliczenia użyłem pochodnej jego odwrotności, policzonej numerycznie przy użyciu metody pięciu punktów (ang. five-point centered-difference formula) [23]:

$$\frac{d\,f(x)}{d\,x} = \frac{f(x-h) - 8f(x-h/2) + 8f(x+h/2) - f(x+h)}{6h} \tag{4.9}$$

Mając wyznaczone krytyczne stopnie zapełnienia φ_c^k dla różnych wartości k, możemy obliczyć wartość graniczną, przy $k \to \infty$, progu perkolacji φ_c^∞ hipersfer w przypadku ciągłym.

Jak zostało pokazane w pracach [10] i [18] zbieżność $\varphi_c^k \to \varphi_c^\infty$ opisana jest zależnością wykładniczą:

$$1 - \varphi_c^k - \exp\left[\ln(1-\varphi_c^\infty)\frac{V_k}{V_{ex,k}}\right] \propto k^{-\theta}, \tag{4.10}$$

gdzie V_k jest objętością, $V_{ex,k}$ wyłączoną objętością obiektu o rozmiarze k.

4.3.1 Oszacowanie błędów

Dane otrzymane z symulacji dla każdego zestawu parametrów D, L oraz k podzieliłem na 10 podzbiorów o jednakowej wielkości. Niezależnie dla każdego z nich, w sposób opisany powyżej, policzyłem wartość $\varphi_c^k(L)$. Do kolejnych obliczeń wykorzystałem wartość średnią $\varphi_c^k(L)$ z dokładnością wyznaczoną jako błąd standardowy.

Do wyznaczenia wartości wykładnika krytycznego ν z równań 4.6 i 4.8, progu perkolacji φ_c^k z równania 4.6 oraz progów perkolacji w przypadku ciągłym φ_c^∞ z równania 4.10, wykorzystałem nieliniowe dopasowanie przy użyciu algorytmu *Levenberga-Marquardta*. Następnie błędy wyznaczenia tych wielkości oszacowałem jako pierwiastek kwadratowy z elementów diagonalnych macierzy kowariancji pomnożony przez $\max(1, \sqrt{\chi^2/\mathrm{dof}})$, gdzie $\sqrt{\chi^2/\mathrm{dof}}$ jest wynikiem zredukowanego testu chi-kwadrat, a dof liczbą stopni swobody.

4.4 Wyniki

4.4.1 Perkolacja węzłów dla $k = 1$

Najmniejsze rozpatrywane przeze mnie hipersfery miały średnicę $k = 1$. Zdyskretyzowaną postacią takich hipersfer jest pojedynczy węzeł sieci, czyli mamy do czynienia ze standardową perkolacją węzłów, którą opisałem w rozdziale 2.2. Dzięki temu mogłem wykorzystać wyniki otrzymane dla hipersfer o średnicy $k = 1$ do weryfikacji poprawności zastosowanego algorytmu, poprzez porównanie ich z dostępnymi danymi dla perkolacji węzłów.

Dla każdej z trzech z definicji perkolacji osobno (przypadki A, B i C opisane w rozdziale 4.3) wyznaczyłem progi perkolacji φ_c^1 w granicy dużych systemów, przy użyciu

równania 4.6 z odcięciem $M = 2$. Wartości wykładnika krytycznego ν wyznaczyłem niezależnie z wykorzystaniem równań 4.6 oraz 4.8, w obydwu przypadkach z odcięciem $M = 2$. Metoda druga dała dużo dokładniejsze wyniki. Dla każdego z rozpatrywanych wymiarów $D = 3, 4, 5$ policzyłem średnią z wartości otrzymanych dla przypadków A, B i C. Ze względu na to, że dane dla wszystkich trzech przypadków pochodzą z tych samych symulacji (nie są niezależne), w celu skompensowania możliwych korelacji, dokładność wyznaczenia średnich została dodatkowo przemnożona przez czynnik $\sqrt{3}$. W **Tabeli 4.1** zamieściłem zestawienie dotychczas znanych wartości (ν dotychczasowe) z wynikami przeprowadzonych przeze mnie obliczeń (ν obecne).

D	3	4	5
ν dotychczasowe	0,8764(12) [24]	0,6852(28) [10]	0,5723(18) [10]
	0,8751(11) [25]	0,6845(06) [26]	0,5757(06) [26]
	0,8762(12) [6]	0,6845(23) [27]	0,5737(33) [27]
	0,8774(13) [10]	0,686 (2) [9]	0,5738(01) [9]
ν obecne	0,8762(07)	0,6842(16)	0,5720(43)

Tabela 4.1: Wartości wykładnika krytycznego ν dla $k = 1$.

Dla przypadku $D = 3$ otrzymane wyniki są idealnie zgodne z dotychczas znanymi wartościami, a ich dokładność została nawet zwiększona o współczynnik $1,6$. Dla pozostałych wymiarów, $D = 4$ oraz $D = 5$, odchylenia od dotychczasowych wyników są większe, ale pozostają w rozsądnym zakresie.

Otrzymane wartości wykładnika krytycznego ν wykorzystałem następnie do wyznaczenia progów perkolacji przy użyciu równania 4.6. Otrzymane wyniki obliczeń przedstawiłem w **Tabeli 4.2**.

Dane zaprezentowane w tabelach wskazują, że wszystkie otrzymane wartości są ze sobą spójne oraz zgodne z dotychczas znanymi. Jest to potwierdzeniem poprawności kodu użytego do przeprowadzenia symulacji.

D	3	4	5
φ_c^1 dotychczasowe	0,311 60768(15) [6]	0,196 88561(3) [7]	0,140 79633(4) [7]
φ_c^1 obecne A	0,311 597(14)	0,196 949(37)	0,140 790(09)
B	0,311 606(04)	0,196 908(79)	0,140 804(05)
C	0,311 606(14)	0,196 964(56)	0,140 794(11)

Tabela 4.2: Progi perkolacji dla $k = 1$.

4.4.2 Perkolacja hipersfer

Wartości wykładnika krytycznego można wstawić do równania 4.6, które posłuży do wyznaczenia progów perkolacji w granicy nieskończenie dużych układów φ_c^k. Wraz z $k \to \infty$ wartości tych progów zbliżają się do progu perkolacji hipersfer w przypadku ciągłym φ_c^∞. Dla każdej rozpatrywanej średnicy hipersfer k wyznaczyłem numerycznie objętości zdyskretyzowanych hipersfer V_k oraz ich objętości wyłączone $V_{ex,k}$. Zakładając, że wzór 4.10 jest poprawny także dla hipersfer, użyłem go do wyznaczenia progów perkolacji φ_c^∞.

Otrzymane w ten sposób wartości progów perkolacji w przypadku ciągłym dla wszystkich trzech definicji perkolacji (A, B i C) są ze sobą zgodne w każdej badanej liczbie wymiarów $D = 3, 4, 5$. Zostały one zaprezentowane w **Tabeli 4.3** (φ_c^∞ obecne) wraz z ich średnimi, wyznaczonymi tak jak to opisałem dla przypadku $k = 1$ (φ_c^∞ uśrednione) oraz najdokładniejszymi wartościami znanymi dotychczas (φ_c^∞ dotychczasowe).

D	3	4	5
φ_c^∞ dotychczasowe	0,289 573(2) [28]	0,122 26(44) [29]	0,052 98(7) [29]
φ_c^∞ obecne A	0,289 5690(020)	0,121 0260(18)	0,052 2538(057)
B	0,289 5702(024)	0,121 0269(35)	0,052 2457(067)
C	0,289 5358(228)	0,121 0272(15)	0,052 2636(104)
φ_c^∞ uśrednione	0,289 5693(026)	0,121 0268(19)	0,052 2524(069)

Tabela 4.3: Progi perkolacji dla nachodzących na siebie hipersfer w przypadku ciągłym.

Warto zauważyć, że tak jak dla przypadku $D = 3$ jest dostępnych wiele estymat progów perkolacji [28] [29] [30], to dla większej liczby wymiarów ($D = 4, 5$) istnieje tylko jedna [29].

W przypadku trójwymiarowym otrzymany przeze mnie próg perkolacji oraz dokładność jego wyznaczenia są zbliżone do dotychczas znanych wartości [28]. Dla wyższej liczby wymiarów wyznaczyłem te wartości z dokładnością większą niż dotychczasowe. Niepewność progów perkolacji jest mniejsza od dotychczas znanych [29] odpowiednio o czynnik 230 ($D = 4$) oraz 10 ($D = 5$). Jednak wyznaczone progi różnią się od dotychczasowych wyników o 3σ ($D = 4$) i o 10σ ($D = 5$), gdzie σ jest sumą niedokładności obu estymat. To oznacza, że któreś z niepewności są zbyt niskie. Co więcej, w niedawno opublikowanej pracy dotyczącej perkolacji z rozszerzonym sąsiedztwem w czterech wymiarach [31], autorzy otrzymali wartość gęstości liczbowej dla hipersfer wynoszącą $\eta = 0,1291(7)$. Zgodnie ze wzorem 2.5, jest to równoważne wartości krytycznego stopnia zapełnienia $\varphi = 0,1211(6)$. Można to uznać za potwierdzenie przedstawionego powyżej, otrzymanego przeze mnie wyniku. Jednak wartość z [29] także nie może być na tej podstawie odrzucona, ponieważ mieści się ona (prawie) w sumie odchyleń standardowych.

W pracy [10], analiza podobna do przedstawionej tutaj pokazała, że progi perkolacji dla pięciowymiarowych hipersześcianów, także te przedstawione w [29], mają zbyt optymistyczne oszacowane dokładności. Poniżej przedstawię argumenty wspierające uzyskane przeze mnie wartości.

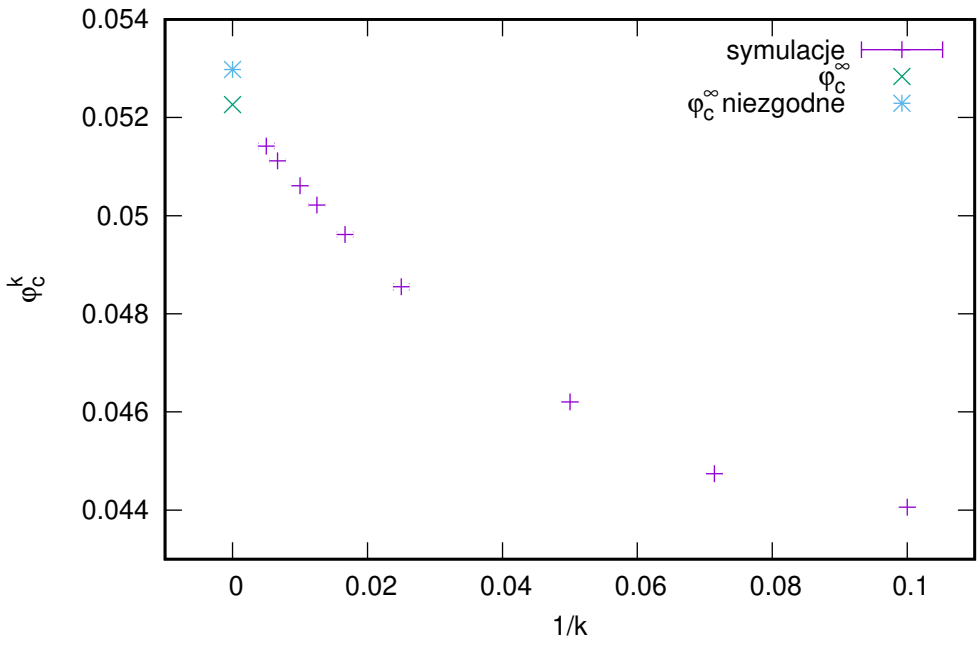
Po pierwsze kod programu używanego do przeprowadzenia symulacji jest jednakowy, niezależnie od wymiaru – wartość D jest argumentem szablonu klas napisanych w języku C++. Uzyskane na podstawie jego wyników wartości, dla $D = 3$ pokrywają się idealnie z dotychczas znanymi progami perkolacji, dla $D = 4$ różnica jest zauważalna, a dla $D = 5$ występuje widoczna niezgodność. Dla każdego z wymiarów, wyniki uzyskane dla trzech definicji perkolacji — A, B i C są ze sobą zgodne. Co więcej, w przypadku $k = 1$ uzyskane wyniki są zgodne z dostępnymi, bardzo dokładnymi wartościami [6] [7]. Wyliczone progi perkolacji φ_c^∞ znajdują się w teoretycznych granicach policzonych w [29].

Wyniki przeprowadzonych przeze mnie dopasowań zweryfikowałem przy pomocy testu zgodności chi-kwadrat. Dla praktycznie wszystkich z nich otrzymałem wyniki zredukowanego testu chi-kwadrat o wartościach $\sqrt{\chi^2/dof} < 2$. Wskazuje to na bardzo

dobrą zgodność danych wejściowych oraz ich dokładności z zastosowanym modelem teoretycznym.

Jak już opisałem w rozdziale 4.2.4, sprawdziłem także zastosowany podczas symulacji generator liczb pseudolosowych — *Mersenne Twister*, porównując otrzymane za jego pomocą wyniki z otrzymanymi przy użyciu innego — *RANLUX24*, co potwierdziło niezależność otrzymanych z symulacji wartości.

Na **Rysunku 4.4** zaznaczyłem otrzymane przeze mnie wartości progów perkolacji φ_c^k dla różnych rozmiarów hipersfer w przypadku $D = 5$. Widoczna jest na mim również ekstrapolacja φ_c^∞ dla $k \to \infty$ zgodnie z równaniem 4.10. Dopasowanie jest dobre, z bardzo małymi, pojedynczymi niepewnościami. Próg perkolacji z [29] jest w sposób oczywisty niezgodny z danymi pochodzącymi z przeprowadzonych przeze mnie symulacji.



Rysunek 4.4: Progi perkolacji φ_c^k hipersfer w $D = 5$ i ich ekstrapolacja dla $k \to \infty$ zgodnie z równaniem 4.10: φ_c^∞. Wartość z [29] niepasująca do danych: φ_c^∞ niezgodne.

Wyniki dopasowania danych do równania 4.10 pokazały, że wartość graniczna progu perkolacji φ_c^∞ jest prawie niezależna od wyboru odcięcia k_{min}, poniżej którego dane nie były brane pod uwagę.

W przeciwieństwie do progu perkolacji, wartość wykładnika θ nie tylko została wyznaczona z dużą niepewnością, ale mocno fluktuuje wraz ze zmianą k_{min}. Z tego względu nie było możliwe zweryfikowanie przypuszczenia wysuniętego dla hipersześcianów w [10], w przypadku hipersfer. Takie zachowanie może tłumaczyć jedna z głównych różnic pomiędzy hipersześcianami, a zdyskretyzowanymi hipersferami — a mianowicie brzeg tych drugich jest dużo bardziej nieregularny. W przeciwieństwie do nich, hipersześciany o różnych rozmiarach mają dokładnie ten sam kształt, jedynie przeskalowany.

Rozdział 5

Podsumowanie

Powyżej przedstawiłem analizę perkolacji nachodzących na siebie zdyskretyzowanych hipersfer w wymiarach od 3 do 5.

Jej przeprowadzenie było możliwe dzięki zastosowaniu opracowanych przeze mnie wydajnych struktur danych i algorytmów. Umożliwiły one przeprowadzenie dużej liczby symulacji procesów perkolacji hipersfer dla rozległych systemów, co było konieczne do uzyskania danych niezbędnych do dalszej analizy.

Podejście, polegające na wyznaczeniu wartości granicznej, przy $k \to \infty$, progów perkolacji φ_c^k hipersfer o różnych średnicach k, pozwoliło na wyznaczenie progu perkolacji w przypadku ciągłym ze znacznie większą dokładnością niż zostało to dotychczas zrobione przy użyciu innych metod. Ponadto otrzymana przeze mnie wartość wykładnika krytycznego długości korelacji ν dla $D = 3$ ma większą dokładność niż dotychczas znana.

W przypadku perkolacji hipersześcianów wartość wykładnika $\theta = 3/2$ wydaje się być uniwersalna we wszystkich wymiarach rozpatrywanych przez autorów pracy [10], czyli dla $D = 2, \ldots, 7$. Wykorzystanie tej wartości umożliwiło na wyznaczenie, z bardzo dużą dokładnością, wartości progu perkolacji φ_c^∞ w przypadku ciągłym dla hipersześcianów.

Ze względu na duże fluktuacje wartości wykładnika θ, nie byłem w stanie potwierdzić hipotezy o jego uniwersalnej wartości w granicy $k \to \infty$ dla użytego modelu perkolacji hipersfer. Prawdopodobną tego przyczyną jest duża nieregularność zdyskretyzowanych hipersfer w porównaniu z hipersześcianami. Możliwe, że można ten problem rozwiązać przeprowadzając symulacje dla dużo większych układów, ale wy-

magałoby to dużo większych zasobów obliczeniowych (w szczególności ilości pamięci) niż mają komputery z których korzystałem.

Mimo, że nie byłem w stanie potwierdzić uniwersalności wartości wykładnika θ, podejście z zastosowaniem objętości i wyłączonych objętości zdyskretyzowanych hipersfer pozwoliło mi na uzyskanie bardzo dokładnych estymat progów perkolacji dla hipersfer w przypadku ciągłym. W przypadku pominięcia wartości V_k oraz $V_{ex,k}$ i użycia prostego dopasowania wartości φ_c^k otrzymuje się wartości o dokładności mniejszej o co najmniej rząd wielkości.

Użyta przeze mnie metoda oraz uzyskane wyniki zostały opublikowane w artykule [32], który napisałem wspólnie z dr hab. Grzegorzem Kondratem.

Bibliografia

[1] Jesper Lykke Jacobsen
Critical points of Potts and O(N) models from eigenvalue identities in periodic Temperley-Lieb algebras
2015 *Journal of Physics A.: Mathematical and Theoretical* **48** 454003

[2] Michael Ellis Fisher
Critical Probabilities for Cluster Size and Percolation Problems
1961 *J. Math. Phys.* **2** 620

[3] Don R. Baker, Gerald Paul, Sameet Sreenivasan and H. Eugene Stanley
Continuum percolation threshold for interpenetrating squares and cubes
2002 *Phys. Rev. E* **66**(4) 046136

[4] Peter J. Reynolds, H. Eugene Stanley and William Klein
Ghost fields, pair connectedness, and scaling: exact results in one-dimensional percolation
1977 *J. Phys. A: Math. Gen.*
textbf10 L203–L210

[5] Dietrich Stauffer and Amnon Aharony
Introduction to Percolation Theory. Revised Second Edition
1992 *Taylor & Francis*

[6] Xiao Xu, Junfeng Wang, Jian-Ping Lv, Youjin Deng
Simultaneous analysis of three-dimensional percolation models
2014 *Frontires of Physics* **9** 113–119

[7] Stephan Mertens and Cristopher Moore
Percolation thresholds and Fisher exponents in hypercubic lattices
2018 *Phys. Rev. E* **98** 022120

[8] John A. Gracey
Four loop renormalization of φ^3 theory in six dimensions
2015 *Phys. Rev. D* **92** 025012

[9] Michael Borinsky, John A. Gracey, Mikhail V. Kompaniets and Oliver Schnetz
Five-loop renormalization of φ^3 theory with applications to the Lee-Yang edge singularity and percolation theory
2021 *Phys. Rev. D* **103** 116024

[10] Zbigniew Koza and Jakub Poła
From discrete to continuous percolation in dimensions 3 to 7
2016 *J. Stat. Mech.* 103206

[11] Grzegorz Kondrat, Zbigniew Koza and Piotr Brzeski
Jammed systems of oriented needles always percolate on square lattices
2017 *Phys. Rev. E* **96** 022154

[12] Mark E. J. Newman and Robert M. Ziff
Efficient Monte Carlo Algorithm and High-Precision Results for Percolation
2001 *Phys. Rev. Lett.* **85** 4104

[13] Paulo Murilo C. de Oliveira, Rafaella A. Nórbrega and Dietrich Stauffer
Corrections to Finite Size Scaling in Percolation
2003 *Brazilian Journal of Physics* **33** 616–618 ISSN 0103-9733

[14] Mark E. J. Newman and Robert M. Ziff
Fast Monte Carlo algorithm for site or bond percolation
2001 *Phys. Rev. E* **64**(1) 016706

[15] Isaac Balberg, Charles H. Anderson, S. Alexander and N. Wagner
Excluded volume and its relation to the onset of percolation
1984 *Phys. Rev. B* **30**(7) 3933–3943

[16] Isaac Balberg
Recent developments in continuum percolation
1987 *Philosophical Magazine B* **56** 991–1003

[17] Vinod K. S. Shante, Scott Kirkpatrick
An introduction to percolation theory
1971 *Advances in Physics* **20** 325–357

[18] Zbigniew Koza, Grzegorz Kondrat and Karol Suszczyński
Percolation of overlapping squares or cubes on a lattice
2014 *J. Stat. Mech.* P11005

[19] Zhipeng Xun, Dapeng Hao and Robert M. Ziff
Site percolation on square and simple cubic lattices
with extended neighborhoods and their continuum limit
2021 *Phys. Rev. E* **103** 022126

[20] Makoto Matsumoto and Takuji Nishimura
Mersenne twister: a 623-dimensionally equidistributed
uniform pseudo-random number generator
1998 *ACM Transactions on Modeling and Computer Simulations* **8** 3–30

[21] Frederick James
RANLUX: A Fortran implementation of the high-quality
pseudorandom number generator of Lüscher
1994 *Computer Physics Communications* **79** 111–114

[22] Mark D. Rintoul and Salvatore Torquato
Precise determination of the critical threshold and exponents
in a three-dimensional continuum percolation model
1997 *Journal of Physics A: Mathematical and General* **30** L585

[23] Timothy Sauer
Numerical Analysis. Second edition
2012 *Pearson Education* 250

[24] Junfeng Wang, Zongzheng Zhou, Wei Zhang,
Timothy M. Garoni and Youjin Deng
Bond and site percolation in three dimensions
2013 *Phys. Rev. E* **87** 052107

[25] Hao Hu, Henk W. J. Blöte, Robert M. Ziff and Youjin Deng
Short-range correlations in percolation at criticality
2014 *Phys. Rev. E* **90** 042106

[26] Xiao-Jun Tan, You-Jin Deng and Jesper Lykke Jacobsen
N-cluster correlations in four- and five-dimensional percolation
2020 *Frontiers of Physics* **15** 41501

[27] Zhongjin Zhang, Pengcheng Hou, Sheng Fang, Hao Hu, Youjin Deng
Critical exponents and universal excess cluster number
of percolation in four and five dimensions
2021 *Physica A* **580** 126124

[28] Christian D. Lorenz and Robert M. Ziff
Precise determination of the critical percolation threshold
for the three–dimensional "Swiss cheese" model using a growth algorithm
2001 *J. Chem. Phys.* **114**(8) 3659

[29] Salvatore Torquato
Effect of dimensionality on the continuum percolation
of overlapping hyperspheres and hypercubes
2012 *J. Chem. Phys.* **136**(4) 054106

[30] Giacomo Gori and Andrea Trombettoni
Conformal invariance in three dimensional percolation
2015 *J. Stat. Mech.* P07014

[31] Pengyu Zhao, Jinhong Yan, Zhipeng Xun, Dapeng Hao and Robert M. Ziff
Site and bond percolation on four-dimensional simple hypercubic lattices
with extended neighborhoods
2022 *J. Stat. Mech.* 033202

[32] Piotr Brzeski and Grzegorz Kondrat
Percolation of hyperspheres in dimensions 3 to 5: from discrete to continuous
2022 *J. Stat. Mech.* 053202