

学位論文

Renormalization-Group Analysis of
Classical Spins on Networks

(ネットワーク上の古典スピンの
繰り込み群解析)

平成 28 年 12 月博士(理学)申請

東京大学大学院理学系研究科
物理学専攻
堀田義仁

Abstract

In this thesis, we present both analytical and numerical treatment of classical spins on networks. This thesis consists of renormalization-group studies of two systems. The first one is the self-avoiding walk on a complex network and the second one is the Ising model on a square lattice. At first glance, the first system is not a spin model and the second system is not on a network but on a lattice, so that the title of this thesis may not be appropriate. We, however, explain that both problems can be actually treated as spin models on networks. Although renormalization-group study of classical systems have a long history and have been studied by many people, we will delve into the problem from a new viewpoint.

In the study of the first model, we focus on an analytical aspect. We treat complex networks, which have been a hot topic for the last fifteen years. The network has a repeated fractal structure like a matrioshka and is especially suitable to study by renormalization-group technique. We study the self-avoiding walk on the complex fractal networks called the (u, v) -flower by mapping it to the \mathcal{N} -vector model in a generating-function formalism and carrying out the renormalization-group calculation of the generating function. First, we analytically calculate the critical exponent ν and the connective constant by a renormalization-group analysis in arbitrary fractal dimensions. We find that the exponent ν is equal to the displacement exponent, which describes the speed of diffusion in terms of the shortest distance. Second, by obtaining an exact solution for the (u, u) -flower, we provide an example which supports the conjecture that the universality class of the self-avoiding walk on graphs is *not* determined only by the fractal dimension.

In the study of the second model, we focus on a numerical aspect. We discuss dynamics of the model by using a new data structure, called tensor network. We propose a tensor-network algorithm for discrete-time stochastic dynamics of a homogeneous system in the thermodynamic limit. We map a d -dimensional nonequilibrium Markov process to a $(d + 1)$ -dimensional infinite tensor network by using a higher-order singular-value decomposition. As an application of the algorithm, we compute the nonequilibrium relaxation from

a fully magnetized state to equilibrium of the one- and two-dimensional Ising models with the periodic boundary condition. Utilizing the translational invariance of the systems, we analyze the behavior in the thermodynamic limit directly. We estimated the dynamical critical exponent $z = 2.16(5)$ for the two-dimensional Ising model. Our approach fits well with the framework of the nonequilibrium-relaxation method. On one hand, our algorithm can compute the time evolution of the magnetization of a large system precisely for a relatively short period. In the nonequilibrium-relaxation method, on the other, one needs to simulate dynamics of a large system for a short time. The combination of the two provides a new approach to the study of critical phenomena.

Acknowledgements

Foremost, I would like to show my greatest appreciation to Prof. Naomichi Hatano. He gave me an opportunity to study statistical physics in my Ph.D course. He kindly taught me numerical simulations and we have discussed content of this thesis for four years. Without his support, this thesis would be impossible.

In the study of the first part of this thesis, Dr. Tomotaka Kuwahara and Dr. Tatsuro Kawamoto helped me a lot. In particular, Dr. Kawamoto gave me guidance of complex networks. I am grateful to them.

I also thank Prof. Tota Nakamura for introducing me to the nonequilibrium-relaxation method. His advice led me to the study of the latter half of this thesis. His comment on analysis of numerical results was also very helpful.

I am also grateful to the members of the Hatano laboratory. I learned physics of various fields in weekly lab meetings. They were always enjoyable to attend.

Finally, I thank friends at the department of physics. Dinners with them at Hongo were the source of powers to write this thesis.

Contents

Table of contents	6
1 Introduction	11
1.1 The self-avoiding walk on complex networks	12
1.2 Simulation of Glauber dynamics by tensor networks	14
1.3 Outline of the thesis	17
2 Review of complex network and self-avoiding walk	19
2.1 Fractal	19
2.1.1 Fractal dimension	20
2.1.2 Similarity dimension	20
2.1.3 Box-counting dimension	21
2.1.4 Cluster dimension	22
2.2 Complex network	24
2.2.1 Graph	24
2.2.2 Complex network	24
2.2.3 Degree	25
2.2.4 Fractal complex networks	26
2.2.5 The (u, v) -flower	27
2.3 Self-avoiding walk	30
2.3.1 Self-avoiding walk in a Euclidian space	30
2.3.2 \mathcal{N} -vector model	32
3 Self-avoiding walk on fractal complex network	37
3.1 Ensemble of fixed length paths	38
3.2 Exact renormalization-group analysis of the general (u, v) -flowers	39
3.2.1 Generating function	39
3.2.2 Renormalization-group analysis	40
3.2.3 Existence and uniqueness of a nontrivial fixed point	43
3.2.4 Range of ν	44

3.2.5	Exact results	45
3.2.6	Comparison to the mean-field theory	46
3.3	Exact solution of the (u, u) -flower	47
3.4	Numerical simulations	49
3.4.1	The number of paths	49
3.4.2	The displacement exponent	50
4	Review of tensor network	55
4.1	Matrix	55
4.1.1	Colon notation	55
4.1.2	Matrix norms	56
4.1.3	The singular-value decomposition	57
4.1.4	The Kronercker product	58
4.2	Tensor	59
4.2.1	Notation of tensors	59
4.2.2	Unfolding	60
4.2.3	Outer product and canonical rank	60
4.2.4	Tensor network diagram	61
4.2.5	Modal product	62
4.3	Decomposition of tensors	63
4.3.1	The higher-order singular-value decomposition	64
4.3.2	Canonical decomposition	65
4.4	Alternating least-square algorithm	65
4.4.1	Example of ALS: Image compression	67
4.5	Tensor networks	67
4.5.1	Tensor-train decomposition	69
4.5.2	The density-matrix renormalization group	70
4.5.3	Tensor network of lattice structure	71
5	Nonequilibrium-relaxation method with tensor networks	73
5.1	Model and notation	73
5.1.1	Kinetic Ising model	74
5.1.2	Diagrammatic representation of the Markov chain	75
5.2	One-dimensional kinetic Ising model	78
5.2.1	Transition matrix as matrix-product operator	78
5.3	Two-dimensional kinetic Ising model	82
5.3.1	Transition operator as tensor-network operator	84
5.3.2	Nonequilibrium relaxation	85
6	Summary	89

CONTENTS

Appendix A Additional results of numerical simulation	93
A.1 The number of paths	93
A.2 The exponent of displacement ν' by the depth-limited search .	93
Appendix B Analysis of the Monte-Carlo simulation	97
Bibliography	100

Chapter 1

Introduction

The critical phenomenon is one of the central interests of statistical mechanics. Systems at criticality show various interesting behavior because the typical spatial and the temporal length scales vanish. One of the most important tools to understand critical phenomena is the renormalization group; we repeat coarse-graining and observe the change of physical quantities. Despite of many years after the first proposal, it is still important to broaden the range of application of the renormalization group for deeper understanding of critical phenomena.

To this end, we study by means of the renormalization group classical spin systems on networks both theoretically and numerically. The new feature here is that the spins are on the nodes (vertices) of networks (graphs). This leads us to renormalization of networks, and opens ways to new renormalization-group analyses of classical spins.

The first half of the thesis presents a theoretical study of the self-avoiding path in terms of zero-component spins on fractal networks. We will show that this model is particularly suitable for exact and rigorous renormalization-group study. The second half presents development of a numerical algorithm for analyzing dynamics of classical spin systems. We will represent the dynamics of the probability distribution of an n -dimensional system as an $(n + 1)$ -dimensional network. This network representation enables us to study dynamical critical phenomena of infinite systems by renormalization of infinite networks.

1.1 The self-avoiding walk on complex networks

In Chap. 3, we will delve into the study of classical spin models on networks. We will consider the self-avoiding walk on fractal graphs called the (u, v) -flower. The self-avoiding walk is important both in graph theory and statistical mechanics. In graph theory, a self-avoiding walk is usually called a path. Enumeration of paths is a classical problem in computer science [1] and enumeration algorithms of all paths connecting two nodes (s - t paths) have been actively studied [2]. Furthermore, paths appear in many graph algorithms such as the depth-first search [3]. On the other hand, the self-avoiding walk in the Euclidian spaces is one of the simplest models of a polymer and its scaling properties have been of statistical physicists' interest [4]. This model of a polymer can be mapped to the $\mathcal{N} \rightarrow 0$ limit of the \mathcal{N} -vector model as de Gennes pointed out [5, 4]; the connection enabled us to understand the model from the viewpoint of critical phenomena of usual spin systems.

The properties of the self-avoiding walk is poorly understood in the Euclidian space, in particular in two, three, and four dimensions. For the problems of the self-avoiding walk on fractals embedded in the Euclidian space, analysis becomes easier and we can obtain exact solutions on some fractals, *e.g.*, the Sierpinski gasket [6, 7]. Such exact solutions have helped us deepen our understanding of the self-avoiding walk. For example, the three-dimensional Sierpinski gasket and the two-dimensional square lattice have the same fractal dimension, while the critical exponents ν of the self-avoiding walk on them are different (Table 1.1).

Table 1.1: The fractal dimension and the critical exponent ν of various models. For models on regular lattices, it is widely believed that ν depends only on the Euclidian dimension and is independent of details of lattices. For models on fractals, on the other hand, it is conjectured that ν is dependent on many parameters other than the fractal dimensionality. Notice that ν is different for the two-dimensional regular lattice and the three-dimensional Sierpinski gasket although their (fractal) dimensionality is the same.

	Lattice	Fractal dimension	Critical exponent ν
Regular lattice [8]	1D lattice	1	1
	2D lattice	2	0.75
	3D lattice	3	0.59
	4D lattice and higher	$d \geq 4$	0.5
Fractal [6]	Branching Koch curve	1.46	0.891
	Sierpinski gasket in 2D	1.58	0.798
	Sierpinski gasket in 3D	2	0.729

This suggests that there is no one-to-one correspondence between the fractal dimension and the universality class. Fractals on which the model is exactly solvable in arbitrary fractal dimensions have not been obtained to the author's knowledge. In this thesis, generalizing the problem to the self-avoiding walk on fractal *graphs*, we obtain exact results in arbitrary fractal dimensions. We thereby verify that critical exponents in fractal graphs are *not* solely determined by the fractal dimension. This fact was conjectured in the 1980s [6, 9, 10], but has never been proved because of the lack of exactly solvable models in arbitrary fractal dimensions.

Song *et al.* found that a few graphs in real networks are indeed fractal [11, 12]. They noticed that complex networks that had been studied many times were fractal, *e.g.*, the WWW network, actors' collaboration network, and biological networks of protein-protein interactions. After the discovery in the real networks, several artificial fractal complex networks have been devised [13]. One of such networks is the (u, v) -flower [14]. As deterministic fractals such as the Sierpinski gasket and the Cantor set helped us understand real fractals in the Euclidian spaces, deterministic fractal complex networks can deepen our understanding of dynamics on fractal complex networks in the real world.

Is it possible to understand the scaling properties of paths on graphs in terms of critical phenomena? In Chap. 3, through the mapping to the \mathcal{N} -vector model, we extend the theory of the self-avoiding walk on regular lattices to that on graphs by using the shortest distance as a distance rather than the Euclidian distance. We perform an exact renormalization-group analysis on the self-avoiding walk on the (u, v) -flower in order to obtain the effective coordination number of a walker (namely, the connective constant) and a critical exponent of the mapped \mathcal{N} -vector model in the limit $\mathcal{N} \rightarrow 0$, namely the zero-component ferromagnet. We thereby answer the questions as to (i) how the number of paths with a fixed length and a fixed starting point increases and (ii) how the mean shortest distance between the two end points grows as the path length increases.

1.2 Simulation of Glauber dynamics by tensor networks

In Chap. 5, we develop algorithms to analyze stochastic processes of spin systems using tensor networks. Stochastic processes often appear in statistical mechanics. Monte Carlo methods are most often used to simulate stochastic processes. The density-matrix renormalization group (DMRG) is also some-

times used to analyze them [15, 16, 17, 18, 19, 20]. Tensor-network algorithms are generalization of DMRG and transfer-matrix methods to higher dimensions [21, 22, 23] and can handle models in two and higher dimensions straightforwardly. We combine our algorithm with the nonequilibrium-relaxation method [24, 25] to evaluate critical exponents. The computational time of our algorithm does not depend on the system size when the system is homogeneous, whereas the computational time of Monte Carlo simulation generally depends on the system size.

Monte Carlo methods are stochastic processes that are often used in studies of statistical mechanics. Although Monte Carlo methods have advantages, such as wide applicability and easiness of implementation, they also have drawbacks, *e.g.*, the dependence of computational complexity on the system size. As another drawback, equilibrium Monte Carlo analysis of critical phenomena become extremely difficult as the system approaches a critical point because of the divergence of the relaxation time. The nonequilibrium-relaxation method [24, 25], on the other hand, determines critical exponents including dynamical ones by observing relaxation from an ordered state to an equilibrium state. This method is especially suitable for systems with large fluctuation and long relaxation, *e.g.*, frustrated and random systems [26].

Other than Monte Carlo simulations, DMRG is also popular, having been very successful in one-dimensional quantum systems [27]. Recently, developments in the field of quantum information have stimulated extensions of DMRG to higher-dimensional systems. Tensor-network algorithms are such extension [21, 22, 23]. One of the reasons why tensor-network algorithms have called attention is that they can handle systems with large degrees of freedom with small computational cost as long as the system is homogeneous. For example, static critical exponents of two- and three-dimensional Ising models have been obtained in high accuracy by tensor renormalization-group methods [28, 29, 30, 31, 32, 33]. On the other hand, calculation of a two-dimensional quantum system by tensor networks is not so successful as classical cases [34].

DMRG studies of stochastic processes [15, 16, 17, 18, 19, 20] have been limited to one-dimensional systems until recently. T. H. Johnson *et al.* studied nonequilibrium stochastic processes in one and two dimensions using a tensor-network algorithm called time-evolving block decimation [35, 36, 37]. It discretizes the time of the dynamics of a finite system, using the Suzuki-Trotter decomposition [38]. They showed that their algorithm can compute in high accuracy observables with large-variances that strongly depend on the time-evolving path of configuration, while Monte Carlo methods require a large number of samples for such variables.

One of the most important dynamics of classical spin systems is the

Glauber dynamics [39], for which equilibrium distribution is usually studied. It is important to study the Glauber dynamics for two reasons. First, the Glauber dynamics in discrete time can be regarded as an approximation of dynamics of real condensed matters, which actually obey the Schrödinger equation. Although it is not *a priori* ensured that the approximation is good, the Glauber dynamics shows interesting behavior, such as the divergence of correlation time and the spatial correlation [40], which are experimentally observable [41]. Therefore, the Glauber dynamics is a good starting point of studying nonequilibrium critical phenomena. Second, study of the Glauber dynamics in one dimension is a good test bench of numerical algorithms, because it is exactly solvable. In this thesis, we compare the exact solution of relaxation of one-dimensional Ising model [42, 43] with our numerical calculation, and assess the performance of our algorithm. Third, combining it with nonequilibrium relaxation [24, 25], we can derive static critical exponents as well as the dynamical critical exponent. Thus, considering the Glauber dynamics is also useful to study static properties of critical phenomena.

We can consider either a continuous-time version [39] or a discrete-time version [42] of the Glauber dynamics. Physically speaking, both versions are approximations of reality, so that it is just a matter of preference which version we study. However, both of them have pros and cons. The continuous-time version is mathematically equivalent to the Schrödinger equation, and hence we can directly apply algorithms of quantum many-body systems to the classical dynamics, *e.g.*, the time-evolving block decimation [35, 36, 37]. It can, however, have the difficulty of numerical error associated with the discretization of time, which is inevitable in numerical simulation. On the other hand, the discrete-time version is different from the time evolution of quantum systems, and therefore we need to develop a special algorithm that is not applicable to quantum systems. Since it is formulated as the discrete Markov process, we do not have numerical error associated with time discretization.

Simulation of Markov processes of physical systems with tensor networks is a promising approach because the data structure of tensor networks represents the spatial structure of the system and its correlation. It is thus suited to computation of time evolution of systems with spatial correlations, in particular systems at criticality. The dynamical critical exponent is a quantity that characterizes dynamical critical phenomena [41]. We are usually interested in dynamical critical phenomena for systems with dimensions greater than two because the dynamical critical exponents take nontrivial values there. Their analytical calculation is usually intractable, and we need to rely on numerical methods.

In Chap. 5, we propose a tensor-network algorithm for discrete-time

Markov chains in d -dimensional infinite homogeneous systems. Representing the probability distribution with a tensor-network state and the transition probability with a tensor-network operator, we map d -dimensional nonequilibrium processes to $(d + 1)$ -dimensional infinite tensor networks. While other tensor-network algorithms for dynamics usually make use of the Suzuki-Trotter decomposition [35, 36, 37], we construct a tensor-network operator of the transition probability in an entirely different way. We construct a tensor-network operator for a sublattice-flip update, using a higher-order singular-value decomposition [44]. Taking advantage of the homogeneity of the systems, we treat infinite systems directly just as the infinite time-evolving block decimation algorithm [45, 46]. The correlation length diverges in a critical system, for which we need to study a large system. Our algorithm is especially suitable for the study of dynamical critical phenomena because the computational complexity of our algorithm does not depend on the system size.

We analyze nonequilibrium relaxation of the magnetization of the one- and two-dimensional Ising models as an application of our algorithm. In particular, we determine the dynamical critical exponent z of the two-dimensional Ising model. Our algorithm of time evolution particularly goes well with the nonequilibrium-relaxation method [25], for which one prepares a large system and compute time evolution for a relatively short time. Our method has common features with the nonequilibrium-relaxation method, which also prepares a systems so large that it can be treated as the thermodynamic limit and computes the time evolution for a short period.

1.3 Outline of the thesis

In Chap. 2, we briefly review the self-avoiding walk and complex networks. Based on this prerequisite knowledge, we present our theory of the self-avoiding walk on complex networks in Chap. 3. In Chap. 4, we review prerequisite knowledge that we need in Chap. 5, in particular about tensor networks from the point of numerical computation. In Chap. 5, we develop tensor network algorithms to simulate dynamics of classical spin systems. By combining the nonequilibrium relaxation method, we show that we can estimate the dynamical critical exponent. Finally, in Chap. 6, we summarize the achievements of this thesis.

1.3. OUTLINE OF THE THESIS

Chapter 2

Review of complex network and self-avoiding walk

In this chapter, we briefly review previous studies and introduce the minimum amount of concepts which are required to understand Chap. 3. The present chapter is made as self-contained as possible.

First, we introduce the concept of the fractal. Because distinguishing various fractal dimensions is important to understand the fractality of complex networks, we introduce several definitions of fractal dimension. Next, we review the basics of complex networks and present a model of fractal networks called the (u, v) -flower. We will consider the self-avoiding walk on the (u, v) -flower in the following chapters. Finally, we describe well-known conjectures on the self-avoiding walk and review the mapping of the self-avoiding walk on a graph to a zero-component ferromagnet.

2.1 Fractal

Structures that appear in nature are really rich in variety [47, 48, 49]. Crystals possess discrete translational and rotational symmetries, and are classified by the point groups. On the other hand, molecules in gas and liquid are randomly distributed. Not all structures that appear in nature, however, are categorized to these two extreme classes. Many materials, such as polymers, liquid crystals, and glasses, indeed fall in between these two classes; they possess partly a periodic structure and partly a random one. If we consider objects in a wide sense, say, branching of trees, shapes of coastlines and rivers, and wrinkles of brains, most of them probably fall into the third class. When we discuss complexity, we do not say that objects with complete periodicity or complete randomness are complex; we regard objects which have

both order and randomness as complex.

2.1.1 Fractal dimension

Among the interesting properties of complex systems, a notable one is the self-similarity. The self-similarity is a symmetry in which a part of a system is similar to the whole part. Of course we cannot expect that real objects in nature are self-similar in a mathematically rigorous sense, but many are so in a statistical sense. For instance, if we enlarge a picture of a ria coast, it will be as complex as the original picture is. If we are not told which is an enlarged one, we will not be able to answer which one is which. This means that a ria coast lacks a typical length. If there were a typical length, the ria coast would look completely different after magnification beyond the length scale. On the other hand, if we magnify a picture of a coastline and the picture looks different when the picture is bigger than some size, it tells us that that size is the typical length of the coastline. Therefore, the self-similarity and lack of the typical length scale are equivalent.

When the ‘size’ M of an object is related with the ‘length’ L as

$$M \propto L^{d_f} \quad , \quad (2.1)$$

we say that the fractal dimension of that object is d_f . There are many mathematically rigorous definitions, but we introduce three definitions relevant to this thesis: the similarity dimension, the box-counting dimension, and the cluster dimension [48].

2.1.2 Similarity dimension

Let us consider how to define a dimension of an object consisting of many small components. For instance, a cubic lattice is a collection of small cubes of edge length l . We use the smallest component as a unit to measure the ‘volume’ of the whole object; we consider that the ‘volume’ of the whole object is proportional to the number of the smallest components contained in the object. Let $N(b)$ be the number of the smallest components needed to fill a cube of edge length $L = bl$. We immediately see that $N(b)$ and b are related as

$$N(b) = b^3 \quad (2.2)$$

in three dimensions. This is consistent with Eq. (2.1).

Generalizing this argument, we want to define a dimension which is applicable to objects without the smallest unit, such as the Sierpinski gasket



Figure 2.1: The Cantor set. Taken from Wikipedia [50]. One third of a line segment is eliminated at each step. The similarity dimension of this set is $\log_3 2$.

and the Cantor set. If an object of length scale L consists of $b^{d_{\text{sim}}}$ pieces of objects of length scale L/b , then we call d_{sim} the similarity dimension. For instance, the Cantor set (Fig. 2.1) is created by deleting the middle one third of a line segment repeatedly. Thus, the original set is restored by collecting two sets scaled down by $1/3$. As $2 = 3^{\log_3 2}$, the similarity dimension of the Cantor set is $\log_3 2$.

The definition of the similarity dimension is applicable only to mathematical models, since fractals in nature possess the self-similarity only in a statistical sense.

2.1.3 Box-counting dimension

The similarity dimension is applicable only in limited cases as we explained. We would like to introduce another dimension which can be used more generally. We define such a dimension by borrowing the concept of the outer measure.

Let the minimum number of cubes of edge length l necessary to cover an object be $N(l)$. If $N(l)$ and l are related as

$$N(l) \propto l^{-d_{\text{BC}}}, \quad (2.3)$$

we can measure the ‘volume’ of the object because we know the volume of the cubes without ambiguity. We refer to d_{BC} as the box-counting dimension. Precisely speaking, the box-covering dimension is defined as

$$d_{\text{BC}} = \lim_{l \searrow 0} \frac{\log N(l)}{\log(1/l)}. \quad (2.4)$$

Unlike the similarity dimension, the definition (2.4) is directly applicable to fractals in nature as well as artificial fractals such as the Sierpinski gasket. It has been indeed known since long years ago that the length of a coastline depends on the precision of measurement [51] (Fig. 2.2). This reflects the fact that the fractal dimensions of coast lines are greater than unity.

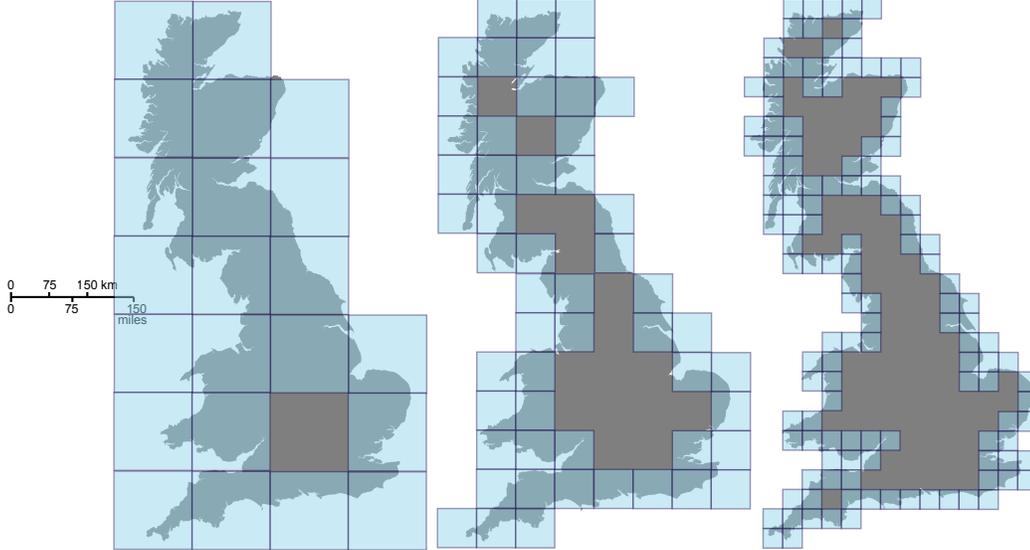


Figure 2.2: The Coastline of the UK. Taken from Wikipedia [52]. The box-counting dimension of the coastline is determined by the minimum number of boxes to cover it.

2.1.4 Cluster dimension

As explained above, we can use the similarity dimension only for artificial fractals with a rigorous self-similarity. It would be convenient if the similarity dimension could be used for objects with a self-similarity in a statistical sense as the box-covering dimension. Let us stipulate that a fractal has a minimum length scale. Let $\tilde{N}(L)$ be the average number of the minimum units inside a cube of edge length L . As the similarity dimension is based on the number of the smaller units, we define a cluster dimension in terms of the average number of the minimum units:

$$\tilde{N}(L) \propto L^{d_c}. \quad (2.5)$$

We call d_c the cluster dimension. We can rephrase Eq. (2.5) as

$$\tilde{N}(L) = b^{d_c} \tilde{N}(L/b). \quad (2.6)$$

Figure 2.3 is the simulation of a model of cluster formation, called the diffusion-limited aggregation [53]. We let each particle perform random walk starting at a random position on the four edges. The number of particles is 10,000 and the length of the edges is 500 in the simulation. By measuring the number of particles inside circles of various radii, we can estimate the cluster dimension. The estimated cluster dimension is $d_c = 1.7$ [53].

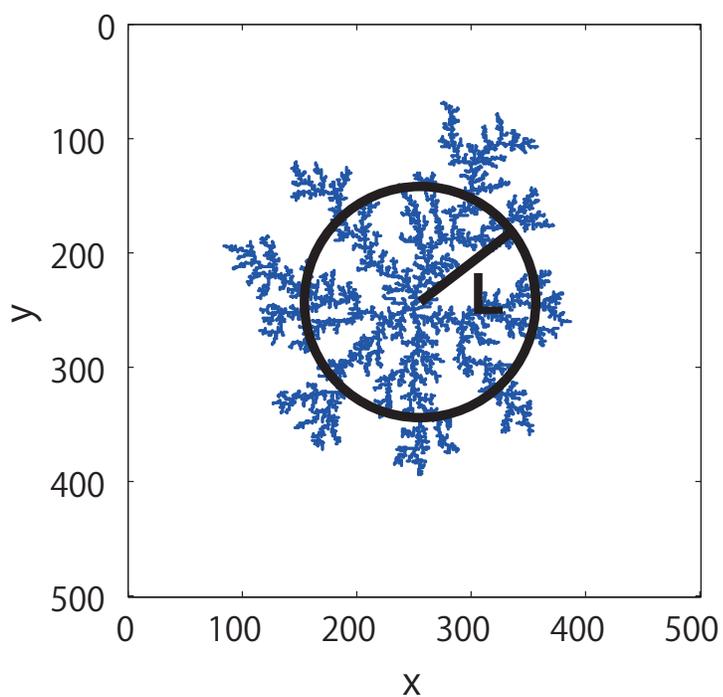


Figure 2.3: The diffusion-limited aggregation. 10,000 particles aggregate to form a cluster. By counting the number of particles inside circles of various radii, we can estimate the cluster dimension numerically.

While there are many definitions of fractal dimensions, it is empirically known that fractal dimensions of fractals in nature seldom depend on the choice of the type of the fractal dimensionality. Hence, the definitions are usually not distinguished and just called ‘the fractal dimension d_f ’. There are, however, cases where fractal dimensions strongly depend on the choice in complex networks as we will explain later.

2.2 Complex network

2.2.1 Graph

Graph theory has a long history. It began in the eighteenth century when a great mathematician Leonhard Euler visited Königsberg. He asked himself whether there is a route to visit every bridge in the city exactly once and to go back to the starting point (Fig. 2.4). The map of the city is originally a two-dimensional one, but in order to solve this problem we do not need the Euclidian distance; we can abstract the map. The abstracted map is represented by and curves connecting them. The black circles and curves are called nodes and edges, respectively. In this case, the nodes represent lands and the edges do bridges.

The nodes and edges are not necessarily associated with physical objects; this kind of abstraction of problems is often useful. For instance, graphs often appear in problems of computer algorithms, which have clearly nothing to do with physical objects. For glossary of graph theory, refer to textbooks or web dictionaries [54].

2.2.2 Complex network

Though there is no rigorous definition of complex networks, graphs which appear in real systems are usually called complex networks. The adjective ‘complex’ is used because the real systems usually have a complex structure. Real networks possess both randomness and order to some extent. Their properties are different from Erdős-Rényi graphs [55], which are completely random, and at the same time different from periodic lattices [56, 57, 58, 59, 60, 61, 62, 63, 49, 64, 65]. Conditions of theorems of graph theory do not often hold in a rigorous sense, and hence we have to resort to some approximations.

Statistical physics has historically treated interactions of components that lie on a lattice with a translational symmetry and studied cooperative phenomena. Attempts to understand real materials have prompted physicists

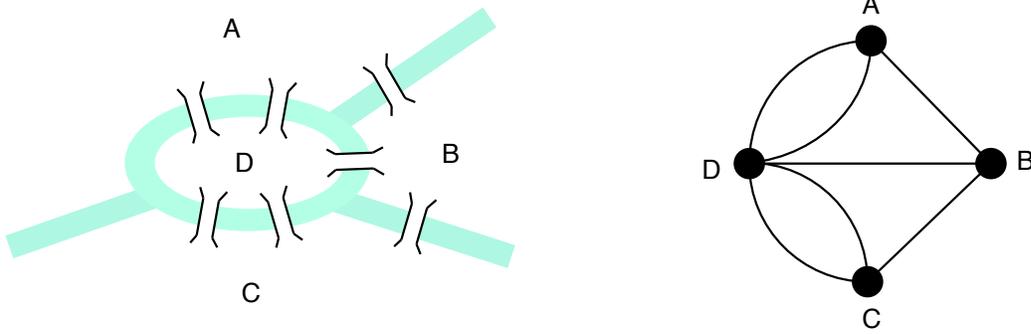


Figure 2.4: The Königsberg bridge problem. The city of Königsberg has seven bridges across rivers (left). The problem is to find a route to pass every bridge in the city once and only once. The left figure can be abstracted into the right figure by replacing each land and bridge with a node and an edge, respectively.

to develop numerous calculation techniques. Physicists have noticed that methodology of statistical mechanics is useful to understand networks, which have nothing to do with materials and had traditionally been thought to be outside the realm of physics.

2.2.3 Degree

The number of edges connected to a node i is called the degree of the node i and denoted as k_i . Let N be the total number of nodes and M be the total number of edges. We have

$$\sum_{i=1}^N k_i = 2M. \quad (2.7)$$

The average degree is

$$\langle k \rangle = \frac{1}{N} \sum_{i=1}^N k_i = \frac{2M}{N}. \quad (2.8)$$

We denote by $P(k)$ the probability that the degree of a randomly extracted node is k , which is called the degree distribution function. Using the degree distribution $P(k)$, we can rewrite the average degree as

$$\langle k \rangle = \sum_{k=0}^{\infty} kP(k). \quad (2.9)$$

In many real networks, degree distributions are power functions for large k as in $P(k) \propto k^{-a}$ with $a > 0$, which is often called the scale-free property.

2.2.4 Fractal complex networks

Only fractals embedded in the Euclidian spaces have been studied until recently. In order to consider fractals in a space without the Euclidian distance, we have to introduce another distance because the fractal is fundamentally associated with the question as to “how a volume grows as the system size increases”.

There are many choices of a distance in graphs, but there is no standard choice as in the Euclidian space. Throughout this thesis, we use *the shortest distance as the distance on graphs*.

The diameter of a graph is the largest of the shortest distances between all the pairs of two nodes. The mean shortest distance is the average over all pairs of nodes. Let N be the number of the nodes and L be a diameter. The fractal dimension of the graph d_f may be intuitively given by

$$N \propto L^{d_f}. \quad (2.10)$$

On the other hand, many real complex networks have a small-world property; the number of nodes and the mean shortest distances are related as

$$\langle l \rangle \propto \log N. \quad (2.11)$$

Because Eq. (2.11) is not the form of Eq. (2.10), it seems that most of real complex networks are not fractals at a glance.

Song *et al.* found that a few graphs in real networks are indeed fractal (Fig. 2.5) [11, 12]. They noticed that complex networks that had been studied many times were fractal, *i.e.*,

1. a part of the WWW composed of 325,729 web pages, which are connected if there is a URL link from one page to another;
2. a social network where the nodes are 392,340 actors, who are linked if they were cast together in at least one film;
3. the biological networks of protein-protein interactions found in *Escherichia coli* and *Homo sapiens*, where proteins are linked if there is a physical binding between them.

Song *et al.* [11] argued that because of the long-tail distribution of degrees of nodes the cluster dimension d_c and d_{BC} are not identical in scale-free networks.

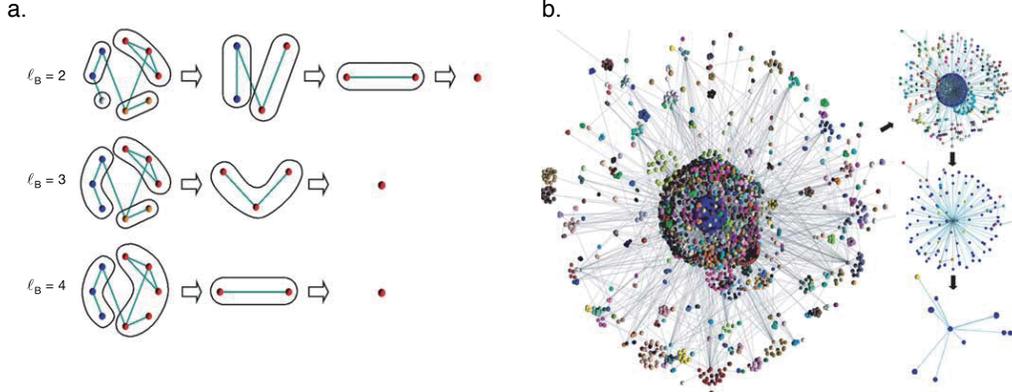


Figure 2.5: The renormalization-group procedure applied to a real complex network. Taken from Song *et al.* [11]. **a.** The box-covering method for a graph. We tile the graph with subgraphs whose diameter is less than l_B . Then we replace each subgraph with a single node; two renormalized nodes are connected if there is at least one edge between the subgraphs. Thus we obtain the network shown in the second column. The decimation is repeated until the graph is reduced to a single node. **b.** The renormalization is applied to the WWW network. The renormalized network is as complex as the original one. This indicates that the WWW network is a fractal.

2.2.5 The (u, v) -flower

After the discovery in the real networks, several artificial fractal complex networks have been devised [13]. One of such networks is the (u, v) -flower (Fig. 2.6) [14]. As deterministic fractals such as the Sierpinski gasket and the Cantor set helped us understand real fractals in the Euclidian spaces, deterministic fractal complex networks can deepen our understanding of fractal complex networks in the real world. As with many other artificial fractals, the (u, v) -flower is a graph with a hierarchical structure [66, 60].

The (u, v) -flower is defined in the following way. First, we prepare a cycle of length $u + v$ as the first generation. Second, given a graph of generation n , we obtain the $(n + 1)$ th generation by replacing each link by two parallel paths of length u and v . We can assume $1 \leq u \leq v$ without losing generality.

Let M_n and N_n be the numbers of edges and nodes, respectively. From the definition of the (u, v) -flower, it straightforwardly follows that

$$M_n = w^n, \quad (2.12)$$

$$N_n = wN_{n-1} - w = \dots = \frac{w-2}{w-1} \times w^n + \frac{w}{w-1}, \quad (2.13)$$

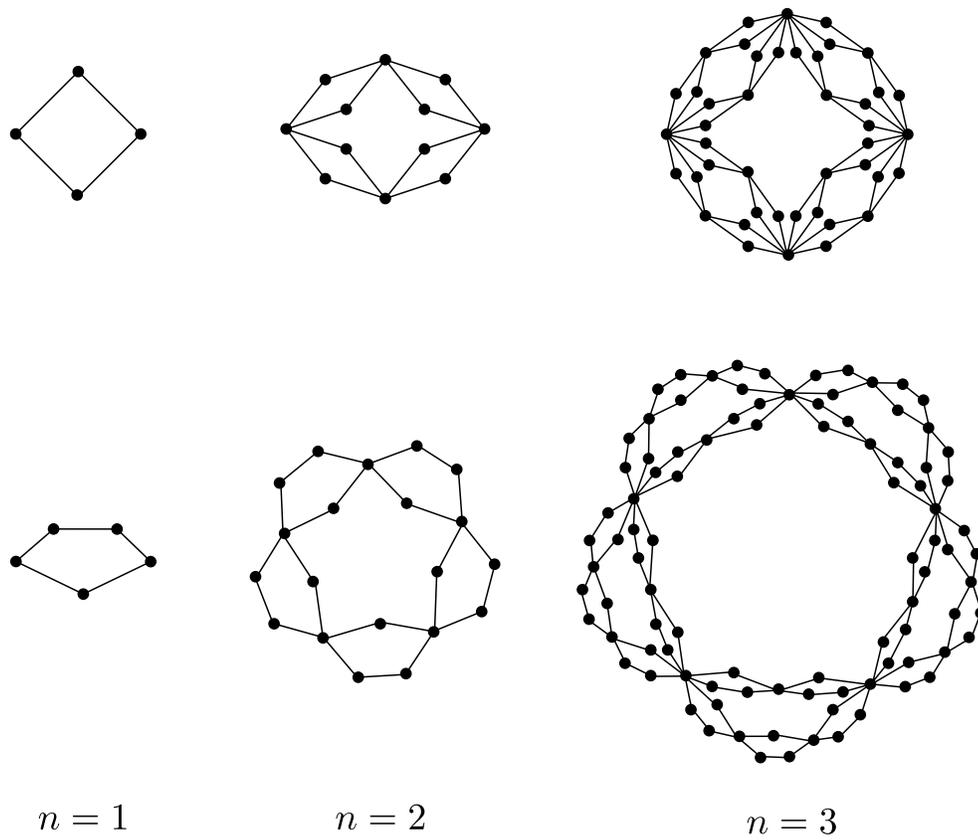


Figure 2.6: The (2,2)-flower and the (2,3)-flower in the first, second and third generations. Each line is replaced by parallel lines of length u and v in construction of the next generation.

where

$$w = u + v. \quad (2.14)$$

The mean degree of (u, v) -flower in the n th generation is

$$\langle k \rangle = \frac{2M_n}{N_n}. \quad (2.15)$$

Similar consideration tells us about the degree distribution. The (u, v) -flowers only have nodes of degree $k = 2^m$, where $m = 1, 2, \dots, n$. Let $N_m(m)$ be the number of nodes of degree 2^m in the n th generation. We thereby have

$$N_n(m) = \begin{cases} N_{n-1}(m-1) & \text{for } m > 1, \\ (w-2)w^{n-1} & \text{for } m = 1. \end{cases} \quad (2.16)$$

Solving this recurrence relation under the initial condition $N_1(1) = w$, we have

$$N_n(m) = \begin{cases} (w-2)w^{n-m} & \text{for } m < n, \\ w & \text{for } m = n, \end{cases} \quad (2.17)$$

which is related to the degree distribution $P(k)$ in the form $|N_n(m)dm| = |P(k)dk|$. We therefore have the degree distribution of the (u, v) -flower with $u, v \geq 1$ as

$$P(k) \propto k^{-\gamma} \quad \text{with } \gamma = 1 + \frac{\ln(u+v)}{\ln 2}. \quad (2.18)$$

The dimensionality of the (u, v) -flowers is totally different for $u = 1$ and $u > 1$ [49]. When $u = 1$ the diameter d_n of the n th generation is proportional to the generation n , while the diameter d_n is a power of u when $u > 1$:

$$d_n \sim \begin{cases} (v-1)n & \text{for } u = 1, \\ u^n & \text{for } u > 1. \end{cases} \quad (2.19)$$

Since $N_n \sim w^n$, we can transform Eq. (2.19) to

$$d_n \sim \begin{cases} \ln N_n & \text{for } u = 1, \\ N_n^{\ln u / \ln(u+v)} & \text{for } u > 1. \end{cases} \quad (2.20)$$

This means that the (u, v) -flowers have a small-world property only when $u = 1$, while the flowers have finite fractal dimensions for $u > 1$; see the definition of fractal dimension Eq. (2.10) and the small-world property Eq. (2.11).

When $u > 1$, it is clear from the construction of flowers that the similarity dimension of the (u, v) -flower is

$$d_{\text{sim}} = \frac{\ln(u+v)}{\ln u} \text{ for } u > 1. \quad (2.21)$$

Because the cluster dimension is an extension of the similarity dimension, the cluster dimension of the (u, v) -flower is the same as that of the similarity dimension for $u > 1$:

$$d_c = d_{\text{sim}} = \frac{\ln(u+v)}{\ln u} \text{ for } u > 1. \quad (2.22)$$

2.3 Self-avoiding walk

A self-avoiding path, which is called a simple path or just a path in graph theory, is a path on a lattice (graph) that is forbidden to visit the same point more than once [67]. This path is referred to as the self-avoiding path throughout this thesis in order to distinguish it from other stochastic processes.

Though the definition is quite easy, many important questions are still open in the Euclidian spaces even today [8]. For example,

1. How many possible self-avoiding paths of length k are there?
2. How long is the typical distance from the starting point?

The goal of the next chapter is to find a graph on which these questions are answered.

2.3.1 Self-avoiding walk in a Euclidian space

In a Euclidian space, the number of paths of length k , which is written as C_k , on \mathbb{R}^n is believed to behave as [8]

$$C_k \sim \mu^k k^{\gamma-1} \quad (2.23)$$

and the mean square distance of paths of length k , which is denoted as $\langle R_k^2 \rangle$, is hypothesized to be [8]

$$\langle R_k^2 \rangle \sim k^{2\nu}. \quad (2.24)$$

Here the sign \sim denotes the asymptotic form of the function as $k \rightarrow \infty$. The constant μ is called the connective constant, which roughly means the effective coordination number, *i.e.*, the number of nodes to which a walker

CHAPTER 2. REVIEW OF COMPLEX NETWORK AND SELF-AVOIDING WALK

can go next. The exponent γ is a critical exponent associated with the susceptibility and ν is one associated with the correlation length from the viewpoint of the correspondence between the self-avoiding walk and the \mathcal{N} -vector model. Thus, μ is sensitive to the specific form of the lattice, while γ and ν are universal quantities, that is, they are insensitive to the specific form of the lattice and are believed to depend only on the Euclidian dimension. The critical exponents are conjectured to be

$$\gamma = \begin{cases} \frac{43}{32} & \text{for } d = 2, \\ 1.162\dots & \text{for } d = 3, \\ 1 \text{ with a logarithmic correction} & \text{for } d = 4, \\ 1 & \text{for } d = 5, \end{cases} \quad (2.25)$$

$$\nu = \begin{cases} \frac{3}{4} & \text{for } d = 2, \\ 0.59\dots & \text{for } d = 3, \\ 1/2 \text{ with a logarithmic correction} & \text{for } d = 4, \\ 1/2 & \text{for } d = 5. \end{cases} \quad (2.26)$$

The upper critical dimension of the self-avoiding walk is $d = 4$, above which the critical exponents are given by a mean-field model. The mean-field model of the self-avoiding walk is the random walk, whose critical exponent ν is $1/2$ as is well known.

Going beyond the Euclidian dimension, the self-avoiding walk in fractal dimensions has also been actively studied since the 1980s [6, 9, 10]. It has been conjectured that the universality class of the self-avoiding walk of fractals are *not* determined just by a fractal dimension (precisely speaking the similarity dimension).

Physicists have tried to express the exponent ν by the similarity dimension as an extension of Flory's approximation in the Euclidian space [6, 9, 10].

$$\nu = \frac{3}{2+d} \quad \longrightarrow \quad \nu = \frac{3}{2+d_{\text{sim}}}. \quad (2.27)$$

They, however, found that replacement of the Euclidian dimension of Flory's approximation with the similarity dimension sometimes gives a deteriorated accuracy. It was concluded that there is no simple formula for a fractal as in the Euclidian space.

2.3.2 \mathcal{N} -vector model

This subsection describes the correspondence between the self-avoiding walk and a zero-component ferromagnet. The connection was first discovered by de Gennes [5, 4], and opened a way to study a polymer in terms of the standard theory of critical phenomena. Shapiro [68] introduced a generating function, whose divergence near a pole governs the behavior of the zero-component ferromagnet at the critical point. We here follow the discussion by Madras and Slade [8]. Their mapping of the self-avoiding walk to the \mathcal{N} -vector model is straightforward and can be directly applied to graphs as well as to usual lattices.

Assume that spins are on a graph $G = (V, E)$. The spins have \mathcal{N} components and the tip of each spin is on a sphere of radius $\sqrt{\mathcal{N}}$:

$$\mathbf{S}^{(x)} = (S_1^{(x)}, S_2^{(x)}, \dots, S_{\mathcal{N}}^{(x)}) \in \mathcal{S}(\mathcal{N}, \sqrt{\mathcal{N}}), \quad (2.28)$$

where $\mathcal{S}(m, r)$ is the sphere of radius r in \mathbb{R}^m :

$$\mathcal{S}(m, r) = \{(a_1, a_2, \dots, a_m) \in \mathbb{R}^m : a_1^2 + a_2^2 + \dots + a_m^2 = r^2\}. \quad (2.29)$$

We consider the Hamiltonian with a ferromagnetic interaction given by

$$H = - \sum_{\langle x, y \rangle} \mathbf{S}^{(x)} \cdot \mathbf{S}^{(y)}, \quad (2.30)$$

where x and y are nodes, and $\langle x, y \rangle$ is the edge connecting x and y . The sum runs over all edges. The expectation value of any quantity A is

$$\langle A \rangle = \frac{1}{Z} E(Ae^{-\beta H}) \quad (2.31)$$

with

$$Z = E(e^{-\beta H}), \quad (2.32)$$

where $E(\cdot)$ is the expectation value with respect to the product of the uniform measures on $\mathcal{S}(\mathcal{N}, \sqrt{\mathcal{N}})$.

The quantity of our interest is the correlation function in the limit $\mathcal{N} \rightarrow 0$:

$$\lim_{\mathcal{N} \rightarrow 0} \langle \mathbf{S}_i^{(x)} \cdot \mathbf{S}_j^{(y)} \rangle. \quad (2.33)$$

The limit $\mathcal{N} \rightarrow 0$ is an extrapolation and not a mathematically justified procedure. We therefore have to explain its meaning. The limit should be

CHAPTER 2. REVIEW OF COMPLEX NETWORK AND SELF-AVOIDING WALK

defined so as to be consistent with the following lemma [8]:

Fix an integer $\mathcal{N} \geq 1$. Let $\mathbf{S} = (S_1, S_2, \dots, S_{\mathcal{N}})$ denote a vector which is uniformly distributed on $\mathcal{S}(\mathcal{N}, \sqrt{\mathcal{N}})$. Given nonnegative integers $k_1, \dots, k_{\mathcal{N}}$,

$$E(S_1^{k_1} S_2^{k_2} \dots S_{\mathcal{N}}^{k_{\mathcal{N}}}) = \begin{cases} \frac{2\Gamma(\frac{\mathcal{N}+2}{2}) \prod_{l=1}^{\mathcal{N}} \Gamma(\frac{k_l+1}{2})}{\pi^{\mathcal{N}/2} \Gamma(\frac{k_1+\dots+k_{\mathcal{N}}+\mathcal{N}}{2})} \mathcal{N}^{(k_1+\dots+k_{\mathcal{N}}-2)/2} & \text{when all } k_l \text{ are even,} \\ 0 & \text{otherwise.} \end{cases} \quad (2.34)$$

□

We can prove it by mathematical induction.

We define the limit $\mathcal{N} \rightarrow 0$ in the following way. First, the following trivial equality holds:

$$E(1) = 1. \quad (2.35)$$

Second, since $E(S_1^2 + \dots + S_{\mathcal{N}}^2) = \mathcal{N}$, it follows from the symmetry that

$$E(S_i^2) = 1. \quad (2.36)$$

Third, when $k_1 + \dots + k_{\mathcal{N}} > 2$, the exponent of $\mathcal{N}^{(k_1+\dots+k_{\mathcal{N}}-2)/2}$ is positive. For these three reasons, we define the limit $\mathcal{N} \rightarrow 0$ as follows.

$$\lim_{\mathcal{N} \rightarrow 0} E(S_1^{k_1} S_2^{k_2} \dots S_{\mathcal{N}}^{k_{\mathcal{N}}}) = \begin{cases} 1 & \text{all } k_l = 0, \text{ or one } k_l = 2 \text{ and } k_j = 0 \ (j \neq l), \\ 0 & \text{otherwise.} \end{cases} \quad (2.37)$$

In order to evaluate Eq. (2.32), we expand the Boltzmann factor in the following power series:

$$e^{-\beta H} = \prod_{\langle x,y \rangle} \exp[\beta \mathbf{S}^{(x)} \cdot \mathbf{S}^{(y)}] = \prod_{\langle x,y \rangle} \sum_{m_{xy}=0}^{\infty} \frac{\beta^{m_{xy}}}{m_{xy}!} (\mathbf{S}^{(x)} \cdot \mathbf{S}^{(y)})^{m_{xy}}. \quad (2.38)$$

Let us label the edges as $e_1, \dots, e_{|E|}$. In this notation, Eq. (2.38) can be rewritten as

$$e^{-\beta H} = \sum_{m_1, \dots, m_{|E|}=0}^{\infty} \frac{\beta^{\sum_{\alpha \in E} m_{\alpha}}}{\prod_{\alpha \in E} m_{\alpha}!} \prod_{\alpha \in E} (\mathbf{S}^{(e_{\alpha}^-)} \cdot \mathbf{S}^{(e_{\alpha}^+)})^{m_{\alpha}}. \quad (2.39)$$

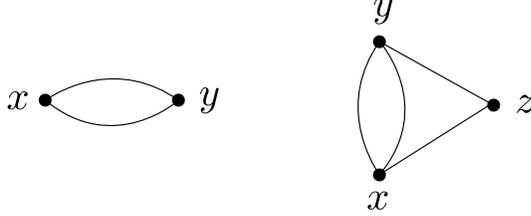


Figure 2.7: Examples of the graphical representation of terms in Eq. (2.40). The left diagram corresponds to the term $E((\mathbf{S}^{(x)} \cdot \mathbf{S}^{(y)})^2)$ and is called a two-edge polygon. The right diagram represents the term $E((\mathbf{S}^{(x)} \cdot \mathbf{S}^{(y)})^2 (\mathbf{S}^{(y)} \cdot \mathbf{S}^{(z)}) (\mathbf{S}^{(x)} \cdot \mathbf{S}^{(z)}))$.

Consider now the partition function

$$Z = \sum_{m_1, \dots, m_{|E|=0}}^{\infty} \frac{\beta^{\sum_{\alpha \in E} m_{\alpha}}}{\prod_{\alpha \in E} m_{\alpha}!} E \left(\prod_{\alpha \in E} (\mathbf{S}^{(e_{\alpha}^-)} \cdot \mathbf{S}^{(e_{\alpha}^+)})^{m_{\alpha}} \right). \quad (2.40)$$

A graphical interpretation of the sum in Eq. (2.40) can be obtained by associating to each term in the sum a graph whose each edge e_{α} is duplicated m_{α} times (if $m_{\alpha} = 0$, then it means that the edge is removed) (Fig. 2.7). It follows from Eq. (2.37) that any term whose corresponding graph has a node from which other than two or zero edges emanate will approach zero in the limit as $\mathcal{N} \rightarrow 0$. Therefore, the only terms which may contribute are one with no edges and ones with self-avoiding polygons.

A two-edge polygon with nearest-neighbor nodes x, y (Fig. 2.7, left) contributes the amount

$$\frac{\beta^2}{2} E((\mathbf{S}^{(x)} \cdot \mathbf{S}^{(y)})^2) = \frac{\beta^2}{2} \mathcal{N}. \quad (2.41)$$

Thus, the two-edge polygon is irrelevant in the limit $\mathcal{N} \rightarrow 0$. A non-degenerate polygon, in other words a polygon consisting of at least three edges, also does not contribute according to a similar argument. The only term which is relevant in Eq. (2.40) is a graph with no edges. We therefore have

$$\lim_{\mathcal{N} \rightarrow 0} Z = 1. \quad (2.42)$$

For the correlation function, the analysis is similar. We would like to compute the limit $\mathcal{N} \rightarrow 0$ of the expectation value for $x \neq y$:

$$\sum_{m_1, \dots, m_{|E|=0}}^{\infty} \frac{\beta^{\sum_{\alpha \in E} m_{\alpha}}}{\prod_{\alpha \in E} m_{\alpha}!} E \left(S_i^{(x)} S_j^{(y)} \prod_{\alpha \in E} (\mathbf{S}^{(e_{\alpha}^-)} \cdot \mathbf{S}^{(e_{\alpha}^+)})^{m_{\alpha}} \right). \quad (2.43)$$

CHAPTER 2. REVIEW OF COMPLEX NETWORK AND SELF-AVOIDING WALK

Terms corresponding to graphs with self-avoiding polygons do not contribute because of the same reason. The only surviving terms are ones with self-avoiding paths from x to y . Contribution due to the self-avoiding path $(x, v_1, \dots, v_{k-1}, y)$ is

$$\beta^k E(S_i^{(x)}(\mathbf{S}^{(x)} \cdot \mathbf{S}^{(v_1)})(\mathbf{S}^{(v_1)} \cdot \mathbf{S}^{(v_2)}) \dots (\mathbf{S}^{(v_{k-1})} \cdot \mathbf{S}^{(y)}) S_j^{(y)}) = \beta^k \delta_{i,j}. \quad (2.44)$$

All the contributing terms can be summed using the generating function of the s - t paths connecting nodes s and t :

$$G_z(s, t) := \sum_{\omega: s \rightarrow t} z^{|\omega|}. \quad (2.45)$$

Here ω is a simple path from s to t , and $|\omega|$ denotes the length of the path ω . The generating function $G_z(s, t)$ is often called the two-point function.

Using the generating function $G_z(x, y)$ and (2.42), we have

$$\lim_{\mathcal{N} \rightarrow 0} \langle \mathbf{S}_i^{(x)} \cdot \mathbf{S}_j^{(y)} \rangle = \sum_{m_1, \dots, m_{|E|} = 0}^{\infty} \frac{\beta^{\sum_{\alpha \in E} m_{\alpha}}}{\prod_{\alpha \in E} m_{\alpha}!} E \left(S_i^{(x)} S_j^{(y)} \prod_{\alpha \in E} (\mathbf{S}^{(e_{\alpha}^-)} \cdot \mathbf{S}^{(e_{\alpha}^+)})^{m_{\alpha}} \right) \quad (2.46)$$

$$= \delta_{i,j} \sum_{\omega: x \rightarrow y} \beta^{|\omega|} = \delta_{i,j} G_{\beta}(x, y). \quad (2.47)$$

Now, the correspondence between the self-avoiding walk and the zero-component ferromagnet is established:

$$\lim_{\mathcal{N} \rightarrow 0} \langle \mathbf{S}_i^{(x)} \cdot \mathbf{S}_j^{(y)} \rangle = \delta_{i,j} G_{\beta}(x, y). \quad (2.48)$$

This relation holds on any graphs as well as on usual lattices.

Chapter 3

Self-avoiding walk on fractal complex network

The content of this chapter is based on Ref. [69]. This chapter is organized as follows. In Sec. 3.1, we define an ensemble of paths of a fixed length, the connective constant and the displacement exponent of the self-avoiding walk on a graph. In Sec. 3.2, we extend the theory of the self-avoiding walk in the Euclidian spaces [68, 7, 6] to the graphs of the (u, v) -flower, on which the shortest distance is used as a distance instead of the Euclidian distance. By conducting a renormalization-group analysis, we calculate the connective constant μ and the critical exponent ν , which is associated with the correlation length of the zero-component ferromagnet on the (u, v) -flowers with $\forall u, v \geq 2$. We thereby write down the critical exponent ν in *arbitrary* fractal dimensions greater than unity. We also compare the results with a tree approximation, which is usually referred to as a mean-field approximation in the context of the study of complex networks. In Sec. 3.3, we exactly obtain the two-point function for the (u, u) -flower and prove that $\nu = 1$ regardless of the fractal dimension between one and two. In Sec. 3.4, we carry out numerical simulations of the self-avoiding walk and observe that the number of paths starting from a hub s behaves as in $C_k^{(s)} \propto \mu^k k^{\gamma-1}$ and that the mean shortest distance between the starting point and the end point increases as $\overline{d_k^{(s)}} \approx k^\nu$, where k is the length of a path and γ is the critical exponent associated with the susceptibility of the mapped zero-component ferromagnet.

3.1 Ensemble of fixed length paths

In the model of the self-avoiding walk, all the paths of the same length starting from a fixed node without self-intersection appear with the same probability. In order to describe this accurately, let us define an ensemble. Let $G = (V, E)$ be a connected finite graph. A path of length k is defined as

$$\omega = (\omega_0, \omega_1, \dots, \omega_k), \quad \omega_i \in V, \quad (3.1)$$

$$\omega_i \neq \omega_j \text{ for } i \neq j, \quad (3.2)$$

$$(\omega_i, \omega_{i+1}) \in E. \quad (3.3)$$

Let us denote by $\Omega_k^{(s)}$ the set of the paths of length k for a fixed starting node and a free end node.

In order to consider the typical end-to-end distance, we define a probability distribution. Fixing a path length k and a starting node s , we introduce a uniform measure such that

$$P(\omega) = \frac{1}{\#\Omega_k^{(s)}}, \quad \forall \omega \in \Omega_k^{(s)}. \quad (3.4)$$

Though this is not a serious problem, note that when the graph is too small, it may not contain a path of length k and hence $\#\Omega_k^{(s)} = 0$.

In order to discuss the speed of diffusion on a graph, we next define a distance on a graph. For any nodes $v_1, v_2 \in V$, we let $d(v_1, v_2)$ denote the shortest distance (the length of the shortest path(s)) between the nodes v_1 and v_2 . The mean shortest distance between the ends of paths whose length is k and which start from node s is then given by

$$\overline{d_k^{(s)}} := \frac{\sum_{(\omega_0, \omega_1, \dots, \omega_k) \in \Omega_k^{(s)}} d(\omega_0, \omega_k)}{\#\Omega_k^{(s)}}, \quad (3.5)$$

where we took the average over the uniform distribution (3.4). We define the connective constant μ and the displacement exponent ν as

$$\mu := \lim_{k \rightarrow \infty} \left(\#\Omega_k^{(s)} \right)^{1/k}, \quad (3.6)$$

$$\nu := \lim_{k \rightarrow \infty} \frac{\ln \overline{d_k^{(s)}}}{\ln k}. \quad (3.7)$$

Here we assumed that $\#\Omega_k^{(s)}$ increases as the product of an exponential μ^k and a power function k^ν and that $\overline{d_k^{(s)}}$ increases as a power function k^ν in the same form as for the self-avoiding walk in \mathbb{R}^n . Thus, by obtaining μ and ν , we can tell the number of paths on the graph and the typical distance from the starting point. The goal of this chapter is to calculate these two quantities on graphs in arbitrary fractal dimensions.

3.2 Exact renormalization-group analysis of the general (u, v) -flowers

In this section, we first define a two-point function and apply an exact renormalization to it. We thus obtain the connective constant μ and the critical exponent ν of the zero-component ferromagnet. Next we consider the mean-field theory of the self-avoiding walk and compare the result with the prediction from the exact renormalization.

3.2.1 Generating function

Let O and R be nodes which are separated by a distance u in the first generation (Fig. 2.6) and r_n be the shortest distance between O and R in the n th generation. The nodes O and R have the largest degree and hence are called hubs. Because each edge is replaced by two parallel lines of length u and v in the construction, the shortest distance r_n increases as

$$r_n = r_{n-1} \times u = \cdots = u^{n-1} r_1 = u^n. \quad (3.8)$$

Defining $C_k^{(n)}(R)$ as the number of self-avoiding paths of length k starting from the node O and ending at the node R in the n th generation, we can construct the two-point function as

$$G_n(x) = \sum_{k=1}^{\infty} C_k^{(n)}(R) x^k. \quad (3.9)$$

The correspondence between the self-avoiding walk and the \mathcal{N} -vector model [8] tells us that x corresponds to the inverse temperature β of the ferromagnet and that the two-point function is the correlation function of the \mathcal{N} -component spins placed at O and R in the limit of $\mathcal{N} \rightarrow 0$:

$$\lim_{\mathcal{N} \rightarrow 0} \langle S_i^{(O)} S_j^{(R)} \rangle = \delta_{ij} G_n(\beta), \quad (3.10)$$

where $S_i^{(v)}$ denotes the i th component of the spin at a node v .

This suggests that the two-point function becomes in the thermodynamic limit

$$G_n(x) \sim \exp(-r_n/\xi(x)) \quad \text{as } n \rightarrow \infty, \quad (3.11)$$

where $\xi(x)$ is the correlation length, which should behave as [8]

$$\xi(x) \sim (x_c - x)^{-\nu} \quad \text{as } x \nearrow x_c, \quad (3.12)$$

3.2. EXACT RENORMALIZATION-GROUP ANALYSIS OF THE GENERAL (U, V) -FLOWERS

where x_c is a critical point. The critical point x_c is therefore equal to the ferromagnetic transition temperature β_c .

Let us assume that $C_k^{(n)}(R)$ in Eq. (3.9) behaves asymptotically as

$$C_k^{(n)}(R)^{1/k} \sim \mu, \quad (3.13)$$

because at each step a walker has μ options to go next on average. Under the assumption of Eq. (3.13), the convergence disk of (3.9) is $|x| < 1/\mu$. We thereby identify x_c as $1/\mu$. In the theory of the self-avoiding walk in the Euclidian space, it is conjectured that the exponents ν defined in Eqs. (3.7) and (3.12) are the same, using the discussion of the scaling theory [8]. For the same reason, we expect that the two exponents are also equivalent in our case. However, the conjecture has never been proved in the Euclidian space, nor can we rigorously prove the equivalence for the (u, v) -flower. Furthermore, we do not *a priori* know whether we can use the shortest distance for a distance on the (u, v) -flower for scaling theory. We assume that μ and ν defined in the two different ways are equal on the (u, v) -flower too. We will check the validity of this assumption in Sec. 3.4.

We can calculate the two-point function in the following way. The two-point function of the first generation is $G_1(x) = x^u + x^v$ by definition. Since the $(n + 1)$ th generation can be regarded as a cycle of $(u + v)$ pieces of the n th generation graphs, we have

$$G_{n+1}(x) = G_n(x)^u + G_n(x)^v. \quad (3.14)$$

The diagrammatic representation of Eq. (3.14) is shown in Fig. 3.1. Therefore, we obtain

$$G_{n+1}(x) = G_1(G_n(x)). \quad (3.15)$$

Repeated use of this relation yields

$$G_n(x) = \underbrace{G_1 \circ G_1 \circ \cdots \circ G_1}_n(x). \quad (3.16)$$

3.2.2 Renormalization-group analysis

We define a renormalization procedure for the (u, v) -flower as the inverse transformation of the constructing procedure of the flower (Fig. 3.2). When the $(n + 1)$ th generation is given, we coarse-grain the minute structure and obtain the n th generation. Every cycle of length $(u + v)$ is therefore replaced

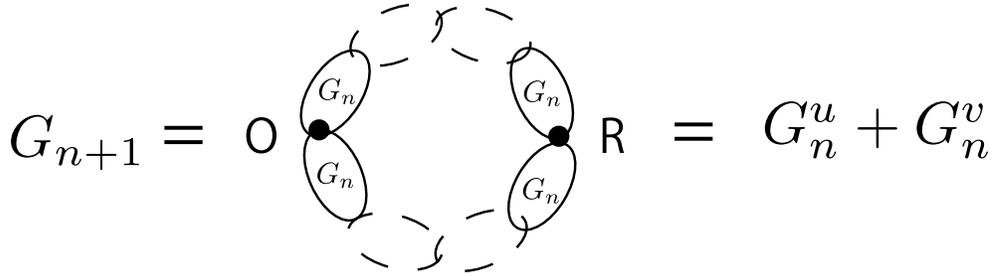


Figure 3.1: Diagrammatic representations of the two-point function $G_n(x)$. The $(n+1)$ th generation can be regarded as a cycle of $(u+v)$ pieces of graphs in the n th generation.

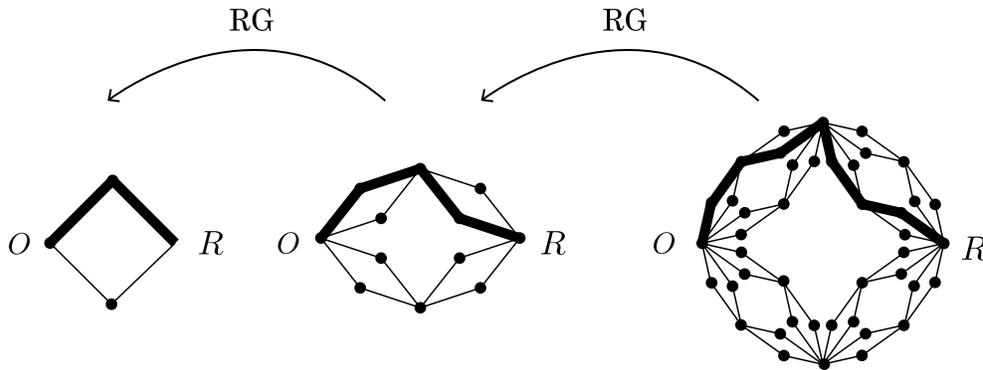


Figure 3.2: An example of renormalization of a self-avoiding path on the $(2, 2)$ -flower. The decimation is carried out by erasing a smaller structure.

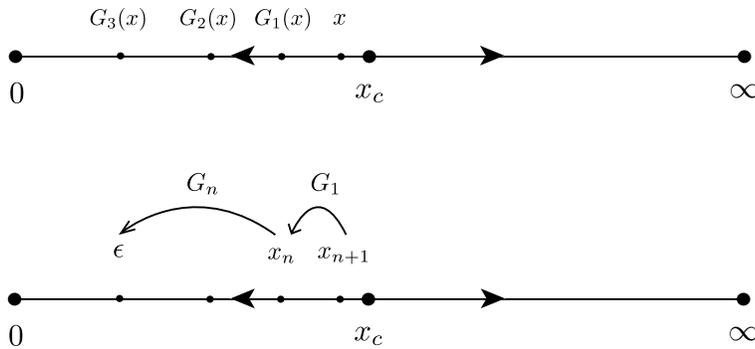


Figure 3.3: The renormalization-group flow. The top figure illustrates how $G_n(x)$ changes as the generation n gets larger with x fixed. The bottom figure shows the flow of the scaling variable x . Here, x_{n+1} is the scaling variable in the $(n+1)$ th flower and x_n is the one in a coarse-grained flower.

3.2. EXACT RENORMALIZATION-GROUP ANALYSIS OF THE GENERAL (U, V) -FLOWERS

by a single edge. Renormalization of self-avoiding paths is also defined in a similar way.

Let ϵ be a sufficiently small positive number. We define the variable x_n such that

$$G_n(x_n) := \epsilon \text{ for all } n. \quad (3.17)$$

The variable x_n is the scaling variable of our theory; we observe how it transforms under the renormalization transformation (Fig. 3.3). We will prove in Sec. 3.2.3 the unique existence of x_n which satisfies Eq. (3.17).

The two-point function of the $(n + 1)$ th generation and that of the n th generation are related as

$$G_{n+1}(x_{n+1}) = \epsilon = G_n(x_n). \quad (3.18)$$

This specifies how the scaling variable x is renormalized. From Eqs. (3.16) and (3.18), we obtain

$$G_n(x_n) = G_{n+1}(x_{n+1}) = G_n(G_1(x_{n+1})) = G_n(x_{n+1}^u + x_{n+1}^v). \quad (3.19)$$

The scaling variable therefore changes under the renormalization transformation as

$$x_n = x_{n+1}^u + x_{n+1}^v. \quad (3.20)$$

We will show in Sec. 3.2.3 that the scaling variable x_n changes as shown in Fig. 3.3. The nontrivial fixed point x_c is given by

$$x_c = x_c^u + x_c^v. \quad (3.21)$$

Next, we obtain the critical exponent ν by studying $\xi(x)$ near the fixed point x_c . From Eqs. (3.11) and (3.19) we should have

$$\frac{r_{n+1}}{\xi(x_{n+1})} = \frac{r_n}{\xi(x_n)}, \quad (3.22)$$

and hence

$$(x_c - x_{n+1})^{-\nu} \sim \frac{r_{n+1}}{r_n} (x_c - x_n)^{-\nu} = u(x_c - x_n)^{-\nu}. \quad (3.23)$$

The Taylor expansion enables us to express ν in terms of x_c :

$$\begin{aligned} x_n - x_c &= x_{n+1}^u + x_{n+1}^v - x_c \\ &\approx x_c^u + ux_c^{u-1}(x_{n+1} - x_c) + x_c^v + vx_c^{v-1}(x_{n+1} - x_c) - x_c \\ &= (ux_c^{u-1} + vx_c^{v-1})(x_{n+1} - x_c), \end{aligned} \quad (3.24)$$

$$\nu = \frac{\ln(u)}{\ln(ux_c^{u-1} + vx_c^{v-1})}. \quad (3.25)$$

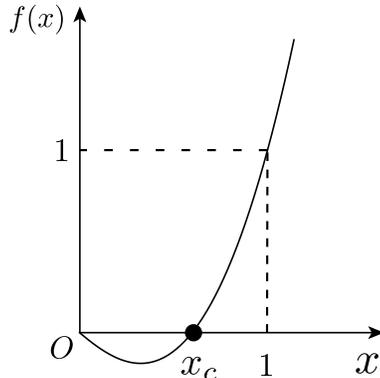


Figure 3.4: The function $f(x) = x^u + x^v - x$, which has a zero point between 0 and 1.

Since Eq. (3.21) cannot be solved explicitly in general, we will rely on a numerical solver when we compare Eq. (3.25) with the value of numerical simulation in Sec. 3.4.

3.2.3 Existence and uniqueness of a nontrivial fixed point

In the above argument, we assumed the existence of a positive fixed point x_c satisfying Eq. (3.21) and the solution x_n which meets Eq. (3.17). We here prove the existence and the uniqueness of $x_c > 0$ and that of x_n as follows.

Let us study how the scaling variable x changes under the renormalization-group equation (3.20). We define the difference of a scaling variable in the original system and a coarse-grained system as

$$f(x) := x^u + x^v - x. \quad (3.26)$$

Because $2 \leq u \leq v$,

$$f(0) = 0, \quad f(1) = 1, \quad f'(0) < 0, \quad (3.27)$$

$$f''(x) = u(u-1)x^{u-2} + v(v-1)x^{v-2} > 0 \quad \text{for } x > 0. \quad (3.28)$$

Therefore, there exists exactly one positive number x_c which satisfies $0 < x_c < 1$ and $f(x_c) = 0$ (Fig. 3.4). In other words, the renormalization-group equation of the self-avoiding walk on the (u, v) -flower has exactly one nontrivial fixed point for $2 \leq u, v$. Under the assumption of Eq. (3.13), on the other hand, we have $\mu = 1/x_c$. We thereby arrive at

$$\mu = \frac{1}{x_c} > 1. \quad (3.29)$$

3.2. EXACT RENORMALIZATION-GROUP ANALYSIS OF THE GENERAL (U, V) -FLOWERS

This result is natural because μ means the effective coordination number. If μ were smaller than unity, a walker would quickly come to a dead end and a path could not spread out.

Next, we show using mathematical induction that $G_n(x)$ is a monotonically increasing function of x in $x > 0$ for $\forall n \in \mathbb{N}$ and that $G_n(x)$ satisfies $G_n(0) = 0$ and $G_n(x_c) = x_c$.

(i) For $n = 1$, we have

$$\frac{dG_1}{dx}(x) = ux^{u-1} + vx^{v-1} > 0 \text{ for } x > 0, \quad (3.30)$$

$$G_1(x_c) = x_c^u + x_c^v = x_c, \quad (3.31)$$

$$G_1(0) = 0. \quad (3.32)$$

Therefore, the statement is true for $n = 1$.

(ii) Suppose that the statement is true for $G_n(x)$. We first prove the monotonicity of $G_{n+1}(x)$, which is given by

$$G_{n+1}(x) = G_n(G_1(x)). \quad (3.33)$$

Since both G_1 and G_n are monotonically increasing functions, the composition of G_n and G_1 is also a monotonically increasing function. Furthermore, we have

$$G_{n+1}(x_c) = G_n(G_1(x_c)) = G_n(x_c) = x_c, \quad (3.34)$$

$$G_{n+1}(0) = G_n(G_1(0)) = G_n(0) = 0. \quad (3.35)$$

Therefore, the statement is also satisfied for G_{n+1} .

Together with the continuity of $G_n(x)$, we have now proved the unique existence of $x_n \in (0, x_c)$ which satisfies Eq. (3.17) for an arbitrary constant $\epsilon \in (0, x_c)$.

3.2.4 Range of ν

We can study the range of the critical exponent ν in Eq. (3.25) by using inequalities. We define x_c as the unique positive solution of Eq. (3.21) from now on:

$$x_c^{u-1} + x_c^{v-1} = 1, \quad 2 \leq u \leq v. \quad (3.36)$$

First, we can obtain the upper bound of ν as

$$\nu = \frac{\ln(u)}{\ln(ux_c^{u-1} + vx_c^{v-1})} \leq \frac{\ln(u)}{\ln(ux_c^{u-1} + ux_c^{v-1})} = \frac{\ln(u)}{\ln(u(x_c^{u-1} + x_c^{v-1}))} = 1. \quad (3.37)$$

The equality holds iff $u = v$.

We next bound ν from below. Since $0 < x_c < 1$, we have

$$\nu \geq \frac{\ln(u)}{\ln(ux_c^{u-1} + vx_c^{u-1})}. \quad (3.38)$$

Since $1 = x_c^{u-1} + x_c^{v-1} \leq 2x_c^{u-1}$, $x_c \geq (1/2)^{1/(u-1)}$. The denominator of the above equation satisfies

$$\ln(ux_c^{u-1} + vx_c^{u-1}) \geq \ln\left(\frac{1}{2}(u+v)\right) \geq \ln u > 0 \quad (3.39)$$

Therefore, $\nu > 0$.

According to its definition (3.7), the exponent ν should satisfy $0 \leq \nu \leq 1$, which is consistent with the prediction of the renormalization-group analysis above.

3.2.5 Exact results

As we noted previously, the solution of Eq. (3.21) cannot be written down explicitly in general. There are, however, exceptional cases where we can obtain x_c , μ , and ν explicitly.

First for the (u, u) -flower, Eq. (3.21) gives

$$x_c = 2^{-\frac{1}{u-1}}, \quad (3.40)$$

from which we obtain

$$\mu = \frac{1}{x_c} = 2^{\frac{1}{u-1}}, \quad (3.41)$$

$$\nu = \frac{\ln(u)}{\ln(u)} = 1. \quad (3.42)$$

We will see in Sec.3.3 that this result coincides with the exact solution which we will derive without using the renormalization-group analysis.

3.2. EXACT RENORMALIZATION-GROUP ANALYSIS OF THE GENERAL (U, V) -FLOWERS

Next for the $(u, 2u - 1)$ -flower, by setting $y = x_c^{u-1}$, we can reduce Eq. (3.21) to the quadratic equation $y^2 + y - 1 = 0$, which yields $y = (-1 + \sqrt{5})/2$ because $y > 0$, and then

$$x_c = \left(\frac{-1 + \sqrt{5}}{2} \right)^{\frac{1}{u-1}}. \quad (3.43)$$

We thereby obtain

$$\mu = \frac{1}{x_c} = \left(\frac{-1 + \sqrt{5}}{2} \right)^{\frac{-1}{u-1}}, \quad (3.44)$$

$$\nu = \frac{\ln(u)}{\ln\left(\frac{5-\sqrt{5}}{2}u + \frac{-3+\sqrt{5}}{2}\right)}. \quad (3.45)$$

In this case, ν is a monotonically increasing function of u and converges to unity in the limit of $u \rightarrow \infty$.

3.2.6 Comparison to the mean-field theory

Let us compare our analytic expressions with mean-field theory. A tree approximation is usually referred to as a mean-field theory when we discuss stochastic processes on complex networks. Under the mean-field approximation, the (u, v) -flower is approximated to a tree whose nodes have the same degree as the mean degree of the original flower.

Let M_n and N_n be the numbers of edges and nodes, respectively. From the definition of the (u, v) -flower, it straightforwardly follows that

$$M_n = w^n, \quad (3.46)$$

$$N_n = wN_{n-1} - w = \dots = \frac{w-2}{w-1} \times w^n + \frac{w}{w-1}, \quad (3.47)$$

where $w = u + v$.

The self-avoiding walk on this tree is identical with the random walk with an immediate return being forbidden (namely, the non-reversal random walk) [70, 71]. Since the connective constant μ is the effective coordination number, the tree approximation gives

$$\mu_{\text{MF}} = \langle k \rangle - 1 = \frac{2M_n}{N_n} - 1 \xrightarrow{n \rightarrow \infty} \frac{u+v}{u+v-2}. \quad (3.48)$$

Since we have neglected loops, the expression in Eq. (3.48) is expected to overestimate the value of μ ; a walker may encounter a visited site on a graph

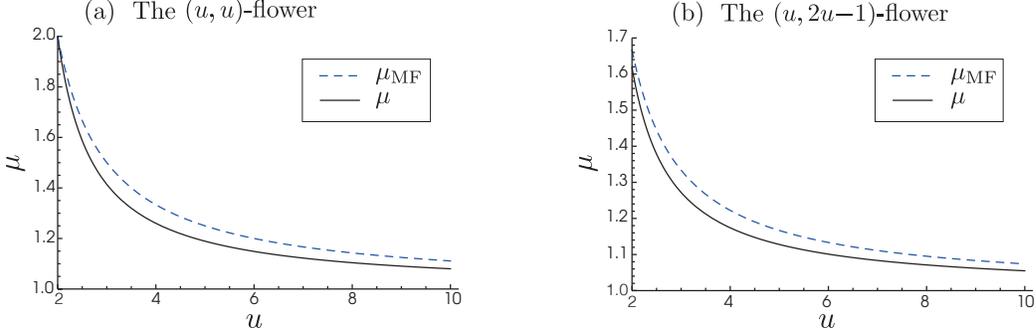


Figure 3.5: Comparison of the connective constants in the mean-field theory and the renormalization-group analysis. The mean-field estimate μ_{MF} always overestimates the true connective constant μ .

with loops, and hence the effective coordination number μ should be smaller compared to a tree with the same average degree. We confirm that our expectation is correct both analytically and numerically. In this section, we first show analytic results (Fig. 3.5). We will explain our numerical simulation in Sec. 3.4.

For the (u, u) -flower, we obtain from Eqs. (3.41) and (3.48)

$$\mu_{\text{MF}} = 1 + \frac{1}{u-1} \geq \mu = 2^{\frac{1}{u-1}}. \quad (3.49)$$

For the $(u, 2u-1)$ -flower, we obtain from Eqs. (3.44) and (3.48)

$$\mu_{\text{MF}} = 1 + \frac{2}{3u-3} > \mu = \left(\frac{\sqrt{5}+1}{2} \right)^{\frac{1}{u-1}}. \quad (3.50)$$

3.3 Exact solution of the (u, u) -flower

We can indeed obtain the exact solution for the (u, u) -flower without relying on the renormalization-group analysis in Sec. 3.2. Using Eq. (3.14) repeatedly, we obtain

$$G_1(x) = x^u + x^u = 2x^u, \quad (3.51)$$

$$G_2(x) = G_1(x)^u + G_1(x)^u = 2^{u+1}x^{u^2}, \quad (3.52)$$

$$G_3(x) = G_2(x)^u + G_2(x)^u = 2^{u^2+u+1}x^{u^3}, \quad (3.53)$$

$$\dots \quad (3.54)$$

$$G_n(x) = 2^{u^{n-1}+u^{n-2}+\dots+1} = 2^{\frac{u^n-1}{u-1}}x^{u^n}, \quad (3.55)$$

3.3. EXACT SOLUTION OF THE (U, U) -FLOWER

which are cast into the form

$$\exp\left(-\frac{r_n}{\xi(x)}\right) = G_n(x) = 2^{\frac{u^n-1}{u-1}} x^{u^n}, \quad (3.56)$$

with

$$\xi(x) = -\frac{r_n}{\ln\left(2^{\frac{u^n-1}{u-1}} x^{u^n}\right)} = -\frac{u^n}{\ln\left(2^{\frac{u^n-1}{u-1}} x^{u^n}\right)}. \quad (3.57)$$

Let $x_c^{(n)}$ be

$$x_c^{(n)} := 2^{\frac{-1+u^{-n}}{u-1}}. \quad (3.58)$$

We then have $0 < \xi(x) < \infty$ when $0 < x < x_c^{(n)}$ and $\xi(x)$ diverges as $x \nearrow x_c^{(n)}$. The Taylor expansion around $x_c^{(n)}$ gives

$$\xi(x) = \frac{2^{\frac{-1+u^{-n}}{u-1}}}{x_c^{(n)} - x + O((x_c^{(n)} - x)^2)}. \quad (3.59)$$

In the thermodynamic limit $n \rightarrow \infty$, we arrive at

$$\lim_{n \rightarrow \infty} x_c^{(n)} = 2^{\frac{-1}{u-1}} =: x_c, \quad (3.60)$$

$$\xi(x) \xrightarrow{n \rightarrow \infty} \frac{2^{\frac{-1}{u-1}}}{x_c - x + O((x_c - x)^2)}. \quad (3.61)$$

The latter yields $\nu = 1$. Therefore, we arrive at the same result as Eqs. (3.40) and (3.42). The critical point $x_c^{(n)}$ in Eq. (3.58) is shifted from x_c because of a finite-size effect. This effect disappears when the system size becomes infinite as $n \rightarrow \infty$ and the critical point reaches the correct value in Eq. (3.60) in the thermodynamic limit.

In this section, we have proved that *the critical exponent ν of the self-avoiding walk on the (u, u) -flower is $\nu = 1$ for $\forall u > 1$* . On the other hand, the fractal dimension of the (u, u) -flower is $d_f = \ln(2u)/\ln(u)$, which takes a value in the range $1 < d_f \leq 2$. We therefore confirm that *there is no one-to-one correspondence between the fractal dimension and the critical exponent ν* .

The critical exponents of the self-avoiding walk in the Euclidian space are considered to be determined only by the dimensionality. It is indeed conjectured that $\nu = 3/4$ in \mathbb{R}^2 [8]. Extension of the self-avoiding walk from the Euclidian space to fractals makes an infinite number of universality classes.

3.4 Numerical simulations

In Sec. 3.2, we used some hypotheses to derive the connective constant μ and the critical exponent ν , *i.e.*:

1. exponential growth of the number of paths of length k in Eq. (3.13);
2. power growth of the mean shortest distance from the starting point with respect to the path length in Eq. (3.7);
3. the equivalence of the definitions of Eqs. (3.6) and (3.13);
4. the equivalence of the definitions of Eqs. (3.7) and (3.12).

In order to confirm the hypotheses, we here present our numerical simulations. Only in this section and Appendix B, we write the displacement exponent ν defined in Eq. (3.7) as ν' so as to distinguish it from the critical exponent ν defined in the other way, Eq. (3.12).

3.4.1 The number of paths

We hypothesized that the number of paths of length k increases exponentially in Eq. (3.13). In order to check the validity of this assumption, we counted up the number of paths of length k which start from a hub and have a *free end point*, using the depth-first search algorithm [3]. Note that the end point was fixed in Eq. (3.13), but we adapt a free end point here. This is because we can expect that the asymptotic form of the number of paths does not depend on whether the end point is fixed or free as in the self-avoiding walk in the Euclidian space [8].

We carry out the depth-first search in the following way [3]. We first define a tree of height k_{\max} whose nodes constitute a self-avoiding path, and next explore the tree by a depth-first manner. Nodes of depth k consist of all self-avoiding paths of length k starting from a node s . Two nodes are connected if the path of the child node can be generated by appending an edge to that of the parent node.

Drawing an analogy to the self-avoiding walk in the Euclidian spaces, we assume that the number of paths of length k starting from a node s with a free end point behaves as

$$C_k^{(s)} = A^{(s)} \mu^k k^{\gamma-1}, \quad (3.62)$$

where γ is the critical exponent associated with the susceptibility. Because $C_k^{(s)}$ increases exponentially, it is easy to acquire the value of μ , but γ , which

is of more interest from the perspective of critical phenomena, is difficult to obtain accurately.

Choosing a node with the largest degree, namely a hub, as a starting point s , we computed $C_k^{(s)}$ for $n = 4, 2 \leq u \leq v \leq 10$, and $1 \leq k \leq 30$ and fitted the series $C_k^{(s)}$ to

$$\ln C_k^{(s)} = A' + k \ln \mu + (\gamma - 1) \ln k. \quad (3.63)$$

We obtained only μ in high precision (Fig. 3.6). The value of μ of the simulation agrees well with the one of the renormalization-group analysis, while the tree approximation overestimate μ because the existence of loops is not taken into consideration. This figure supports the plausibility of Eq. (3.13).

The upper right points of the inset of Fig. 3.6, which correspond to the (2, 2)-flower, deviate from the line. This is because the graph is smallest for $(u, v) = (2, 2)$ and the finite-size effect appears strongly. The number of paths first increases and then starts decreasing due to a finite size effect (Fig. 3.7). What we need to obtain is the asymptotic behavior of the rise in the intermediate region.

3.4.2 The displacement exponent

We hypothesized in Eq. (3.7) that the mean shortest distance from the starting point increases as a power function of the path length with the displacement exponent ν' . We here confirm it by the depth-first search algorithm [3]. Choosing a node with the largest degree, namely a hub, as a starting point s , we computed $\ln \overline{d_k^{(s)}}$ for $n = 4, 2 \leq u \leq v \leq 10$, and $1 \leq k \leq 30$ by enumerating all the paths of length k . We found that $\ln \overline{d_k^{(s)}}$ fluctuates around an asymptotic line and the amplitude of the fluctuation gets smaller as k becomes larger, approaching to the line; see Fig. 3.8 for $(u, v, n) = (3, 5, 5)$ for example. The figure supports our assumption of the form $\overline{d_k^{(s)}} \propto k^{\nu'}$.

Next, we also hypothesized that the critical exponent ν , which is defined through the two-point function by Eq. (3.12), is equal to the displacement exponent ν' in Eq. (3.7): $\nu = \nu'$. We conducted a Monte-Carlo simulation in order to check the validity of this assumption (Fig. 3.9). We need to evaluate long paths to obtain ν' in high precision. Because the number of paths grows exponentially, we cannot use the depth-first search algorithm, which enumerates all the paths. We therefore adapted the biased sampling algorithm [72], which is a kind of Monte-Carlo algorithm, in order to generate long paths. In the biased sampling algorithm, we make a walker select the next site randomly among adjacent unvisited sites, and thereby define $P'(\omega)$

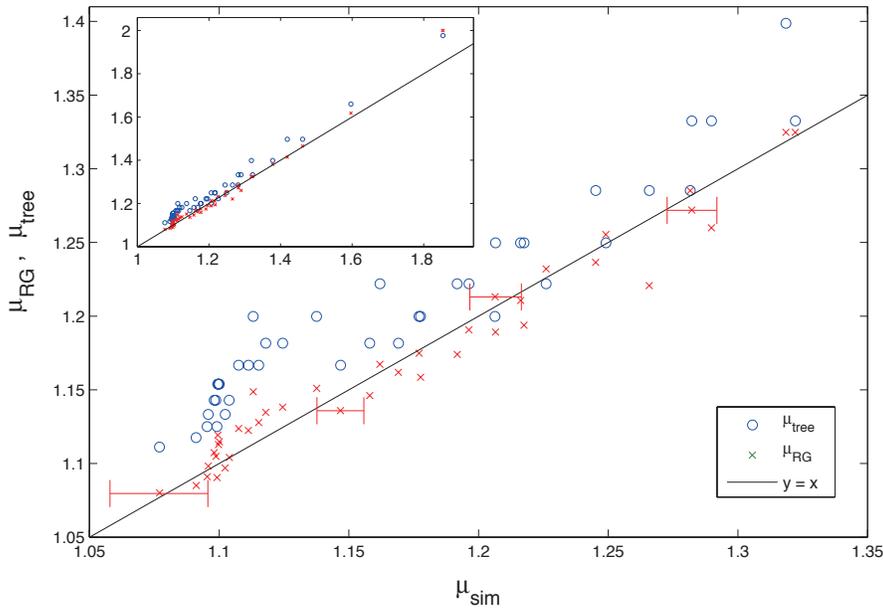


Figure 3.6: Comparison of the connective constant μ obtained by three different methods for various values of u and v . The horizontal axis denotes the estimate of μ in simulation with the fitting in Eq. (3.63), while the vertical axis denotes that of the renormalization-group analysis and the tree approximation (mean-field theory with the common values of u and v). The simulation condition is $n = 4$, $2 \leq u \leq v \leq 10$, and $1 \leq k \leq 30$. A hub was chosen as the starting point. The inset shows the whole range. The lower left part of the inset is magnified and shown as the main figure. All symbols are accompanied by error bars but only the typical size of the error bar is shown.

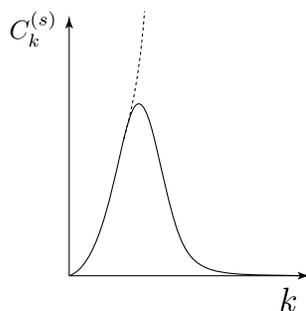


Figure 3.7: A sketch of the number of paths against the path length. The number of paths increases as Eq. (3.62) when k is moderately large, and then it starts decreasing due to the finite-size effect. What we need is the asymptotic behavior in the thermodynamic limit indicated by the dashed curve.

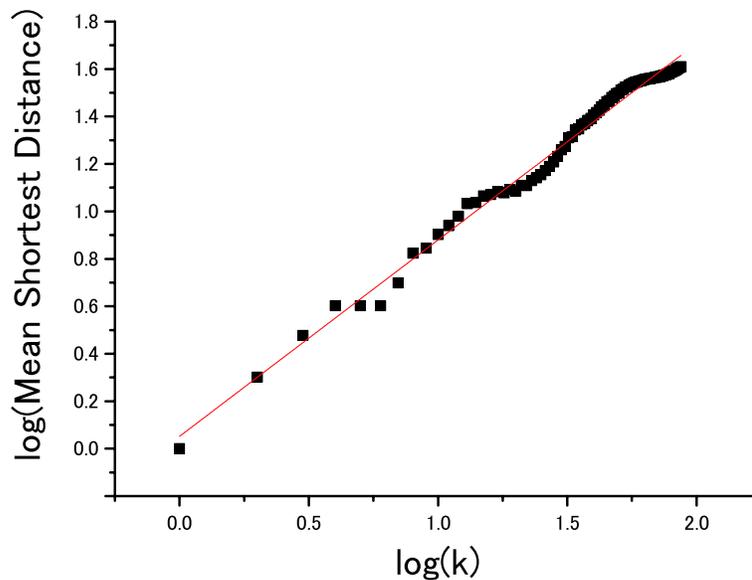


Figure 3.8: The mean shortest distance $\overline{d_k^{(s)}}$ against the path length k . The series of $\overline{d_k^{(s)}}$ computed by the depth-first search is fitted to $\ln \overline{d_k^{(s)}} = A + \nu' \ln k$. We chose a hub as the starting node s . We counted all the paths of $k \leq 87$ for $(u, v, n) = (3, 5, 5)$. The data points of $\ln \overline{d_k^{(s)}}$ fluctuate around an asymptotic line and the amplitude of the fluctuation gets smaller as k becomes larger.

CHAPTER 3. SELF-AVOIDING WALK ON FRACTAL COMPLEX NETWORK

as a distribution that the trajectory of the walker follows. Let l_i be the number of sites to which a walker can go next in the step i . Then a path appears with a probability proportional to $1/\prod_i l_i$. We can express the average of any quantity X in the form

$$\langle X \rangle = \frac{\int_{\Omega_k^{(s)}} X(\omega) \prod_{i=0}^{k-1} l_i(\omega) d\mu'}{\int_{\Omega_k^{(s)}} \prod_{i=0}^{k-1} l_i(\omega) d\mu'} \quad (3.64)$$

$$\approx \frac{X(\omega^{(1)}) \prod_{i=0}^{k-1} l_i(\omega^{(1)}) + \dots + X(\omega^{(M)}) \prod_{i=0}^{k-1} l_i(\omega^{(M)})}{\prod_{i=0}^{k-1} l_i(\omega^{(1)}) + \dots + \prod_{i=0}^{k-1} l_i(\omega^{(M)})}. \quad (3.65)$$

Here $\omega^{(i)}$ for $1 \leq i \leq M$ is a random variable (path) which follows the distribution $P'(\omega)$.

The simulation condition was $2 \leq u \leq 5$ and $2 \leq v \leq 10$. We computed the average $\overline{d_k^{(s)}}$ over 10,000 configurations of paths, using Eq. (3.65) for various values of u and v . We used a hub as the starting node s . Assuming the relation (3.7), we fitted the estimate of the obtained mean shortest distance $\overline{d_k^{(s)}}$ to

$$\ln \overline{d_k^{(s)}} = A + \nu' \ln k. \quad (3.66)$$

We rejected the data point for $(u, v) = (2, 2)$ because the finite-size effect appeared so strongly that fitting could not be done. Detail of analysis is written in Appendix B. Figure 3.9 supports our assumption that the exponents defined in the two ways are indeed equal to each other: $\nu = \nu'$.

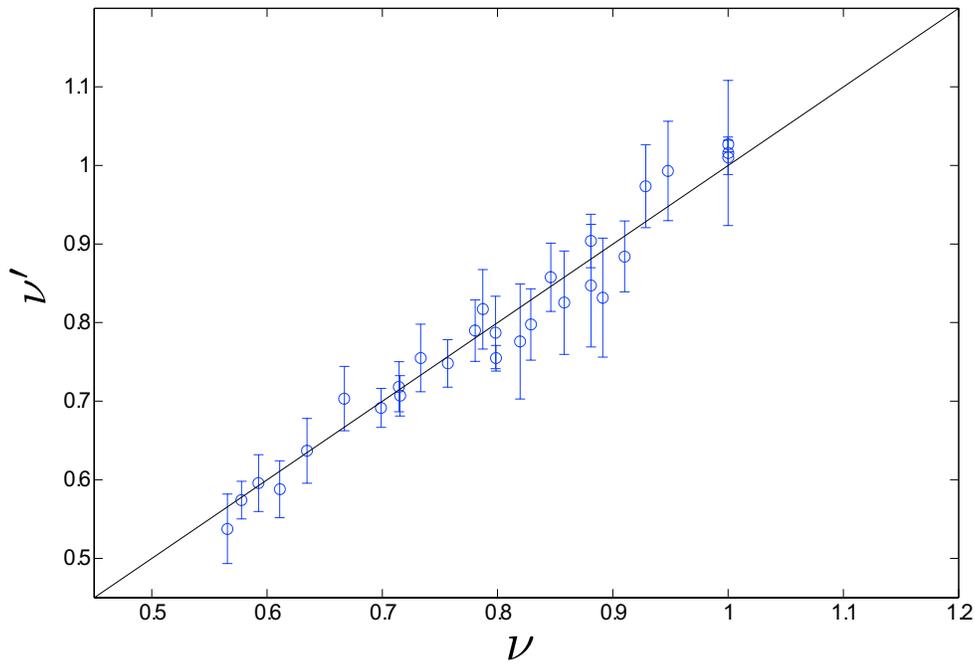


Figure 3.9: The critical exponent ν of the zero-component ferromagnet and the displacement exponent ν' defined in terms of the end-to-end shortest distance estimated by simulations for various values of u and v . We estimate the critical exponent ν by the renormalization-group analysis and computed the displacement exponent ν' by the biased sampling algorithm followed by curve fitting exemplified in Fig. B.1 in Appendix B. We chose a hub as the starting point s . The simulation condition was $n = 4$, $2 \leq u \leq 5$, and $2 \leq v \leq 10$.

Chapter 4

Review of tensor network

In this chapter, we review prerequisite knowledge that we need in Chap. 5. This chapter is intended to be an introduction of tensor networks from the viewpoint of numerical computation. Tensor networks consist of tensors, whose calculation is usually done converting tensors and matrices to each other. We start from the definition of notation of matrices in Sec. 4.1 and see properties of the singular-value decomposition. In Sec. 4.2, we define notations regarding tensors including several kinds of products of tensors. We also introduce diagrammatic representation of tensor decompositions in this section. In Sec. 4.3, we review the higher-order singular-value decomposition, which enables us to see “inside” a tensor. Notations and techniques used in this decomposition are the basis of the theory of tensor networks. In Sec. 4.4, we will explain the alternating least-square (ALS) algorithm. The ALS algorithm is the “workhorse” in this field. Finally, in Sec. 4.5, we explain tensor networks with more complicated structures. We will mainly explain a linear tensor network which is also known as the matrix-product state.

4.1 Matrix

In this section, we review introductory computation of matrices. We basically use linear algebra in tensor networks of liberal arts level, but we need to explain extra topics that we use in computer computation.

4.1.1 Colon notation

We use a colon notation that is used in MATLAB in this chapter. If $A \in \mathbb{R}^{m \times n}$, then $A(k, :)$ designates the k th row:

$$A(k, :) = (a_{k1}, \dots, a_{kn}). \quad (4.1)$$

Similarly, the k th column is specified by $A(:, k)$.

4.1.2 Matrix norms

The analysis of matrix algorithms requires the use of matrix norms. For example, when we approximate a matrix with another matrix, we need a measure of distance on the space of matrices.

It is natural to define matrix norms generalizing vector norms. The most frequently used vector norm is the vector 2-norm:

$$\|x\| = \sqrt{\sum_{i=1}^d x_i^2}, \quad x \in \mathbb{R}^d. \quad (4.2)$$

The matrix 2-norm is defined by using the vector 2-norm:

$$\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}. \quad (4.3)$$

Another frequently used matrix-norm is the Frobenius norm:

$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}, \quad A \in \mathbb{R}^{m \times n}. \quad (4.4)$$

Note that when $A \in \mathbb{R}^{m \times 1}$, the Frobenius norm is reduced to the vector 2-norm. We will generalize the Frobenius norm to tensors shortly.

The matrix 2-norm and the Frobenius norm satisfy $\|A+B\| \leq \|A\| + \|B\|$. Furthermore, these norms are mutually consistent, and therefore we can write

$$\|AB\| \leq \|A\| \|B\|, \quad A \in \mathbb{R}^{m \times k}, \quad B \in \mathbb{R}^{k \times n} \quad (4.5)$$

in the subscript-free norm notation.

These norms satisfy orthogonal invariance. If $A \in \mathbb{R}^{m \times n}$ and there are orthogonal matrices $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$, then

$$\|UAV\|_2 = \|A\|_2, \quad (4.6)$$

$$\|UAV\|_F = \|A\|_F. \quad (4.7)$$

For the 2-norm, the equality is obvious. For the Frobenius norm, it follows from

$$\|UA\|_F = \sum_j \|UA(:, j)\|^2 = \|A\|_F \quad (4.8)$$

and

$$\|UAV\|_F = \|AV\|_F = \|(AV)^T\|_F = \|V^T A^T\|_F = \|A^T\|_F = \|A\|_F. \quad (4.9)$$

4.1.3 The singular-value decomposition

For any real matrices, the singular-value decomposition (SVD) is defined as follows.

Theorem 4.1 (Singular-Value Decomposition (SVD)) *If A is a real $m \times n$ matrix, then there exist orthogonal matrices $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ such that*

$$U^T A V = \Sigma = \text{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m \times n}, \quad p = \min\{m, n\}, \quad (4.10)$$

where $\sigma_1 \geq \dots \geq \sigma_p \geq 0$.

Proof is written in any standard textbook.

From the orthogonal invariance of the norms, it follows that

$$\|A\|_2 = \sigma_1, \quad (4.11)$$

$$\|A\|_F = \sqrt{\sigma_1^2 + \dots + \sigma_p^2}. \quad (4.12)$$

These relations yield the next theorem.

Theorem 4.2 (The Eckart-Young Theorem) *If $k < \text{rank}(A)$ and the SVD of A is given by $A = U S V^T$, then*

$$\min_{\text{rank}(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}, \quad (4.13)$$

$$\min_{\text{rank}(B)=k} \|A - B\|_F = \|A - A_k\|_F = \sqrt{\sigma_{k+1}^2 + \dots + \sigma_{\text{rank}(A)}^2}, \quad (4.14)$$

where

$$A_k = \sum_{i=1}^k \sigma_i U(:, i) V(:, i)^T. \quad (4.15)$$

This theorem says that by throwing away small singular values and the associated orthogonal bases we can approximate a matrix with a matrix of smaller rank and that the error due to approximation is given by the 2-norm or the Frobenius norm. We use this theorem to choose a smaller number of degrees of freedom in the tensor renormalization group.

4.1.4 The Kronecker product

We will use several kinds of matrix products in this chapter. The first one is the Kronecker product. The Kronecker product enables us to generalize linear algebra of matrices to tensors in a natural fashion as we will discuss in a later part of this chapter.

Definition 4.1 (Kronecker product) *The Kronecker product, denoted by \otimes , is a product of matrices $A \in \mathbb{R}^{m_1 \times n_1}$ and $B \in \mathbb{R}^{m_2 \times n_2}$ to a larger matrix with the size of $m_1 m_2 \times n_1 n_2$. It is defined by using the notation of block matrix,*

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n_1}B \\ \vdots & & & \vdots \\ a_{m_1 1}B & a_{m_1 2}B & \cdots & a_{m_1 n_1}B \end{pmatrix}. \quad (4.16)$$

Important properties of the Kronecker product include

$$(A \otimes B)^T = A^T \otimes B^T, \quad (4.17)$$

$$(A \otimes B)(C \otimes D) = AC \otimes BD, \quad (4.18)$$

$$(B \otimes C)^{-1} = B^{-1} \otimes C^{-1}. \quad (4.19)$$

$$(4.20)$$

To understand the linear map $A \otimes B$ in terms of A and B , we define an operation that converts a matrix into a long vector.

Definition 4.2 (Vectorization) *Suppose that X is an $m \times n$ matrix. Its vectorization, which is denoted by vec , is defined by*

$$\text{vec}(X) = \begin{pmatrix} X(:, 1) \\ \vdots \\ X(:, n) \end{pmatrix}. \quad (4.21)$$

We also define the inverse function of vec here.

Definition 4.3 (Reshaping of matrix) *If the vectorization is the same for two matrices A and B , then we call B is the reshape of A . Furthermore, if $B \in \mathbb{R}^{m \times n}$, we write the reshaping operation by*

$$B = \text{reshape}(A, m, n). \quad (4.22)$$

We frequently use the vectorization and reshape in computer calculation of tensors.

If $B \in \mathbb{R}^{m_1 \times n_1}$, $C \in \mathbb{R}^{m_2 \times n_2}$, and $X \in \mathbb{R}^{n_1 \times m_2}$, then

$$Y = CXB^T \iff \text{vec}(Y) = (B \otimes C)\text{vec}(X). \quad (4.23)$$

4.2 Tensor

In this section, we briefly review tensor decompositions. It is particularly important to set notation clearly when we discuss tensors because calculation of tensors is much more complicated than that of matrices. For more detailed review, refer to other sources [73, 74, 75].

4.2.1 Notation of tensors

If $A \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and $\mathbf{i} = (i_1, \dots, i_d)$ with $1 \leq i_k \leq n_k$ for $k = 1, \dots, d$, then

$$A(\mathbf{i}) := A(i_1, \dots, i_d). \quad (4.24)$$

We say A is an order- d tensor and i_k is the k th mode.

By using the colon notation, we can extract smaller order tensors. For example, if A is an order-3 tensor of $3 \times 4 \times 5$, then

$$B = A(:, :, 2) \in \mathbb{R}^{3 \times 4}, \quad (4.25)$$

$$C = A(:, 1, 2) \in \mathbb{R}^3. \quad (4.26)$$

Extracting an order-2 tensor (matrix) as in Eq. (4.25) is called slicing and extracting an order-1 tensor (vector) is called fibering.

Generalizing the Frobenius norm of matrices, we define the corresponding norm for tensors. If $A \in \mathbb{R}^{n_1 \times \dots \times n_d}$, then its Frobenius norm is given by

$$\|A\|_F = \sqrt{\sum_{i_1=1}^{n_1} \dots \sum_{i_d=1}^{n_d} A(\mathbf{i})^2}. \quad (4.27)$$

The ‘vec’ operation and ‘reshape’ operation for tensors are defined in a similar way as for matrices. We vectorize tensors in a FORTRAN fashion in the column-major order. For example, if $A \in \mathbb{R}^{2 \times 2 \times 2}$, then its vectorization is given by

$$\text{vec}(A) = \begin{pmatrix} A(1, 1, 1) \\ A(2, 1, 1) \\ A(1, 2, 1) \\ A(2, 2, 1) \\ A(1, 1, 2) \\ A(2, 1, 2) \\ A(1, 2, 2) \\ A(2, 2, 2) \end{pmatrix}. \quad (4.28)$$

Alternatively, if we define an integer-valued function by

$$\text{col}(\mathbf{i}, \mathbf{n}) = i_1 + (i_2 - 1)n_1 + (i_3 - 3)n_1n_2 + \cdots + (i_d - 1)n_1 \cdots n_{d-1}, \quad (4.29)$$

then the vectorization is characterized by

$$A(\mathbf{i}) = \text{vec}(A)(\text{col}(\mathbf{i}, \mathbf{n})). \quad (4.30)$$

If two tensors A and B have the same vectorization, then we say that B is a reshape of A . For example, for the above order-3 tensor A ,

$$\text{reshape}(\text{vec}(A), 2, 2, 2) = A. \quad (4.31)$$

Next, we define the transposition for tensors. The transposition of an order- d tensor is denoted by $A^{[p_1 p_2 \cdots p_d]} = A^{[\mathbf{p}]}$, where $[\mathbf{p}]$ is a permutation of $\{1, 2, \cdots, d\}$. For instance, for an order-3 tensor A , $A(i_1, i_2, i_3) = A^{[3 2 1]}(i_3, i_2, i_1)$. For a general order- d tensor, the transposition is defined by

$$A(i_1, \cdots, i_d) = A^{[\mathbf{p}]}(i_{p_1}, \cdots, i_{p_d}). \quad (4.32)$$

4.2.2 Unfolding

We often need to convert back and forth between a tensor and a matrix. We can perform this conversion by combining the transposition and the reshape operation. There is a special conversion from a tensor to a matrix that we call modal unfolding. If $A \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $N = n_1 \cdots n_d$, then the mode- k unfolding $A_{(k)}$ is an $n_k \times N/n_k$ matrix whose columns are the mode- k fibers:

$$A_{(k)}(i_k, \text{col}(\tilde{\mathbf{i}}, \tilde{\mathbf{n}})) = A(\mathbf{i}), \quad (4.33)$$

where $\tilde{\mathbf{i}} = (i_1, \cdots, i_{k-1}, i_{k+1}, \cdots, i_d)$ and $\tilde{\mathbf{n}} = (n_1, \cdots, n_{k-1}, n_{k+1}, \cdots, n_d)$. For example, the mode-2 unfolding of an order-3 tensor $A \in \mathbb{R}^{2 \times 2 \times 2}$ is

$$A_{(2)} = \begin{pmatrix} a_{111} & a_{211} & a_{112} & a_{212} \\ a_{121} & a_{221} & a_{122} & a_{222} \end{pmatrix}. \quad (4.34)$$

We will use the modal unfolding to define tensor decompositions.

4.2.3 Outer product and canonical rank

The outer product is the operation to combine an order- d tensor A and an order- f tensor B to produce an order- $(d + f)$ tensor C . It is denoted by

$$C(\mathbf{i}, \mathbf{j}) = C(i_1, \cdots, i_d, j_1, \cdots, j_f) = A(\mathbf{i}) \circ B(\mathbf{j}). \quad (4.35)$$

The rank, more specifically the canonical rank, of a tensor is defined by the minimal number of outer products to represent the tensor as their summation. For example, an order- d tensor A is rank 1 iff there exists a decomposition

$$A = b_1 \circ b_2 \circ \cdots \circ b_d, \quad (4.36)$$

where b_k ($k = 1, \dots, d$) is a vector. Note that the above equation is equivalent to

$$\text{vec}(A) = b_d \otimes \cdots \otimes b_2 \otimes b_1. \quad (4.37)$$

We define the canonical rank as follows:

Definition 4.4 (Canonical rank) *Let A be an order- d tensor. If A is decomposed by a sum of some vectors f_j^k ($j = 1, \dots, r$, $k = 1, \dots, d$) as*

$$A = \sum_{j=1}^r \lambda_j f_j^1 \circ \cdots \circ f_j^d \quad (4.38)$$

and no shorter summation of rank-1 tensors exists, then we say that A is a canonical rank- r tensor.

4.2.4 Tensor network diagram

We are going to define a tensor contraction and its diagrammatic representation. Contraction of indices of two tensors are essentially a matrix multiplication of unfolded matrices. For example, let us consider a contraction of two order-4 tensors A and B :

$$\sum_{k=1}^{n_2} A(i_1, k, i_3, i_4) B(j_1, j_2, j_3, k) =: C(i_1, i_3, i_4, j_1, j_2, j_3). \quad (4.39)$$

We can calculate this using matrix-by-matrix product after carrying out modal unfolding of A and B , *i.e.*,

$$\tilde{C} := A_{(2)}^T B_{(4)}, \quad (4.40)$$

followed by reshape of \tilde{C} to C . We can also contract more than two indices at the same time with the sequence of permute, reshape, matrix-by-matrix product, reshape, and permute.

To denote the sequence of contraction simply, we introduce tensor network diagrams. The simplest diagram is the contraction of two tensors.

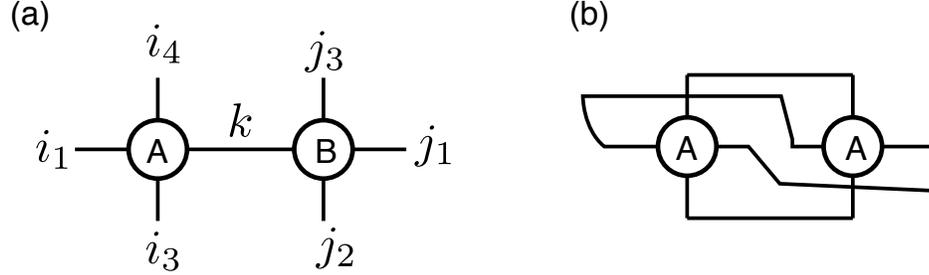


Figure 4.1: Diagrammatic representation of index contraction between two tensors. (a) Contraction of a single index (Eq. (4.39)). (b) Contraction of four indices. This is the square of the Frobenius norm (Eq. (4.27)).

Figure 4.1 (a) is the diagrammatic representation of Eq. (4.39). The nodes denote order-4 tensors A and B and the linked bond is the one over which summation is taken. The diagram has 6 open bonds, and thus it means that the resulting tensor is of order 6. We also show in Fig. 4.1 (b) a diagrammatic representation of the Frobenius norm for an order-4 tensor. The corresponding equation is $\|A\|_F^2 = \text{vec}(A)^T \text{vec}(A)$.

4.2.5 Modal product

We now define a product of a tensor and a matrix.

Definition 4.5 (Modal product) Let $S \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and $M \in \mathbb{R}^{m_k \times n_k}$. The mode- k product of S and M is given by

$$A_{(k)} = MS_{(k)}, \quad (4.41)$$

where the subscript k denotes the mode- k unfolding. We denote this operation by

$$A = S \times_k M. \quad (4.42)$$

The diagrammatic representation of Eq. (4.42) is Fig. 4.2 (a).

The modal product has the following properties:

$$(S \times_k F) \times_j G = (S \times_j G) \times_k F \quad (j \neq k), \quad (4.43)$$

$$(S \times_k F) \times_k G = S \times_k (GF). \quad (4.44)$$

These relations are clear if we draw diagrams (Fig. 4.2 (a) and (b)).

We use the next theorem to derive a kind of tensor decomposition (HOSVD) in Sec. 4.3.

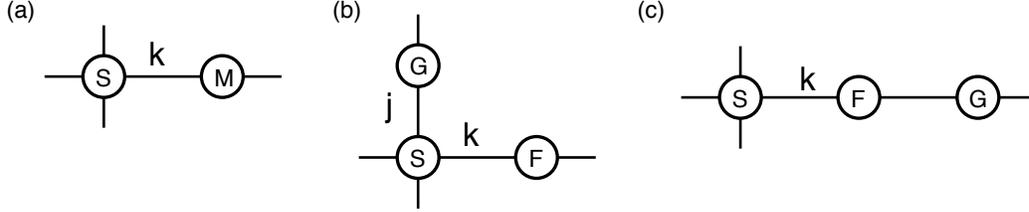


Figure 4.2: Diagrams regarding modal products. (a) Definition of modal product (Eq. (4.42)). (b) Eq. (4.43). (c) Eq. (4.44).

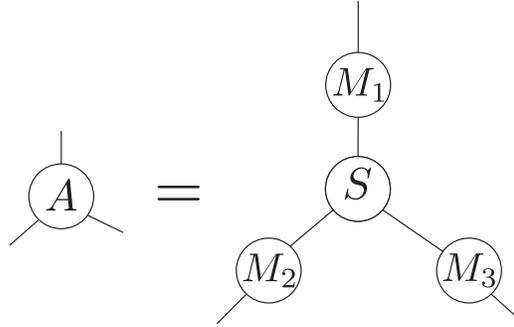


Figure 4.3: The diagrammatic representation of Eq. (4.45) for an order-3 tensor A .

Theorem 4.3 Suppose $S \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and $M_k \in \mathbb{R}^{m_k \times n_k}$ for $k = 1, \dots, d$. The following equations are equivalent:

$$A = S \times_1 M_1 \times_2 M_2 \cdots \times_d M_d \quad (4.45)$$

$$\iff A_{(k)} = M_k S_{(k)} (M_d \otimes \cdots \otimes M_{k+1} \otimes M_{k-1} \otimes \cdots \otimes M_1). \quad (4.46)$$

Furthermore, if M_1, \dots, M_d are all nonsingular, then

$$S = A \times_1 M_1^{-1} \times_2 M_2^{-1} \cdots \times_d M_d^{-1}. \quad (4.47)$$

The equivalence is easy to be shown if we use Eq. (4.23). The latter half of the theorem is manifest by drawing a diagram (Fig. 4.3).

4.3 Decomposition of tensors

We have generalization of SVD to tensors in two ways so as to preserve some properties of SVD. The first generalization of SVD that we present here is the higher-order singular-value decomposition (HOSVD). The next one is the canonical decomposition.

4.3.1 The higher-order singular-value decomposition

The higher-order singular-value decomposition (HOSVD) generalizes SVD so as to preserve its orthonormal properties of row and column matrices [44].

Theorem 4.4 (HOSVD) *If $A \in \mathbb{R}^{n_1 \times \dots \times n_d}$ and SVD of its modal unfoldings are given by $A_k = U_k \Sigma_k V_k^T$ ($k = 1, \dots, d$), then its HOSVD is*

$$A = S \times_1 U_1 \times_2 U_2 \cdots \times_d U_d, \quad (4.48)$$

where the core tensor S is defined by

$$S = A \times_1 U_1^T \times_2 U_2^T \cdots \times_d U_d^T. \quad (4.49)$$

Furthermore, the norm of the modal unfolding of S are structured in the following sense:

$$\|S_{(k)}(i, :)\|_F = \sigma_i(A_{(k)}). \quad (4.50)$$

Here $\sigma_i(A_{(k)})$ is the i th largest singular value of $A_{(k)}$.

The existence of the decomposition follows from Theorem 4.3. The norm property Eq. (4.50) is shown as follows. The vectorization of A satisfies

$$\text{vec}(S) = (U_d^T \otimes \cdots \otimes U_1^T) \text{vec}(A). \quad (4.51)$$

The relation (4.23) gives the mode-1 unfolding of the core tensor as

$$S_{(1)} = U_1^T A_{(1)} (U_d \otimes \cdots \otimes U_2) \quad (4.52)$$

$$= \Sigma_1 V_1^T (U_d \otimes \cdots \otimes U_2). \quad (4.53)$$

Because the Frobenius norm is invariant under orthogonal transformations, Eq. (4.50) holds $k = 1$. For $k \neq 1$, the same argument holds for the permuted tensor of A , namely $A^{[k \ 1 \ 2 \cdots k-1 \ k+1 \cdots d]}$.

Each column vector of U_k ($k = 1, \dots, d$) can be interpreted as the principal component in each mode and the elements of the core tensor represents interactions between different components.

The multilinear rank denoted by $\text{rank}_*(A)$ is the rank of modal unfoldings:

$$\text{rank}_*(A) = (\text{rank}(A_{(1)}), \dots, \text{rank}(A_{(d)})). \quad (4.54)$$

The HOSVD leads to a naive approximation with a lower multilinear tensor. Suppose that $r_k < \text{rank}_*(A_{(k)})$ for $k = 1, \dots, d$. We define \tilde{A} by

$$\tilde{A} = \sum_{j_1=1}^{r_1} \cdots \sum_{j_d=1}^{r_d} S(\mathbf{j}) U_1(:, j_1) \circ \cdots \circ U_d(:, j_d). \quad (4.55)$$

Its absolute error is bounded by

$$\|A - \tilde{A}\| \leq \min_{1 \leq k \leq d} \sum_{i=r_k+1}^{\text{rank}(A_{(k)})} \sigma_i(A_{(k)})^2. \quad (4.56)$$

See Ref. [44] for proof.

4.3.2 Canonical decomposition

One of the convenient attributes of SVD is that the singular matrix is diagonal. HOSVD, on the other hand, does not inherit this property; the core matrix S is not a diagonal tensor. There is a route to generalize SVD to tensors other than HOSVD. The canonical decomposition [76, 77] generalizes SVD to tensors so as to preserve the diagonal property of the singular matrix.

The canonical decomposition of an order- d tensor is given by

$$A = \sum_{j=1}^r \lambda_j f_j^1 \circ \cdots \circ f_j^d. \quad (4.57)$$

We can consider approximating a tensor with the best canonical rank- r decomposition in terms of the Frobenius norm. However, this framework has drawbacks: (1) finding the canonical rank is NP-hard [78]; (2) there exist tensors that can be approximated in arbitrary precision by a lower canonical rank [73]; (3) even if $X_{r+1} = \sum_{j=1}^{r+1} \lambda_j f_j^1 \circ \cdots \circ f_j^d$ is the best canonical rank- $(r+1)$ approximation, it does not necessarily follow that $X_r = \sum_{j=1}^r \lambda_j f_j^1 \circ \cdots \circ f_j^d$ is the best canonical rank- r approximation [79]. Simply put, for tensors we cannot obtain a beautiful theorem like the Eckart-Young theorem even if we generalize SVD so as to keep the diagonal property.

4.4 Alternating least-square algorithm

We noted that by throwing away small elements of a core tensor as in Eq. (4.55) we can approximate a tensor with a smaller multilinear rank. However, this approximation is not optimal. Let us consider the following optimization problem:

$$\min_X \|A - X\|_F, \quad (4.58)$$

where A is the original tensor of order- d and X is an order- d tensor with a smaller multilinear rank than A . By varying X , we would like to minimize the

4.4. ALTERNATING LEAST-SQUARE ALGORITHM

above objective function. As we noted above, Eq. (4.55) is not the optimal solution of this optimization problem.

We explain it using order-3 tensors. Suppose that $A \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, while $S \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ and $U_k \in \mathbb{R}^{n_k \times r_k}$ ($k = 1, 2, 3$) are initialized randomly. We define a tensor of a lower multilinear rank X as $X = S \times_1 U_1 \times_2 U_2 \times_3 U_3$. We search S , U_1 , U_2 , and U_3 that minimize the following function:

$$\|\text{vec}(A) - (U_3 \otimes U_2 \otimes U_1)\text{vec}(S)\|. \quad (4.59)$$

We can solve this problem using the alternating least-square (ALS) algorithm [80]. The ALS algorithm works by freezing all but one of the unknowns and minimize the objective function varying the single free tensor. Then we optimize another tensor fixing the other tensors. This process keeps going until some convergence criteria is met. The ALS algorithms are widely used in tensor decomposition. As we will see later, the density-matrix renormalization group is also an ALS algorithm.

Because $\{U_k\}$ are orthogonal, Theorem 4.3 gives the optimal S as

$$\text{vec}(S) = (U_3^T \otimes U_2^T \otimes U_1^T)\text{vec}(A). \quad (4.60)$$

Thus, we only need to search the optimal U_k ($k = 1, 2, 3$). The square of the objective function is

$$\|A - X\|_F^2 = \|A\|_F^2 + \|X\|_F^2 - 2\langle A, X \rangle, \quad (4.61)$$

where $\langle A, X \rangle$ denotes the inner product of the two tensors, which is defined as

$$\langle A, X \rangle := \text{vec}(A)^T \text{vec}(X). \quad (4.62)$$

We can transform the inner product as

$$\langle A, X \rangle = \langle A, S \times_1 U_1 \times_2 U_2 \times_3 U_3 \rangle \quad (4.63)$$

$$= \langle A \times_1 U_1^T \times_2 U_2^T \times_3 U_3^T, S \rangle \quad (4.64)$$

$$= \langle S, S \rangle \quad (4.65)$$

$$= \|S\|_F^2. \quad (4.66)$$

Because the Frobenius norm is invariant under orthogonal transformations, we have

$$\|X\|_F^2 = \|S\|_F^2. \quad (4.67)$$

Therefore, the objective function (4.61) becomes

$$\|A - X\|_F^2 = \|A\|_F^2 - \|S\|_F^2. \quad (4.68)$$

Because the first terms is a constant, we need to maximize the norm of Eq. (4.60). The Frobenius norm of S is

$$\|S\|_F = \|(U_3^T \otimes U_2^T \otimes U_1^T)\text{vec}(A)\|_F = \begin{cases} \|U_1^T A_{(1)}(U_3 \otimes U_2)\|_F, \\ \|U_2^T A_{(2)}(U_3 \otimes U_1)\|_F, \\ \|U_3^T A_{(3)}(U_2 \otimes U_1)\|_F. \end{cases} \quad (4.69)$$

Here we used the relation (4.23). In the step of optimization of U_1 , we change only U_1 fixing U_2 and U_3 . We optimize U_1 by SVD; if SVD of $A_{(1)}(U_3 \otimes U_2)$ is $P\Sigma Q$, then the optimal U_1 is given by $U_1 = P(:, 1 : r_1)$, where $1 : r_1$ means that we only keep the first r_1 columns of P . In the steps of optimization of U_2 and U_3 , we do similar processes. We repeat SVD until the norm stops changing. Although it is not guaranteed to converge to the global optimum, it works in many cases in practice.

4.4.1 Example of ALS: Image compression

As a simple example of lower-rank approximation of tensor, we explain the compression of a color image. The image of Lena is an order-3 tensor of size $512 \times 512 \times 3$, where each slice corresponds to R, G, and B color scales. We exploit the fact that every slice is correlated to each other in order to compress it. Let $A \in \mathbb{R}^{512 \times 512 \times 3}$ be the original image and $X \in \mathbb{R}^{r \times r \times 3}$ be the compressed image. Our goal is to minimize the objective function (4.58); we used the ALS algorithm to do this. We show in Fig. 4.4 compressed images for various r with the relative error $\|A - X\|_F / \|A\|_F$. The approximation error decreases monotonically with respect to r .

4.5 Tensor networks

The HOSVD requires $O(r^d)$ memory for large d , and hence is useless. The approximation with the canonical decomposition often fails because of the difficulties mentioned above. Therefore, it is a good idea to look for an alternative of the canonical decomposition by increasing the number of parameters. A natural candidate is the tensor-train decomposition, which is also called the matrix-product state. The tensor-train format takes storage of $O(d)$ to represent order- d tensors; its implementation is also simple. As a famous algorithm of tensor-train decomposition, we explain the density-matrix

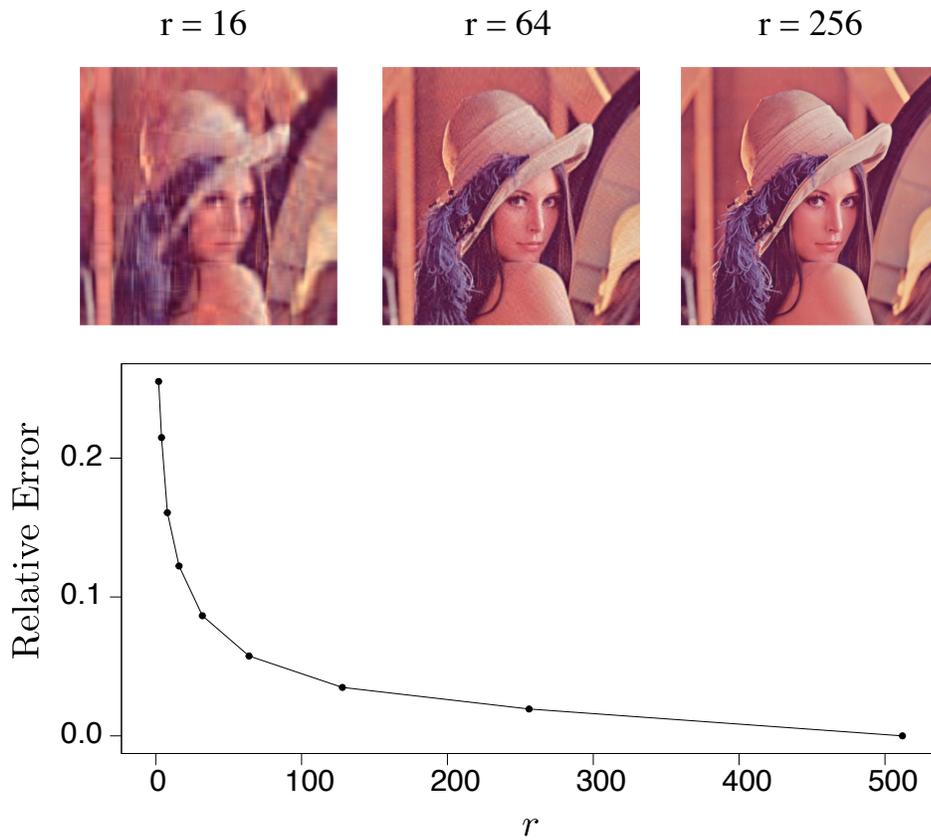


Figure 4.4: Image compression by the ALS algorithm. The top images are the compressed ones with $X \in \mathbb{R}^{r \times r \times 3}$ ($r = 16, 64, 256$, respectively). The bottom figure is the relative error of the approximation.

renormalization group. The diagrammatic representation of the decomposition motivates us to consider more complicated structures of networks. For a detailed review of the tensor networks, see Refs. [81, 22, 21], for example.

4.5.1 Tensor-train decomposition

Let us try to approximate a tensor with a product of matrices.

Definition 4.6 (Tensor-train decomposition) *Let A be an order- d tensor. We refer to the following decomposition by a product of d matrices as a tensor-train decomposition (TT decomposition):*

$$A(i_1, \dots, i_d) = G_1(i_1) \cdots G_d(i_d), \quad (4.70)$$

where $G_k(i_k)$ is an $r_{k-1} \times r_k$ matrix and the boundary conditions are $r_0 = r_d = 1$. We call G_k the core matrix and r_k the TT-rank.

Clearly, the TT decomposition is a generalization of the canonical decomposition with a larger number of parameters. If a tensor is decomposed by a canonical decomposition with canonical rank r , then we can represent it with a TT format with a TT-rank r . Finding the TT-rank is easy in contrast to the canonical rank according to the following theorem.

Theorem 4.5 *Let A_k be the unfolding of an order- d tensor A :*

$$A_k(i_1 \cdots i_k, i_{k+1} \cdots i_d) = A(\mathbf{i}). \quad (4.71)$$

If $\text{rank} A_k = r_k$, then there exists a TT decomposition with a TT-rank not higher than r_k .

See Ref. [81] for a proof. Furthermore, we can represent any tensor with TT decomposition exactly with the repeated use of SVD. This fact is, however, not very useful in practice because this procedure requires an exponential number of parameters with respect to d . In some cases, we can guarantee that we can represent a tensor with small TT-ranks from physical argument (*e.g.* entanglement entropy) [22] or from direct construction of TT decomposition, but in general we do not know whether there is a low-rank approximation or not until carrying out computation. Therefore, the more practical approach is that we continue calculation hypothesizing that a tensor has a low-rank TT approximation. After we obtain a solution, we check the validity of the approximation.

The TT decomposition does not become difficult for high dimensionality; its storage is linear in d . Furthermore, if vectors and matrices are prepared

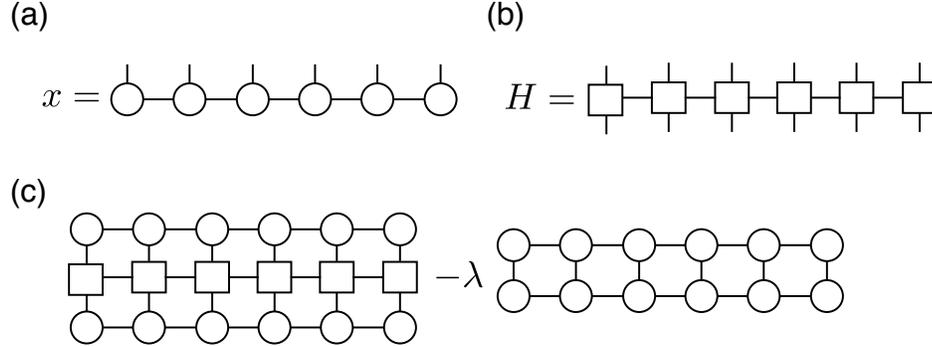


Figure 4.5: (a) TT format of the vector x . (b) TT format of the matrix H . (c) Maximization of the Rayleigh quotient (Eq. (4.72)).

in TT format, we can do many basic operations of linear algebra with the complexity linear in d , for example, addition of vectors, matrix-by-vector multiplication, calculation of the Frobenius norm, matrix tranposition, and trace of matrix. As we will see below, we can also calculate extreme eigenvalues of the TT matrix efficiently.

4.5.2 The density-matrix renormalization group

The density-matrix renormalization group is the minimization of the Rayleigh quotient with the ALS algorithm. The Rayleigh quotient is defined by

$$\frac{x^T H x}{x^T x}, \quad (4.72)$$

where H is the Hermitian matrix. It was first formulated using the density matrix [82, 83], but is also possible with the TT format [84]. Here we assume that both x and H are in the TT format (Fig. 4.5 (a) and (b)). Minimization of Eq. (4.72) is equivalent to extremizing

$$x^T H x - \lambda x^T x, \quad (4.73)$$

where λ is the Lagrange multiplier. We show a diagrammatic representation of this equation in Fig. 4.5 (c). We freeze $(d - 1)$ pieces of cores and change the $(k - 1)$ th core tensor so as to minimize Eq. (4.73) in the k th step. The derivative of (4.73) with respect to the k th core tensor is indicated in a diagram (Fig. 4.6 (a)). Except for the k th core tensor, the values of the tensors are known in this step. Using the definition of \tilde{H} and N in Fig. 4.6

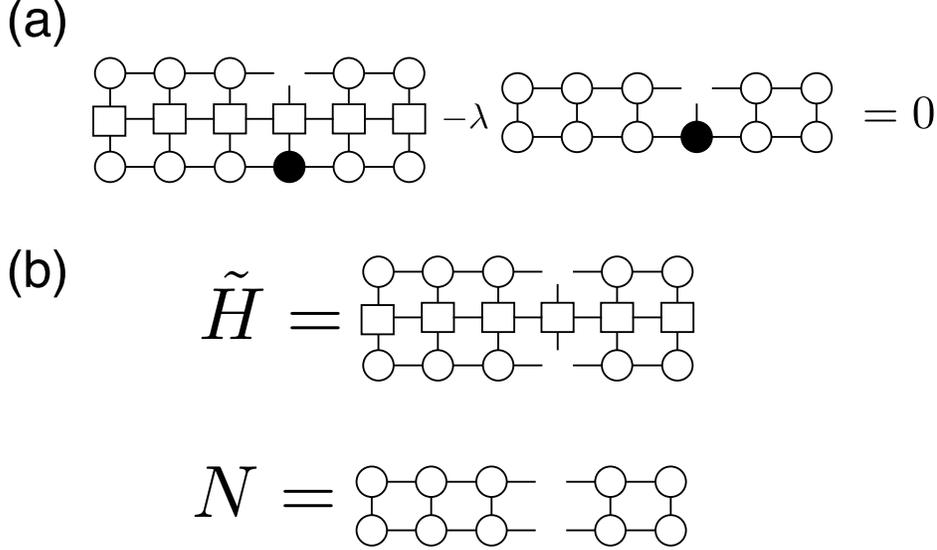


Figure 4.6: (a) Stationary condition of the objective function in the DMRG. (b) The definition of \tilde{H} and N in Eq. (4.74).

(b) and (c), we can write down the condition of the stationary point as the following generalized eigenvalue problem:

$$\tilde{H}v = \lambda Nv, \quad (4.74)$$

where v is the vectorization of the k th core, $v = \text{vec}(G_k)$. By applying local orthogonal transformations to x we can change N to the identity matrix, and thus we can reduce the generalized eigenvalue problem (4.74) to the standard eigenvalue problem $\tilde{H}v = \lambda v$. We sweep from left to right and from right to left several times until a convergence criteria is met, thereby obtaining the lowest eigenvalue of H and the associated eigenvector. As we mentioned in Sec. 4.4, ALS algorithms including the present one do not guarantee the global minimum. We can obtain not only the lowest eigenvalues but also a few extreme eigenvalues by minimizing the block Rayleigh quotient in the same way [85].

4.5.3 Tensor network of lattice structure

In Sec. 4.5, we have reviewed the TT format, which is a one-dimensional structure with the open boundary condition. We can generalize networks to arbitrary graph structures so as to allow periodic boundary condition, an infinite chain, a lattice structure, and so on (Fig. 4.7). It has been ap-

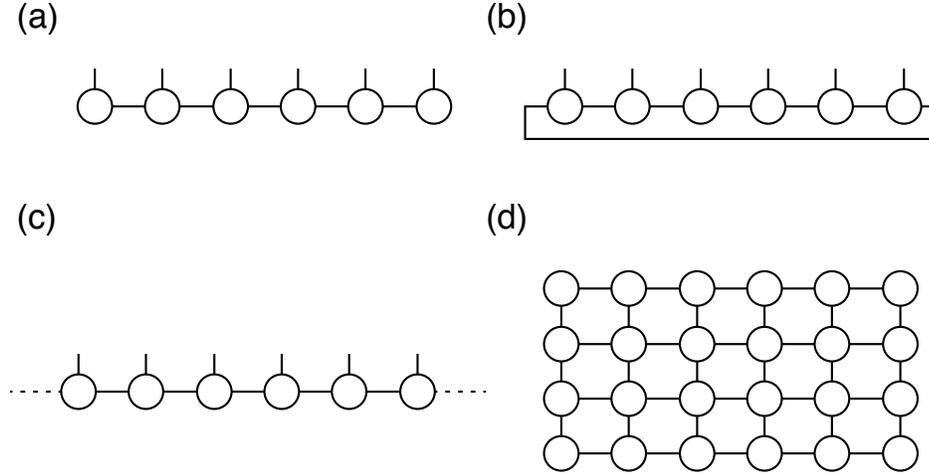


Figure 4.7: Zoo of tensor networks. (a) One-dimensional structure with the open boundary condition. (b) One-dimensional structure with the periodic boundary condition. (c) Infinite one-dimensional structure. (d) Square lattice representing a scalar.

plied to various problems of many-body physics [86, 21, 23], including calculation of partition functions of classical spin systems [28], calculation of ground-state energy of quantum spin systems [87], time evolution of quantum systems [88, 89, 37], and stochastic processes [35, 36]. Physical intuition of correlation and entanglement properties have motivated physicists to develop many approximate schemes. For example, we can represent a partition function of two-dimensional classical spin models on square lattices using a diagram of Fig. 4.7 (d) at any temperatures [28]. This stems from the fact that the two-dimensional network can represent polynomially decaying correlations [21]. On the other hand, any two-point function represented by the tensor-train format (Fig. 4.7 (a)) decays exponentially [22]. Thus, we expect that a two-dimensional classical system at the criticality is well described by a two-dimensional network. Indeed, it has been reported that tensor networks of lattice structures produce quite accurate results for classical systems [28, 31, 29, 30, 90]. As another big advantage of the tensor networks other than the tensor-train format is that they can represent an infinite system as long as the system is translational invariant [91, 92, 45, 46, 93]. This scheme is suitable especially for system with long-range correlations.

Chapter 5

Nonequilibrium-relaxation method with tensor networks

The content of this chapter is based on Ref. [94]. This chapter is organized as follows. In Sec. 5.1, we explain notations and the model that we use as an example of our algorithm. We consider the Glauber dynamics [39] in discrete time throughout this chapter. We borrow notations from quantum mechanics, expressing the probability distribution as a ket and the transition probability as an operator. In Sec. 5.2, we explain how to construct a tensor-network operator for the transition probability, using a higher-order singular-value decomposition. We calculate the relaxation of the magnetization from the all-spin-up state to the equilibrium with various bond dimensions and compare the results with an analytic result. In Sec. 5.3, we analyze a system at the critical point and estimate the dynamical critical exponent z using the nonequilibrium-relaxation method.

5.1 Model and notation

Throughout this chapter, we focus on kinetic Ising models, whose definition we give in this section; we can easily generalize our algorithm to any sublattice-update dynamics with nearest-neighbor interactions on a bipartite graph. We first define the update rule of spins and next explain how to describe a Markov process, borrowing the notation of quantum mechanics and using diagrams of tensor networks.

Unlike continuous time evolution, we do not need to perform the Suzuki-Trotter decomposition for the time evolution; we instead construct an operator of time evolution using the higher-order singular-value decomposition.

5.1.1 Kinetic Ising model

The Glauber dynamics [39] is a kind of kinetic Ising model whose transition rule is a heat-bath type. We consider the Glauber dynamics in discrete time although the original Glauber dynamics is in continuous time. The Glauber dynamics in a computer simulation is usually the one in discrete time because such dynamics is simulated by Monte Carlo methods in many cases. Our task is to implement heat-bath algorithms using tensor networks.

We explain the Glauber dynamics taking the one-dimensional case as an example; generalizing it to higher dimensions is straightforward. Let us consider a one-dimensional spin chain of length $2N$ whose Hamiltonian is given by

$$H = - \sum_{i=1, \dots, 2N} \sigma_i \sigma_{i+1}. \quad (5.1)$$

We use periodic boundary conditions throughout this chapter.

We update a single spin with the following transition probability of the heat-bath-type :

$$w(\sigma_i \rightarrow \sigma'_i) = \frac{e^{\beta \sigma'_i (\sigma_{i-1} + \sigma_{i+1})}}{\sum_{\sigma''_i = \pm 1} e^{\beta \sigma''_i (\sigma_{i-1} + \sigma_{i+1})}}. \quad (5.2)$$

Here, σ denotes a spin variable, which takes the values ± 1 , i is the spin that we try to update, and σ_i and σ'_i are the values of the spin i at the previous and new time steps, respectively. We adopt the two-sublattice multi-spin-flip dynamics, in which we divide the whole system into two sublattices and flip all the spins on each sublattice simultaneously. In the odd time steps, the transition probability of the whole system is given as follows:

$$\begin{aligned} & w(\sigma_1, \dots, \sigma_{2N}; \sigma'_1, \dots, \sigma'_{2N}) \\ &= \prod_{i=1, 3, \dots, 2N-1} \frac{e^{\beta \sigma'_i (\sigma_{i-1} + \sigma_{i+1})}}{\sum_{\sigma''_i = \pm 1} e^{\beta \sigma''_i (\sigma_{i-1} + \sigma_{i+1})}} \prod_{i=2, 4, \dots, 2N} \delta_{\sigma_i, \sigma'_i}, \end{aligned} \quad (5.3)$$

where δ_{ij} is Kronecker's delta. The above transition probability means that spins at even sites are frozen and act as a heat bath to odd spins during the time evolution. This is a sufficient condition for relaxation to the equilibrium. In the even time steps, roles of spins at odd and even sites are exchanged. The transition probability becomes

$$\begin{aligned} & w(\sigma_1, \dots, \sigma_{2N}; \sigma'_1, \dots, \sigma'_{2N}) \\ &= \prod_{i=2, 4, \dots, 2N} \frac{e^{\beta \sigma'_i (\sigma_{i-1} + \sigma_{i+1})}}{\sum_{\sigma''_i = \pm 1} e^{\beta \sigma''_i (\sigma_{i-1} + \sigma_{i+1})}} \prod_{i=1, 3, \dots, 2N-1} \delta_{\sigma_i, \sigma'_i}. \end{aligned} \quad (5.4)$$

We obtain the probability distribution by applying the transition operators (5.3) and (5.4) alternatively.

5.1.2 Diagrammatic representation of the Markov chain

Consider a spin chain of length M ($M \in \mathbb{N}$) and denote the spin variables as $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_M)$. We express the probability distribution $P(\boldsymbol{\sigma})$ of a spin configuration borrowing the notation of quantum mechanics:

$$P(\boldsymbol{\sigma}) = \langle \boldsymbol{\sigma} | P \rangle. \quad (5.5)$$

We now describe the ket of the probability distribution using a matrix-product state [22]:

$$|P\rangle = \sum_{\boldsymbol{\sigma}} \text{Tr} [A_1(\sigma_1) \cdots A_M(\sigma_M)] |\sigma_1 \cdots \sigma_M\rangle, \quad (5.6)$$

where $A_i(\sigma_i)$ represents a matrix state at site i . For example, consider the fully magnetized state, in which all spins point up. We can represent it as a product of matrices of 1×1 :

$$A_i(+1) = 1, \quad A_i(-1) = 0 \quad (i = 1, \dots, M). \quad (5.7)$$

For another instance, let us consider a fully magnetized state whose spins are all up or down with the same probability. In this case, the state is more complex than the previous example, and we cannot express it as a product of 1×1 matrices; we need 2×2 matrices to express the state as a matrix product:

$$A_i(+1) = 2^{-1/M} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad A_i(-1) = 2^{-1/M} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \quad (i = 1, \dots, M). \quad (5.8)$$

Generally speaking, the more complex a state becomes, the larger matrices we need to express it as a matrix-product state. The dimensionality of matrices to express a state is called the bond dimension of a matrix-product state; the bond dimension of the first example is thus one and that of the second example is two. We can represent this matrix-product state as a diagram in Fig. 5.1 (a).

Next, consider the transition probability $W(\boldsymbol{\sigma} \rightarrow \boldsymbol{\sigma}')$ from a configuration $\boldsymbol{\sigma}$ to a new configuration $\boldsymbol{\sigma}'$; it is an order- $2M$ tensor, which we can represent as a matrix-product operator:

$$\hat{W} = \sum_{\boldsymbol{\sigma}, \boldsymbol{\sigma}'} |\boldsymbol{\sigma}'\rangle \langle \boldsymbol{\sigma}' | \hat{W} | \boldsymbol{\sigma} \rangle \langle \boldsymbol{\sigma} | := \sum_{\boldsymbol{\sigma}, \boldsymbol{\sigma}'} W(\boldsymbol{\sigma} \rightarrow \boldsymbol{\sigma}') |\boldsymbol{\sigma}'\rangle \langle \boldsymbol{\sigma} |. \quad (5.9)$$

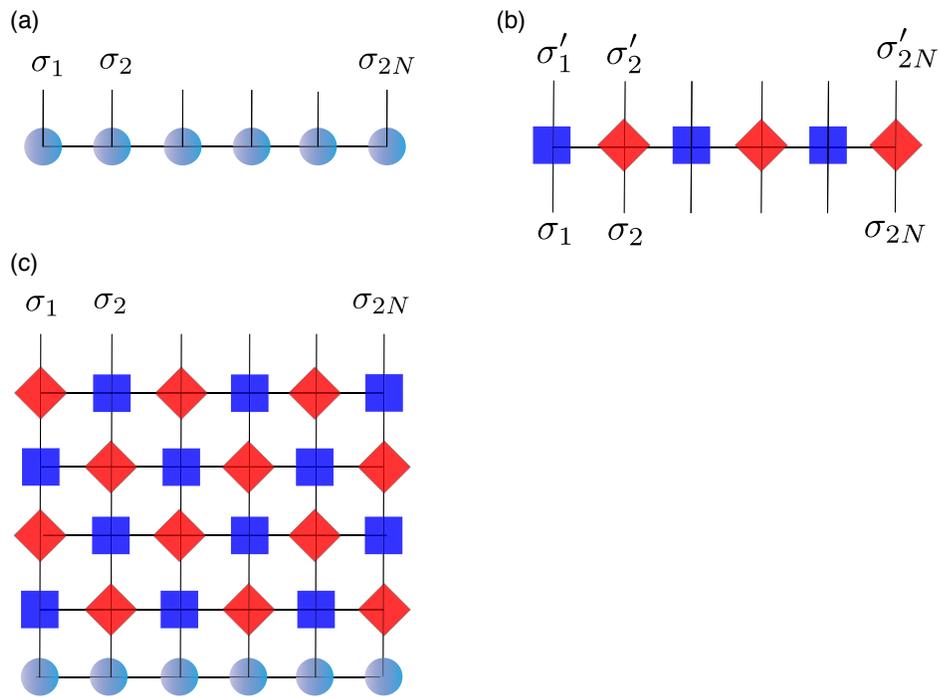


Figure 5.1: Diagrams in the tensor-network algorithm for a one-dimensional system. (a) A matrix-product-state representation of the probability distribution. (b) A matrix-product-operator representation of the transition probability. The squares and rhombuses indicate Y and X tensors defined in Eqs. (5.14) and (5.13), respectively. (c) A diagrammatic representation of the time evolution (5.11).

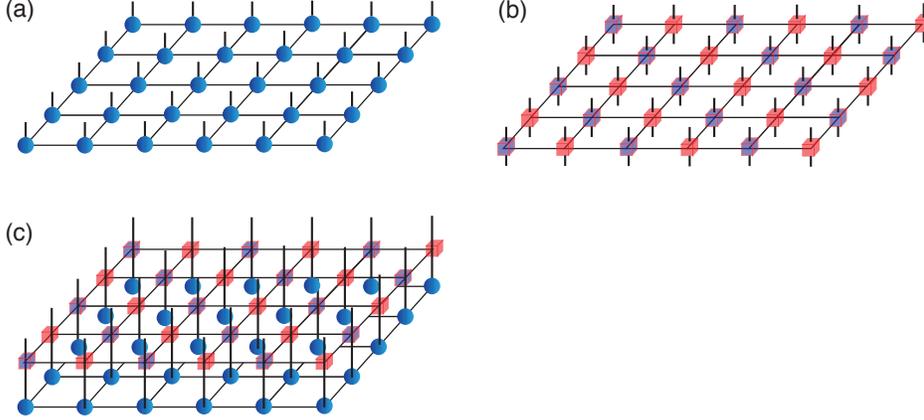


Figure 5.2: Diagrams in the tensor-network algorithm for a two-dimensional system. (a) A tensor-network-state representation of the probability distribution. (b) A tensor-network-operator representation of the transition probability. (c) Time evolution in a single time step, which updates half of the spins (Eq. (5.10)).

Figure 5.1 (b) is a diagrammatic representation of this matrix-product operator. Combining it with the matrix-product-state representation of the probability distribution, we can represent the time evolution of a probability distribution in the form

$$|P(t+1)\rangle = \hat{W}|P(t)\rangle, \quad (5.10)$$

which is followed by

$$|P(t)\rangle = \hat{W}^t|P(0)\rangle. \quad (5.11)$$

The corresponding diagram is Fig. 5.1 (c).

In a two-dimensional system, the probability distribution and the transition probability become a tensor-network state and a tensor-network operator, respectively (Fig. 5.2). We can represent any transition probabilities with tensor-network operators in principle. However, writing down its form is not a trivial problem in practice and requires ingenuity. For the two-sublattice multi-spin-flip dynamics, we can write down the tensor-network operator explicitly as we will explain in the following two sections.

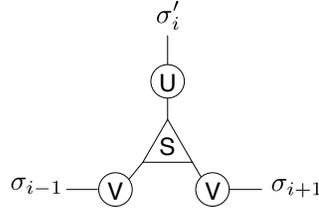


Figure 5.3: A diagrammatic representation of the higher-order singular-value decomposition (HOSVD) of a local transition probability Eq. (5.12).

5.2 One-dimensional kinetic Ising model

We numerically analyze the one-dimensional kinetic Ising model in this section. The update rule is the one explained in Sec. 5.1.1. We prepare all the spins to be $\sigma_i = 1$ at the initial time and observe the relaxation to equilibrium. N. Ito *et al.* derived an asymptotic form of the relaxation of the magnetization analytically [42, 43]. We calculate it numerically and compare the result with their asymptotic form.

5.2.1 Transition matrix as matrix-product operator

We use the transition probabilities given by the order- $4N$ tensors (5.3) and (5.4). We first decompose the local transition probability by using the higher-order singular-value decomposition [44, 73] as

$$w(\sigma_i \rightarrow \sigma'_i) = \frac{e^{1/T \times \sigma'_i (\sigma_{i-1} + \sigma_{i+1})}}{\sum_{\sigma''_i = \pm 1} e^{1/T \times \sigma''_i (\sigma_{i-1} + \sigma_{i+1})}} = \sum_{\alpha, \beta, \gamma=1,2} S_{\alpha\beta\gamma} U_{\sigma'_i\alpha} V_{\sigma_{i-1}\beta} V_{\sigma_{i+1}\gamma}, \quad (5.12)$$

where S is an order-3 tensor called the core tensor, while U and V are 2×2 orthogonal matrices and $1/T$ is the inverse temperature. We perform this decomposition numerically. We can represent this equation using a diagram in Fig. 5.3.

We next define the following order-4 tensors:

$$X(\sigma'_j \sigma_j)_{pq} := V_{\sigma_j p} V_{\sigma'_j q} \delta_{\sigma'_j \sigma_j}, \quad (5.13)$$

$$Y(\sigma'_i)_{\beta\gamma} = Y(\sigma'_i \sigma_i)_{\beta\gamma} := \sum_{\alpha=1,2} S_{\alpha\beta\gamma} U_{\sigma'_i\alpha}. \quad (5.14)$$

We let Y have an index σ_i because we need Y to have both indices σ_i and σ'_i in the definition of a matrix-product operator below. We obtain matrix-product-operator representations of the whole system by lining up X and Y

(Fig. 5.1 (b)). For instance, in the odd time steps,

$$W(\boldsymbol{\sigma} \rightarrow \boldsymbol{\sigma}') = \text{Tr}[Y(\sigma'_1\sigma_1)X(\sigma'_2\sigma_2)Y(\sigma'_3\sigma_3)X(\sigma'_4\sigma_4)\cdots]. \quad (5.15)$$

In the even time steps, the order of X and Y in the trace is interchanged. We thus represent the time evolution by piling up the two matrix-product operators alternatively to the initial matrix-product state (Fig. 5.1 (c)). We set the initial state to be the all-spin-up state, which is a matrix-product state of bond dimension $D = 1$. The bond dimensionality of the matrix-product operator is two.

We calculate $|P(t+1)\rangle$ from $|P(t)\rangle$ using Eq. (5.10). We can represent $|P(t+1)\rangle$ as a matrix-product state since $|P(t)\rangle$ is also a matrix-product state and \hat{W} is a matrix-product operator. Our initial state $|P(0)\rangle$ is the matrix-product state of Eq. (5.7). We can represent the procedure of time evolution graphically. Suppose that Fig. 5.1 (a) is the state at $t = 0$. Applying the matrix-product operator of Fig. 5.1 (b), we obtain the state at $t = 1$. Repeating the procedure, we can temporally evolve the stochastic process. For example, Fig. 5.1 (c) is the state at $t = 4$. In numerical computation, we contract vertical bonds every time we apply a matrix-product operator to a state. Therefore, states are always expressed as matrix-product states. The shape of the state is the same as in Fig. 5.1 (a) but all tensors are not the same for $t \geq 1$ because the applied matrix-product operators consist of the two kinds of matrices (5.13) and (5.14). Each time we apply a matrix-product operator to a state, the bond dimensionality of the state doubles, as the bond dimensionality of the matrix-product operator is two. We would be able to express the state at t exactly as a product of matrices of size 2^t . However, as the amount of computer memory is limited, of course, we need to perform approximation; we truncate the bond dimensions at a dimensionality D by using the singular-value decompositions and keep the D largest singular values. When the dimensionality of every connected bond is less than D , we say that the tensor network has bond dimensions D .

We explain this procedure in detail here, following Ref. [22]. Let us consider approximating the bond connected by the two tensors A_i and A_{i+1} in Eq. (5.6). The product before approximation is

$$\sum_{j=1}^{D'} A_i(\sigma_s)_{i,j} A_{i+1}(\sigma_t)_{j,k}. \quad (5.16)$$

The dimensions of the indices σ_s and σ_t are two, and we assume that the dimensions of the indices i, j , and k are D' ($D' > D$). Our task is to approximate the contraction (5.16), which is the summation of D' terms, by a

5.2. ONE-DIMENSIONAL KINETIC ISING MODEL

summation of D terms. First, we rewrite the tensors with three indices to tensors with two indices (matrices):

$$A_i(\sigma_s)_{i,j} = A'_i(i\sigma_s, j), \quad A_{i+1}(\sigma_t)_{j,k} = A'_{i+1}(j, k\sigma_t). \quad (5.17)$$

Then the contraction (5.16) becomes a multiplication of the new two matrices: $A'_i A'_{i+1}$.

We now approximate the product of two order- D' matrices by a product of two order- D matrices as follows. We first decompose $A'_i A'_{i+1}$ by using the singular-value decomposition:

$$A'_i A'_{i+1} = U \Lambda V^\dagger. \quad (5.18)$$

Here Λ is a diagonal matrix of size D' and we assume that the eigenvalues are sorted in the non-ascending order. We keep only the D largest eigenvalues of Λ and omit the other small eigenvalues:

$$U \Lambda V^\dagger \approx \tilde{U} \tilde{\Lambda} \tilde{V}^\dagger, \quad (5.19)$$

where $\tilde{\Lambda}$ is the diagonal matrix of size D . We left out the last $D' - D$ columns of U and V , and defined the two new matrices \tilde{U} and \tilde{V} . We now approximate A'_i and A'_{i+1} as follows:

$$A'_i \approx \tilde{A}'_i := \tilde{U} \sqrt{\tilde{\Lambda}}, \quad (5.20)$$

$$A'_{i+1} \approx \tilde{A}'_{i+1} := \sqrt{\tilde{\Lambda}} \tilde{V}^\dagger. \quad (5.21)$$

We reshape the matrices $\tilde{A}'_i(i\sigma_s, j)$ and $\tilde{A}'_{i+1}(j, k\sigma_t)$ to tensors with three indices again:

$$\tilde{A}'_i(i\sigma_s, j) = \tilde{A}'_i(\sigma_s)_{i,j}, \quad (5.22)$$

$$\tilde{A}'_{i+1}(j, k\sigma_t) = \tilde{A}'_{i+1}(\sigma_t)_{j,k}, \quad (5.23)$$

where i and k runs over $1, \dots, D'$, j runs over $1, \dots, D$, and each of σ_s and σ_t takes ± 1 . The contraction of the D' terms is finally approximated by the contraction of the D terms:

$$\sum_{j=1}^{D'} A_i(\sigma_s)_{i,j} A_{i+1}(\sigma_t)_{j,k} \approx \sum_{j=1}^D A_i(\sigma_s)_{i,j} A_{i+1}(\sigma_t)_{j,k}. \quad (5.24)$$

The computational complexity of the time evolution is of $O(D^3)$ because the singular-value decomposition takes CPU time of $O(mn^2)$ for an $m \times n$ ($m \geq n$) matrix.

We finally calculate the average magnetization at the odd sites and that at the even sites separately. Let us consider calculating the average of σ_1 as a representative of the odd spins. Suppose that the state at time t is $|P(t)\rangle = \sum_{\boldsymbol{\sigma}} \text{Tr}[A(\sigma_1)B(\sigma_2)A(\sigma_3)B(\sigma_4)\cdots]|\boldsymbol{\sigma}\rangle$. We first compute the marginal distribution of $|\sigma_1\rangle$:

$$P(\sigma_1; t) := \sum_{\sigma_2, \dots, \sigma_{2N}} \text{Tr}[A(\sigma_1)B(\sigma_2)A(\sigma_3)B(\sigma_4)\cdots]. \quad (5.25)$$

The average magnetization of σ_1 is then given by

$$\begin{aligned} \langle \sigma_1 \rangle &= \sum_{\sigma_1=\pm 1} \sigma_1 P(\sigma_1; t) \\ &= \text{Tr}[(A(+1) - A(-1)) \sum_{\sigma_2} B(\sigma_2) \sum_{\sigma_3} A(\sigma_3) \sum_{\sigma_4} B(\sigma_4)\cdots]. \end{aligned} \quad (5.26)$$

We can represent this equation as the diagram shown in Fig. 5.4. Each open circle denotes $\sum_{\sigma=\pm 1} A(\sigma)$, each solid circle denotes $\sum_{\sigma=\pm 1} B(\sigma)$, and the solid rhombus denotes $A(+1) - A(-1)$.

At each time step, we repeat a sufficient number of renormalization until the result converges. We explain the procedure of renormalization in detail. In the first step, we define the two matrices

$$C = (A(+1) - A(-1)) \sum_{\sigma=\pm 1} B(\sigma), \quad (5.27)$$

$$E = \sum_{\sigma_a=\pm 1} A(\sigma_a) \sum_{\sigma_b=\pm 1} B(\sigma_b). \quad (5.28)$$

In the second step, namely the renormalization step, we update C and E as

$$C \leftarrow CE, \quad (5.29)$$

$$E \leftarrow EE. \quad (5.30)$$

Here, the arrow denotes a substitution and CE and EE are matrix-by-matrix multiplications. Every time these substitutions are done, the system size is halved and the targeted matrix C changes taking the effect of its environment matrix E into consideration. We call this step the renormalization step because this procedure is similar to the renormalization.

The sum of probabilities $\sum_{\boldsymbol{\sigma}} \text{Tr}[A(\sigma_1)B(\sigma_2)A(\sigma_3)B(\sigma_4)\cdots]$ deviates from unity as time evolves because of truncations by singular-value decompositions. To preserve the normalization of the probability, we divide the magnetization by the norm of the probability distribution at time t as in Fig. 5.4.

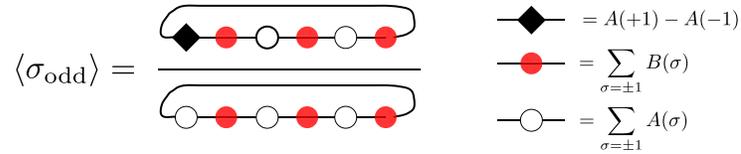


Figure 5.4: A diagrammatic representation of the average of the spins at odd sites. The denominator is the norm of probability. The right figure shows the definition of each symbol.

The average magnetization at odd sites is

$$\langle \sigma_{\text{odd}} \rangle = \frac{\text{Tr } C}{\text{Tr } E}. \quad (5.31)$$

We repeat this renormalization procedure until the average magnetization $\langle \sigma_{\text{odd}} \rangle$ converges.

The calculation of the magnetization at even sites is similar. The magnetization of the whole system is the average of these two averaged magnetizations. The evaluation of the diagram in Fig. 5.4 takes CPU time of $O(D^3)$. The total computational complexity of our algorithm is thus of $O(D^3)$.

We show the results of the calculation in Fig. 5.5. The broken line in the figure is the analytic asymptotic form [42, 43]; the magnetization decays exponentially with the correlation time

$$\xi_t = \frac{1}{\log(\coth(2/T))}. \quad (5.32)$$

As the bond dimension D increases, the range of time for which we can calculate the magnetization precisely also expands. We were able to reproduce the exponential decay up to around $t \approx 100$ in one dimension with $D = 1024$.

We update half of the tensors at each step of time. An advantage of our algorithm is that we can update half of the system at the same time, making use of the translational variance of the system. We just need to update a single tensor of a sublattice because the tensors of the same sublattice are all the same. In Monte Carlo simulation, on the other hand, flipping a half of the system takes the CPU time that increases linearly in the system size.

5.3 Two-dimensional kinetic Ising model

We numerically analyze the two-dimensional kinetic Ising model in this section. In particular, by using the nonequilibrium-relaxation method, we estimate the dynamical critical exponent z of the two-dimensional Ising model.

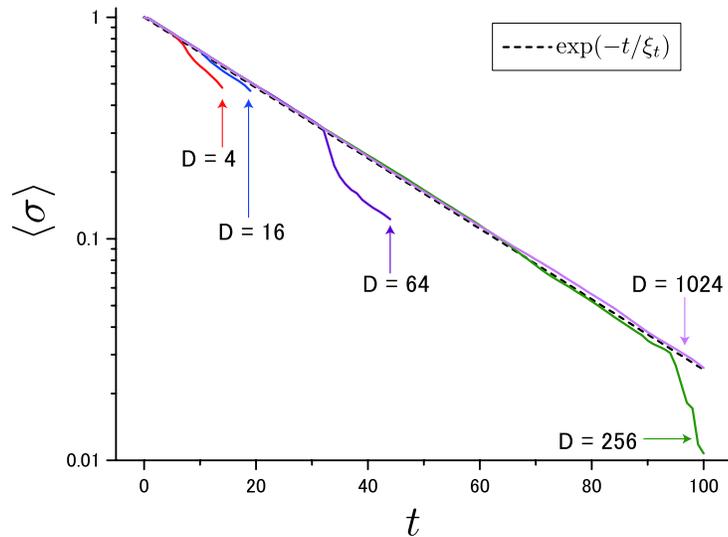


Figure 5.5: The relaxation of the magnetization of the one-dimensional Ising model. The horizontal axis indicates the number of the time steps, while the vertical axis indicates the magnetization. We changed the maximum bond dimension D . The broken line is the analytic asymptotic form [25]; the magnetization decays exponentially with the correlation time ξ_t defined in Eq. (5.32).

5.3.1 Transition operator as tensor-network operator

The update rule of a single spin in the one-dimensional Glauber dynamics, Eq. (5.2), changes in the two-dimensional case to

$$w(\sigma_i \rightarrow \sigma'_i) = \frac{e^{1/T \times \sigma'_i (\sigma_L + \sigma_R + \sigma_D + \sigma_U)}}{\sum_{\sigma''_i = \pm 1} e^{1/T \times \sigma''_i (\sigma_L + \sigma_R + \sigma_D + \sigma_U)}} \quad (5.33)$$

$$= \sum_{\alpha, \beta, \gamma, \delta, \epsilon = 1, 2} S_{\alpha\beta\gamma\delta\epsilon} U_{\sigma'_i\alpha} V_{\sigma_L\beta} V_{\sigma_R\gamma} V_{\sigma_D\delta} V_{\sigma_U\epsilon}, \quad (5.34)$$

where the subscripts L, R, U , and D represent spins to the left, right, up, and down of the spin σ_i , respectively. As in the one-dimensional case, we perform the higher-order singular-value decomposition (HOSVD) and define local transition operators:

$$X(\sigma'_j \sigma_j)_{pqrs} := V_{\sigma_j p} V_{\sigma_j q} V_{\sigma_j r} V_{\sigma_j s} \delta_{\sigma'_j, \sigma_j}, \quad (5.35)$$

$$Y(\sigma'_i)_{\beta\gamma\delta\epsilon} := Y(\sigma'_i \sigma_i)_{\beta\gamma\delta\epsilon} = \sum_{\alpha=1,2} S_{\alpha\beta\gamma\delta\epsilon} U_{\sigma'_i\alpha}. \quad (5.36)$$

A diagrammatic representation of the transition operator of the whole system consists of the local transition operators X and Y . The bond dimensionality of the transition operator is two; see Fig. 5.2 (b). We calculate the time evolution by stacking up the tensor-network operators for each of odd and even time steps alternatively (Fig. 5.2 (c)).

We calculate the time evolution of a state by contracting the state and tensor-network operators from the bottom layers. The state at $t = 0$ is shown in Fig. 5.2 (a). Applying the tensor-network operator of Fig. 5.2 (b) to this state, we can evolve the dynamics by one step (Fig. 5.2 (c)). We then contract vertical bonds of Fig. 5.2 (c) and obtain the tensor-network state at $t = 1$. The shape of the tensor network state at $t = 1$ is the same as Fig. 5.2 (a), but all tensors are not the same. The lattice consists of two sublattices; tensors of each sublattice share the common tensor. By repeating the same procedure, we can calculate the states of $t \geq 1$, $|P(t)\rangle$. During the calculation of the time evolution, we truncate bond indices by singular-value decompositions if the bond dimensionality exceeds D . We can do these singular-value decompositions in the time of $O(D^5)$ with a little ingenuity [95].

We finally compute the magnetization by repeating renormalization of the tensor network that contains an ‘‘impurity tensor’’ (the solid rhombus in Fig. 5.4) until convergence as in the one-dimensional case. We also divide the magnetization, which corresponds to the numerator of Fig. 5.4, by the norm

of $|P(t)\rangle$ (the denominator of the same figure) as in the one-dimensional case. We use the algorithm by C. Wang *et al.* [95], in which one truncates bond dimensions by singular-value decompositions during renormalization, whereas the other parts of the procedure are the same as the tensor renormalization group with the higher-order singular-value decomposition [30]. The calculation of the average magnetization by tensor renormalization group takes the time of $O(D^8)$ [95] and the total computational complexity of our algorithm in two dimensions is also of $O(D^8)$.

5.3.2 Nonequilibrium relaxation

We observe the relaxation from the all-spin-up initial state to the equilibrium state. The initial state is all-spin-up, which is a two-dimensional tensor-network state of the bond dimension $D = 1$. The asymptotic behavior of the magnetization depends on which phase the system is in. In our case, since we do not apply a magnetic field, the asymptotic decay becomes as follows:

$$\langle \sigma \rangle = \begin{cases} e^{-t/\xi_t} & T > T_c, \\ t^{-\lambda_m} & T = T_c, \\ m_{\text{eq}} + ce^{-t/\xi_t} & T < T_c, \end{cases} \quad (5.37)$$

where $T_c = 2.269\dots$ [96], ξ_t is the relaxation time, m_{eq} is the spontaneous magnetization, c is a constant, and λ_m is the dynamical critical exponent that characterizes the power-law decay of the magnetization at the critical point. It is related with the standard critical exponents as [24, 25]

$$\lambda_m = \frac{\beta}{z\nu}. \quad (5.38)$$

In the two-dimensional Ising model, the critical exponents β and ν have been obtained analytically at $1/8$ and 1 , respectively, and therefore we can evaluate z from the decay of the magnetization at the critical point. This method is called the nonequilibrium-relaxation method [24, 25].

We calculated the relaxation of systems in the high-temperature phase and at the critical point (Fig. 5.6). In the high-temperature phase, the magnetization decays exponentially in time with the correlation time ξ_t , while at the critical point ξ_t diverges, and the magnetization shows a power-law decay. To calculate the critical exponent z precisely, we calculated “local exponents” and extrapolated the results to the limit of the infinite time [24,

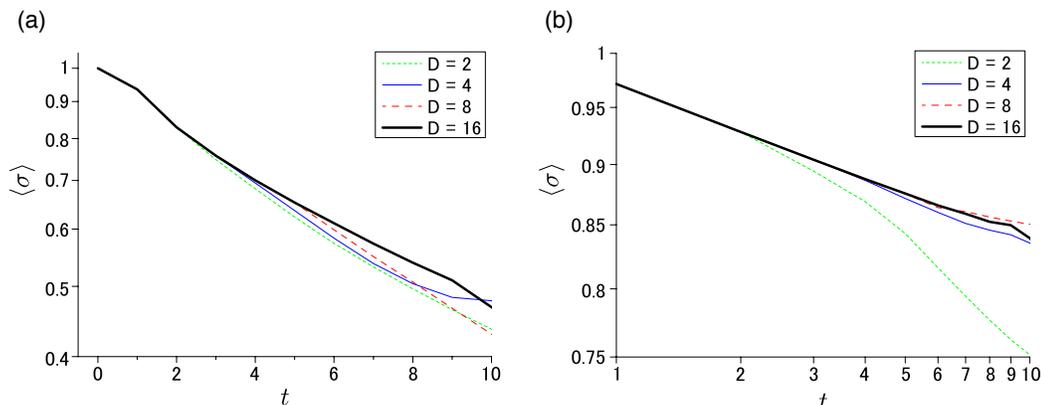


Figure 5.6: Relaxation of the magnetization from the all-spin-up state in the two-dimensional Ising model. The horizontal axis indicates the time steps and the y -axis indicates the magnetization, while D denotes the bond dimensionality of tensors. (a) A semi-logarithmic plot for $T = 3 > T_c$ under no magnetic field. (b) A logarithmic plot at the critical point $T = T_c \approx 2.269$.

25]. We define the local exponents by

$$\lambda_m(t) := -\frac{d \log m(t)}{d \log t} \approx \frac{t}{\Delta t} \left(\frac{m(t - \Delta t)}{m(t)} - 1 \right), \quad (5.39)$$

$$z(t) := \frac{\beta}{\nu \lambda_m(t)}. \quad (5.40)$$

We choose $\Delta t = 1$ and fit the series $z(t)$ to

$$z(t) = a/t + z, \quad (5.41)$$

where a is a constant and z is the final estimate of the critical exponent when the numbers of time steps are extrapolated to infinity. We did the extrapolation by the Bayesian linear regression [97] (Fig. 5.7). The intersection of the line of the best fit and the y -axis is the point estimate of z . We used the 75 % credible interval at the origin of the x -axis as the uncertainty of our estimate of z .

We calculated z for various bond dimensions ranging from $D = 2$ to $D = 20$ and carried out the analysis described as above. In the extrapolation of the local exponent $z(t)$, we did not use the first two data points corresponding to $t = 0, 1$. We used five data points ($t = 2, \dots, 6$) for $D < 17$ and six data points ($t = 2, \dots, 7$) for $D \geq 17$. As Fig. 5.8 shows, the estimates

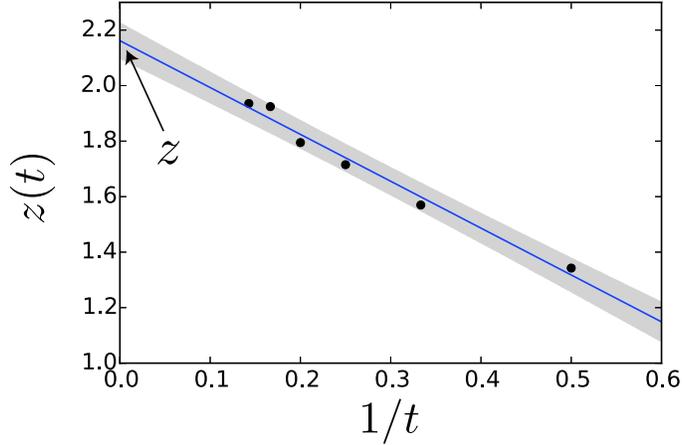


Figure 5.7: Extrapolation of the local exponent $z(t)$ to $t \rightarrow \infty$ according to Eq. (5.41). We performed the Bayesian linear regression. The line of the best fit is the maximum *a posteriori* solution and the shaded area is the 75 % credible interval. The intersection of the line of the best fit and the y -axis is the final estimate of the critical exponent z . The data is for the bond dimension of tensors $D = 20$.

of z converges as D increases. We estimated the critical exponents z to be 2.16(5) using data of $D = 20$. The nonequilibrium-relaxation method with a Monte Carlo method [24] estimated z to be $z = 2.165(10)$ and a series expansion [98] estimated z to be $z = 2.183(5)$. Our result is consistent with the values of these studies.

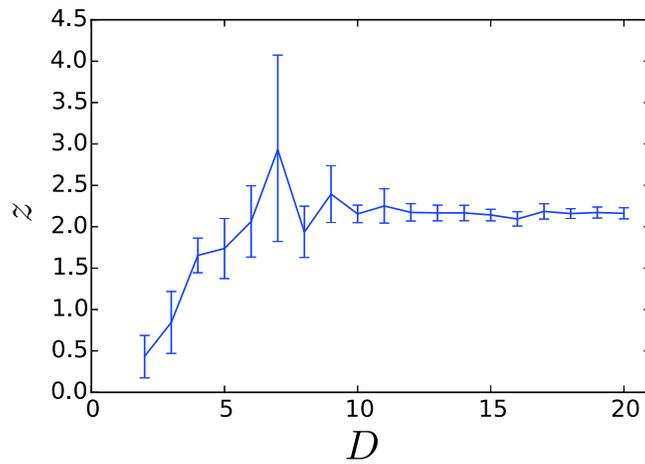


Figure 5.8: Estimates of the dynamical critical exponent z with various bond dimensions D . The error bars indicate the 75 % credible intervals of the predictive distribution at $1/t = 0$ in Eq. (5.41), *i.e.*, the shaded area at the origin of the x -axis in Fig. 5.7.

Chapter 6

Summary

We summarize the results of Chaps. 3 and 5.

In Chap. 3, we studied the scaling properties of the self-avoiding walk on the (u, v) -flower. By obtaining the connective constant μ and the critical exponent ν , we studied (i) how the number of paths with a fixed length and a fixed starting point increases and (ii) how the mean shortest distance between the two end points grows as the path length increases.

In the Euclidian space, it is widely believed that the critical exponents of the self-avoiding walk depends only on the Euclidian dimension. In contrast, for the generalized problem on fractals, it has been conjectured since the 1980s [6, 9, 10] that critical exponents of the self-avoiding walk are not determined only by the similarity dimension [10]. Critical exponents are difficult to obtain exactly except for a few cases in the Euclidian spaces [7, 6] and hence the direct verification of the conjecture had never been done. Adaptation of the (u, v) -flowers, on which we can easily carry out the renormalization-group analysis, enabled us to confirm this conjecture directly. We expect that fractal complex networks are useful in understanding scaling of other stochastic models in fractals as well.

In contrast to scaling theories for Markovian processes on complex networks [58, 99, 100, 101], those for non-Markovian processes, such as the self-avoiding walk, are poorly understood. The present methodology of the renormalization group, however, is also applicable to non-Markovian dynamics on graphs as well as Markovian processes. We believe that this direction of research will deepen our understanding of non-Markovian dynamics on complex networks.

It will be interesting to consider the extension of Flory's approximation of ν to fractal graphs. Flory's approximation is to approximate ν by the Euclidian dimension: $\nu = 3/(2 + d)$. Extension of this formula to fractals embedded in the Euclidian space has been studied [6, 9, 10]. It is impos-

sible to apply the renormalization-group analysis used here to real complex networks because graphs for which the exact renormalization can be applied are limited; this is the fundamental limit of the present renormalization-group analysis. We, however, expect that the scaling properties of paths on networks are determined only by a few parameters including not only the similarity dimension but also other parameters. Our exact results will serve as a basic model to develop a scaling theory which is generally applicable. For example, we found that the speed of the increase of the mean shortest distance between end-to-end points is related with the critical exponent of the zero-component ferromagnet on the (u, v) -flower as in Fig. (3.9). If ν can be expressed as a function of a few parameters, we can predict the number of the s - t paths and the mean shortest end-to-end distance. It is also of interest to study whether the critical exponent ν of the \mathcal{N} -vector model in the limit $\mathcal{N} \rightarrow 0$ is the same on other fractal networks as that of the displacement exponent ν , which is defined by $\overline{d_k^{(s)}} \approx k^\nu$.

In Chap. 5, we have proposed an algorithm to compute sublattice-flip dynamics on an infinite bipartite graph. We split local transition probabilities into tensors that depend only on a single spin variable by a higher-order singular-value decomposition and constructed a tensor-network-operator representation of the transition probability of the whole system. We can apply this approach to any sublattice-update dynamics with nearest-neighbor interactions on a bipartite graph. The order of tensors is determined by the degree of a graph (lattice).

For example, on a honeycomb lattice, the order of tensor is four, because a node is surrounded by three nodes with an additional leg representing the spin index. We can calculate the time evolution of magnetization of a honeycomb lattice Ising model in the following way. We first compute the time evolution of probability distribution by stacking up tensor network operators as in Fig. 5.2. Similarly to the case of a square lattice in Sec. 5.3.1, the bond dimension doubles at each step, and hence we need to cut off the bond dimensionality at D . This procedure takes $O(D^4)$ [95]. Next, we calculate the magnetization by contracting the whole lattice to a single point. This contraction can be done, for instance, by tensor renormalization group [28] with computational time $O(D^6)$. Therefore, the total computational complexity for a honeycomb lattice is $O(D^6)$.

We treat the infinite system utilizing translational invariance and obtained the magnetization in the thermodynamic limit directly without system-size extrapolation. Instead of extrapolating the system size to infinity, we did the bond-dimension extrapolation. We are also able to apply our algorithm to an open boundary system in principle. The system, however, becomes

inhomogeneous, and hence we will need to keep all the tensors that lie on all the sites with the computational costs depending on the system size. Therefore, studying dynamics in the thermodynamics limit directly is only possible when we adopt the periodic boundary condition.

Our algorithm goes together well with the nonequilibrium-relaxation method. In the nonequilibrium-relaxation method, one prepares a large system and simulate it for a rather small number of time steps [25]. In calculation of the time evolution with tensor networks, we cannot compute dynamics for an arbitrary long time because an error due to singular-value decompositions accumulates during the time evolution. We can, however, update an infinite number of spins at once utilizing translational invariance. We calculated the time evolution of the infinite system for a short period indeed with good precision and were able to determine the critical exponent z .

We estimated only z among critical exponents because the other exponents have been analytically known. The nonequilibrium-relaxation method, however, can calculate the other critical exponents as well as the critical temperature, and thus we can use it even for systems for which analytical calculation is intractable. The nonequilibrium-relaxation method combined with our algorithm is a new direction of study of critical phenomena with tensor networks.

A shortcoming of our algorithm at present is that we can utilize it for a limited range of time. This problem is serious in two dimensions; we obtained the results only for $t < 10$ because it became difficult to increase bond dimensions further as the dimensionality increased. The accuracy of our estimate of the critical exponent z is worse than the estimate of the nonequilibrium-relaxation method with a Monte Carlo simulation [24] because we used a smaller number of time steps. In order to compute longer and to improve the accuracy of the estimate, we need to develop a better scheme to truncate bond dimensions during the time evolution.

In summary, we proposed both exact and approximate theories of renormalization group that becomes possible only after treating systems as networks. Viewing systems as networks enabled us to propose novel application of renormalization group. We showed two examples of such new applications in this thesis. In the first example, we revealed the compatibility of renormalization group and network structures. Because renormalization group repeatedly changes scales of systems, it is natural to consider networks with repeated structure, that is, fractal networks. The combination of renormalization group and the fractal networks led us to exact solutions in arbitrary fractal dimension because we were able to sum up degrees of freedom exactly during the scale changes. In the second half of this thesis, we emphasized that lattices are also networks. The renormalization-group study of lattices

is much more difficult than that of fractal networks because it becomes impossible to sum up degrees of freedom exactly during the scale changes. In such cases, viewing lattices as networks was beneficial. We were able to consider not only finite lattice systems but also infinite lattice systems by constructing networks consisting of many tensors. We summed up degrees of freedom approximately during renormalization. This approximation method is basically the same as in equilibrium systems, non-equilibrium systems, finite systems, and infinite systems. This systematic treatment stems from the network structure of the systems. They are all networks.

Appendix A

Additional results of numerical simulation

We described all important results in Chap. 3, but we here present several additional results.

A.1 The number of paths

A shortcoming of an enumeration algorithm is that long paths are difficult to sample, though what we need is a quantity for large k in Eq. (3.7).

As we noted, the number of paths increases as (3.62) when k is small, and then it starts decreasing due to a finite-size effect (Fig. 3.7). We verified this for various generations of the (2,3)-flower (Fig. A.1).

A.2 The exponent of displacement ν' by the depth-limited search

Longer paths should be obtained in order to determine the exponent ν' accurately. To estimate ν' , therefore, the biased sampling method was more suitable than the depth-limited search. We also estimated ν' by the enumeration algorithm for cross-check (Fig. A.2). The systematic error discussed later is not taken into consideration.

A.2. THE EXPONENT OF DISPLACEMENT ν' BY THE DEPTH-LIMITED SEARCH

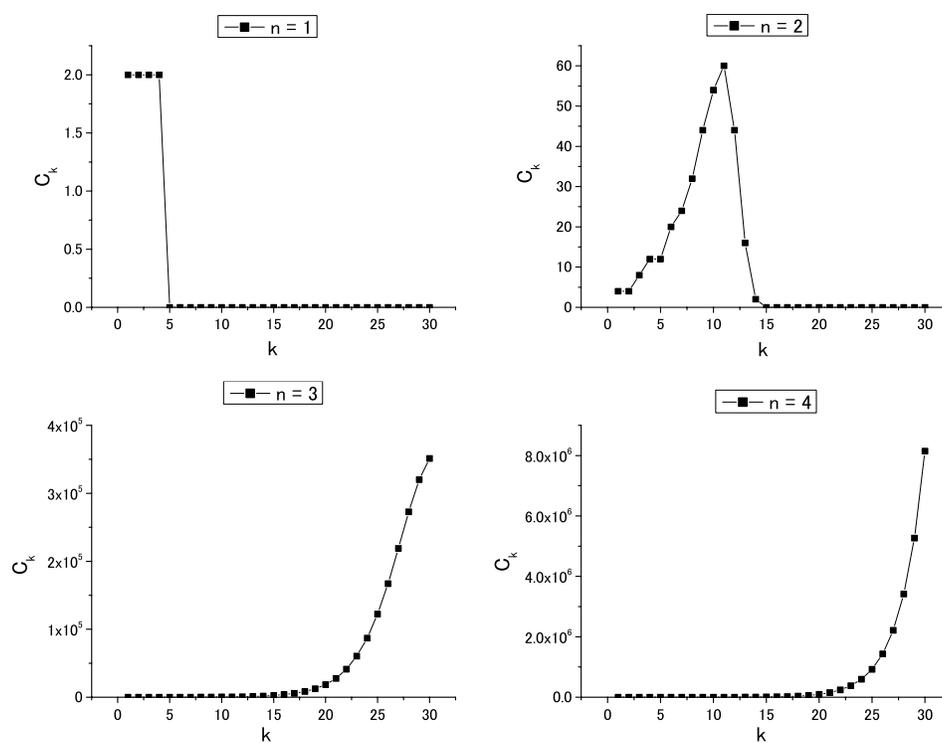


Figure A.1: The number of paths of length k for various generations of the $(2, 3)$ -flower. The number of paths increases as in Eq. (3.62) when k is small, and then it starts decreasing due to a finite-size effect.

APPENDIX A. ADDITIONAL RESULTS OF NUMERICAL SIMULATION

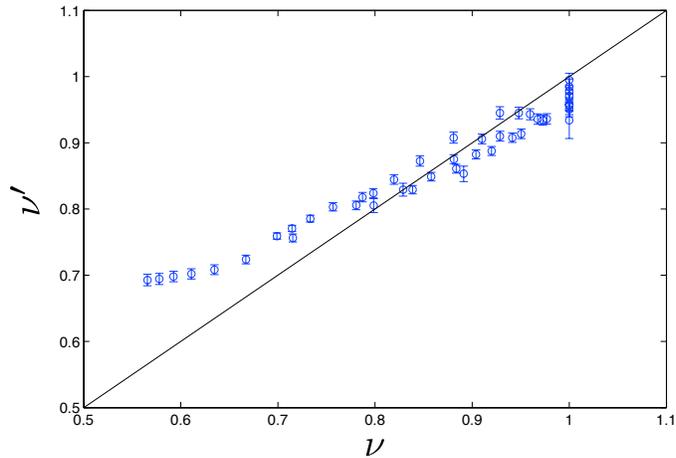


Figure A.2: The critical exponent ν calculated by the renormalization-group analysis and the exponent of displacement ν' estimated by the enumeration algorithm followed by a curve fitting. The points on the left deviate from the line $y = x$ because the mean end-to-end shortest distance $\overline{d_k^{(s)}}$ does not still reach an asymptotic region.

A.2. THE EXPONENT OF DISPLACEMENT ν' BY THE
DEPTH-LIMITED SEARCH

Appendix B

Analysis of the Monte-Carlo simulation

In this appendix, we explain the method used in data analysis of estimation of the exponent of displacement ν' in Fig. 3.9.

The mean shortest distance $\overline{d_k^{(s)}}$ for various values of u and v computed by the biased sampling method is accompanied by an error:

$$(\text{The mean shortest distance from the starting point}) = \overline{d_k^{(s)}} \pm \sigma_k. \quad (\text{B.1})$$

Here $\overline{d_k^{(s)}}$ is the sample mean of the shortest distances $d_k^{(s)}$ over M realizations and σ_k is the sample standard deviation of $d_k^{(s)}$.

Each estimate of ν' is accompanied by two kinds of errors: a statistical error and a systematic error. The statistical error, which is indicated by σ_k , comes from fluctuation of random numbers. The systematic error, on the other hand, is due to insufficient path lengths in our case: for example,

- $\overline{d_k^{(s)}}$ has not reached the asymptotic region;
- $\ln \overline{d_k^{(s)}}$ ripples around the asymptotic line.

What we need to obtain is the asymptotic behavior of $\overline{d_k^{(s)}}$ as $k \rightarrow \infty$. Therefore, ν' estimated by the fitting (3.66) is not accurate if k is too small. Furthermore, we found that $\ln \overline{d_k^{(s)}}$ ripples around the asymptotic line as in Fig. B.1 (a), and the amplitude of oscillation gets smaller as k becomes larger. In other words, the model (3.66) is not correct in a strict sense, because the average $\overline{d_k^{(s)}}$ does not converge to $e^A k$ in the limit $M \rightarrow \infty$. We define σ'_k as

the standard deviation of $\ln \overline{d_k^{(s)}}$:

$$\sigma'_k := [\ln(\overline{d_k^{(s)}} + \sigma_k) - \ln(\overline{d_k^{(s)}} - \sigma_k)]/2. \quad (\text{B.2})$$

Thus, simply fitting $\ln k$ and $\ln \overline{d_k^{(s)}}$ with a weight $1/\sigma_k'^2$ has two downsides. First, because the least-square method minimizes the total of residuals, the fitting is done mainly using the data points of small k , whose oscillation is large, while the data with large values of k are little taken into consideration. Second, the error of the estimate of ν' is underestimated because the systematic error of oscillation is neglected. The problem is that we do not know the amplitude of oscillation, and hence we resort to an *ad hoc* prescription to take the systematic error into consideration.

We obtain the estimates of ν' for each (u, v) -flower in the following procedure:

1. Generate M pieces of paths and compute the sample mean shortest distance $\overline{d_k^{(s)}}$ and the sample mean standard deviation of $\overline{d_k^{(s)}}$, *i.e.*, σ_k .
2. Remove the first data of $\overline{d_k^{(s)}}$ whose sample mean standard deviations are zero.
3. If there still remains $\overline{d_k^{(s)}}$ whose sample mean standard deviation is zero, then set σ_k to the average of the sample mean standard deviations of neighboring data points. Compute σ'_k defined by Eq. (B.2).
4. Divide the data points into n_b bins of the same width in $\ln k$.
5. Do fitting inside each bin using Eq. (3.66) with a weight $1/\sigma_k'^2$ and calculate the root mean square of the residuals of $\ln \overline{d_k^{(s)}}$. We denote this root mean square by s_i . Let the mean of $\ln k$ in each bin be $\ln k_i$ and that of $\ln \overline{d_k^{(s)}}$ be $\ln \overline{d_{k_i}^{(s)}}$.
6. Fit the data $\ln k_i$ and $\ln \overline{d_{k_i}^{(s)}}$ ($i = 1, \dots, n_b$) in all bins with a weight $1/s_i^2$ using the model (3.66) (Fig. B.1 (b)). We denote the error of the estimate of ν' as $\delta\nu'$.

An example of plot before and after the coarse-graining procedure is given in Fig. B.1. Step 2 is intended to remove the region where the distance increases linearly. Otherwise s_1 would become zero in Step 5. Step 3 is necessary to fit $\ln \overline{d_k^{(s)}}$ with finite weights in Step 5. Steps 5 and 6 are done to coarse-grain

APPENDIX B. ANALYSIS OF THE MONTE-CARLO SIMULATION

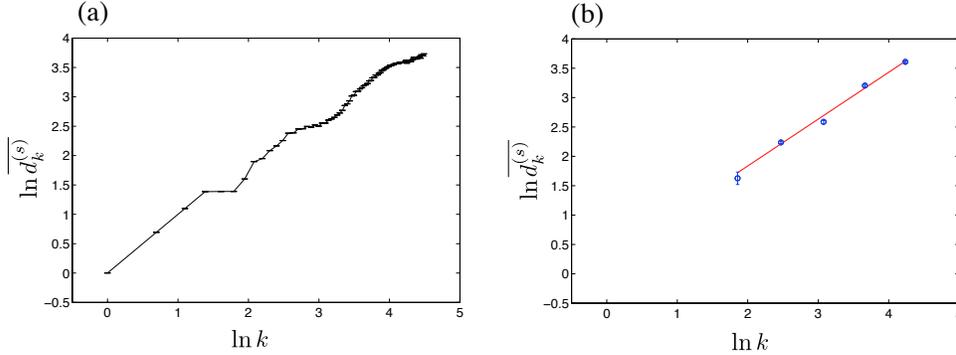


Figure B.1: An example of coarse-graining procedure. The data (a) before and (b) after coarse-graining. We computed $\ln \overline{d_{k_i}^{(s)}}$ by the biased-sampling algorithm. The simulation condition was $M = 10,000$, $n = 4$, $u = 3$, and $v = 5$. After the coarse-graining described in the text, we fit representative points in five bins ($n_b = 5$) to the linear function (3.66).

data so as to assign a larger weight to the data with larger k and to avoid the underestimation of the error of the estimate of ν' by taking account of the effect of oscillation as a systematic error.

Thus obtained estimates of ν' are plotted in Fig. 3.9 when the number of bins is $n_b = 5$.



Bibliography

- [1] D. Knuth. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*. Pearson Education India, 2011.
- [2] M. Bousquet-Mélou, A. J. Guttmann, and I. Jensen. Self-avoiding walks crossing a square. *Journal of Physics A: Mathematical and General*, 38(42):9159, 2005.
- [3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, 2009.
- [4] P. G. de Gennes. *Scaling concepts in polymer physics*. Cornell university press, New York, 1979.
- [5] P. G. de Gennes. Exponents for the excluded volume problem as derived by the wilson method. *Phys Lett A*, 38(5):339–340, 1972.
- [6] R. Rammal, G. Toulouse, and J. Vannimenus. Self-avoiding walks on fractal spaces: exact results and flory approximation. *Journal de Physique*, 45(3):389–394, 1984.
- [7] D. Dhar. Self-avoiding random walks: Some exactly soluble cases. *Journal of Mathematical Physics*, 19(1):5–11, 1978.
- [8] N. N. Madras and G. Slade. *The self-avoiding walk*. Birkhäuser, Boston, 1996.
- [9] S. Havlin and D. Ben-Avraham. Diffusion in disordered media. *Advances in Physics*, 36(6):695–798, 1987.
- [10] A. Aharony and A. B. Harris. Flory approximant for self-avoiding walks on fractals. *Journal of Statistical Physics*, 54(3-4):1091–1097, 1989.
- [11] C. Song, S. Havlin, and H. A. Makse. Self-similarity of complex networks. *Nature*, 433(7024):392–395, 2005.

- [12] L. K. Gallos, C. Song, and H. A. Makse. A review of fractality and self-similarity in complex networks. *Physica A: Statistical Mechanics and its Applications*, 386(2):686–691, 2007.
- [13] C. Song, S. Havlin, and H. A. Makse. Origins of fractality in the growth of complex networks. *Nature Physics*, 2(4):275–281, 2006.
- [14] H. D. Rozenfeld, S. Havlin, and D. Ben-Avraham. Fractal and transfractal recursive scale-free nets. *New J Phys*, 9(6):175, 2007.
- [15] Y. Hieida. Application of the density matrix renormalization group method to a non-equilibrium problem. *Journal of the Physical Society of Japan*, 67(2):369–372, 1998.
- [16] E. Carlon, M. Henkel, and U. Schollwöck. Density matrix renormalization group and reaction-diffusion processes. *Eur. Phys. J. B*, 12(1):99–114, 1999.
- [17] E. Carlon, M. Henkel, and U. Schollwöck. Critical properties of the reaction-diffusion model $2 \vec{A} \rightarrow 3a, 2 \vec{A} \rightarrow 0$. *Phys Rev E*, 63(3):036101, 2001.
- [18] Z. Nagy, C. Appert, and L. Santen. Relaxation times in the asep model using a dmrg method. *Journal of Statistical Physics*, 109(3-4):623–639, 2002.
- [19] A. Kemper, A. Schadschneider, and J. Zittartz. Transfer-matrix density-matrix renormalization group for stochastic models: the domany-kinzel cellular automaton. *Journal of Physics A: Mathematical and General*, 34(19):L279, 2001.
- [20] K. Temme and F. Verstraete. Stochastic matrix product states. *Phys Rev Lett*, 104(21):210502, 2010.
- [21] R. Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics*, 349(0):117–158, 2014.
- [22] U. Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011.
- [23] F. Verstraete, V. Murg, and J. I. Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57(2):143–224, 2008.

BIBLIOGRAPHY

- [24] N. Ito. Non-equilibrium relaxation and interface energy of the ising model. *Physica A: Statistical Mechanics and its Applications*, 196(4):591–614, 1993.
- [25] Y. Ozeki and N. Ito. Nonequilibrium relaxation method. *Journal of Physics A: Mathematical and Theoretical*, 40(31):R149, 2007.
- [26] Y. Ozeki and N. Ito. Nonequilibrium relaxation study of ising spin glass models. *Physical Review B*, 64(2):024416, 2001.
- [27] U. Schollwöck. The density-matrix renormalization group. *Reviews of Modern Physics*, 77(1):259, 2005.
- [28] M. Levin and C. P. Nave. Tensor renormalization group approach to two-dimensional classical lattice models. *Phys Rev Lett*, 99(12):120601, 2007.
- [29] H. H. Zhao, Z. Y. Xie, Q. N. Chen, Z. C. Wei, J. W. Cai, and T. Xiang. Renormalization of tensor-network states. *Physical Review B*, 81(17):174411, 2010.
- [30] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang. Coarse-graining renormalization by higher-order singular value decomposition. *Physical Review B*, 86(4):045139, 2012.
- [31] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang. Second renormalization of tensor-network states. *Phys Rev Lett*, 103(16):160601, 2009.
- [32] T. Nishino. Density matrix renormalization group method for 2d classical models. *Journal of the Physical Society of Japan*, 64(10):3598–3601, 1995.
- [33] T. Nishino, Y. Hieida, K. Okunishi, N. Maeshima, Y. Akutsu, and A. Gendiar. Two-dimensional tensor product variational formulation. *Progress of Theoretical Physics*, 105(3):409–417, 2001.
- [34] Z.-C. Gu, H.-C. Jiang, D. N. Sheng, H. Yao, L. Balents, and X.-G. Wen. Time-reversal symmetry breaking superconducting ground state in the doped mott insulator on the honeycomb lattice. *Physical Review B*, 88(15):155112, 2013.
- [35] T. H. Johnson, S. R. Clark, and D. Jaksch. Dynamical simulations of classical stochastic systems using matrix product states. *Phys Rev E*, 82(3):036702, 2010.

-
- [36] T. H. Johnson, T. J. Elliott, S. R. Clark, and D. Jaksch. Capturing exponential variance using polynomial resources: Applying tensor networks to nonequilibrium stochastic processes. *Phys Rev Lett*, 114(9):090602, 2015.
- [37] G. Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Phys Rev Lett*, 93(4):040502, 2004.
- [38] M. Suzuki. Relationship between d-dimensional quantal spin systems and (d+ 1)-dimensional ising systems equivalence, critical exponents and systematic approximants of the partition function and spin correlations. *Progress of Theoretical Physics*, 56(5):1454–1469, 1976.
- [39] R. J. Glauber. Timedependent statistics of the ising model. *Journal of Mathematical Physics*, 4(2):294–307, 1963.
- [40] H. Yahata. Critical relaxation of stochastic ising model. *Journal of the Physical Society of Japan*, 30(3):657–666, 1971.
- [41] P. C. Hohenberg and B. I. Halperin. Theory of dynamic critical phenomena. *Reviews of Modern Physics*, 49(3):435–479, 1977.
- [42] N. Ito. Discrete-time and single-spin-flip dynamics of the ising chain. *Progress of Theoretical Physics*, 83(4):682–692, 1990.
- [43] N. Ito and T. Chikyu. Multi-spin-flip dynamics of the ising chain. *Physica A: Statistical Mechanics and its Applications*, 166(2):193–205, 1990.
- [44] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
- [45] G. Vidal. Classical simulation of infinite-size quantum lattice systems in one spatial dimension. *Phys Rev Lett*, 98(7):070201, 2007.
- [46] R. Orús and G. Vidal. Infinite time-evolving block decimation algorithm beyond unitary evolution. *Physical Review B*, 78(15):155117, 2008.
- [47] B. B. Mandelbrot. *The fractal geometry of nature*. The Fractal Geometry of Nature, New York, 1983.
- [48] K. Falconer. *Fractal geometry: mathematical foundations and applications*. Wiley, New York, 2007.

BIBLIOGRAPHY

- [49] K. Yakubo. *Complex network and its structure (in Japanese)*. Kyoritsu Shuppan, Tokyo, 2013.
- [50] Wikipedia. Cantor set — wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Cantor_set; Online accessed 13-December-2016.
- [51] B. B. Mandelbrot. How long is the coast of britain. *Science*, 156(3775):636–638, 1967.
- [52] Wikipedia. Coastline of the united kingdom — wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Coastline_of_the_United_Kingdom; Online accessed 27-November-2016.
- [53] T. A. Witten and L. M. Sander. Diffusion-limited aggregation, a kinetic critical phenomenon. *Phys Rev Lett*, 47(19):1400–1403, 1981.
- [54] R. J. Wilson. *Introduction to Graph Theory*. Prentice Hall, Essex, 2010.
- [55] P. Erdős and A. Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.
- [56] S. N. Dorogovtsev and J. F. F. Mendes. *Evolution of networks: From biological nets to the Internet and WWW*. Oxford University Press, Oxford, 2003.
- [57] D. J. Watts. *Small worlds: the dynamics of networks between order and randomness*. Princeton University Press, Princeton, 1999.
- [58] A. Barrat, M. Barthélemy, and A. Vespignani. *Dynamical processes on complex networks*, volume 1. Cambridge University Press, Cambridge, 2008.
- [59] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47, 2002.
- [60] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks. *Advances in Physics*, 51(4):1079–1187, 2002.
- [61] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.
- [62] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. Critical phenomena in complex networks. *Reviews of Modern Physics*, 80(4):1275, 2008.

- [63] M. E. J. Newman, A.-L. Barabási, and D. J. Watts. *The structure and dynamics of networks*. Princeton University Press, Princeton, 2006.
- [64] D. J. Watts and S. H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442, 1998.
- [65] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
- [66] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. Pseudofractal scale-free web. *Phys Rev E*, 65(6):066122, 2002.
- [67] P. J. Flory. *Principles of polymer chemistry*. Cornell University Press, New York, 1953.
- [68] B. Shapiro. A direct renormalisation group approach for the self-avoiding walk. *Journal of Physics C: Solid State Physics*, 11(13):2829, 1978.
- [69] Y. Hotta. Self-avoiding walk on fractal complex networks: Exactly solvable cases. *Phys Rev E*, 90(5):052821, 2014.
- [70] C. P. Herrero and M. Saboyá. Self-avoiding walks and connective constants in small-world networks. *Phys Rev E*, 68(2):026106, 2003.
- [71] C. P. Herrero. Self-avoiding walks on scale-free networks. *Phys Rev E*, 71(1):016103, 2005.
- [72] K. Binder and D. W. Heermann. *Monte Carlo Simulation in Statistical Mechanics: An Introduction*. Springer, Heidelberg, 5th edition, 2010.
- [73] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.
- [74] L. De Lathauwer. *Signal processing based on multilinear algebra*. PhD thesis, 1997.
- [75] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, 4 edition, 2012.
- [76] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition. *Psychometrika*, 35(3):283–319, 1970.
- [77] R. A. Harshman. Foundations of the parafac procedure: Models and conditions for an “explanatory” multi-modal factor analysis. 1970.

BIBLIOGRAPHY

- [78] J. Håstad. Tensor rank is np-complete. *Journal of Algorithms*, 11(4):644–654, 1990.
- [79] T. G. Kolda. Orthogonal tensor decompositions. *SIAM Journal on Matrix Analysis and Applications*, 23(1):243–255, 2001.
- [80] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the best rank-1 and rank-(r_1, r_2, \dots, r_n) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.
- [81] I. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.
- [82] S. R. White. Density matrix formulation for quantum renormalization groups. *Phys Rev Lett*, 69(19):2863–2866, 1992.
- [83] S. R. White. Density-matrix algorithms for quantum renormalization groups. *Physical Review B*, 48(14):10345–10356, 1993.
- [84] F. Verstraete, D. Porras, and J. I. Cirac. Density matrix renormalization group and periodic boundary conditions: A quantum information perspective. *Phys Rev Lett*, 93(22):227205, 2004.
- [85] S. V. Dolgov, B. N. Khoromskij, I. V. Oseledets, and D. V. Savostyanov. Computation of extreme eigenvalues in higher dimensions using block tensor train format. *Computer Physics Communications*, 185(4):1207–1216, 2014.
- [86] J. I. Cirac and F. Verstraete. Renormalization and tensor product states in spin chains and lattices. *Journal of Physics A: Mathematical and Theoretical*, 42(50):504004, 2009.
- [87] F. Verstraete and J. I. Cirac. Renormalization algorithms for quantum-many body systems in two and higher dimensions. *arXiv preprint cond-mat/0407066*, 2004.
- [88] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal. Time-dependent density-matrix renormalization-group using adaptive effective hilbert spaces. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(04):P04005, 2004.
- [89] G. Vidal. Efficient classical simulation of slightly entangled quantum computations. *Phys Rev Lett*, 91(14):147902, 2003.

-
- [90] G. Evenbly and G. Vidal. Tensor network renormalization. *Phys Rev Lett*, 115(18):180405, 2015.
- [91] I. P. McCulloch. Infinite size density matrix renormalization group, revisited. *arXiv preprint arXiv:0804.2509*, 2008.
- [92] G. M. Crosswhite, A. C. Doherty, and G. Vidal. Applying matrix product operators to model systems with long-range interactions. *Physical Review B*, 78(3):035116, 2008.
- [93] J. Jordan, R. Orús, G. Vidal, F. Verstraete, and J. I. Cirac. Classical simulation of infinite-size quantum lattice systems in two spatial dimensions. *Phys Rev Lett*, 101(25):250602, 2008.
- [94] Y. Hotta. Tensor-network algorithm for nonequilibrium relaxation in the thermodynamic limit. *Phys Rev E*, 93(6):062136, 2016.
- [95] C. Wang, S. Qin, and H. Zhou. Topologically invariant tensor renormalization group method for the edwards-anderson spin glasses model. *Physical Review B*, 90(17):174201, 2014.
- [96] L. Onsager. Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Physical Review*, 65(3-4):117–149, 1944.
- [97] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Taylor & Francis, Boca Raton, 2013.
- [98] B. Dammann and J. D. Reger. High-temperature series expansion for the relaxation times of the two dimensional ising model. *Zeitschrift für Physik B Condensed Matter*, 98(1):97–110, 1995.
- [99] S. Condamin, O. Benichou, V. Tejedor, R. Voituriez, and J. Klafter. First-passage times in complex scale-invariant media. *Nature*, 450(7166):77–80, 2007.
- [100] L. K. Gallos, C. Song, S. Havlin, and H. A. Makse. Scaling theory of transport in complex biological networks. *Proceedings of the National Academy of Sciences*, 104(19):7746–7751, 2007.
- [101] S. Condamin, V. Tejedor, R. Voituriez, O. Benichou, and J. Klafter. Probing microscopic origins of confined subdiffusion by first-passage observables. *Proceedings of the National Academy of Sciences*, 105(15):5675–5680, 2008.