Universality and Scaling in the Random-Field Ising model

Mainou, Argyro

Award date:
2021

Awarding institution:
Coventry University

Coventry University

Doctoral Thesis

Universality and Scaling in the Random-Field Ising model

Author:
Argyro Mainou

Director of study:
Dr. Nikolaos Fytas

Supervisor:
Dr. Martin Weigel

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Faculty of Engineering, Environment and Computing,
Coventry University

7th December 2021

Certificate of Ethical Approval

Applicant:

Argyro Mainou

Project Title:

Universality and scaling in the random-field Ising model

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

23 September 2020

Project Reference Number:

P110308

Please add a copy of this section within the first few pages of your thesis, after your title page. Refer to 'Thesis Information Guidance' for more information.

Section 3 Submission Declaration

Have materials contained in your thesis/submission been used for any other submission for an academic award?	Yes	No
	☐	☒

If you have answered Yes to above please state award and awarding body and list the material:

To the best of my knowledge, there are no health reasons that will prevent me from undertaking and completing this assessment and I will ensure to notify my Director of Studies and the Doctoral College if there is any change to these circumstances	Agree	Disagree
	☒	☐

Ethical Declaration: I declare that my research has full University Ethical approval and evidence of this has been included within my thesis/submission. Please also insert ethics reference number below Project Reference: **P**110308	Yes	No
	☒	☐

Freedom of Information:

Freedom of Information Act 2000 (FOIA) ensures access to any information held by Coventry University, including theses, unless an exception or exceptional circumstances apply.

In the interest of scholarship, theses of the University are normally made freely available online in CURVE, the Institutions Repository, immediately on deposit. You may wish to restrict access to your thesis for a period of three years. Reasons for restricting access to the electronic thesis should be derived from exemptions under FOIA. (Please also refer to the University Regulations Section 8.12.5)

Do you wish to restrict access to thesis/submission: No

If Yes please specify reason for restriction:

Does any organisation, other than Coventry University, have an interest in the Intellectual Property Rights to your work?

No

If Yes please specify Organisation:

Please specify the nature of their interest:

Candidates Signature: This item has been removed	**Date:** 15/4/2021

Coventry University | Doctoral College | Centre for Research Capability and Development

Library Declaration and Deposit Agreement

Section 1: Candidate Information

PGR ID:	Forename:	Family Name:
7380224	**Argyro**	**Mainou**

Section 2: Research Details

Faculty/URC:	Faculty of Engineering, Environment and Computing
Award:	**PhD**
Thesis Title:	Universality and scaling in the random-field Ising model

Freedom of Information:

Freedom of Information Act 2000 (FOIA) ensures access to any information held by Coventry University, including theses, unless an exception or exceptional circumstances apply.

In the interest of scholarship, theses of the University are normally made freely available online in the Institutions Repository, immediately on deposit. You may wish to restrict access to your thesis for a period of up to five years. Reasons for restricting access to the electronic thesis should be derived from exemptions under FOIA.
(Please also refer to the University Regulations Section 8.11.10)

Do you wish to restrict access to thesis/submission:	No

Please note: If your thesis includes your publications in the appendix, please ensure you seek approval from the publisher first, and include their approval with this form. If they have not given approval, they will need to be removed from the version of your thesis made available in the Institutional Repository.

If Yes please specify reason for restriction:

Length of restriction:	
Does any organisation, other than Coventry University, have an interest in the Intellectual Property Rights to your work?	No
If Yes please specify Organisation:	
Please specify the nature of their interest:	
Signature: This item has been	**Date:** 13/10/2021

For Doctoral College and Centre for Research Capability and Development use

Date Final Thesis Submitted	
Date of Thesis release to Library	

In the loving memory of George Cox. . .

'*In science, convictions have no rights of citizenship, as is said with good reason. Only when they decide to descend to the modesty of a hypothesis, of a provisional experimental point of view, of a regulative fiction, maybe they be granted admission and even a certain value within the realm of knowledge - though always with the restriction that they remain under police supervision, under the police of mistrust. But does this not men, more precisely considered, that a conviction may obtain admission to science only when it ceases to be a conviction? Would not the discipline of the scientific spirit begin with this, no longer to permit oneself any convictions? Probably that is how it is. But one must still ask whether it is not the case that, in order that this discipline could begin, a conviction must have been there already, and even such a commanding and unconditional one that it sacrificed all other convictions for its own sake. It is clear that science too rests on a faith; there is no science 'without presuppositions'. The question whether truth is needed must not only have been affirmed in advance, but affirmed to the extend that the principle, the faith, the conviction is expressed: 'nothing is needed more than truth, and in relation to it everything else has only second-rate value'.*

...

Consequently, 'will to truth' does not mean 'I will not let myself be deceived' but - there is no choice - 'I will not deceive, not even myself': and with this we are on the ground of morality. For one should ask oneself carefully: 'Why don't you want to deceive?' especially if it should appear - and it certainly does appear - that life depends on appearance; I mean, on error, simulation, deception, self-deception; and when life has, as a matter of fact, always shown itself to be on the side of the most unscrupulous polytropoi. Such an intent charitably interpreted, could perhaps be a quixotism, a little enthusiastic impudence; but it could also be something worse, namely, a destructive principle, hostile to life. 'Will to truth' - that might be a concealed will to death.'

Friedrich Nietzsche

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my director of studies Ass. Prof. Nikolaos Fytas and my supervisor Prof. Martin Weigel for their continuous support throughout my entire research experience. The completion of my thesis would not be carried out without their persistence guidance and motivation. They both constitute examples of exceptional scientists and supervisors who every PhD student desires to have and apart from helping me achieve my academic goals, they have taught me, whether intentionally or not, rare virtues like academic ethos, altruistic teamwork and contribution to science, patience and gratefulness. Above all, they have vividly revealed to me the romantic and beautiful sides of science and they have generously allowed me to take a glimpse into them through their knowledge, character and experience. It has been an honour to work with both of them.

Furthermore, I wish to thank all members of the Statistical Physics Group of the Fluids and Complex Systems Research Centre of Coventry University for creating the best possible environment to complete my studies. In particular I feel indebted to Prof. Alban Potherat and Dr. Lorna Everall for their invaluable support regarding my health and financial issues during the whole period of my postgraduate studies. Also, I would like to acknowledge the allocation of significant CPU time on the supercomputers Zeus and Pluto of Coventry University and the precious help of our administrator Ass. Prof. Alex Pedcenko. My gratitude also goes to Ass. Prof. Abhishek Kumar, not only for his essential help with computational techniques, but also for standing by me as a friend, motivating me and supporting me during difficult times. What is more, I want to particularly acknowledge the contribution of Dr. Gerasimos Politis for assisting me with LaTeX. A special thank you goes to Dr. Sanjay Singh, who stood by my side as a great friend throughout my whole experience inside and outside of work, handling me at my best and at my worst times, encouraging me and supporting me all the way. Furthermore, I feel the need to thank my former PhD fellows Dr. Petro Sarkanyach, Dr. Robin de Regt and Dr. Joe Yoz for their helpful advice and discussions regarding not only physics, but life in general. My gratitude also goes to my good friends Evangelia Bisketzi, Evi Triantafyllidou and

Ioannis Sidiropoulos for their continuous support and for being loving and caring during all these years. I also wish to particularly thank Dr. Irini Tsirka and Kimon Spyridopoulos for their exceptional professional help with my health issues throughout this whole period. Last but not least, I wish to thank my family for their support during my research.

Abstract

In this thesis, we examine various aspects of phase transitions and critical phenomena of pure and disordered magnetic systems using a variety of numerical approaches. In particular, we employ the Metropolis and Wolff algorithms in order to study the finite-size scaling of the interfacial adsorption of the two-dimensional Blume-Capel model at both its first- and second-order transition regimes, as well as at the vicinity of the tricritical point. What is more, we review the size dependence of the interfacial adsorption under the presence of quenched bond randomness at the originally first-order transition regime and the relevant self-averaging properties of the system. Following, we turn our focus on exact ground-state calculations with the use of graph cut methods for the investigation of the critical behaviour of the two-dimensional random-field Ising model. We illustrate the effectiveness of the Boykov-Kolmogorov algorithm and implement it for carrying out a thorough research on the breakup length scale problem of the square and triangular lattice models. We address questions such as which law governs the scaling of the breakup length of the random-field Ising model and whether this law depends on the definition used for the ratio of ferromagnetic ground states over the overall number of samples or on the lattice geometry. Finally, an alternative robust approach based on the second-moment correlation length ξ of the model as obtained from a recently developed fluctuation-dissipation formalism is undertaken and provides a clear-cut resolution of the model's scaling description.

Contents

List of Figures

List of Tables

List of Abbreviations

BC	**B**lume-**C**apel
BFS	**B**readth-**f**irst **S**earch
BK	**B**oykov-**K**olmogorov
DAFF	**D**iluted **A**nti**f**erromagnets in a **F**ield
FIFO	**F**irst-**I**n-**F**irst-**O**ut
FM	**F**erro**m**agnetic
FSS	**F**inite **S**ize **S**caling
GS	**G**round **S**tate
MC	**M**onte **C**arlo
PM	**P**ara**m**agnetic
RFIM	**R**andom-**F**ield **I**sing **M**odel
RFPM	**R**andom-**F**ield **P**otts **M**odel
RG	**R**enormalization-**G**roup
RNG	**R**andom **N**umber **G**enerator
TCP	**T**ri**c**ritical **P**oint

Introduction

In order to illustrate the theory of phase transitions, in statistical physics we offer two main categories of physical systems: fluids and magnets. The study of simplified models of such systems led to the concepts of universality, scaling and renormalization, which are the basic foundations of the theory of critical phenomena (Stanley, 1971; Yeomans, 1992; Goldenfeld, 1995; Cardy, 1996; Newman and Barkema, 1999; Landau and Binder, 2000). Experimental checks of the theory can be performed by various systems, some of which (Guggenheim, 1945; Heller and Benedek, 1962) constitute the modern era of critical phenomena Stanley (1971).

In particular, regarding magnetic systems, there have been many attempts by experimental physicists in order to reduce the defects of the models under study. However, after Imry and Ma (1975), disordered materials have given rise to new research paths during the last 45 years (Fisher, Grinstein et al., 1988; Nattermann and Villain, 1988; Belanger and Nattermann, 1998). Disordered magnets are particularly useful for the study of systems of quenched, as well as homogeneous randomness.

Let us here highlight the difference between quenched and annealed disorder. Starting with a system that is in thermal equilibrium with its environment, there are two ways to cool the system and therefore observe how it changes. We can either quench the system, that is, rapidly change the environment to its final value, or we can anneal the system, namely slowly cool its environment down. Therefore, the system will have the same state as the original one, but will be subject to a different environment, or it will be in equilibrium with its environment at all times, respectively. In other words, thinking about a system with quenched or annealed disorder, its disorder variables do not or do evolve in time, respectively. A quenched average is therefore the average keeping the random variables fixed, while an annealed average is an average which is also carried out over all the possible values that the random variables can take. In the case of quenched disorder, the thermal averaging comes before the disorder configuration averaging. In this thesis, we exclusively study systems of quenched randomness.

In general, random systems are studied through the scope of random field as well as random bond (Belanger and Nattermann, 1998) systems, both for theoretical and experimental cases. The main theoretical interest focuses on how the behaviour of pure systems changes when they are subjected to disorder. Many questions remain open to date, particularly regarding the difference of the nature of phase transitions and the scaling properties of the disordered system, compared to its pure equivalent. Such questions have been addressed theoretically with the use of the renormalization group and mean-field theories. However, many predictions have been proven inadequate, contrasting or even wrong.

For the above reason, the importance of computational statistical physics for the study of such systems should be strongly emphasized. In this field, random sampling, that is, Monte Carlo (MC) (Newman and Barkema, 1999; Landau and Binder, 2000) simulations, have been the most predominant numerical methods for studying phase transitions in the presence of quenched disorder. What is more, appropriate techniques are required in order to expand the simulation results to systems large enough to perform finite-size scaling (FSS) (Privman, 1990; Barber, 1983) and therefore extract accurate results. However, as far as it concerns the numerical methods, the lack of consistency in the results, especially for large system sizes, combined with the bulk of open questions yet to be approached, has led to the development of alternative numerical techniques, namely, graph cut methods at zero temperature (Hartmann and Rieger, 2004).

In this thesis, we take advantage of both numerical approaches, focusing strongly on the latter and we study the critical behaviour of complex systems, such as the Blume-Capel and the random-field Ising models. More precisely, in Chap. 1, a brief theoretical background regarding phase transitions, disordered systems, finite size scaling and the RFIM is included, while in Chap. 2, some of the most popular Monte Carlo numerical techniques are listed, along with an application of such methods for the Blume-Capel model. Chap. 3 focuses on the methodology and implementation of applying graph cut methods for solving the energy minimization problem for the RFIM, followed by Chap. 4, where the main application of our research on graph cut methods is presented, that is, the breakup length scale of the RFIM.

Chapter 1

Theoretical Background

In this chapter, we introduce the necessary background theory of phase transitions and finite size scaling, along with the concept of disordered systems and the FSS for such systems. In particular, in Sec. 1.1, the fundamental elements of the theory of phase transitions and critical phenomena are presented, while in Sec. 1.2 the FSS method is described, together with examples. Sec. 1.3 includes the Harris criterion, as well as an illustration of the FSS method, this time for disordered systems, followed by Sec. 1.4, where we finish our theoretical background description by introducing the RFIM.

1.1 Phase transitions and critical phenomena

Let us consider a large piece of material and examine and write down its macroscopic properties (e.g. density, magnetisation, etc.). Next, let us split the piece into two almost equal pieces and re-examine the macroscopic properties of the new pieces, while keeping the external parameters of the experiment (e.g. pressure - P, temperature - T, etc.) constant. We observe that the macroscopic properties of the two pieces remain the same as the ones of the original piece. We shall observe the same case as we continue dividing the small pieces into two smaller ones and so on.

In the long run, after many repetitions of the above process, we expect to observe something different, given the fact that the material consists of atoms and molecules whose characteristic properties differ significantly from the material's ones. The length scale at which the overall properties of the small pieces of the material begin to be notably different from the ones of the original macroscopic piece define a measure called the correlation length ξ of the material (Cardy, 1996). In other words, it is the length scale over which the fluctuations of the microscopic degrees of freedom are important and therefore referring to the theory

about the behavior of many-particle systems (central limit theorem) ceases to be a reliable method. Since the value of the correlation length depends on the external conditions of the experiment (such as temperature), its thermal dependence plays an important role in the macroscopic behaviour of the system.

It is well established that the macroscopic behaviour of many systems is remarkably different and as a matter of fact unexpectedly during the smooth variation of thermodynamic variables such as P and T. The points at which these abrupt changes take place are called critical points of the phase transition and they indicate the phase transition from a state of matter to another.

Phase transitions commonly occur in two different ways. According to the first case, the two (or more) states on both sides of the critical point coexist exactly on it, however they remain distinct from each other due to their different macroscopic properties. Nonetheless, hardly away from the critical point there is a characteristic unique phase whose properties are continuously connected to one of the co-existent phases at the critical point. In this case, the first derivatives of the free energy exhibit a discontinuous behaviour as the critical point is crossed from one phase to the other. Such transitions, where the correlation length is finite, are called discontinuous or first-order phase transitions (Stanley, 1971; Yeomans, 1992).

Now as far as it concerns the second basic category of phase transitions, which is the one of exclusive interest in the scope of our research, things are quite different. This is the case where the correlation length is infinite and we have continuous or second-order phase transitions, with the system's degrees of freedom being correlated over all length scales. This causes the system to be in a unique critical phase. During a second-order phase transition, the two (or more) phases on both sides of the critical point should gradually coincide while approaching the critical point. Therefore, the correlation length tends to infinity. In contrast, the differences among the thermodynamic quantities (e.g. energy density, magnetisation) of the two (or more) phases converge continuously to zero.

The occurrence of many correlated degrees of freedom makes the understanding of second-order phase transitions a particularly difficult process. However, during the last 45 years and with the help of renormalization theory, the approach of such concepts takes place through a different mindset which has undoubtedly helped the recognition and comprehension of the complex methods involved in critical phenomena of matter (Goldenfeld, 1995; Cardy, 1996).

One of the most substantial conclusions of modern theory of critical phenomena is that, despite the fact that systems corresponding to large correlation length values seem to be quite complex, at the same time they conceal an extraordinary simplicity, whose use makes the study of such systems easier to some extent. One such case is the universality (Goldenfeld, 1995; Cardy, 1996), according to which many properties of a system close to a critical point of a continuous phase transition depend only on few features of the system and not on the microscopic details of the interactions among the individual particles. Instead, they belong to one of the few different universality classes existing, where each one of them is governed by some common elements, such as the symmetries of the Hamiltonian and of the ground state (GS), the spatial dimension D, etc.

Let us point out here that the concepts of universality and scaling were initially highlighted through the numerical data analysis of simple models (Ising model) during the 60's. They were subsequently physically interpreted in the scope of renormalization theory and of relative Kadanoff-type arguments (Goldenfeld, 1995).

At this point, the basic concepts of statistical physics and theory of critical phenomena which are essential for the understanding of the current thesis are discussed. The most important quantity in statistical physics is the partition function $\mathcal{Z}$, which describes the statistical properties of a system in thermodynamic equilibrium. In general, the partition function is a function of temperature and of other system parameters, such as volume in case of a gas or magnetic field in case of a magnetic system. The basic thermodynamic quantities, such as internal energy, free energy and entropy can be appropriately expressed with terms and derivatives of the partition function. There are many ways to express the partition function depending on the statistical ensemble under consideration. For example, the canonical partition function refers to the canonical ensemble, where the system exchanges energy with a heat bath, under fixed temperature, volume and number of particles.

Let us now define the Hamiltonian of the most fundamental spin model of Statistical Physics, that is the Ising model:

$$\mathcal{H} = -J \sum_{\langle x,y \rangle} s_x s_y - h \sum_x s_x, \tag{1.1}$$

where $s_x = \pm 1$ and $J > 0$ is the interaction constant. $\langle x, y \rangle$ represent the short-range, nearest-neighbour interactions and h is an external magnetic field. Now let s ($s = 1, 2, 3, ...$) be the precise (micro)states of the system and E_s be the total energy of the system when in state s. Then, the partition function of the system

can be written as:

$$\mathcal{Z} \equiv \mathcal{Z}(T, h) = \sum_s e^{-\beta E_s}, \tag{1.2}$$

where $\beta \equiv \frac{1}{k_B T}$ and k_B is the Boltzmann constant. The free energy occurs as the logarithm of the partition function:

$$\mathcal{F} = -k_B T \ln \mathcal{Z}. \tag{1.3}$$

All macroscopic thermodynamic properties of the system are derived by appropriately differentiating Eq. (1.3).

Before proceeding to reproducing the basic thermodynamic quantities, let us re-express the partition function through the system's density of states, $g(E)$, which is the number of all possible states of the system corresponding to each energy E. In this case, Eq. (1.2) writes:

$$\mathcal{Z} = \sum_E g(E)\, e^{-\beta E}. \tag{1.4}$$

The density of states does not depend on the temperature and therefore can be used in order to construct normal distributions at each temperature. This fact has allowed the development of a series of MC algorithms whose main goal is to directly calculate a system's density of states. Such algorithms will be described in detail in Chap. 3.

We can now calculate the internal energy U of the system:

$$U = -\frac{\partial \ln \mathcal{Z}}{\partial \beta} = \langle E \rangle = \frac{\sum_E E g(E)\, e^{-\beta E}}{\sum_E g(E)\, e^{-\beta E}}. \tag{1.5}$$

In general, the expectation value of the k^{th} power of the system energy is:

$$\langle E^k \rangle = \frac{\sum_E^k E^k g(E)\, e^{-\beta E}}{\sum_E^k g(E)\, e^{-\beta E}}. \tag{1.6}$$

Therefore, the fluctuations of internal energy yield the specific heat C:

$$C = \left(\frac{\partial U}{\partial T}\right)_h = k_B \beta^2 \left| \langle E^2 \rangle - \langle E \rangle^2 \right|. \tag{1.7}$$

Appropriate differentiation of the free energy generates the magnetisation M and the magnetic susceptibility χ:

$$M = \left(\frac{\partial \mathcal{F}}{\partial h}\right)_T \tag{1.8}$$

and

$$\chi = \left(\frac{\partial M}{\partial h}\right)_T = \beta^{-1}\frac{\partial^2}{\partial h^2}\ln \mathcal{Z} = \beta\left|\langle M^2\rangle - \langle M\rangle^2\right|. \tag{1.9}$$

Up to now, starting from the partition function, we have defined the basic thermodynamic quantities which describe the macroscopic behaviour of a magnetic system. The special importance of the density of states of the system should be highlighted, since if $g(E)$ is known, Eqs. (1.5) to (1.7) can be easily calculated.

Now let us focus on the point where a continuous phase transition takes place, that is, the critical point. Normally, close to a critical point, the correlation length and the rest of the thermodynamic parameters demonstrate power-law dependencies on the parameters describing the distance from the critical point. This distance is expressed through the reduced critical temperature t, where $t = \frac{T-T_c}{T_c}$. Such power laws are called scaling laws and the corresponding exponents are called critical exponents. Critical exponents are simple, usually rational numbers and in many cases they exclusively depend on the universality class to which they belong (Stanley, 1971; Privman, 1990).

A critical exponent μ for a thermodynamic quantity $\mathcal{A}$ is defined as:

$$\mu = \lim_{t\to 0}\frac{\ln \mathcal{A}(t)}{\ln t}, \tag{1.10}$$

yielding the following expression for the dependence of the quantity $\mathcal{A}$ on the critical exponent μ:

$$\mathcal{A}(t) \propto t^{\mu}. \tag{1.11}$$

Eq. (1.11) is true only for the case where $t \to 0$. In general, taking into account correction terms as well, one can get $\mathcal{A}(t) = At^{\mu}\,(1 + bt^{\mu_1} + ...)$. In addition, given that the approach to the critical point can be done in two ways, that is, from below or from above, two critical exponents can be defined as μ and μ' for $t \to 0^-$ and $t \to 0^+$, respectively.

Now regarding magnetisation, its critical exponent β is given by:

$$M(t, h = 0) \simeq (-t)^{\beta}, t \to 0^-, \tag{1.12}$$

while the critical exponents γ and γ' of the magnetic susceptibility by:

$$\chi(t, h = 0) \simeq \begin{cases} (t)^{-\gamma}, & t \to 0^- \\ (-t)^{-\gamma'}, & t \to 0^+ \end{cases}. \tag{1.13}$$

As for the specific heat's critical exponents α and α', we have:

$$C(t, h=0) \simeq \begin{cases} (t)^{-\alpha}, & t \to 0^- \\ (-t)^{-\alpha'}, & t \to 0^+ \end{cases}. \tag{1.14}$$

Finally, the critical exponent δ of the critical isotherm for small field variations is defined as:

$$M(0, h) \simeq h^{\frac{1}{\delta}} \operatorname{sign} h, h \to 0 \tag{1.15}$$

and the critical exponents ν and ν' for the correlation length write:

$$\xi(t, h=0) \simeq \begin{cases} (t)^{-\nu}, & t \to 0^- \\ (-t)^{-\nu'}, & t \to 0^+ \end{cases}. \tag{1.16}$$

Let us proceed to a universality example, namely, Widom's scaling hypothesis, according to which (Barber, 1983) the non-analytical part (singular, s) of the free energy per particle (f_s) of a magnetic system near its critical point is a homogeneous function with respect to the system parameters. This two-parameter scaling theory can be expressed as follows (Stanley, 1971):

$$f_s(\lambda^p t, \lambda^q \mathrm{h}) = \lambda f_s(t, \mathrm{h}) \Rightarrow \begin{cases} \lambda^q M(\lambda^p t, \lambda^q \mathrm{h}) = \lambda M(t, \mathrm{h}) \\ \lambda^{2q} \chi(\lambda^p t, \lambda^q \mathrm{h}) = \lambda \chi(t, \mathrm{h}) \\ \lambda^{2p} C(\lambda^p t, \lambda^q \mathrm{h}) = \lambda C(t, \mathrm{h}) \end{cases}, \tag{1.17}$$

where the equations regarding magnetisation, magnetic susceptibility and specific heat occur after appropriate differentiation of the left-hand side of Eq. (1.17) and $\mathrm{h} = \frac{h - h_c}{h_c}$. There exist many different formulas for Eq. (1.17) (Stanley, 1971; Privman, 1990; Barber, 1983), which uses the fundamental properties of homogeneous functions of two variables (Stanley, 1971).

Now setting $\mathrm{h} = 0$ and $\lambda = (-t)^{-\frac{1}{p}}$ for the magnetisation equation in Eq. (1.17), we get $M(t, 0) = (-t)^{\frac{1-q}{p}} M(-1, 0), T \to 0^-$ and therefore, Eq. (1.12) yields $\beta = \frac{1-q}{p}$. In the same manner, setting $t = 0$ and $\lambda = \mathrm{h}^{-\frac{1}{q}}$, again for the magnetisation equation, we compare with Eq. (1.15) and get $\delta = \frac{q}{1-q}$. Following similar techniques for the susceptibility and specific heat equations, we obtain $\gamma = \gamma' = \frac{2q-1}{p}$ and $\alpha = \alpha' = 2 - \frac{1}{p}$, respectively. Therefore, an outcome of Widom's scaling is that the critical exponents are not independent but correlated to each other via two numbers, $p, q \in \mathbb{R}$:

$$\begin{aligned} \alpha = \alpha' = 2 - 2\beta(\delta + 1) = 2 - \frac{1}{p} \\ \gamma = \gamma' = \beta(\delta - 1). \end{aligned} \tag{1.18}$$

Rushbrooke:	$\alpha' + 2\beta + \gamma' \geq 2$
Griffith:	$\alpha' + \beta(1+\delta) \geq 2$
Fisher:	$\nu(2-\eta) \geq 2\gamma$
Josephson:	$D\nu' \geq 2-\alpha'$, $d\nu \geq 2-\alpha$

TABLE 1.1: Scaling laws.

In Table 1.1 one can check the most known inequalities for the critical exponents (Stanley, 1971; Privman, 1990), which emerge from the renormalization theory (Stanley, 1971). Via Widom's scaling hypothesis, the equalities $\alpha' = \alpha$, $\gamma' = \gamma$, $\nu' = \nu$ come out and consequently we shall consider the exponents equal from now on.

Two more critical exponents are introduced in Fisher and Josephson inequalities, namely ν and η. ν determines the power law of the correlation length and η the critical behaviour of the connected correlation function $\Gamma(r) = \langle s_0 s_r \rangle - \langle s_0 \rangle \langle s_r \rangle$, which calculates the correlation of the spin values fluctuations from their averages.

Far from the critical temperature, the spin values fluctuations from their average values are particularly small and decay exponentially and analogously to distance. Therefore, the following behaviour for the connected correlation function is expected:

$$\Gamma(r) \approx r^{-\tau} e^{-\frac{r}{\xi}}, \tag{1.19}$$

where exponent τ plays no important role as well as the function behaves exponentially, given that the dominant behaviour is the one to be determined by the exponential function.

According to the fluctuation theorem, the isothermal susceptibility is the spatial integral of the correlation function, that is $\chi \sim N \int d^d r r^{d-1} \Gamma(r)$, which emerges as a result of the behaviour described by Eq. (1.19) and hence converges to a finite value. Nevertheless, we know that as we approach either from above or below the critical temperature, the isothermal susceptibility per particle diverges to infinity, as an outcome of the fluctuations of the order parameter. Close to the critical point the order parameter converges to 0 and its fluctuations, although expanding to big distances, are not particularly strong.

Mathematically speaking, the divergence of the isothermal susceptibility at the critical point can be explained as follows: As the critical temperature is

approached, the correlation length increases according to a power law:

$$\xi \simeq \begin{cases} (t)^{-\nu}, & t \to 0^- \\ (-t)^{-\nu'}, & t \to 0^+ \end{cases}. \tag{1.20}$$

Thus, as the correlation length increases, the behaviour of the correlation function given by Eq. (1.19) is not valid anymore. This implies that there is no characteristic length for the problem and all the length scales participate to the establishment of the behaviour. In this case, the correlation function is described by a power law of the form:

$$\Gamma(r) \approx \frac{1}{r^{d-2+\eta}}. \tag{1.21}$$

1.2 Finite size scaling

The number of particles (spin variables) N in a graph cut (or MC) simulation is of the order of 10^2–10^7. However, a real system consists of $N \sim N_{\text{real}} \sim 10^{23}$ particles. Therefore, it is clear that one needs to come up with a technique that calculates the properties of the real, large system, by studying the properties of smaller, finite, computationally feasible systems. A suitable method for this purpose is the finite-size scaling method (FSS) (Barber, 1983), where the values of the critical exponents are figured out, based on the study of the change of the thermodynamic quantities of a system as a function of the size lattice L.

Let us re-write Widom's scaling hypothesis regarding the non-analytical part of the free energy (Eq. (1.17)):

$$f_s(t, \mathrm{h}) = l^{-D} \tilde{f}\left(l^{\frac{1}{\nu}} t, l^{\frac{\beta+\gamma}{\nu}} \mathrm{h}\right) = l^{-\frac{2-\alpha}{\nu}} \tilde{f}\left(l^{\frac{1}{\nu}} t, l^{\frac{\beta+\gamma}{\nu}} \mathrm{h}\right), \tag{1.22}$$

where now the parameter λ is equal to the length of the Kadanoff unit cell, i.e., $\lambda = l^D$, where l is a length scale. The correlation between Eq. (1.17) and Eq. (1.22) is obvious via Eq. (1.18), since $\lambda^p = l^{Dp} = l^{\frac{D}{2-\alpha}} = l^{\frac{1}{\nu}}$ and $\lambda^q = l^{Dq} = l^{\frac{\beta+\gamma}{\nu}}$, respectively. The scaling function $\tilde{f}$ is non-dimensional. The same holds for its arguments, which are called scaling parameters (Barber, 1983) or scaling fields (Goldenfeld, 1995).

As a result of the above, the simplest way to follow an FSS process is to introduce the linear size of the lattice to Eq. (1.22) as one additional scaling parameter:

$$f_s(t, \mathrm{h}, L^{-1}) = l^{-D} \tilde{f}\left(l^{\frac{1}{\nu}} t, l^{y_{\mathrm{h}}} \mathrm{h}, \frac{l}{L}\right), \tag{1.23}$$

where the dependence on the system size does no longer hold at $L \to \infty$, as desired. As shown by Barber (1983), the most sensible choice of the length scale is the one equal to the linear size of the system ($l = L$), due to the rounding of the thermodynamic quantities. Then, the scaling of the free energy takes the form:

$$f_s(t, \mathrm{h}, L^{-1}) = L^{-D} \tilde{f}\left(L^{\frac{1}{\nu}} t, L^{y_\mathrm{h}} \mathrm{h}, 1\right). \tag{1.24}$$

At this point, Eq. (1.24) allows us to determine the L-dependence of all thermodynamic quantities of the system. Assume that we are interested in the properties of the system when the external magnetic field is zero. Then, one can write, for example, for the magnetisation M, magnetic susceptibility χ and specific heat C the following formulas:

$$M = L^{-\frac{\beta}{\nu}} \tilde{M}(x_t), \tag{1.25}$$

$$\chi = L^{\frac{\gamma}{\nu}} \tilde{\chi}(x_t), \tag{1.26}$$

$$C = L^{\frac{\alpha}{\nu}} \tilde{C}(x_t), \tag{1.27}$$

where $x_t = L^{\frac{1}{\nu}} t$ is the temperature scaling parameter. The scaling functions $\tilde{M}, \tilde{\chi}$ and $\tilde{C}$ are almost independent of the linear size of the lattice, provided that we omit the higher order corrections.

A typical method for calculating the critical temperature T_c and the critical exponent ν of the correlation length, in the scope of the FSS theory, is based on the estimation of the position of the peak of a thermodynamic derivative of the free energy, such as the specific heat or the magnetic susceptibility. For a finite lattice, the peak corresponds to the temperature where the scaling function, e.g. $\tilde{C}(x_t)$, is maximum, i.e.:

$$\left.\frac{d\tilde{C}(x_t)}{dx_t}\right|_{x_t = x_t^*} = 0. \tag{1.28}$$

This temperature is called pseudo-critical T_L^* and is defined via $x_t = x_t^*$ to be changing in accordance to the linear size of the lattice as:

$$\left[\frac{T_L^* - T_c}{T_c}\right] L^{\frac{1}{\nu}} = x_t^* \implies T_L^* = T_c + b L^{-\frac{1}{\nu}}, \tag{1.29}$$

where $b = T_c x_t^*$. Eq. (1.29), as well as Eqs. (1.25) to (1.27) are precise only in the case where the system size is very big and the temperature is close to the critical temperature. In the case of small lattice sizes, one needs to include corrections, typically of the form $\sim (1 + cL^{-\omega} + \ldots)$ (Ferrenberg and Landau, 1991; Aharony and Fisher, 1983).

Let us focus on the critical exponent ν, whose calculating with FSS with the use of numerical data is a costly process. For this reason, there are various alternative ways for determining the exponent in the bibliography, such as functions related to the order parameter M. In the scope of FSS, the probability density function $P_L(M)$ for the order parameter of a finite size system of linear size L does not depend on each of the three variables L, ξ and M, but it is a generalised homogeneous function of two of them (Binder, 1981a). Thus, one can define the probability density function $P_L(M)$ as (Binder, 1981a):

$$P_L(M) = L^{\frac{\beta}{\nu}} \tilde{P}\left(\frac{\xi}{L}, ML^{\frac{\beta}{\nu}}\right), \; \xi \to \infty, \; L \to \infty, \; \frac{\xi}{L}\text{: finite}, \tag{1.30}$$

where $\tilde{P}$ is a scaling function, similar to Eqs. (1.25) to (1.27). From the above equation, the value of the k^{th} power of the order parameter is:

$$\begin{aligned} \langle |M|^k \rangle &= L^{\frac{\beta}{\nu}} \int dM \, |M|^k \, \tilde{P}\left(\frac{\xi}{L}, ML^{\frac{\beta}{\nu}}\right) \\ &\sim L^{-\frac{k\beta}{\nu}} \int dz z^k \tilde{P}\left(\frac{\xi}{L}, z\right) = L^{\frac{-k\beta}{\nu}} \tilde{f}_k\left(\frac{\xi}{L}\right), \end{aligned} \tag{1.31}$$

where $z = ML^{\frac{\beta}{\nu}}$ and $\tilde{f}_k$ is a new scaling function.

We can now define, via Eq. (1.31), the fourth-order Binder's cumulant (Binder, 1981a; Binder, 1981b):

$$U_4 = 1 - \frac{\langle M^4 \rangle}{3\langle M^2 \rangle^2}, \tag{1.32}$$

where the maximum value of its slope scales as:

$$\left.\frac{dU_4}{dK}\right|_{\max} \sim L^{\frac{1}{\nu}}, \tag{1.33}$$

with $K = \frac{1}{T}$. Furthermore, independent estimations of the critical exponent ν are given by logarithmic derivatives of various orders of the order parameter (Ferrenberg and Landau, 1991):

$$\frac{\partial \ln \langle M^k \rangle}{\partial K} = \frac{1}{\langle M^k \rangle} \frac{\partial \langle M^k \rangle}{\partial K} = \frac{\langle M^k E \rangle}{\langle M^k \rangle} - \langle E \rangle, \tag{1.34}$$

whose maximum values scale, in accordance to Eq. (1.33), as:

$$\left.\frac{\partial \ln \langle M^k \rangle}{\partial K}\right|_{\max} \sim L^{\frac{1}{\nu}}. \tag{1.35}$$

In addition, alternative estimations of the critical exponents β and ν are provided by Ferrenberg and Landau (1991), via the scaling of the maximum

values of the derivative of the absolute value of the order parameter

$$\frac{\partial\langle|M|\rangle}{\partial K} = \langle|M|\,E\rangle - \langle|M|\rangle\langle E\rangle, \tag{1.36}$$

given by:

$$\left.\frac{\partial\langle|M|\rangle}{\partial K}\right|_{\max} \sim L^{\frac{1-\beta}{\nu}}. \tag{1.37}$$

Finally, the temperatures corresponding to the maximum values of the derivatives given by Eq. (1.32), Eq. (1.34) and Eq. (1.36) define additional pseudo-critical temperatures which follow the scaling expression seen in Eq. (1.29) and yield supplementary estimations for the critical temperature and the critical exponent ν.

The scaling of the quantities in Eqs. (1.25) to (1.27), the displacement of the pseudo-critical temperatures [Eq. (1.29)] and the alternative methods for figuring out the critical exponents ν and β [Eq. (1.35) and Eq. (1.37)] make up a basic tool for the study of random-field models and the determination of their critical exponents in the next chapters.

1.3 Disordered systems and FSS

All of the above fundamental concepts of statistical physics are well established for the case of the phase transitions in pure systems. However, the study of the critical behaviour of disordered spin systems is a more recent and difficult research field. These systems' physical content is of great interest and it brings out new and complex phenomena, many of which are directly related to experiments (Fisher, Grinstein et al., 1988; Belanger and Nattermann, 1998).

The immense difficulty of answering the basic questions about the changes of the critical behaviour of a pure system when disorder is introduced has led to the development of new scientific pathways, which are not only related to statistical physics but also to condensed matter physics, computer science, mathematics and even biology (Belanger and Nattermann, 1998; Hartmann, 2004)

There are different disorder forms which can be introduced in magnetic spin systems, such as random-field disorder, site and bond dilution and bond disorder. For every case, the introduction of disorder induces a series of new phenomena, many of which have not been studied yet or are not congruent with each other (Rieger, 1998). A significant progress in the study of such systems has been achieved with the use of the mean field theory (Belanger and Nattermann, 1998), however the need for a more precise investigation of their properties has led to the

development of complex numerical techniques, such as MC algorithms at finite temperatures ($T > 0$) and optimization algorithms at $T = 0$.

The above methods have undoubtedly contributed to the understanding of the behaviour of disordered systems and they constitute, to date, the largest part of the research activity in the field of statistical physics of disordered systems. Still, the numerical simulations both at $T > 0$ and $T = 0$ do face challenges due to the complex form of the spectrum of the free energy of disordered systems, where the case of consecutive local energy minima being separated by high energy boundaries is quite common. This leads to exponentially increasing restoration time in the MC ($T > 0$) simulations and to the NP-hardness problem in the optimization algorithms ($T = 0$) (Hartmann, 2004; Rieger, 1998).

1.3.1 The Harris criterion

As previously mentioned, the formulation of reliable theoretical predictions for random systems continues to be a challenge for the scientific community. The most known and crucial theoretical prediction regarding the effect of disorder to the critical behaviour of spin models is the Harris criterion (Harris, 1974), which refers to the effect of the weak, uncorrelated randomness in systems whose pure equivalents exhibit a second-order phase transition.

Let us consider a disorder in the interaction energy J of a system, which can be regarded equivalently as a spatial variation to the transition temperature of the system. Thus, we introduce a spatially varying, local, critical temperature $T_c(r)$ and we study its variation with regard to the overall critical temperature T_c of the disordered system in the region of some volume of the correlation length $V_\xi = \xi^D$, where $\xi \sim t^{-\nu}$:

$$\delta T_c(r) \equiv T_c(r) - T_c. \tag{1.38}$$

The Harris criterion can therefore be proposed as follows: the pure fixed point will remain intact from the introduction of disorder and the critical behaviour of the system will not change, if the square root of the absolute value of the square of the fluctuations $\delta T_c(r)$ inside the volume V_ξ is sufficiently small compared to $t \sim \xi^{-\frac{1}{\nu}}$, as $t \to 0$. Alternatively, the introduction of disorder will affect the pure fixed point, the system will move to a new universality class and its behaviour will be determined by the new random fixed point.

An exceptional case laid aside the Harris criterion is the case where the critical exponent of the specific heat is zero. This is the case for the $2D$ Ising model, about

which there exist theoretical predictions that the effect of the weak, uncorrelated randomness leads to the creation of logarithmic corrections to the behaviour of the pure system. We shall examine the critical behaviour of the $2D$ Ising model in the presence of random-field disorder in Chap. 4.

1.3.2 Finite size scaling in disordered systems

In general, the quantities of interest that are studied in the scope of finite size scaling are the distributions $P(Z)$ of the quantities Z_q that have occurred from calculations over some collection of random realisations $\{1, ..., q, ..., Q\}$ of the disorder. $Z(T)$ is any thermodynamic quantity, such as specific heat, magnetic susceptibility etc. and its mean value over the many realisations is computed as:

$$\langle Z \rangle_{\text{dis}} = \frac{1}{Q} \sum_{q=1}^{Q} Z_q(T). \tag{1.39}$$

Following, the FSS is applied to its maximum:

$$\langle Z \rangle_{\text{dis}}^{\star} = (\langle Z \rangle_{\text{dis}})^{\star}. \tag{1.40}$$

An alternative method of averaging and calculating the mean value is:

$$\langle Z^{\star} \rangle_{\text{dis}} = \frac{1}{Q} \sum_{q=1}^{Q} Z_q^{\star}, \tag{1.41}$$

where the averaging over the disorder takes place after the computation of the maximum value of a thermodynamic quantity. This process is called averaging over individual maxima and up to date, it is not clear whether its application on FSS is equivalent to Eq. (1.40), at least for the case of averaging over a finite number of realisations of the disorder.

Respectively to the mean values given by Eq. (1.40) and Eq. (1.41), one can define the pseudo-critical temperatures $T_{\langle Z \rangle_{\text{dis}}^{\star}}$ and $\langle T_Z^{\star} \rangle_{\text{dis}}$. These temperatures are expected to scale according to the scaling law described by Eq. (1.29), where T_c is now the critical temperature of the random system and ν the respective exponent of the correlation length.

In particular, it has been shown (Chayes et al., 1986) that the critical exponent ν satisfies the inequality given by Eq. (1.42). More specifically, a random system of disorder parameter λ, $0 \leq \lambda \leq 1$ was considered, which goes under a second-order phase transition at some value $0 < \lambda_c \leq 1$. Given the assumption that this transition is characterised by a change in the scaling behaviour of the probability of a finite-volume event, it was concluded that this change can be used for the

appropriate definition of a correlation length $\xi_f(\lambda)$. It was hence shown that if the finite-size scaling correlation length $\xi_f(\lambda)$ diverges as $\xi_f(\lambda) \sim |\lambda_c - \lambda|^{-\nu}$, as $\lambda \to \lambda_c$, then the exponent ν satisfies:

$$\nu \geq \frac{2}{D}. \tag{1.42}$$

The proof of the above, given by Chayes et al. (1986) is general, that is, it is only assumed that the system exhibits a second-order phase transition with a divergent correlation length and the randomness is uncorrelated at large length scales. It arises from a mathematical theorem that states that the derivative with respect to the disorder parameter λ of the chosen finite-volume event is bounded from above.

Eq. (1.42) has been validated for a series of classic and quantum disordered systems (Chayes et al., 1986; Huse and Fisher, 1987; Fisher, 1995) and for the systems under study in the current thesis. Still, the concept of the correlation length ξ_f is not trivial in random systems; it is in no case clear if it is equivalent, critical-exponent-wise, to the intrinsic correlation length ξ. However, the general arguments of the renormalization theory imply that most critical transitions are characterised by a scaling behaviour of appropriately defined variables, such that the equivalence of ξ_f and of the intrinsic correlation length ξ is considered reliable (Fisher, Grinstein et al., 1988).

1.4 The Random-Field Ising model

The RFIM is described by the Hamiltonian:

$$\begin{aligned} \mathcal{H} = -J \sum_{\langle x,y \rangle} s_x s_y - \sum_x h_x s_x \Leftrightarrow \\ \Leftrightarrow E_{\text{total}} = E_{\text{bond}} + E_{\text{field}}, \end{aligned} \tag{1.43}$$

where, as we already know, $s_x = \pm 1$ and J is the nearest-neighbour interaction constant, which, for the ferromagnetic model, is $J > 0$. In our study, we particularly set it to be $J = 1$. When referring to the simple Ising model, the external field h in Eq. (1.1) is equal to zero. For the RFIM, the disorder is introduced via h_x, which are independent random magnetic fields acting on each spin s_x and following some distribution $P(h_x)$.

Various random-field distributions $P(h_x)$ have been studied in the literature, such as the Gaussian and the bimodal distributions (Belanger and Young, 1991; Belanger and Nattermann, 1998; Rieger, 1995b) with more emphasis being given

on the Gaussian distribution (Dotsenko, 2007). In the current thesis we consider quenched random fields following the Gaussian distribution ($\mu = 0$, $\sigma^2 = h^2$), where h is the disorder strength.

The RFIM is one of the simplest and characteristic disordered systems (Imry and Ma, 1975; Aharony, Imry et al., 1976; Young, 1977; Fishman and Aharony, 1979; Parisi, 1979; Cardy, 1984; Imbrie, 1984; Schwartz and Soffer, 1985; Gofman et al., 1992; Esser et al., 1997; Barber and Belanger, 2001). Its study not only allows the investigation of basic theories in Statistical Physics and of a series of complex phenomena, but also is directly connected to a significant number of experiments in condensed matter physics (Belanger and Young, 1991; Rieger, 1995a; Belanger and Nattermann, 1998; Belanger, King et al., 1983; Vink et al., 2006). These facts have established it as one of the main platform models for the study of collective behaviour under quenched disorder.

The most widely studied experimental realizations of the model are the diluted antiferromagnets in a uniform external field (DAFF) (Fishman and Aharony, 1979; Cardy, 1985; Belanger and Nattermann, 1998). Further experimental realizations of the model are fluids in porous media (Gennes, 1984), colloid-polymer mixtures (Vink et al., 2006; Annunziata and Pelissetto, 2012), mixed Jahn-Teller systems (Graham et al., 1987), colossal magnetoresistance oxides (Dagotto, 2005; Burgy et al., 2001), phase coexistence in the presence of quenched disorder (Cardy and Jacobsen, 1997; Fernández et al., 2008b; Fernández et al., 2008a), mixed crystals undergoing structural or ferroelectric transitions (Nattermann, 1990), nonequilibrium phenomena such as the Barkhausen noise in magnetic hysteresis (Sethna et al., 1992; Perković et al., 1999), the design of switchable magnetic domains (Silevitch et al., 2010), etc (Fytas and Martín-Mayor, 2016).

In general, theoretical and numerical studies play the most crucial role in the investigation of the critical behaviour of the RFIM. Theoretical methods include mean field applications, renormalization-group (RG) theory and series expansions of high and low temperatures (Dotsenko, 2007). As for the numerical techniques, many attempts have been performed for the development of efficient numerical methods for the study of the phase transition properties of the model (Rieger, 1995a). Apart from the typical MC techniques at finite temperature ($T > 0$), the zero-temperature fixed-point scenario (Nattermann and Villain, 1988) has led to the switch of a significant proportion of the research activity to the study of the model at zero temperature, via ground state finding approaches (Ford-Fulkerson, Push-relabel algorithms) (Rieger, 1998). These techniques have yielded various reliable results (Middleton and Fisher, 2002) for some of the features of the critical

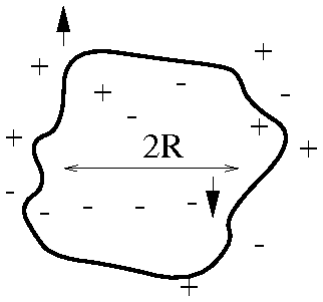

FIGURE 1.1: Schematic representation of the Imry and Ma argument (Nattermann, 1997). Domain of reversed spins.

behaviour of the RFIM. We shall discuss these methods in more detail in the following sections.

1.4.1 Lower critical dimension and phase diagram

It is well known that the pure Ising model has a ferromagnetically ordered phase for spatial dimensions $D > 1$. The additional random field term [second term of Eq. (1.43)] acts against this order and shifts the value of the lower critical dimension (D_l) to higher values. In the case where the strength of the random field h is much larger than the coupling constant J, the system is disordered at low temperatures (Beretti, 1985). From now on we shall consider only the case where $h \ll J$.

Imry and Ma were the first to show that there are indeed no ferromagnetically ordered phases in the RFIM for $D \leq 2$ (Imry and Ma, 1975). Namely, they showed that the lower critical dimension of the model is $D_l = 2$, below which there is no phase transition. This initial prediction by Imry and Ma was thoroughly confirmed by Grinstein (1976), Grinstein and Ma (1983), Imbrie (1984) and Bricmont and Lebowitz (1987), who proved, using powerful mathematical arguments, the existence of a phase transition at $D = 3$, for $h \leq h_c$, where h_c is the critical field (see, also, Fig. 1.2).

In short, the Imry and Ma argument goes as follows: Let us consider a region of radius R at zero temperature in the ferromagnetic phase and let us assume that there exists excess random field energy in this domain. The percentage of this energy is estimated according to the central limit theorem and for a region of volume R^d it is of the order $E_h \sim hR^{\frac{D}{2}}$. When flipping the spins inside this area (see Fig. 1.1), the energy cost E_J is proportional to the surface of the area, that is $E_J \sim JR^{D-1}$. Comparing E_h and E_J, it is clear that for $D > 2$ and for large R the energy change is positive and therefore the system exhibits a stable

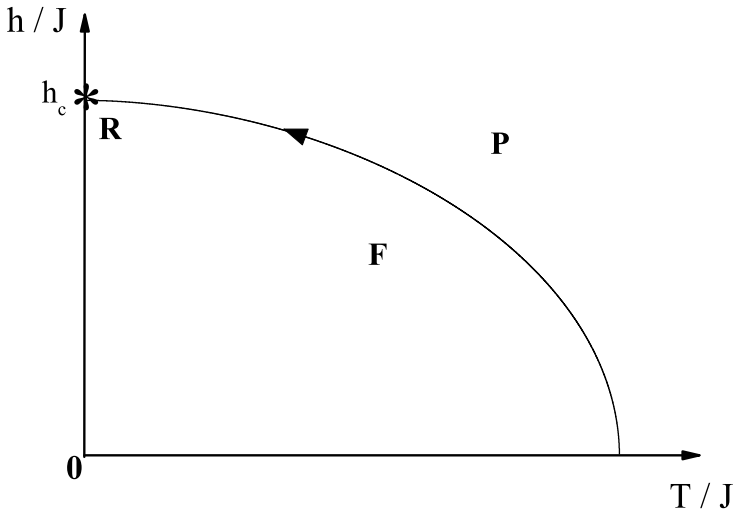

FIGURE 1.2: A sketch of the phase diagram of the RFIM. The solid line (phase boundary) separates the ferromagnetic (**F**) and paramagnetic (**P**) phases. The black arrow shows the flow to the random fixed point (**R**) at $T = 0$ and $h = h_c$, as marked by an asterisk.

state of ferromagnetic order. On the other hand, for $D \leq 2$ and for large R the energy change is negative and hence the ferromagnetic state becomes unstable with respect to domain formation (Nattermann, 1997). It is evident from the above that the lower critical dimension of the RFIM is $D_l = 2$.

As we saw, it emerges from the argument of Imry and Ma (Imry and Ma, 1975) that there exists an ordered ferromagnetic phase for the RFIM at low temperature and weak disorder for $D > 2$ (Villain, 1984; Bray and Moore, 1985a; Fisher, 1986; Berker and McKay, 1986; Bricmont and Lebowitz, 1987). This fact has allowed a qualitative sketching of the phase diagram of the RFIM, which has been used in many studies of the model (Newman, Roberts et al., 1992; Machta et al., 2000; Newman and Barkema, 1996; Itakura, 2001; Fytas and Malakis, 2008; Aharony, 1978). There also exist closed form quantitative expressions, emerging from mean-field calculations (Aharony, 1978), however they are not precise approximations (Fytas and Martín-Mayor, 2016).

A qualitative, schematic phase diagram and renormalization-group flow of the RFIM is shown in Fig. 1.2. At low temperatures and weak disorder the system is in a stable state of ferromagnetic order (**F**), while at high temperatures the system is paramagnetic (**P**). The phase boundary crosses the randomness axis at

the critical value of the disorder strength, $h = h_c$, away from which there is no phase transition of the system.

1.4.2 Critical behaviour and universality aspects

Given the fact that the RFIM has an ordered phase in more than two space dimensions, the question regarding the nature of the transition of the model from a ferromagnetic to a paramagnetic phase naturally emerges. This question was initially addressed with mean-field theory and in particular with the use of the replica technique by Schneider and Pytte (1977) for the Gaussian distribution of the random fields and later by Aharony (1978) for the bimodal distribution. The results concluded a second-order phase transition for all the values of the disorder strength, for the Gaussian distribution, while for the case of the bimodal distribution, the existence of a first-order phase transition and of a tricritical point (TCP) was predicted, if the disorder strength is larger than a critical value.

Later, Swift et al. (1996), via performing an exact determination of the ground states at $D = 4$, found a discontinuous transition for the bimodal field distribution. What is more, strong evidence of a first-order phase transition have been reported for high values of the disorder strength by Hernández and Ceva (2008). Therefore, despite the fact that the model is expected to have a continuous phase transition for $D > 2$ and for the disorder strength following the Gaussian distribution, the situation remains still not clear regarding the nature of the model's transition for the case of the bimodal distribution. In addition, fundamental issues such as the existence or not of a universality class of the model (Sourlas, 1996), the values of the critical exponents and their dependence on the distribution and on the values of the disorder strength (Sourlas, 1996; Anglés d' Auriac and Sourlas, 1997; Hartmann and Nowak, 1999), as well as the effect of complex phenomena to the critical behaviour of the model, such as the lack of self-averaging (Dayan et al., 1992; Parisi and Sourlas, 2002) have prevented a full and clear resolution of the model's critical behaviour.

At this point let us point out that it is well established that the upper critical dimension of the RFIM is $D_u = 6$ (Aharony, 1978). The calculations of critical exponents above this dimension are performed by mean-field theory techniques and the critical exponents exhibit logarithmic corrections. No numerical simulations had been performed in order to confirm that the upper critical dimension of the RFIM is $D_u = 6$ until very recently. Ahrens and Hartmann (2011), using graph cut methods at zero temperature, calculated the critical exponents for $D = 5, 6, 7$ for the purpose of validating that $D_u = 6$ and all exponents are mean-field for

$D \geq 6$, if corrections to scaling are taken into account. However, a concrete, extensive study that will unveil the presence and role of logarithmic corrections at the upper critical dimension is yet to be undertaken.

The above discussion implies that the physically relevant dimensions of the RFIM are $3 \leq D < 6$. Fytas, Martín-Mayor, Picco et al. (2017b) carried out a thorough investigation of the critical properties of the RFIM at these dimensions and their results constitute the most recent establishment of the model's critical behaviour. A summary of the team's findings (Fytas, Martín-Mayor, Picco et al., 2016; Fytas and Martín-Mayor, 2013b; Fytas and Martín-Mayor, 2013a; Fytas, Martín-Mayor, Picco et al., 2017a) can be found in the paper by Fytas, Martín-Mayor, Picco et al. (2017b), together with new results for the critical exponents, critical points and universal ratios of the RFIM at $D = 3, 4, 5$. The calculation of these quantities, along with their numerical estimates for the critical exponents α, β and γ satisfying the Rushbrooke relation, constitute an unarguable claim of universality and a verification of scaling relations, and provide a complete picture of the model's critical behavior for $D < D_{\mathrm{u}}$.

Another important finding of Fytas, Martín-Mayor, Picco et al. (2017b) is related to the dimensional reduction property of the RFIM (Young, 1977), a consequence of supersymmetry, as introduced in the seminal paper by Parisi and Sourlas (Parisi and Sourlas, 1979). Dimensional reduction states that the critical properties of the RFIM in D dimensions are expected to be the same as those of the pure Ising ferromagnet at $D-2$ dimensions. It is by now conclusively proved that this prediction is not true in three dimensions because the $3D$ RFIM orders (Bricmont and Kupiainen, 1987), while the $1D$ pure Ising ferromagnet does not. The property does not hold at $D = 4$ either (Fytas, Martín-Mayor, Picco et al., 2016). As for the $5D$ RFIM, they discovered that its critical exponents are compatible to those of the pure $3D$ Ising ferromagnet up to relatively small simulation errors and in agreement to the results obtained via functional renormalisation group by Tissier and Tarjus (2011), Tissier and Tarjus (2012) and Tarjus et al. (2013), pointing to a possible restoration of dimensional reduction at (or close to) $D = 5$. The first numerical evidence in favour of supersymmetry at the $D = 5$ RFIM have been presented very recently by Fytas, Martín-Mayor, Parisi et al. (2019).

Chapter 2

Monte Carlo methods

In this chapter, we describe the Monte Carlo (MC) numerical techniques used in the current thesis. More specifically, after a general introduction to the MC simulations in Sec. 2.1, the importance sampling and in particular the Metropolis algorithm are outlined in Sec. 2.2, followed by a description of the Wolff algorithm in Sec. 2.3. Subsequently, Sec. 2.4 includes an application of the MC methods (especially, Metropolis and Wolff algorithms) for the study of the interfacial adsorption of the Blume-Capel model. Finally, in Sec. 2.5, the limitations on MC methods are discussed.

2.1 Monte Carlo method description

It is well-known that for the most physical systems there are no accurate analytical solutions that predict their behaviour and characteristic properties. Therefore, one has to turn to approximate methods, which can be either analytical or computational. The continuous increment of the computational power since the invention of the first computational machines has led to the development of approximate numerical methods for the simulation of systems with complex structures and interactions.

One such method, which uses probability and random numbers theories in order to calculate the mean values of various quantities, is the MC method. The whole idea is based on the fact that using random numbers, one can estimate all the physical quantities of a system, such as the internal energy or the magnetisation. Statistical Physics provides the necessary tools for the calculation of the mean values of physical quantities of our interest. In the scope of the canonical ensemble, the mean thermal value of a quantity Q is given by:

$$\langle Q \rangle = \frac{\sum_i Q_i e^{-\beta E_i}}{\sum_i e^{-\beta E_i}}, \tag{2.1}$$

where E_i is the energy of the system when in micro-state i and Q_i is the value of the quantity in this micro-state. The probability of the micro-state is proportional to the Boltzmann factor $e^{-\beta E_i}$.

The estimation of Eq. (2.1) is a quite difficult procedure, given that not only a system's phase space may be large, but also an infinite number of states exist at the thermodynamic limit. For this reason, one resorts to approximations and calculates the sums in Eq. (2.1) in an appropriately defined subset of the system's states, hoping that this subset is representative of the overall behaviour of the system.

Namely, let us consider M micro-states $\{\mu_1, \mu_2, ..., \mu_M\}$ of the system. Then, the thermal average Q_M of the physical quantity Q is:

$$Q_M = \frac{\sum_{i=1}^{M} Q_{\mu_i} p_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}}{\sum_{i=1}^{M} p_{\mu_i}^{-1} e^{-\beta E_{\mu_i}}}, \tag{2.2}$$

where p_{μ_i} is the choice probability of micro-state μ_i and μ_i is the value of the thermodynamic quantity when in state Q_{μ_i}. It is clear that the restriction of the sums in the above subset of the micro-states leads to the introduction of statistical errors. What is more, the bigger and the more representative the above subset is, the more reliable the results of Eq. (2.2) are. When $M \to \infty$, then there holds:

$$\lim_{M \to \infty} Q_M = \langle Q \rangle. \tag{2.3}$$

In a Monte Carlo simulation, the basic subject under question is the choice of a representative subset of micro-states of the system's phase space. The answer to this question is given with the help of Markov processes, which are stochastic processes that generate random micro-states. A first-order Markov process yields a random micro-state, where its probability depends only on the micro-state attained in the previous event. First-order Markov processes are the ones which are extensively used in the MC techniques of Statistical Physics.

Given a state i, the probability of moving to the next state j is the transition probability $P_{i \to j}$. In a Markov process, all transition probabilities satisfy two conditions. Firstly, they do not change with time and secondly, they only depend on the properties of the initial micro-state i and final micro-state j. In addition, the transition probabilities satisfy the following equation:

$$\sum_j P_{i \to j} = 1, \tag{2.4}$$

due to the fact that in a Markov process, given the initial micro-state i, the final micro-state j has to be produced.

In a MC procedure, the role of a Markov process is to generate a sequence of micro-states of the system that occur according to a stationary distribution, after a certain equilibration time. Two more conditions need to be satisfied in order for this to be achieved: ergodicity and detailed balance (Newman and Barkema, 1999).

The first condition, namely, the ergodicity, expresses the requirement that starting from any possible micro-state of the system, a Markov process will eventually generate all the possible micro-states of the phase space. Without doubt, in macroscopic systems, the time scales needed for producing all the possible micro-states of the system are very large and lead to ergodicity breaking of the thermodynamic equilibrium state (Newman and Barkema, 1999). Disordered systems, in particular the ones with glassy disorder, exhibit ergodicity breaking of their phase space and it is common for the systems to be trapped for long time to non representative subsets of the phase space. Therefore, the Markov process fails to evolve. Moreover, the computational time as well as the errors in the configuration of the mean values of the physical quantities increase.

The second condition, i.e. the detailed balance, ensures that the Markov process finally approaches the stationary distribution and is mathematically described by:

$$\frac{P_{i\to j}}{P_{j\to i}} = \frac{p_j}{p_i}. \tag{2.5}$$

At this point one may wonder which is the most appropriate Markov process and the corresponding transition probabilities that lead as fast as possible to equilibrium. The answer is provided by the acceptance ratio, which is based on the idea that if $i = j$ in Eq. (2.5), then $1 = 1$. This identity implies that the detailed balance is always satisfied for $P_{i\to i}$ and does not depend on its value. This in turn indicates that one is flexible in the choice of values for the rest transition probabilities, $i \neq j$. Hence, we can adjust the value of any $P_{i\to j}$, while Eq. (2.4) is satisfied, by appropriately selecting the value of $P_{i\to i} \in [0, 1]$. This would obviously induce the corresponding adjustment of $P(j \to i)$, in order to satisfy Eq. (2.5).

From the above discussion, it is evident that the ability to change the transition probabilities $P_{i\to i}$ provides us with the facility to select the values of $P_{i\to j}$. In turn, this fact allows the re-formulation of the expression for the transition probabilities $P_{i\to j}$ as follows:

$$P_{i\to j} = S_{i\to j} A_{i\to j}. \tag{2.6}$$

$S_{i\to j}$ is the selection probability and corresponds to the probability that the algorithm yields a new micro-state j, given that the system is in a micro-state i and $A_{i\to j} \in [0,1]$ is the acceptance ratio that corresponds to the probability of accepting the transition of the system from the state i to the state j, produced by the algorithm. Via Eq. (2.6), Eq. (2.5) now writes:

$$\frac{P_{i\to j}}{P_{j\to i}} = \frac{S_{i\to j}A_{i\to j}}{S_{j\to i}A_{j\to i}}, \tag{2.7}$$

where $\frac{A_{i\to j}}{A_{j\to i}} \in [0,\infty)$ and hence the selection probabilities $S_{i\to j}$ and $S_{j\to i}$ can be assigned to any value we desire.

2.2 The Metropolis algorithm

The simplest way to carry out a Monte Carlo sampling is the random choice of micro-states of the system's phase space with the same probability. It appears that this simple sampling is correct at the limit $M \to \infty$. However, since the sampling takes place on a finite number of micro-states of the system and since there is no way to predict which are the appropriate states for each temperature, it is obvious that the simple sampling shall fail to correctly determine the properties of the system.

In contrast to simple sampling, the importance sampling ensures that for every temperature T, the biggest proportion of the chosen micro-states belongs to the area of the maximum of the energy distribution.

Let us consider a system in thermodynamic equilibrium with its environment at some temperature T. Then, according to the importance sampling, the choice probability of a micro-state μ_i of the system with energy E_{μ_i} is:

$$p_{\mu_i} = \frac{e^{-\beta E_{\mu_i}}}{\mathcal{Z}}, \tag{2.8}$$

where $\mathcal{Z}$ is the partition function. Now via Eq. (2.8), Eq. (2.2) writes:

$$Q_M = \frac{1}{M}\sum_{i=1}^{M} Q_{\mu_i}, \tag{2.9}$$

where Q_M is the thermal average. In other words, in importance sampling, the Boltzmann distribution is the equilibrium distribution. Thus, Eq. (2.5) leads to:

$$\frac{P_{i\to j}}{P_{j\to i}} = \frac{p_j}{p_i} = e^{-\beta(E_j - E_i)}. \tag{2.10}$$

The Metropolis algorithm (Metropolis et al., 1953) was the first importance sampling algorithm to be developed and its idea is based on the MC simulations and importance sampling discussed above. As we shall see, it guarantees that both ergodicity and detailed balance are satisfied.

Let us initially examine the choice probabilities $S_{i\to j}$ for the Metropolis algorithm. In general, the choice probabilities must be selected in a way that ergodicity is satisfied. It is clear that this can be achieved in many ways. For example, given an initial micro-state i, one can generate a number of nominated micro-states j, by flipping several subsets of the spins of the system under study. However, according to the classic theory of Statistical Physics, the relative variance of an extended thermodynamic quantity with N degrees of freedom is $\sim \sqrt{N}$. This indicates that systems in thermodynamic equilibrium are most of the time at states whose energies are quite similar to each other. Therefore, the simplest way is to flip a simple spin each time, ensuring that the energy does not significantly change, compared to the initial energy and that there is no waste of computational time. An algorithm that carries out this procedure is said to have single spin-flip dynamics.

It is obvious that all algorithms that include single spin-flip dynamics are ergodic, since in a finite size lattice the system can go from any micro-state to any other micro-state via consecutive spin-flips. Metropolis algorithm falls into this category and the choice probabilities $S_{i\to j}$ for any of the possible states j are picked to be equal to each other. In other words, in a system of N spin variables, given an initial micro-state i, there exist N different micro-states j and hence N choice probabilities $S_{i\to j}$, where $S_{i\to j} = \frac{1}{N} \neq 0$ and their sum equals to 1.

Based on this choice probability, the detailed balance condition given by Eq. (2.10) can now be written as:

$$\frac{P_{i\to j}}{P_{j\to i}} = \frac{S_{i\to j}}{S_{j\to i}} \frac{A_{i\to j}}{A_{j\to i}} = \frac{\frac{1}{N}}{\frac{1}{N}} \frac{A_{i\to j}}{A_{j\to i}} = e^{-\beta(E_j - E_i)}, \tag{2.11}$$

which yields the expression for the acceptance ratio $A_{i\to j}$ used in the Metropolis algorithm:

$$A_{i\to j} \equiv \begin{cases} e^{-\beta(E_j - E_i)}, & E_j - E_i > 0 \\ 1, & \text{otherwise} \end{cases}. \tag{2.12}$$

This implies that if during the random walk a state with energy less or equal to the initial one is chosen, then the step is acceptable. On the contrary, if the energy of the state chosen is larger than the energy of the initial state, then the

step is acceptable according to the probability given by Eq. (2.12). Let us point out here that every algorithm that uses acceptance ratios like the one described by Eq. (2.12) can be considered to be of Metropolis type.

The success of the Metropolis algorithm in the study of simple models in Statistical Physics has resulted in its establishment as the fundamental tool for performing MC simulations. However, the difficulties that arose from implementing the Metropolis algorithm in the study of the Ising model have provoked the development of new, more efficient algorithms (Newman and Barkema, 1999). These problems are related to the increase of the critical fluctuations and of the corresponding correlation times, as $T \rightarrow T_c$ (see, also, Sec. 2.5). This phenomenon is known as critical slowing down and it is described via the dynamic exponent z , where

$$\tau \sim \xi^z \tag{2.13}$$

at the critical point and τ is the correlation time.

It is well-known that as the critical temperature is approached, the system tends to formulate same-spin clusters that induce large fluctuations on the values of the fundamental thermodynamic quantities, such as energy and magnetisation, when flipped. Such fluctuations, combined with large correlation times, lead to the increase of statistical errors close to the phase transition. Despite the fluctuations being an innate physical feature of each model, the correlation times do depend on the algorithm used (Newman and Barkema, 1999). Therefore, in an attempt to reduce the exponent z, various new and more complex cluster spin-flipping algorithms were developed during the 80's. The most important are the Wolff (Wolff, 1989) and the Swendsen-Wang (Swendsen and Wang, 1987) algorithms.

2.3 The Wolff algorithm

As discussed in Sec. 2.2, as we approach the critical temperature of the Ising model, single spin-flip algorithms, such as the Metropolis algorithm, exhibit large correlation times that contribute to the growth of statistical errors. Conversely, cluster spin-flipping algorithms, such as the Wolff algorithm, seem to weaken significantly the error effect.

2.3.1 The dynamic exponent z

Before proceeding to the description of the Wolff algorithm, let us focus our attention in some more detail on the role of the dynamic exponent z appearing in Eq. (2.13). As mentioned in Sec. 2.2, the critical slowing down is measured via z, which takes different values for each different MC algorithm. According to Eq. (2.13), a small or large value of z indicates a small or large correlation time τ, respectively, while $z = 0$ implies that there is no critical slowing down at all.

Note that Eq. (2.13) indicates that close to a phase transition, the correlation time increases as the correlation length ξ diverges. Nonetheless, in a finite-size system, the correlation length, which denotes the typical size of the spin clusters formulated during the equilibration of an Ising model, does not ever really diverge. In a system of lattice size L, the largest value a spin cluster can attain is L^D, where D is the dimension of the system. Thus, close to a phase transition where $\xi > L$, Eq. (2.13) writes:

$$\tau \sim L^z. \tag{2.14}$$

Given the fact that we know the critical temperature T_c of a system, such as the case of the $2D$ Ising model, we can use Eq. (2.14) to exactly calculate the dynamic exponent z, by performing a FSS-type technique, namely, performing simulations at $T = T_c$ for different system sizes L and plotting τ as a function of L on logarithmic scales. The slope of the occurring plot will then be the value of z.

Many MC estimations of the dynamic exponent z have been carried out, specifically for the case of the Metropolis algorithm. The most accurate so far has been performed by Nightingale and Blöte (1996) and reads $z = 2.1665 \pm 0.0012$.

The above value for z, which is quite high compared to other MC algorithms for the $2D$ Ising model, implies that the Metropolis algorithm is not the most efficient algorithm for studying the behaviour of the Ising model close to its critical point. Therefore, in Sub. 2.3.2 we introduce the Wolff algorithm, which despite being more complex, has a significantly lower dynamic exponent.

2.3.2 Wolff algorithm description

As we saw in Sub. 2.3.1, the dynamic exponent z for the Metropolis simulation of the $2D$ Ising model is fairly high compared to other MC algorithms. This is mainly due to the fact that inside the critical area the correlation length diverges

and critical fluctuations emerge. Then, large same-spin clusters are formulated, namely, large same-spin domains.

In order for the Metropolis algorithm to flip a whole spin domain, a spin-by-spin flip has to be performed. This is a quite difficult and time-consuming process, since the ferromagnetic interactions between nearest neighbours lower the choice probability of each spin flip. In other words, a spin with 4 same-spin neighbours has low chances to be flipped, since such a move would have a high energy cost. Despite the chances of flipping a spin at the edge of the domain, where the same-spin neighbours are less than 4 and hence the energy cost is lower, are higher, it is a very slow process that gets slower and slower as we approach the phase transition where the domains become larger.

An algorithm capable of tackling the above problem was suggested by Wolff (1989), where instead of single spin-flips, cluster spin-flips are performed. The clusters which are flipped consist of spins of the same orientation. Such cluster spin-flipping algorithms have become the most predominant among the MC algorithms and in the case of the Ising model, they almost completely eliminate the critical slowing down phenomenon.

Let us examine how the clusters to be flipped are determined. Starting from any random spin in the lattice, the algorithm checks whether any of its neighbours is of the same spin. If it is, it is added to the same cluster with the initial spin. Now the same check is repeated for the new spin until an entire cluster of spins is built up, that is, until there exist no more same-spin neighbours. However, we do not want the cluster that is about to be flipped to include all of the lattice spins with the same orientation as the initial one.

The number of the spins that the algorithm will flip depends on the temperature. For example, in the Ising model, at high temperatures, the spins form very small clusters, namely, they are strongly uncorrelated, while when moving close and below the critical temperature T_c, the spin clusters are becoming larger. In order to mimic this physical behaviour, Wolff algorithm chooses whether or not to add the (same-oriented) neighbouring spins to the same cluster as the initial spin, according to some probability P_{add} which increases as the temperature decreases. When no more neighbouring spins of the same orientation satisfy the probability P_{add}, the growth of the cluster stops and the cluster is flipped. The flip is performed with some acceptance ratio that depends on the energy cost of the flip.

In order to define the acceptance ratio, let us consider two micro-states of the

Metropolis:	$z = 2.1665 \pm 0.0012$
Wolff:	$z = 0.25 \pm 0.02$

TABLE 2.1: Best estimations of the dynamic exponents z for the Metropolis and Wolff algorithms.

system, namely, i and j, which are distinct from each other by a single same-spin cluster flip. Then, the acceptance ratio is given by:

$$\frac{A_{i\to j}}{A_{j\to i}} = [e^{2\beta J}(1 - P_{\text{add}})]^{n-m}, \tag{2.15}$$

where m and n are the bonds between neighbouring spins at the edges of the cluster which need to be broken in order for the cluster to be flipped during a forward or a backward move, respectively.

At this point we should also mention the ergodicity of the algorithm, by noticing that for every move, there is a finite chance for a spin to be chosen from a cluster which is afterwards flipped. Repeating moves in a similar matter will apparently get us from a state to another in finite time. This, together with the detailed balance condition, implies that the algorithm will yield consecutive states that will appear with their correct Boltzmann probabilities (Newman and Barkema, 1999).

Let us finally examine how Wolff algorithm's dynamic exponent z compares to the dynamic exponent calculated by Nightingale and Blöte for the Metropolis algorithm. The best estimation for the exponent z for the Wolff algorithm was given by Coddington and Ballie (1992) and it writes $z = 0.25 \pm 0.02$. As we can clearly observe in Table 2.1, the dynamic exponent of the Wolff algorithm is significantly lower that the one calculated for the Metropolis algorithm, fact that implies that the Wolff algorithm is undoubtedly better that the Metropolis algorithm for studying systems close to their phase transitions.

2.4 An application of MC methods: Interfacial adsorption of the Blume-Capel model

In this section, we present an application of the Monte Carlo numerical methods and in particular, of the Metropolis and Wolff algorithms outlined in Sec. 2.2 and Sec. 2.3, respectively. For this purpose, we investigate the scaling of the interfacial adsorption of the $2D$ Blume-Capel (BC) model using MC simulations.

More specifically, we study the FSS behavior of the interfacial adsorption of the pure model at both its first- and second-order transition regimes, as well as at the vicinity of the tricritical point. Our analysis benefits from the currently existing quite accurate estimates of the relevant (tri)critical-point locations. In all studied cases, the numerical results verify to a level of high accuracy the expected scenarios derived from analytic free-energy scaling arguments.

We also investigate the size dependence of the interfacial adsorption under the presence of quenched bond randomness at the originally first-order transition regime (disorder-induced continuous transition) and the relevant self-averaging properties of the system. For this ex-first-order regime, where strong transient effects are shown to be present, our findings support the scenario of a non-divergent scaling, similar to that found in the original second-order transition regime of the pure model.

The above work, which is thoroughly discussed in this section was published by Fytas, Mainou, Theodorakis et al. (2019) and the rest of the current section is largely a reproduction of the aforementioned publication.

Let us start by pointing out that critical interfacial phenomena have been studied extensively over the last decades, both experimentally and theoretically (Abraham, D. B., 1986; Dietrich, S., 1988; Bonn et al., 2009; Ralston et al., 2008). A well-known example is wetting, where the macroscopically thick phase, for example, the fluid, is formed between the substrate and the other phase, say, the gas. Liquid and gas are separated by the interface.

An interesting complication arises when one considers the possibility of more than two phases. A third phase may be formed at the interface between the two other phases. An experimental realization is the two-component fluid system in equilibrium with its vapor phase (Dietrich, S., 1988; Moldover and Cahn, 1980).

Both of the above scenarios may be mimicked in statistical physics in a simplified fashion, by either the two-state Ising model in wetting - with the state "+1" representing, say, the fluid, and "−1" the gas - or for the case of a third phase via multi-state spin models, simply by fixing distinct boundary states at the opposite sides of the system. In this latter case, the formation of the third phase with an excess of the non-boundary states has been called as interfacial adsorption (Selke and Pesch, 1982; Selke and Huse, 1983; Fisher, 1984).

Throughout the years, various aspects of the interfacial adsorption have been investigated via MC methods and density renormalization group calculations on the basis of specific multi-state spin models, namely Potts and BC models (Selke

and Pesch, 1982; Selke and Huse, 1983; Selke and Yeomans, 1983; Selke, Huse and Kroll, 1984; Yamagata, 1991; Yamagata and Kasono, 1992; Carlon et al., 1999; Albano and Binder, 2012; Fytas and Selke, 2013; Trobo and Albano, 2014; Albano and Binder, 2014).

Additional scaling and analytic arguments have been presented (Selke and Huse, 1983; Selke, Huse and Kroll, 1984; Carlon et al., 1999; Bricmont and Lebowitz, 1987; Messager et al., 1991; Cardy, 2000), though not all of them have been confirmed numerically, due to the restricted system sizes studied and, in some cases, the uncertainty in the location of (tri)critical points. However, notable results in the field include the determination of critical exponents and scaling properties of the temperature and lattice size dependencies, as well as the clarification of the fundamental role of the type of the bulk transition, with isotropic scaling holding at continuous and tricritical bulk transitions and anisotropic scaling at bulk transitions of first-order type.

More recently, a formulation of the field theory of phase separation by Delfino and colleagues has provided new insight into the problem (Delfino and Viti, 2012; Delfino and Squarcini, 2013; Delfino and Squarcini, 2014a; Delfino and Squarcini, 2014b; Delfino and Squarcini, 2015; Delfino, 2016; Delfino and Squarcini, 2016; Delfino, Selke et al., 2018) and what is more, the role of randomness has been scrutinized on the basis of the disordered Potts model (Monthus and Garel, 2008; Brener, 2010; Fytas, Malakis et al., 2015; Fytas, Theodorakis et al., 2017).

Clearly the Potts model offers the unique advantage that if one considers the system at its self-dual point, then, the phase-transition temperatures between the ordered ferromagnetic phase and the high-temperature disordered phase are known exactly from self-duality for arbitrary values of the internal states q and particular implementations of the randomness distribution (Kinzel and Domany, 1981).

On the other hand, for the Blume-Capel model, one relies upon the existing estimates for the locations of (tri)critical and transition points and this may be a source of systematic error when uncovering the scaling behavior of the interfacial adsorption, as has already been underlined in the literature (Selke, Huse and Kroll, 1984).

However, quite recently, important progress has been reported with respect to an accurate reproduction of the phase diagram of the model for a wide range of its critical parameters (Silva et al., 2006; Malakis, Berker, Hadjiagapiou et al., 2009; Malakis, Berker, Hadjiagapiou et al., 2010; Kwak et al., 2015; Zierenberg

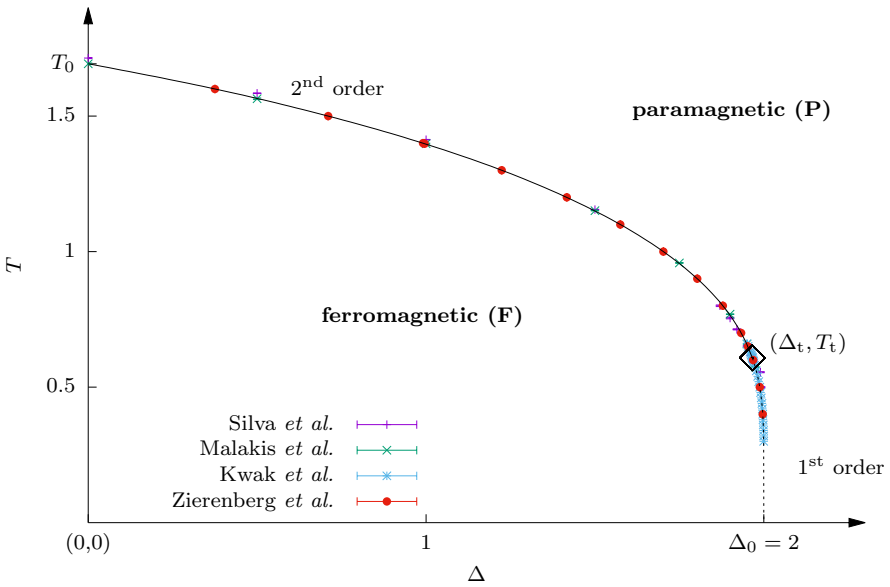

FIGURE 2.1: Phase diagram of the square-lattice zero-field Blume-Capel model in the Δ - T plane. The phase boundary separates the ferromagnetic (F) phase from the paramagnetic (P) phase. The solid line indicates continuous phase transitions and the dotted line marks first-order phase transitions. The two lines merge at the tricritical point (Δ_t, T_t), as highlighted by the black diamond. The data shown are selected estimates from previous numerical studies.

et al., 2017; Fytas, Zierenberg et al., 2018), thus motivating the current study. In particular, in the present thesis we investigate the FSS behavior of the interfacial adsorption of the $2D$ square-lattice BC model, at both the continuous and first-order transition regimes of its phase diagram, as well as at the vicinity of the tricritical point. Furthermore, we study the effect of quenched bond randomness on the interfacial adsorption at the disorder-induced continuous transition.

Our discussion below follows the seminal works by Selke and collaborators (Selke and Yeomans, 1983; Selke, Huse and Kroll, 1984), where the first MC results for the pure BC model have been presented, corroborated by analytical scaling arguments, which we will also outline for the benefit of the reader in cases where direct comparison with the numerical data is possible.

In a nutshell, the main objectives of the current work are as follows: For the pure case, previous numerical findings (Selke and Yeomans, 1983; Selke, Huse and Kroll, 1984) based on less extensive simulations are scrutinized, confirmed and refined to a high-level of numerical accuracy, especially for the areas around the tricritical point and the first-order transition line in the $\Delta - T$ plane (as will

be explicitly elaborated in the discussion of Fig. 2.2 and Fig. 2.3 below). Completely new results are presented for the random case, an aspect that has not been previously considered in the relevant literature, where an intriguing crossover behaviour, with a finite interfacial adsorption, at the randomness-induced continuous transition is observed and explained.

We consider the Blume-Capel model (Blume, 1966; Capel, 1966) defined by the Hamiltonian:

$$\mathcal{H} = -J \sum_{\langle xy \rangle} s_x s_y + \Delta \sum_x s_x^2. \tag{2.16}$$

The spin variable s_x takes on the values -1, 0, or $+1$, $\langle ij \rangle$ indicates summation over nearest neighbors and $J > 0$ is the ferromagnetic exchange interaction. The parameter Δ denotes the crystal-field coupling and controls the density of vacancies ($s_x = 0$). For $\Delta \to -\infty$ vacancies are suppressed and the model becomes equivalent to the Ising model.

The phase diagram of the BC model in the crystal-field - temperature plane consists of a boundary that separates the ferromagnetic from the paramagnetic phase (see Fig. 2.1). The ferromagnetic phase is characterized by an ordered alignment of ± 1 spins. The paramagnetic phase, on the other hand, can be either a completely disordered arrangement at high temperature or a ± 1-spin gas in a 0-spin dominated environment for low temperatures and high crystal fields. At high temperatures and low crystal fields, the ferromagnetic-paramagnetic transition is a continuous phase transition in the Ising universality class, whereas at low temperatures and high crystal fields the transition is of first-order character (Blume, 1966; Capel, 1966).

The model is thus a classical and paradigmatic example of a system with a tricritical point (Δ_t, T_t) (Lawrie and Sarbach, 1984), where the two segments of the phase boundary meet. At zero temperature, it is clear that ferromagnetic order must prevail if its energy $\frac{zJ}{2}$ per spin exceeds that of the penalty Δ for having all spins in the ± 1 state. Note that z is the coordination number and $z = 4$ in the present case (see, also, Subsec. 3.1.1). Hence the point $(\Delta_0 = \frac{zJ}{2}, T = 0)$ is on the phase boundary (Capel, 1966).

For zero crystal-field Δ, the transition temperature T_0 is not exactly known, but well studied for a number of lattice geometries. A most recent reproduction of the phase diagram of the model can be found in Zierenberg et al., 2017 and is also given here in Fig. 2.1, where a summary of results is presented from various works in the literature. We set $J = 1$ and $k_B = 1$, to fix the temperature scale. A

recent accurate estimation of the location of the tricritical point has been given by Kwak et al. (2015): $(\Delta_t, T_t) = [1.9660(1), 0.6080(1)]$.

In order to study the interfacial adsorption, denoted hereafter as W and following the work of Selke and collaborators (Selke and Yeomans, 1983; Selke, Huse and Kroll, 1984) we shall employ special boundary conditions, distinguishing the cases $[1:1]$ and $[1:-1]$ that will favor the formation of an interface within the system. For the case $[1:1]$, the spin variable is set at all boundary sites equal to 1, while for the case $[1:-1]$ the variable is set equal to 1 at one half of the boundary sites and to -1 at the opposite half of the boundary sites.

Typical equilibrium configurations have verified that under these special boundary conditions an excess of the non-boundary states, $s_i = 0$, is generated at the interface (see for instance Fig. 1 in Selke and Yeomans, 1983). This phenomenon is described quantitatively by the net adsorption per unit length of the interface, that is defined with the help of the following mathematical expression (Selke and Yeomans, 1983):

$$W = \frac{1}{L} \sum_x \left[\langle \delta_{0,s_x} \rangle_{[1:-1]} - \langle \delta_{0,s_x} \rangle_{[1:1]} \right], \tag{2.17}$$

where the angular brackets denote thermal averages and L is the linear dimension of the square lattice.

The critical behavior of W is characterized by the critical exponents x and ω via Selke, Huse and Kroll (1984):

$$W_L \sim L^x \quad (T = T_c), \tag{2.18}$$

and

$$W_{t_c} \sim t_c^{-\omega} \quad (L = \infty), \tag{2.19}$$

where $t_c = \frac{T_c - T}{T_c}$ is the reduced critical temperature for the standard case of a critical point. Although the above Eqs. (2.18) and (2.19) are expressed for the usual case of continuous transitions, they can be similarly generalized for the case of a tricritical point, where $t_t = \frac{T_t - T}{T_t}$, or for a first-order phase transition, $t^* = \frac{T^* - T}{T^*}$, where T^* denotes now the corresponding transition temperature.

In this thesis we have studied the interfacial properties of the system at three values of the crystal-field coupling Δ, including both the first- and second-order lines of the transition, but also the tricritical point of the phase diagram shown in Fig. 2.1. We have considered the values $\Delta = 1$ (second-order regime), $\Delta = \Delta_t = 1.966$ (tricritical point) and $\Delta = 1.975$ (first-order regime). The corresponding

transition temperatures for the cases $\Delta = 1$ and $\Delta = 1.975$ have been estimated to be $T_c = 1.398$ and $T^* = 0.574$, respectively (Malakis, Berker, Hadjiagapiou et al., 2010), whereas for the case of the tricritical point we have used the most recent estimate $T_t = 0.608$ (Kwak et al., 2015).

Additionally, for the case $\Delta = 1.975$ of the originally first-order transition regime, we have also considered the disordered version of the Hamiltonian (2.16) by selecting ferromagnetic couplings $J \rightarrow J_{ij}$ between nearest-neighbor sites i and j, to be either J_1, with probability p, or J_2 with probability $1 - p$. In the case $J_1 > J_2$, one has either strong or weak bonds. Then, the ratio $r = \frac{J_2}{J_1}$ defines the disorder strength, where $\frac{J_1+J_2}{2} = 1$. Clearly, the value $r = 1$ corresponds to the pure model. For the needs of the present work we fixed the ratio $r = 0.6$, for which the critical temperature of the disorder-induced continuous transition has been estimated to be $T_c = 0.626$ (Malakis, Berker, Hadjiagapiou et al., 2010).

Our numerical protocol consists of canonical MC simulations, employing a combination of a Wolff single-cluster update (Wolff, 1989) of the ± 1 spins and a single-spin flip Metropolis update that enables the necessary updates of the vacancies $s_i = 0$ (Blöte et al., 1995; Hasenbusch, 2010; Malakis, Berker, Fytas et al., 2012).

We adapted the relative frequencies of using the two updates to optimize the performance and discarded the initial part of each time series to ensure equilibration. Using this approach, we simulated for both versions of the model and for all values of Δ system sizes in the range $L = 8 - 96$, which, as will be shown below, is enough for a safe estimation of the asymptotic behavior, in accordance with the expected scaling arguments.

For the pure model, we performed several independent runs to increase statistical accuracy, whereas for the disordered system an extensive averaging over the disorder [...] has been undertaken, varying from 5×10^3 realizations for the smaller system sizes down to 1×10^3 for the larger sizes studied. For the disordered case, error bars were computed from the sample-to-sample fluctuations which in all cases were found to be larger than the statistical errors of the single disorder realizations.

For the various cases of phase transitions in the BC model along the Δ - T plane, some very useful analytic and scaling arguments for the interfacial adsorption have been presented in the early work of Selke, Huse and Kroll (1984). In what follows, we shall only provide the main results of this discussion that are also

relevant for comparison with our numerical data; for more details we refer the reader to Selke, Huse and Kroll (1984).

The main point in this description is the reformulation of the interfacial adsorption W with the help of the interface tension σ. According to Selke, Huse and Kroll, 1984, using that $\langle \delta_{0,s_x} \rangle = 1 - \langle s_x^2 \rangle$, the interface adsorption may be written in the form $W = \frac{1}{L} \sum_x \left[\langle s_x^2 \rangle_{[1:1]} - \langle s_x^2 \rangle_{[1:-1]} \right]$. Denoting the total free energy for $[1:1]$ boundary conditions by $F_{[1:1]}$ (similarly, $F_{[1:-1]}$ for $[1:-1]$), W can then be expressed in terms of the interface tension, $\sigma = \frac{1}{L}(F_{[1:1]} - F_{[1:-1]})$, as $W = \frac{\beta^{-1} \partial \sigma}{\partial \Delta}$, where $\beta = \frac{1}{k_B T}$.

The presentation of our FSS analysis starts with the most interesting cases referring to the vicinity of the tricritical point and the first-order transition regime. As already mentioned above, the location of the tricritical point of the BC model is known today with very good accuracy (Kwak et al., 2015), thus removing one source of error inherent in previous simulation works (Selke and Yeomans, 1983; Selke, Huse and Kroll, 1984). According to the scaling arguments of Selke, Huse and Kroll (1984), the exponents appearing in Eqs. (2.18) and (2.19) take on the values $x = \frac{4}{5}$ and $\omega = \frac{4}{9}$, respectively, for the case of the tricritical point.

In Fig. 2.2 we present our numerical data and the relevant scaling analysis for the interfacial adsorption W_L (main panel) and W_{t_t} (inset) at $\Delta = \Delta_t = 1.966$. Fits of the form (2.18) and (2.19) shown by the solid lines in the main panel and the corresponding inset respectively, provide us with the estimates $x = 0.802(3)$ and $\omega = 0.4441(5)$, both fully consistent with the expected values $x = \frac{4}{5}$ and $\omega = \frac{4}{9}$.

We should point out here that the numerical estimation of the exponent x for the tricritical point has been reported as a quite difficult task in the literature, due to the imprecise knowledge of the tricritical coordinates (see Fig. 7 in Selke and Yeomans, 1983 where $\Delta_t \approx 1.92(2)$) and the presence of strong finite-size effects for small system sizes (see Fig. 3 in Selke, Huse and Kroll, 1984) where for the actual value of $\Delta_t = 1.966$ an effective exponent of the order of ~ 0.65 is obtained. Both of these adversities have been satisfied in the present thesis, leading to a clear verification of the scaling arguments presented by Selke, Huse and Kroll (1984).

As is well known, at the critical (and tricritical) points, the singularities in the interfacial adsorption are induced by bulk critical fluctuations. On the other hand, at first-order phase transitions there are no bulk critical fluctuations and the divergence of W arises from an interface delocalization transition (Lipowsky et al., 1983).

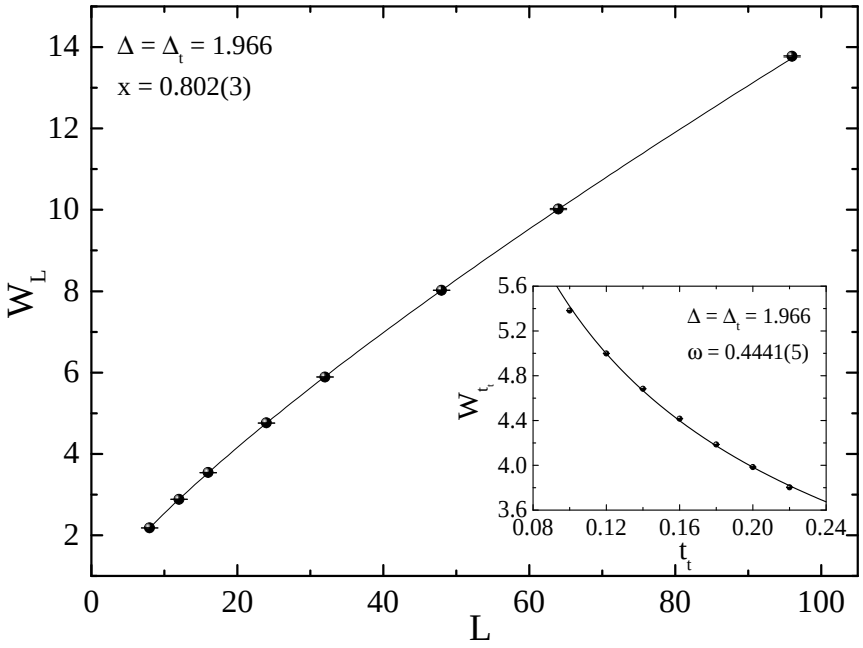

FIGURE 2.2: Finite size scaling of the interfacial adsorption W_L (main panel) and W_{t_t} (inset) at the tricritical point.

In the latter case and for lattices of square shapes a linear divergence of the form $W_L \sim L$ is expected, *i.e.*, $x = 1$ (Selke, Huse and Kroll, 1984). Additionally, the critical exponent ω appearing in Eq. (2.19) is expected to take the value $\frac{1}{3}$, as was originally found in the case of interface unbinding (Abraham and Smith, 1982) and further generalized for first-order phase transitions in $2D$ (Fisher, 1984; Selke and Yeomans, 1983; Selke, Huse and Kroll, 1984; Lipowsky et al., 1983).

For the case of the BC model, the prediction $\omega = \frac{1}{3}$ has been numerically confirmed (Selke and Yeomans, 1983; Selke, Huse and Kroll, 1984), though the numerical data for W_L did not allow for an accurate estimation of the exponent x. In particular, in Selke and Yeomans, 1983 a value of $x = 0.7 \pm 0.05$ has been found that was subsequently explained as an apparent exponent due to strong metastability effects (Selke, Huse and Kroll, 1984).

To fill in the gap with the scaling analysis of W_L at the first-order transition regime of the BC model, we present in Fig. 2.3 our numerical data for the interfacial adsorption obtained at $\Delta = 1.975$. The fitting results using the Eqs. (2.18) and

(2.19) as in Fig. 2.3, give $x = 1.00(2)$ and $\omega = 0.337(6)$, in excellent agreement with the theoretical expectations $x = 1$ and $\omega = \frac{1}{3}$.

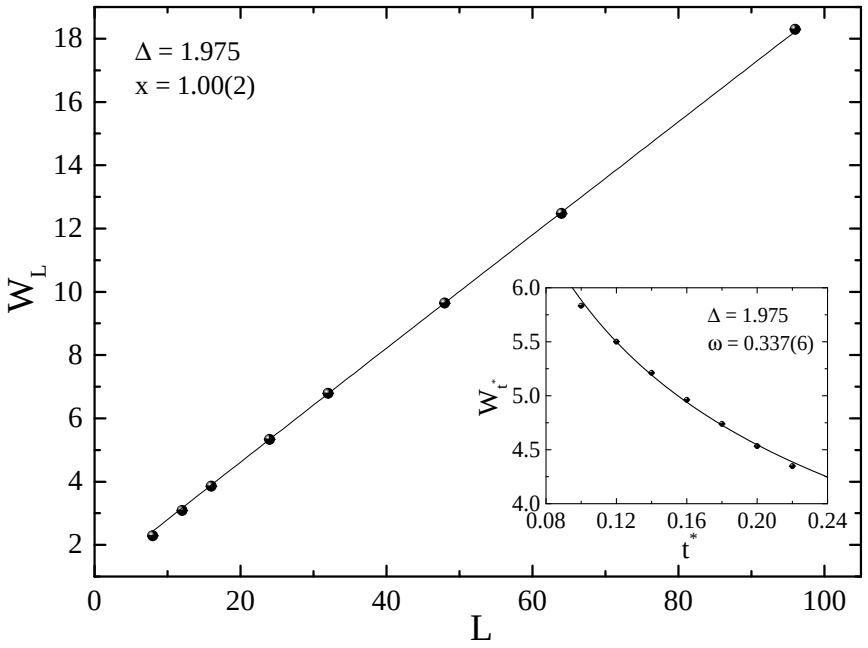

FIGURE 2.3: Finite size scaling of the interfacial adsorption W_L (main panel) and W_{t^*} (inset) at the first-order transition regime.

For the spin-$\frac{1}{2}$ Ising model it is known that $\sigma \sim t_c$ for $t_c \to 0^+$ at the critical point. Given that the BC model for $\Delta < \Delta_t$ belongs to the same universality class, we also expect a similar statement to hold, where now t_c may be the distance from the critical curve. Since Δ is a non-ordering field (Riedel, 1972), as was also concluded by Selke, Huse and Kroll (1984), $W \sim \frac{\partial \sigma}{\partial t_c} \sim$ const.

We present in Fig. 2.4 the FSS behavior of the interfacial adsorption W_L for $\Delta = 1$. Indeed, a simple power-law fit of the form $W_L = W_\infty + bL^x$ gives a negative exponent $x = -1.42(9)$ and a finite value of W_∞, thus a non-divergent behavior, in agreement with the above arguments. Similar results have been presented by Selke and Yeomans (1983) for a few values of Δ in the second-order transition regime but for smaller system sizes and are overall in contrast to the Potts case, where a clear diverging behavior has been observed in many relevant works (Selke and Pesch, 1982; Selke and Huse, 1983; Fytas, Theodorakis et al., 2017; Fytas, Theodorakis et al., 2017). This may be due to the different geometric nature of

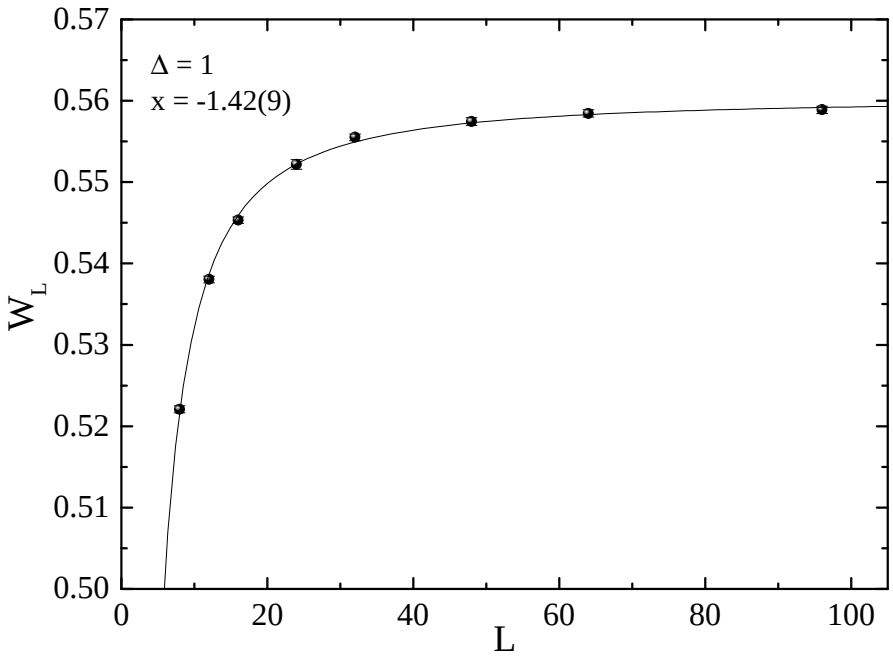

FIGURE 2.4: Finite size scaling of the interfacial adsorption W_L at the second-order transition regime.

the interfacial adsorption among the two models, which in the present BC model occurs in a layer-like fashion as expected on the basis of single spin-flip energy considerations (see Fig. 1 in Selke and Yeomans, 1983), whereas in Potts models a droplet-like adsorption of non-boundary states takes place due to the energetic equivalence of all states (Selke and Pesch, 1982).

The last part of this section is dedicated to the study of the interfacial adsorption under the presence of quenched bond randomness at the originally first-order phase transition regime of the phase diagram and particularly at the crystal-field value $\Delta = 1.975$.

Simulations have been performed for a single value of the disorder strength, namely $r = 0.6$, at the estimated by Malakis, Berker, Hadjiagapiou et al. (2010) critical temperature $T_c = 0.626$. The numerical data for the disorder-averaged $[W]_L$ are shown in the main panel of Fig. 2.5, where a very strong saturation is observed and should be compared to the diverging behavior of the corresponding pure system (see Fig. 2.2). This result is in agreement with the theoretical expectations discussed above for a non-divergent behavior of W in the case of continuous transitions for the present model.

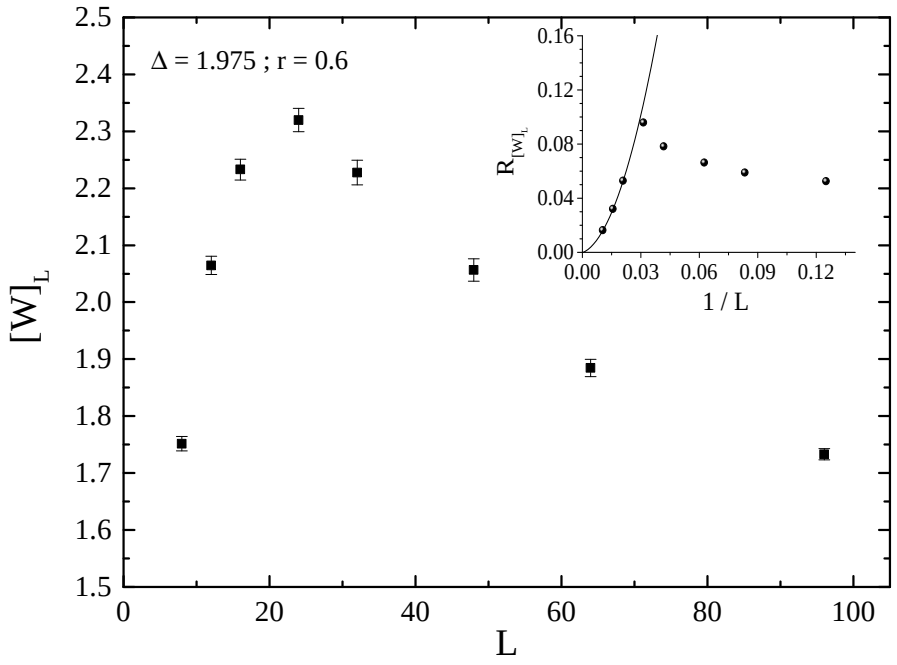

FIGURE 2.5: Finite size scaling of the disorder-averaged interfacial adsorption $[W]_L$ of the random-bond Blume-Capel model at the disorder-induced continuous transition. The inset illustrates the relevant self-averaging properties in terms of the relative-variance ratio $R_{[W]_L}$ as a function of the inverse system size.

In the inset of Fig. 2.5 we present the self-averaging properties of the system using the relative-variance ratio $R_{[W]_L} = \frac{V_{[W]_L}}{[W]_L^2}$, where $V_{[W]_L} = [W^2]_L - [W]_L^2$. The limiting value of this ratio is characteristic of the self-averaging properties of the system (Wiseman and Domany, 1998b; Wiseman and Domany, 1998a). The solid line in the inset illustrates a simple polynomial fit over the larger system sizes, indicating the restoration of self-averaging at the thermodynamic limit, given that $R_{[W]_L} \to 0$ as $L \to \infty$. Similar results have been presented for the case of various random-bond Potts models in $2D$ (Fytas, Theodorakis et al., 2017).

Finally, it is worth noting that the FSS behavior of both $[W]_L$ and $R_{[W]_L}$ is affected by strong transient effects with a crossover length-scale $L^* \approx 32$, where a turnaround in the behavior sets off. This is consistent with previous observations on the scaling behavior of the correlation length and other thermodynamic observables of the system for the same range of parameters (Fytas, Zierenberg et al., 2018). Indeed, in Fytas, Zierenberg et al., 2018 it has been explicitly shown

that $L \approx 32$ is the apparent size where the first-order characteristic signatures of the transition disappear. Of course, we expect that the value of L^* depends on the disorder strength r as well as on the strength of the first-order transition and it would be interesting to investigate the shift of this crossover length-scale as a function of Δ and r. However this is a task that goes beyond the scope of the present work.

To conclude, we have investigated the scaling aspects of the interfacial adsorption of the $2D$ BC model using a combined MC scheme. We presented a detailed analysis of the FSS behavior of the interfacial adsorption of the pure model at both its first- and second-order transition regimes, as well as at the area of the tricritical point, taking advantage of the current high-accuracy estimates of (tri)critical-point locations.

A dedicated part of our work regarding the scaling of the interfacial adsorption under the presence of quenched bond randomness at the originally first-order transition regime (disorder-induced continuous transition) revealed the scenario of a non-divergent scaling, similar to that found in the original second-order transition regime of the pure model. Overall, our results and analysis nicely verified the predicted from analytic arguments scaling scenarios of Selke, Huse and Kroll (1984), overcoming the numerical difficulties highlighted in that seminal work.

2.5 Limitations on MC methods

In this final section of Chap. 2, we summarize the most crucial challenges arising from the use of MC techniques for the simulation of complex systems and in particular of the Ising model. The limitations discussed below have played a crucial role to the development of not only different algorithms of MC type (see, for example, Sec. 2.2, Sec. 2.3) , but also of alternative methods coming from different research fields for the study of such systems, as we shall see in Chap. 3.

2.5.1 Critical fluctuations

As we briefly mentioned in Sec. 2.2, MC-type algorithms and especially the Metropolis algorithm have been proven to be efficient for the study of simple models in Statistical Physics. However, when focusing on the critical behaviour of more complex systems, such as the Ising model, one needs to recall that inside the critical region, large clusters of same-orientation spins are formulated. The

flip of a single large cluster thus leads to large fluctuations in M and E, namely, critical fluctuations (Newman and Barkema, 1999).

One can easily conclude that as we approach the critical temperature, since the size of the clusters diverges, the same shall hold for the critical fluctuations. This implies that M and E also diverge. In turn, Eq. (1.7) and Eq. (1.9) indicate that C and χ will also diverge as $T \to T_c$. The divergence of these thermodynamic quantities close to a phase transition are of particular interest and they often constitute the main or even sole point of focus when performing MC simulations. Nonetheless, this divergence is one of the leading factors that contribute to the inefficiency of MC methods inside the critical region.

Despite the fact that the systems one simulates are of finite size and hence the clusters' size and the thermodynamic quantities do not actually diverge, they can become very large and therefore lead to significant statistical errors. One can tackle this problem by increasing the number of measurements of a thermodynamic quantity, in order to reduce the error bars. However, more measurements require more time and the correlation time τ, which is already very large when approaching the critical temperature, does not allow a large number of measurements to take place. As an overall, the co-existence of large critical fluctuations and of large correlation time close to T_c does not leave scope for an efficient way to reduce the statistical errors of the measured quantities.

With the critical fluctuations emerging as an intrinsic property of the system, any MC algorithm which accurately performs importance sampling via the Boltzmann distribution will also yield critical fluctuations (Newman and Barkema, 1999). Simply put, critical fluctuations are inevitable and cannot be avoided regardless of any MC-type algorithm we may choose to use for studying the critical behaviour of a complex system.

2.5.2 Trapping at local minima and critical slowing down

In addition to the difficulties occurring due to the nature of the system, the large correlation time inside the critical region arises as a property of the MC algorithm we use. The growth of the correlation time as we approach the phase transition is called critical slowing down (Hohenberg and Halperin, 1977; Zinn-Justin, 2005) and it depends on the structure of each algorithm.

Let us examine how the critical slowing down occurs. Recall that our goal is to calculate the mean values of thermodynamic quantities such as the energy E and the magnetisation m, given by Eq. (2.1), for systems in the canonical ensemble.

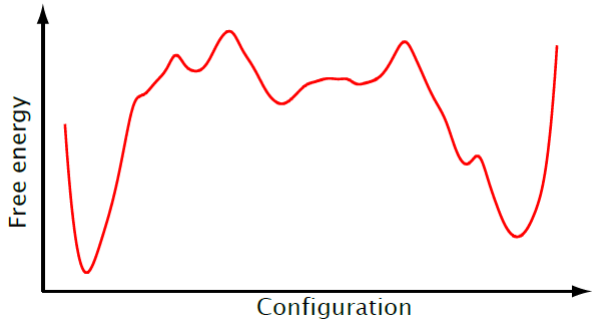

FIGURE 2.6: The free-energy landscape of a complex system. It contains many local minima, which define metastable states with exponentially long escape times.

The probability of each micro-state i is proportional to the Boltzmann factor $e^{-\beta E_i}$. Given an initial random state i, we employ a Markov process that generates a sequence of micro-states of the system that occur according to a stationary distribution, which is the physical distribution describing the equilibrium state of the system.

Reaching the stationary distribution close to a phase transition is particularly difficult, since the thermalisation times of the most typical MC algorithms grow as L^z, where L is the linear size of the system and $z \approx 2$. In cases of rugged free-energy landscapes (see, for example, Fig. 2.6) the phenomenon of critical slowing down becomes even more prevalent, with the thermalisation times growing exponentially with the free-energy barriers.

A fundamental example of multiple local minima is the case of the Random-field Ising model (RFIM), where a free-energy barrier exists between the ordered and the disordered states of the system, with only one of these configurations defining a stable phase (Mosquera, 2011). Now the barriers grow as L^z , where D is the spatial dimension of the system and $z < D - 1$. Thus, the existence of critical slowing down is evident, with $log\tau \sim L^z$ and given the fact that the RFIM is a disordered system, a large number of realisations of the disorder should be considered in order to obtain reliable results.

As we have already mentioned above, the growth of the correlation time τ inside the critical area alternates according to the algorithm in use. Therefore, despite the fact that we cannot avoid the critical fluctuations, the same does not hold for the effect of critical slowing down, since one may figure out appropriate algorithms that reduce or even eliminate this effect. In Chap. 3 we shall discuss such algorithms which are not of MC type and emerge from a research field that

at first glance seems to be irrelevant to the study of the critical behaviour of complex systems.

Chapter 3

Graph cut methods

In this chapter, we turn to the main focus of our research, which is the study of the critical behaviour of the RFIM with quenched disorder, by using graph cut methods at zero temperature. For this purpose, we start in Sec. 3.1 by focusing on the theoretical graph cut background which is necessary for the initial step of the implementation of the methods, namely, mapping the RFIM to a network. After the illustration of the mapping in Sec. 3.1, we proceed to describing some minimum cut-maximum flow algorithms and in particular, the Boykov-Kolmogorov algorithm in Sec. 3.2.

3.1 Mapping the RFIM to a network

When studying phase transitions in the presence of quenched disorder, the predominant methods used for working out the ground states of the Hamiltonians have been approximate techniques, such as MC simulations and simulated annealing. However, due to the limitations of these methods, combined with the fact that a crossing of the phase boundary at $T > 0$, $T < +\infty$ leads to the same critical behaviour as a crossing at $T = 0$ by varying h does, the study of disordered systems via calculating ground states with the use of network theory is gaining an increasing popularity. This has led to the extensive employment of such optimization techniques by various recent works (see Sec. 3.2). As a result, in this section, we focus on the main goal of this thesis, which is to take advantage of the idea of applying graph cut theory in order to address such energy minimization problems (Greig et al., 1989).

The idea of applying graph cut theory for the study of the GS properties of the RFIM stems from results triggered by the max-flow min-cut theorem (Ford and Fulkerson, 1956; Ford and Fulkerson, 1957), which plays a particularly important role in computer science and in network flows and optimization theories.

According to the theorem, the maximum flow from the source to the sink of any network is equal to the capacity of the minimum cut separating the source from the sink.

Hammer (1965) suggested a method for determining the minimum cut and hence the maximum flow through a network, by applying the idea of pseudo-Boolean programming (Hammer et al., 1968). It was shown by Hammer et al. (1968) that given a pseudo-Boolean function $F(x_1, ..., x_n) : L_2^n \rightarrow \mathbb{R}$, where L_2 is the Boolean algebra with the elements 0 and 1, determining all points $(x_1, ..., x_n) \in L_2^n$ is equivalent to minimizing the function F. Motivated by this, Hammer represented any cut separating the source and the sink of a network by using a binary vector and illustrated how the determination of the minimum cut through the network allows the derivation of the minimum cut's capacity, or, in other words, of the network's maximum flow. It was also shown that the converse procedure also holds.

The above results motivated Picard and Ratliff, who introduced the idea of treating energy minimization problems of a class of systems, including the RFIM, as minimum cut problems on a network (Picard and Ratliff, 1975), by mapping the system under consideration into a network and calculating its maximum flow. The value of the maximum flow is then equal to the ground-state energy of the system. The orientations of the spins in the ground state can be easily constructed from the flow values the network takes. Apart from the RFIM, examples of other systems whose ground-state calculation can be mapped to maximum-flow problems in networks are interfaces in random elastic media (Seppäla and Alava, 2000; Seppäla, Alava and Duxbury, 2001) and fracture surfaces in random fuse networks (Seppäla, Raisanen et al., 2000).

3.1.1 Graph theory

In this subsection, we briefly summarize a few elements of graph theory which are necessary for its implementation on ground state problems in the current thesis. The general fundamental concepts of graph and network theory are not outlined in the thesis, as they are considered to be commonly known.

Let us hence note that a graph is called dense when the total number of its edges is close to the maximal number of edges. The opposite, a graph with much less edges than the maximum possible, is called a sparse graph.

A random graph is characterized by two numbers, namely the number of nodes N and the coordination number z, which is the average number of edges incident

to each node, i.e. the graph's average degree (Gros, 2009). Then, for a sparse graph, the maximum number of edges (or links) N_l is given by:

$$N_l = \frac{N \cdot z}{2}, \tag{3.1}$$

while for a dense graph the above equation reads:

$$N_l = \frac{N(N-1)}{2}. \tag{3.2}$$

One way to define a graph's density is by calculating the total number of edges over the total number of nodes (Lawler, 1976), that is:

$$d = \frac{N_l}{N}. \tag{3.3}$$

Eqs. 3.1 and 3.2 lead to

$$d = \frac{z}{2} \tag{3.4}$$

and

$$d = \frac{N-1}{2}, \tag{3.5}$$

for the density of a sparse and of a dense graph, respectively.

Eq. 3.1 implies that for a sparse graph, the number of links for each node does not depend on the total amount of nodes, since the density is constant, while Eq. 3.2 implies that this does not hold for the case of a dense graph, since the density increases analogously to the number of nodes.

We shall look back to the above information when necessary, but for the moment let us carry on with the mapping of the RFIM to a network, which is described in the next subsection.

3.1.2 Mapping

Let $\mathcal{G} = (V, E)$ be a directed, weighted graph consisting of V nodes and a set E of edges, each connecting two nodes. Let the number of nodes be $n+2$. We enumerate the nodes as $V = \{0, 1, \ldots, n, n+1\}$ and we define $0 \longrightarrow$ source ($s \in V$) and $n+1 \longrightarrow$ sink ($t \in V$). The remaining nodes, namely, the inner vertices, will be associated to the lattice points of the RFIM. We can now define a network $\mathcal{N} = (G, c, s, t)$, where $c : V \times V \longrightarrow \mathbb{R}_0^+$ are the capacities of the edges $(i, j) \in E$, with $c_{ij} \equiv c(i, j) = 0$ for $(i, j) \notin E$.

We separate the network with an (s, t)-cut $(S, \overline{S})$, that is a cut separating the source s from the sink t. The cut has the properties of a node partition:

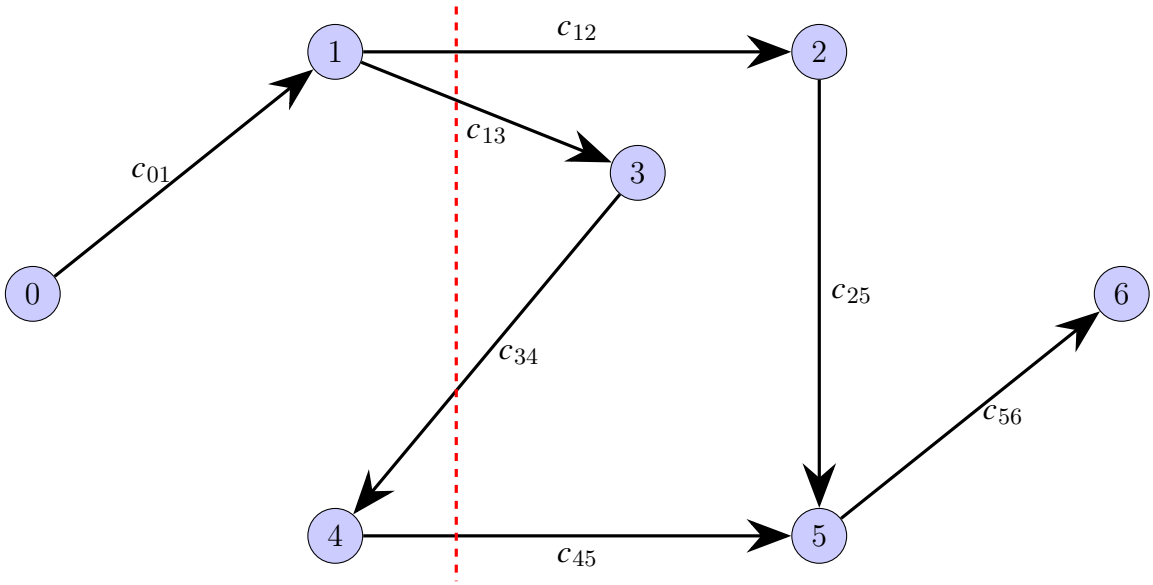

FIGURE 3.1: A cut (red dashed line) separating the source $s = 0$ from the sink $t = 6$. The cut crosses through the edges $(1, 2)$, $(1, 3)$, $(3, 4)$ and $(4, 5)$. However, the capacity of the cut equals to $C(S, \overline{S}) = c_{12} + c_{13} + c_{45}$.

- $S \cup \overline{S} = V$,
- $S \cap \overline{S} = \emptyset$,
- $s \in S$,
- $t \in \overline{S}$.

The capacity $C(S, \overline{S})$ of the cut is defined as the sum of the capacities of the contributing edges, i.e. the edges that cross the cut from S to $\overline{S}$:

$$C(S, \overline{S}) = \sum_{i \in S, j \in \overline{S}} c_{ij}. \tag{3.6}$$

For example, in Fig. 3.1, the edges contributing to the cut $(S, \overline{S})$ are the ones going from the left to the right of the cut, namely $(1, 2)$, $(1, 3)$ and $(4, 5)$.

The central idea to represent the Hamiltonian 1.43 by a network is to represent a (s, t)-cut by a binary vector $\underline{X} = \{x_0, x_1, \ldots, x_n, x_{n+1}\}$ (Hammer, 1965), where:

$$x_i = \begin{cases} 1, & i \in S \\ 0, & i \in \overline{S} \end{cases}. \tag{3.7}$$

Thus an edge (i, j) goes from left to right of the cut if and only if $x_i = 1$ and $x_j = 0$. By definition, since x_0 corresponds to the source and x_{n+1} corresponds to

the sink of the network, we have $x_0 = 1$ and $x_{n+1} = 0$. Therefore, we end up with the following formula for the capacity of any cut represented by Eq. 3.7:

$$C(\underline{X}) = \sum_{i=j=0}^{n+1} x_i(1-x_j)c_{ij} = -\sum_{i=j=0}^{n+1} c_{ij}x_ix_j + \sum_{i=0}^{n+1}\left(\sum_{j=0}^{n+1} c_{ij}\right)x_i. \tag{3.8}$$

Note the structural similarity to Eq. 1.43. This implies that the energy of every configuration of the system will be equal to the capacity of the corresponding cut. As discussed by Picard and Ratliff, 1975; Picard and Ratliff, 1973, finding a binary vector $\underline{X} = \{x_0, x_1, \ldots, x_n, x_{n+1}\}$, such that a quadratic function $F(\underline{X})$ is minimized, is equivalent to finding a minimum cut in F's corresponding network. In our case, we are looking for the configuration of the system that corresponds to the GS energy, that is the minimum cut $\underline{X}$ of the network, which minimizes $\mathcal{H}(\underline{X})$.

Let us now illustrate that every spin configuration of the system corresponds to a cut in the network. This is easy to verify, simply by noticing that for $s_i = \pm 1$, x_i may be expressed as $\frac{s_i+1}{2}$. By plugging this in Eq. 3.7, we obtain:

$$s_i = \begin{cases} 1, & i \in S \\ -1, & i \in \overline{S} \end{cases}. \tag{3.9}$$

In other words, for any cut in the network, when a node belongs to the source set, then its corresponding spin will be looking up and when a node belongs to the sink set, its corresponding spin will be looking down. Looking for the minimum among all cuts that minimizes $\mathcal{H}(\underline{X})$ is equivalent to looking for the spin configuration that minimizes $\mathcal{H}(s_i)$, i.e. the ground state of the system.

We are now only left with choosing the capacities of the network for completing the correspondence between Eqs. 1.43 and 3.8. The capacities of the edges connecting the inner nodes with each other and with the source and the sink, respectively, can be derived by comparing Eqs. 1.43 and 3.8. After some trivial algebra, we obtain for the inner edges:

$$c_{ij} \equiv \begin{cases} 0, & i \geq j \\ 4J_{ij}, & else \end{cases}, \tag{3.10}$$

where $i, j \in 1, ..., n$ and $c_{ii} = 0$, $\forall i \in 1, ..., n$.

Here we need to point out that by definition, network flow problems are described with the constraint that, excluding edges connecting the source with the sink, only non-negative capacities are allowed for the edges of a network. This

is because when we try to find the maximum flow through the network, we start from the source and edge by edge, we push the flow and add it to the flow value until we reach the sink, through a specific path. If negative values are included, then the problem becomes NP hard, and hence we cannot do simulations for reasonably big systems. Only for non-negative capacities there exist simulations that can be performed in polynomial time (Hartmann and Rieger, 2004). Hence, here we choose that the positive capacities will be the ones for edges (i, j) going from left to right, that is, for $i < j$.

As for the edges connected to the source or the sink, comparison of Eqs. 1.43 and 3.8 leads, respectively, to:

$$c_{0i} \equiv \begin{cases} 0, & w_i > 0 \\ -w_i, & \text{else} \end{cases} \tag{3.11}$$

and

$$c_{i,n+1} \equiv \begin{cases} w_i, & w_i > 0 \\ 0, & else \end{cases}, \tag{3.12}$$

where

$$w_i \equiv -2h_i - \frac{1}{2}\sum_j (c_{ij} - c_{ji}). \tag{3.13}$$

Eqs. 3.11 - 3.13 reassure that the source and sink capacities of the network are also non-negative.

Finally, regarding the capacity of the edge connecting the source and the sink of the network (offset capacity), we have:

$$c_{0,n+1} \equiv -\frac{1}{4}\sum_{i<j}(c_{ij} + c_{ji}) - \frac{1}{2}\sum_i (c_{0i} + c_{i,n+1}). \tag{3.14}$$

The above capacity is always negative and hence it is initially ignored when calculating the maximum flow through the network (as seen in Fig. 3.1). However, since it is crossed by any cut separating the source from the sink, it needs to be taken into account in our calculations in order to obtain the correct capacity of a cut in the network and hence the correct energy value of the equivalent system.

One may note that the values of the inner capacities are not dependent on the values of the local magnetic fields. In fact, the distribution and the disorder strength (i.e. the distribution's standard deviation) of the random field of a RFIM only affect the source, sink and offset capacities of its corresponding network.

Now that the correspondence between Eqs. 1.43 and 3.8 is complete, recall

that we are interested in ground states, i.e. in working out the minimum of $\mathcal{H}(s_i)$. Therefore, we are interested in the minimum cut of the network, that is the cut $\underline{X}$ which separates the source from the sink having the smallest capacity. This cut will then minimize $\mathcal{H}(\underline{X})$. As mentioned above, according to the max-flow min-cut theorem, looking for the set of edges that disconnects the network in such a way that the total capacity of these edges is the smallest possible, is equivalent to working out the maximum flow of the network.

For this purpose, we need to employ an appropriate maximum flow algorithm. In Sec. 3.2 we shall mention some of the most known algorithms of such type and we shall describe the algorithm of our choice, that is, the Boykov-Kolmogorov algorithm, for calculating the maximum flow in a network. But before that, let us examine the effectiveness of the mapping described in this subsection for two simple examples given by Hartmann and Rieger (2004).

3.1.3 A 2×2 RFIM without magnetic field

In this subsection, as a test case we consider a system of 4 spins, setting $J = 1$ and the local field magnitudes to $h_1 = 0$, $h_2 = -2h$, $h_3 = 2h$, and $h_4 = 0$. For this specific example, we choose $h = 0$. We construct an equivalent network of 6 nodes; 4 inner nodes (one for each spin), and additionally a source 0 and a sink 5.

We proceed to working out the maximum flow of this network by hand, according to the method described in Subsec. 3.1.2. The capacities of the inner edges are derived from Eq. 3.10:

$$\begin{aligned} c_{12} = c_{13} = c_{24} = c_{34} = 4, \\ c_{ij} = 0 \text{ for all rest inner capacities.} \end{aligned} \tag{3.15}$$

In order to figure out the capacities of the edges connecting the inner vertices to the source or the sink, we calculate the values w_i according to Eq. 3.13 and obtain:

$$\begin{aligned} w_1 &= -4, \\ w_2 &= 4h, \\ w_3 &= -4h, \\ w_4 &= 4. \end{aligned} \tag{3.16}$$

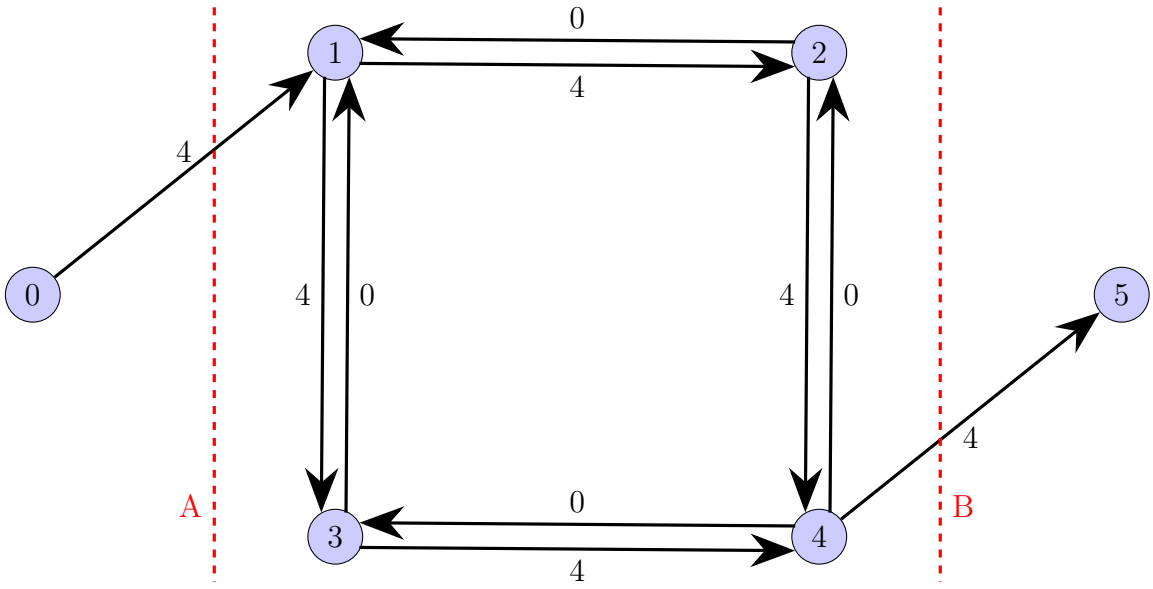

FIGURE 3.2: The network obtained for the local field magnitudes $h_1 = h_2 = h_3 = h_4 = 0$. Note that 2 distinct minimum cuts, i.e. 2 different ground-state energy configurations can be achieved.

Hence, for $h = 0$, the above equation yields:

$$\begin{aligned} w_1 &= -4, \\ w_2 &= 0, \\ w_3 &= 0, \\ w_4 &= 4 \end{aligned} \tag{3.17}$$

and plugging Eq. 3.17 into Eqs. 3.11 and 3.12 leads to the following values for the source and sink capacities:

$$\begin{aligned} c_{01} = 4, \quad c_{02} = c_{03} = c_{04} = 0, \\ c_{15} = c_{25} = c_{35} = 0, \quad c_{45} = 4. \end{aligned} \tag{3.18}$$

Now, what is left to do is calculate the capacity of the edge connecting the source with the sink of the above network. Plugging Eqs. 3.15 and 3.18 into Eq. 3.14 gives $c_{05} = -8$ for the offset capacity. The network corresponding to the small 4-spin random-field system for $h = 0$ is shown in Fig. 3.2.

The network allows two distinct minimum cuts, which both have capacity $C(S, \overline{S}) = 4 + c_{05} = -4$. Hence, the maximum flow, i.e. the GS energy of the above system is $f_{\max} = -4$. The existence of more that one minimum cuts implies the degeneracy of our system, i.e. the existence of more than one spin configurations

yielding the GS energy. In fact, we can precisely determine these configurations.

Cut A separates the source from all rest nodes which, according to Eq. 3.9, implies that $s_i = -1 \ \forall i \in \{1, 2, 3, 4\}$, i.e. cut A yields the GS configuration where all spins are pointing down. Cut B indicates that nodes $0 - 4$ belong to the source set S and node 5 is in the sink set $\overline{S}$. Hence, according to Eq. 3.9, cut B corresponds to the GS configuration where $s_i = 1 \ \forall i \in \{1, 2, 3, 4\}$, i.e. all spins are pointing up.

The above results regarding the system's ground-state energy and behaviour are expected, since a zero field acting on the spins implies the existence of ferromagnetic spin configurations at zero temperature. Indeed, both ground states corresponding to both minimum cuts are ferromagnetically ordered. Since we don't have a magnetic field, the total energy of the system is equal to its bond energy. Note that for this example we don't have any periodic boundary conditions for the lattice sites, that means, we have only 4 bonds between 4 nodes. Since $\operatorname{sign} s_i = \operatorname{sign} s_j \ \forall i, j \in \{1, 2, 3, 4\}$, each bond contributes an amount of $-Js_is_j = -1(1) = -1$. Hence, according to Eq. 1.43, the total GS energy is $E_{\text{total}} = E_{\text{bond}} = -4$, which is the maximum flow we derived for the above network.

3.1.4 A 2×2 RFIM with magnetic field

In this subsection, we examine the behaviour of the previous system for the case where $h = 1$. Again we work out the ground state of this example analytically, following the technique described in Subsec. 3.1.2. Recall that the inner capacities of a network corresponding to a RFIM remain invariant under any changes to the random field. Therefore, the capacities for the inner edges of the corresponding network are again given by Eq. 3.15.

As for the source and sink capacities, Eq. 3.16 for $h = 1$ reads:

$$\begin{aligned} w_1 &= -4, \\ w_2 &= 4, \\ w_3 &= -4, \\ w_4 &= 4, \end{aligned} \tag{3.19}$$

which leads to the following values:

$$\begin{aligned} c_{01} = 4, \quad c_{02} = 0, \quad c_{03} = 4, \quad c_{04} = 0, \\ c_{15} = 0, \quad c_{25} = 4, \quad c_{35} = 0, \quad c_{45} = 4. \end{aligned} \tag{3.20}$$

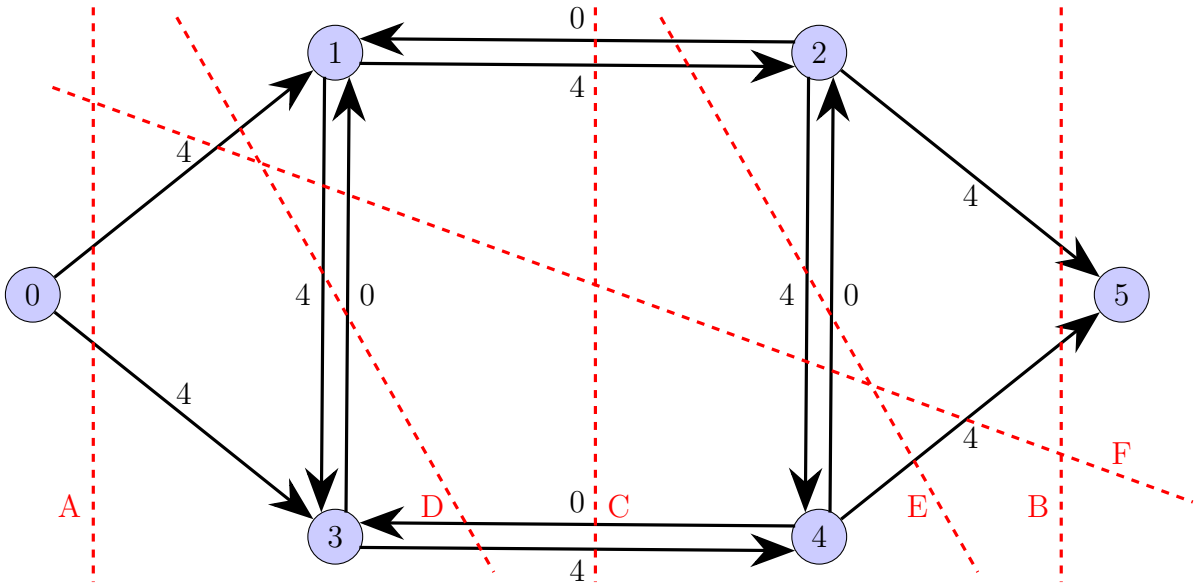

FIGURE 3.3: The network obtained for the local field magnitudes $h_1 = 0$, $h_2 = -2$, $h_3 = 2$, and $h_4 = 0$. Note that a total of 6 distinct minimum cuts, i.e. of 6 different ground-state energy configurations can be achieved.

Finally, plugging Eqs. 3.15 and 3.20 into Eq. 3.14 gives $c_{05} = -12$ for the offset capacity. The network corresponding to the 4-spin random-field system for the case $h = 1$ is shown in Fig. 3.3.

In this case six distinct minimum cuts are allowed, which all have capacity $C(S, \overline{S}) = 8 + c_{05} = -4$. Hence, the maximum flow, i.e. the GS energy of the above system is $f_{\max} = -4$ for all the six different spin configurations. Again, by advising Eq. 3.9, we can determine accurately each one of them.

Cuts A-B correspond to the ordered ground states of our system, namely, the states where all spins point up or all spins point down. Cut A implies that spin 2 has the same direction and spin 3 has the opposite direction from their local magnetic fields, while cut B implies the opposite for spins 2 and 3. Cuts C-F correspond to the disordered ground states of the system, where spins 2 and 3 are oriented in parallel to their local magnetic fields and spins 1 and 4 are randomly oriented in all four different possible combinations. The spin configurations corresponding to the six minimum cuts are presented in Fig. 3.4.

Let us finally validate that the above results regarding the system's ground-state energy and behaviour are expected. For each of the six different ground states corresponding to the six different minimum cuts, we calculate the total

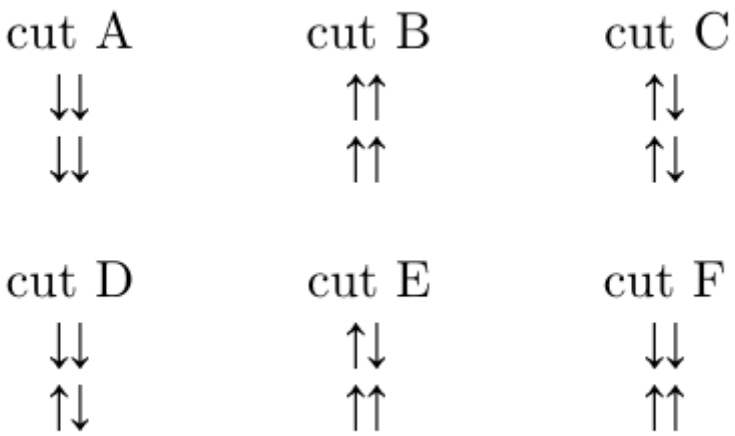

FIGURE 3.4: Spin configurations of the 6 distinct ground states of the RFIM for $h = 1$, corresponding to the 6 distinct minimum cuts of the system's equivalent network.

energy given by Eq. (1.43) and obtain the following results:

- **cut A:**

$$\begin{aligned}\mathcal{H} = -1\Big((-1)(-1) + (-1)(-1) + (-1)(-1) + (-1)(-1)\Big) \\ -\Big(0(-1) - 2(-1) + 2(-1) + 0(-1)\Big) = -4.\end{aligned} \tag{3.21}$$

- **cut B:**

$$\begin{aligned}\mathcal{H} = -1\Big((1)(1) + (1)(1) + (1)(1) + (1)(1)\Big) \\ -\Big(0(1) - 2(1) + 2(1) + 0(1)\Big) = -4.\end{aligned} \tag{3.22}$$

- **cut C:**

$$\begin{aligned}\mathcal{H} = -1\Big((1)(-1) + (-1)(-1) + (-1)(1) + (1)(1)\Big) \\ -\Big(0(1) - 2(-1) + 2(1) + 0(-1)\Big) = -4.\end{aligned} \tag{3.23}$$

- **cut D:**

$$\begin{aligned}\mathcal{H} = -1\Big((-1)(-1) + (-1)(-1) + (-1)(1) + (1)(-1)\Big) \\ -\Big(0(-1) - 2(-1) + 2(1) + 0(-1)\Big) = -4.\end{aligned} \tag{3.24}$$

- **cut E:**

$$\begin{aligned}\mathcal{H} = -1\Big((1)(-1) + (-1)(1) + (1)(1) + (1)(1)\Big) \\ -\Big(0(1) - 2(-1) + 2(1) + 0(1)\Big) = -4.\end{aligned} \tag{3.25}$$

- **cut F:**

$$\begin{aligned}\mathcal{H} = -1\Big((-1)(-1) + (-1)(1) + (1)(1) + (1)(-1)\Big) \\ -\Big(0(-1) - 2(-1) + 2(1) + 0(1)\Big) = -4.\end{aligned} \tag{3.26}$$

Hence, for each case the total GS energy is $E_{\text{total}} = -4$, which is the capacity we derived for all the minimum cuts of the above network.

We have therefore validated analytically the effectiveness of the mapping described in Subsec. 3.1.2 and we are now ready to proceed with its numerical implementation. In Sec. 3.2 we outline some of the most known algorithms for calculating the maximum flow through a network and then carry on with the description of the algorithm of our choice for the current thesis, namely, the Boykov-Kolmogorov algorithm.

3.2 Maximum flow algorithms: The Boykov - Kolmogorov algorithm

To summarize our discussion so far, recall that the RMIF has a fixed point at $T = 0$ (Villain, 1984; Bray and Moore, 1985b; Fisher and Huse, 1986) and therefore its critical behaviour is the same everywhere along the phase boundary. This implies that we can determine its behaviour by staying at zero temperature and crossing the phase boundary at the critical field point, allowing us to figure out exactly the ground states of the system.

The above is carried out through the mapping of the ground state to the maximum flow optimization problem described in Subsec. 3.1.2 (Anglés d'Auriac et al., 1985; Cormen et al., 1990; Papadimitriou, 1994) and subsequently with the use of maximum flow algorithms (Ogielski, 1986; Anglés d'Auriac, 1986; Sourlas, 1998; Hartmann and Usadel, 1995; Hartmann and Nowak, 1999; Anglés d' Auriac and Sourlas, 1997; Swift et al., 1997; Bastea and Duxbury, 1998; Hartmann and Young, 2001; Duxbury and Meinke, 2001; Middleton, 2001; Middleton and Fisher, 2002; Dukovski and Machta, 2003; Wu and Machta, 2005; Fytas and Martín-Mayor, 2013b; Alava et al., 2001; Seppäla and Alava, 2001; Shrivastav,

Krishnamoorthy et al., 2011; Ahrens and Hartmann, 2011; Stevenson and Weigel, 2011).

3.2.1 Minimum cut - maximum flow algorithms

The application of maximum flow algorithms is nowadays well established for the study of the RFIM (Alava et al., 2001). In this subsection we therefore look at the most popular minimum cut - maximum flow algorithms, which, as we shall see, are roughly divided into two main categories.

Let us start with the historically first algorithm for calculating the maximum flow through a network, which was developed by Ford and Fulkerson (1956). The Ford-Fulkerson algorithm's input is a graph, along with a source s and a sink t. The graph is any representation of a weighted, directed, residual graph where vertices are connected by edges of specified weights. Each vertex, except s and t, can receive and send an equal amount of flow through it, while s can only send and t can only receive flow.

The steps of this greedy algorithm can be summarized as follows:

1. Start with an empty network, that is, initialize the flow along every edge to zero.
2. Use a breadth-first search (BFS) to find an augmenting path from the source to the sink, that is, beginning at the source, iteratively visit neighbouring nodes that have not been visited before. An augmenting path is a path that can take some flow the entire way from s to t.

   While there is an augmenting path from the source to the sink, add the capacity of this path to the flow. Let c_P indicate the maximum flow that can go through this path. In order to find the capacity of this path, we need to look at all edges of the path and subtract their current flow, f_{ij} from their capacity, c_{ij}. Since the smallest value of $c_{ij} - f_{ij}$ is a bottleneck for the path, we set c_P equal to this value.
3. Augment the flow along the path by c_P.
4. Repeat the process from step 2 until no augmenting paths are found.
5. The maximum flow has been attained.

The time complexity of the Ford-Fulkerson algorithm has no polynomial bound and is equal to $O(m|C|)$, where n is the number of edges of the network and

$|C|$ is the cost of a minimum cut, which is the value of the maximum flow. A variant of the algorithm was developed by Edmonds and Karp (1972), which does have a polynomial time complexity equal to $O(nm^2)$, where m is the number of nodes. Other algorithms falling into the augmenting-path-style category are Dinic's algorithm (Dinic, 1970) with time complexity $O(n^2m)$, as well as the BK algorithm which we shall outline in more detail in Subsec. 3.2.2.

Now let us turn to the second category of minimum cut - maximum flow algorithms, namely, algorithms of push-relabel style. One of the most efficient network flow algorithms used to solve the RFIM is the general push-relabel algorithm developed by Goldberg and Tarjan (1988), with a computational time complexity of $O(n^2m)$. Proofs and theorems regarding the algorithm have been provided by Cormen et al. (1990) and Papadimitriou (1994). Two efficient implementations of the push-relabel method for the maximum flow problem, namely the H_PRF and Q_PRF techniques have been proposed by Cherkassky and Goldberg (1997), with computational time complexities of $O(n^2\sqrt{m})$ and $O(n^3)$, respectively.

Like Ford-Fulkerson, push-relabel algorithm's input is also a residual graph, along with a source s and a sink t. However, instead of examining the entire residual network to find an augmenting path, push-relabel algorithms work on one node at a time. One can consult the above references for further information regarding the steps of the general push-relabel algorithm. Here we choose to focus on a modification proposed by Middleton et al. (Middleton, 2001; Middleton and Fisher, 2002; Middleton, 2002), which removes the source and sink nodes, reducing memory usage and clarifying the physical connection. As we shall see later, this algorithm is one of the algorithms used for the discussion in Chap. 4.

One can refer to the appropriate bibliography (Middleton, 2001; Middleton and Fisher, 2002; Middleton, 2002) for the details of the algorithm, however a brief summary of the algorithm's steps is given below (Fytas and Martín-Mayor, 2016):

1. Assign an excess x_i to each lattice site i.
   Initially set residual capacities r_{ij} between neighbouring sites to J.

2. Assign to each node a height u_i via a global update step:

   - Each site in the set $\mathcal{T} = \{j \mid x_j < 0\}$ of negative excess sites takes a height value of $u_i = 0$.

- Sites with $x_i \geq 0$ have u_i set to the length of the shortest path, via edges with positive capacity, from i to $\mathcal{T}$.

3. If $x_i > 0$, $r_{ij} > 0$ and $u_j = u_i - 1$, a push operation moves excess from a site i to a lower height neighbour j.

4. When $x_i > 0$ and no further push is possible, the site is relabelled, with $u_i \longrightarrow 1 + \min_{\{j|r_{ij}>0\}} u_j$. This is a single push-relabel step and the number of such steps is the measure of the algorithmic time.

5. Repeat steps $2-4$ until no more pushes or relabels are possible.

6. A final global update determines the ground state:
   Sites which are path connected by bonds with $r_{ij} > 0$ to $\mathcal{T}$ have $s_i = -1$.
   Sites which are disconnected from $\mathcal{T}$ have $s_i = 1$.

The lattice sites i are considered with the use of a first-in-first-out (FIFO) queue (Middleton, 2002), where a push-relabel step for a site i takes place at the front of a list. All active sites (i itself and/or neighbouring sites), if any, after the push-relabel step, are added to the end of the list. Finally, note that the computational efficiency of the algorithm described above has been increased via the use of periodic global updates every L^3 relabels (Middleton and Fisher, 2002; Middleton, 2002), where L is the lattice size.

3.2.2 BK algorithm description

In this subsection, we draw our attention to the Boykov-Kolmogorov min-cut/max-flow algorithm (Boykov and Kolmogorov, 2004), which is being used as a black-box in the current thesis for calculating the maximum flow of appropriate networks in order to study the critical behaviour of the RFIM. The algorithm was developed in order to improve the empirical performance of standard augmenting path algorithms on graphs in vision.

The BK algorithm's input is again a weighted, directed, residual graph, together with a source s and a sink t, where the reverse weighted edges connecting the nodes can be used. The algorithm consists a variation of standard augmenting path algorithms. It involves the construction of two non-overlapping search trees, namely a source-tree and a sink-tree (see Fig. 3.5), which treat the terminals symmetrically and outperform an earlier version of the algorithm where a single tree rooted at the source was used (Boykov and Kolmogorov, 2001). What is more, instead of the usual process of building new augmenting paths at each new iteration, the above trees are reused throughout the algorithm.

FIGURE 3.5: End of the growth stage: A path (yellow line) from s to t is found (Boykov and Kolmogorov, 2004).

The nodes in both search trees are labelled by using the graph colouring technique, according to which tree they belong to. In addition, they are assigned a status flag, A or P, according to whether they are active or passive, respectively. In the beginning of the algorithm only the source and the sink are coloured (red and blue, respectively) and have active status. The algorithm loops over the three following stages:

- **Growth stage**: During this stage, starting from the source s and the sink t, if a neighbour of an active node is connected to it via an edge of positive capacity, it becomes active as well and it now belongs to the search tree of the current active node (S or T tree, respectively).
  If there are no more such neighbours, the current active vertex becomes passive and the next active node is examined.
  The search trees S and T grow until there are no more active nodes left or until they touch. In the latter case a path from the source to the sink is found (see Fig. 3.5).
- **Augmentation stage**: The path that was found in the growth stage is augmented: The residual capacities of the edges and reverse edges belonging to the growth-stage path are updated, by subtracting from them, or by adding to them, respectively, the bottleneck capacity of the path.
  Note that each node belonging to a search tree has a non-saturated connection to a terminal. Therefore, since during the augmentation stage at least one saturated edge is created, the built up search trees can be broken into forests.
- **Adoption stage**: The structures of search trees S and T are restored, i.e. the orphans created during the augmentation stage acquire new parents: If

a neighbouring node of a current orphan has a non-saturated connection to the same terminal as the one the orphan had, then it becomes its new parent and this forest is included again in the search tree.
If there are no such neighbours, the current orphan becomes a free vertex (black nodes in Fig. 3.5) and its children become orphans.
If there are no more orphans, the adoption stage terminates.

The algorithm terminates when the search trees cannot grow any further, i.e. the flow has attained its maximum value. The worst case complexity of the BK algorithm is $O(mn^2|C|)$. As we shall see in Subsec. 3.2.5, despite the fact that theoretically speaking this is worse than the time complexities of other standard minimum cut-maximum flow algorithms (see Subsec. 3.2.1), the BK algorithm is significantly faster when applied to real-size experiments (Boykov and Kolmogorov, 2004).

3.2.3 Implementation of the BK algorithm for simple examples

In order to derive numerical results for the ground state behaviour of the RFIM, we modify and implement the 3.01 Version software for computing the minimum cut/maximum flow in a graph, constructed and provided by Boykov and Kolmogorov (2004) according to their algorithm, which was described in Subec. 3.2.2.

We use a C/C++ environment for implementing the software, which consists of two main source code files, namely `maxflow.cpp` and `graph.cpp`. Both compilation units include the header files `instances.inc` and `graph.h`, which in turn include the header files `graph.h` and `block.h`, respectively. Following the example and instructions given in `README.TXT`, we create a third compilation file in order to write our main program, which includes the header file `graph.h`. We name this file `main.cpp` and each time we modify it, depending on the system under study. After writing our main program, we compile all `.cpp` files and hence, indirectly, all the files of the software, by using the following command:

```
g++ -w *.cpp
```

Appropriate modifications will be done to the above command, according to our needs for linking the necessary libraries.

Let us first validate that the above technique indeed works for the simple example found in the `README.TXT` file given by Boykov and Kolmogorov (2004).

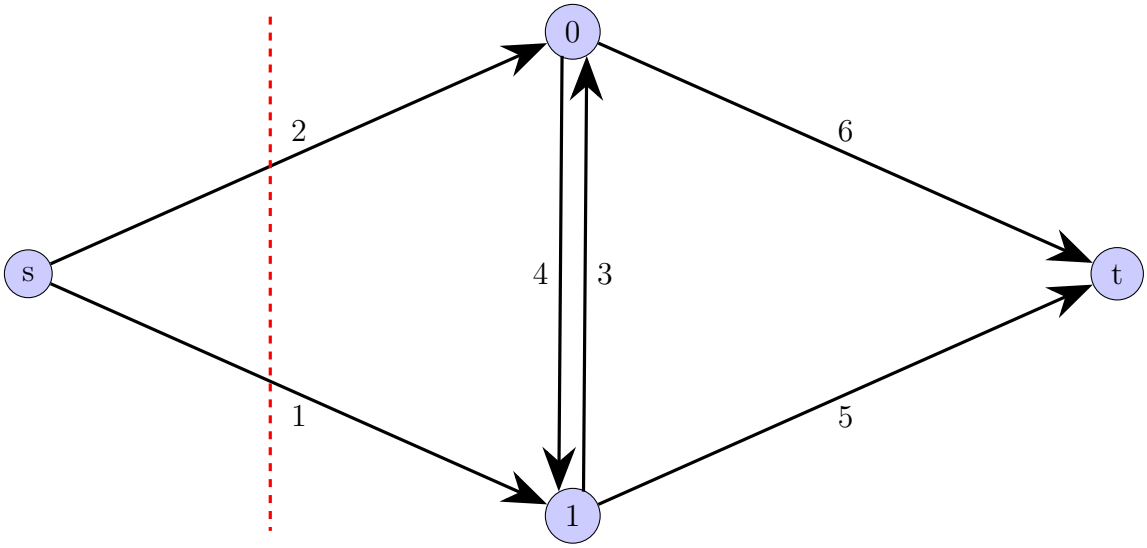

FIGURE 3.6: The example network presented by Boykov and Kolmogorov (2004), which allows a unique minimum cut, represented by the red dashed line. The cut crosses the edges $(s, 0)$ and $(s, 1)$, implying that both inner nodes belong to the sink set. Disregarding the offset edge, the cut's capacity is $C(S, \overline{S}) = c_{s0} + c_{s1} = 3$.

We create the appropriate `main.cpp` file, which includes the code given in `README.TXT`, for the graph given in Fig. 3.6. The code for this elementary example is included in Appx. Sec. A.1.

Note that the algorithm allocates a number to each inner node, starting from 0, without taking into account the source and the sink for the estimated number of total nodes. We therefore use a similar way for numbering the graph's nodes from now on, as seen in Fig. 3.6.

The main program in `main.cpp` starts by assigning to the graph constructor created in `graph.cpp` a new name, that is, `GraphType` and by declaring the types of the inner capacities, the source/sink capacities and the flow:

```
typedef Graph<int,int,int> GraphType;
```

Following, the graph constructor is called, after being pointed to the variable `g`. Its arguments give the maximum number of inner nodes and edges that can be added to the graph:

```
GraphType *g = new GraphType(2, 6);
```

Afterwards, the graph generation begins by adding nodes to the graph:

```
g -> add_node();
```

The above command adds by default one node to the graph and hence it is called as many times as the total number of nodes. Recall, as illustrated in Fig. 3.6, that the algorithm allocates an identity to every inner node, starting from 0 up to $N-1$, where N is the total number of nodes. This is performed by the above command, whose first call returns 0, second call returns 1, and final call returns $N-1$.

Now that the graph's nodes have been generated and have acquired a unique identity, the graph's edges are added, along with their capacities:

```
g -> add_tweights( 0, 2, 6 );
g -> add_tweights( 1, 1, 5 );
g -> add_edge( 0, 1, 4, 3 );
```

The first two lines refer to source and sink edges. Function `add_tweights` adds one source and one sink edge to every node and allocates their corresponding capacities. It has three arguments, namely the node's identity, the capacity of the edge connecting it to the source and the capacity of the edge connecting it to the sink. In the third line, function `add_edge` adds a bidirectional inner edge between two nodes i and j, with its corresponding capacity and reverse capacity. It has four arguments, that is the identity of node i, the identity of node j, the capacity of the inner edge from i to j and the capacity of the inner edge from j to i.

Once the graph's construction is complete, its maximum flow is calculated by calling the `maxflow()` function:

```
int flow = g -> maxflow();
```

Function `maxflow()` has two arguments. The first one is a boolean value, which is true or false according to whether any trees in the graph are reused or not, respectively, from a previous computation of the maximum flow. This value is set by the program's developers to be false by default. The second argument is a pointer variable including a list of any changed nodes in the graph. By default, this list is empty. Therefore, when no changes are performed to the graph, `maxflow()` can be called without arguments for calculating its maximum flow.

After the maximum flow is calculated, the algorithm also determines the node configuration of the minimum cut, by calling the `what_segment()` function, which has two arguments. The first argument refers to the node's identity, while the

second one refers to which graph segment does the node belong to. The second argument is by default set to return the source set and can be skipped when calling `what_segment()`. This implies that in the case where more than one minimum cuts exist, where nodes may occur to belong to the source set for a specific cut and to the sink set for another cut, then calling `what_segment()` for all nodes will let us obtain the minimum cut configuration with the biggest number of nodes belonging to the source set possible.

```
  printf("Minimum cut configuration:\n");
  if (g->what_segment(0) == GraphType::SOURCE)
printf("node0 is in the SOURCE set\n");
  else
printf("node0 is in the SINK set\n");

  if (g->what_segment(1) == GraphType::SOURCE)
printf("node1 is in the SOURCE set\n");
  else
printf("node1 is in the SINK set\n");
```

Finally, the built graph is deleted:

```
  delete g;
```

We compile all `.cpp` files and running `a.out` yields:

```
Flow = 3
Minimum cut configuration:
node0 is in the SINK set
node1 is in the SINK set
```

which are the results we were expecting for the graph given in Fig. 3.6. Recall that the algorithm does not take into account the offset edge capacity.

Let us now proceed with implementing the algorithm for various graph cases representing RFIMs. We start with the two examples presented in Subsecs. 3.1.3 and 3.1.4, where we already know what results to expect, and we solve them numerically. Then we shall proceed to more complicated examples.

- **BK implementation for example given in Subsec. 3.1.3**. The content of the relevant `main.cpp` file is given in Appx. Sec. A.2.

  We now change the estimated number of nodes and inner edges, according to our needs. For any node in the graph, when there exists only one edge connecting it to the source or to the sink of the network, we set the sink or

the source capacity argument of `add_tweights` function to 0, respectively, implying that the corresponding edge does not exist. Finally, we add the capacity of the offset edge to the flow value obtained by calling `maxflow()` and get the following output:

```
Flow = -4
Minimum cut configuration:
node 0 is in the SOURCE set
node 1 is in the SOURCE set
node 2 is in the SOURCE set
node 3 is in the SOURCE set
```

We indeed obtain the expected result for the GS energy, that is $E = -4$. As discussed earlier, among all the minimum cuts, the algorithm is by default set to return the one with the most nodes belonging to the source set. Therefore, between the two minimum cuts shown in Fig. 3.2, cut B is generated.

- **BK implementation for example given in Subsec. 3.1.4.** The content of the relevant `main.cpp` file is given in Appx. Sec. A.3.

  We obtain exactly the same output as in the above example, implying that we have calculated the correct GS energy, $E = -4$ and that among the six minimum cuts shown in Fig. 3.3, the algorithm generates cut F.

Implementation of the BK algorithm for both simple examples yielded the correct results for the GS energy of the small RFIMs. What is more, it allowed us to obtain the exact node configuration of one of the minimum cuts of each network. This motivates us to proceed to our next goal, which is to modify further this code in order to work out the maximum flow of a network corresponding to a $2D$ $L \times L$ RFIM with continuous random-field distribution and the spin configuration of its ground state, for any given lattice size L, either square or triangle.

3.2.4 Implementation of the BK algorithm for a $2D$ $L \times L$ RFIM with Gaussian random-field distribution

In this subsection, we shall attempt to generalize the procedure illustrated in Subsec. 3.2.3 for a $2D$ RFIM with Gaussian random-field distribution, for any $L \times L$ lattice size, where the lattice has a square or a triangle geometry. What

is more, we shall calculate several thermodynamic quantities of our interest and determine the spin configuration of the system's ground state.

Square lattice

We start with the square lattice and focus on a single realization of the quenched disorder. Recall that we have a lattice with $N = L \times L$ spins, which implies that we have 2^N possible spin configurations. The random field follows the Gaussian distribution, which is continuous. This means that for one specific realization of the quenched disorder, only 1 out of 2^N spin configurations yields the GS energy of the system.

For this specific spin configuration, we can calculate the magnetization per lattice site, m, as well as the second and fourth magnetic moments per lattice site, m^2 and m^4, respectively, the field energy per lattice site, $E_{\text{field\_N}}$, the bond energy per lattice site, $E_{\text{bond\_N}}$ and the total energy per lattice site, $E_{\text{total\_N}}$. Calculation of the Binder cumulant and of expectation values of any thermodynamic quantities makes no sense for one single ground state.

For one realization of the quenched disorder, we need N random numbers which are picked randomly from the random-field's distribution. Each one represents the value of the local magnetic field, h_i, acting on one site i of the lattice. Recall that the standard deviation of the distribution is the disorder strength h.

We now attempt our first generalization of the BK algorithm for working out the ground state of a single disorder realization of a $2D$ $L \times L$ square RFIM, where h_i, $i = 0, ..., N-1$ occur from a standard normal distribution, that is, a Gaussian with 0 mean and unitary standard deviation. The latter implies that for this specific example, the disorder strength will be $h = 1$.

Let us start with the local magnetic fields. We need to generate N Gaussian random numbers and allocate one to each lattice site. For this purpose, one needs to look at the GSL libraries in order to pick an appropriate random number generator (RNG) for the local random-field values. We choose the Mersenne Twister generator (Matsumoto and Nishimura, 1998). The libraries need to be included in the code, by including the header file `gsl/gsl_randist.h` and also called properly when compiling the program, by modifying the compiling command accordingly:

```
g++ -w *.cpp -lgsl -lgslcblas -lm
```

The `-lm` flag is used to also link the math library and for this code, which is presented in Appx. Sec. A.4, it is necessary for using the `pow()` function.

We use the function `gsl_ran_gaussian` in order to call the random number generator. The function returns a Gaussian random variate, with fixed mean equal to 0. We can choose the standard deviation, i.e. the disorder strength h as desired, by modifying accordingly the function's second argument when calling it. For our example, h will be equal to 1. As for its first argument, it is the `const` pointer to which the random numbers are allocated by the random number generator. Once the random numbers are generated, they are dynamically allocated to `h`, which is the array of the random fields h_i. We prefer dynamic allocation to static, since we want to avoid segmentation faults occurring at big lattice sizes.

We now proceed to the graph generation. We start by changing the types of the inner capacities, the source/sink capacities and the flow from `int` to `double`, since we now want to allow these variables to take any real value:

```
typedef Graph<double,double,double> GraphType;
```

Next, one needs to take into account the fact that our graph is not dense. A dense graph would imply that each spin interacts with every other spin in the lattice. This would mean, for the square $2D$ case, $N-1$ instead of 4 neighbours. Therefore, the graph we are generating here is a sparse graph with coordination number $z = 4$ (see Subsec. 3.1.1). The sparse graph's total number of inner edges can be easily calculated by Eq. 3.1, where N is the total number of inner nodes, i.e. the total number of spins. As for the maximum number of edges connecting each node with the source and the sink, we have $N + N = 2N$. Finally, one should also include the offset edge. We therefore change the estimated number of nodes and edges accordingly:

```
GraphType *g = new GraphType(N+2, 2*N+N*z/2+1);
```

The above fact implies that the usual methods for representing a graph, that is, with an adjacency matrix or an adjacency list, would not be the wisest choice for our case. This is because the corresponding adjacency matrix would be half-empty (or, more precisely, with half of its elements equal to 0) and hence our code would be computationally costful for no reason. As a result, we shall represent our graph in a different way. Thus, we choose to represent every inner node with a number, namely, from 0 to $N-1$, as mentioned in Subsec. 3.2.3. As one may also notice in Fig. 3.8, every node has four neighbours, corresponding to four numbers. In our code, we name the neighbours `up`, `down`, `left` and `right` and define them according to Fig. 3.7 and to the code given below:

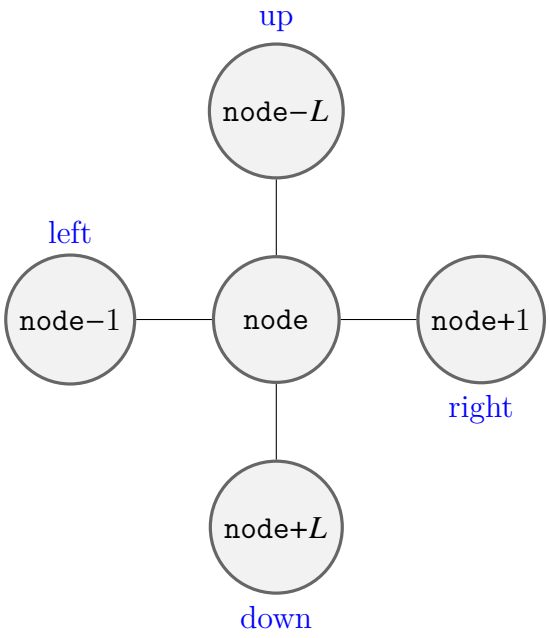

FIGURE 3.7: Square lattice: Expressions for a single node's four nearest neighbours ($z = 4$).

```
for(int node=0; node<N; node++) {
    int right = node + 1;
    int left = node - 1;
    int up = node + L;
    int down = node - L;
}
```

One needs to take into consideration the fact that in the $2D$ case, our lattice wraps around creating a donut shape and hence the appropriate boundary conditions should be included when defining the neighbours. By looking at Fig. 3.8, we can easily see that the rules followed in Fig. 3.7 do not apply for the borders of the lattice. For example, the right neighbour of 2 should be 0 and not 3, the right neighbour of 5 should be 3 and not 6 etc. This implies that any `right` neighbour generated according to Fig. 3.7, which is a multiple of L, should be reduced by L in order to obtain the correct number identity. Following the same mindset for the rest of the lattice borders, we redefine the neighbours for the four border cases with four `if` conditions:

```
for(int node=0; node<N; node++) {
    if (right % L == 0){right -= L;}
    if (node % L == 0){left +=L;}
```

```
        if (up > (N-1)){up -=N;}
        if (down < 0){down +=N;}
    }
```

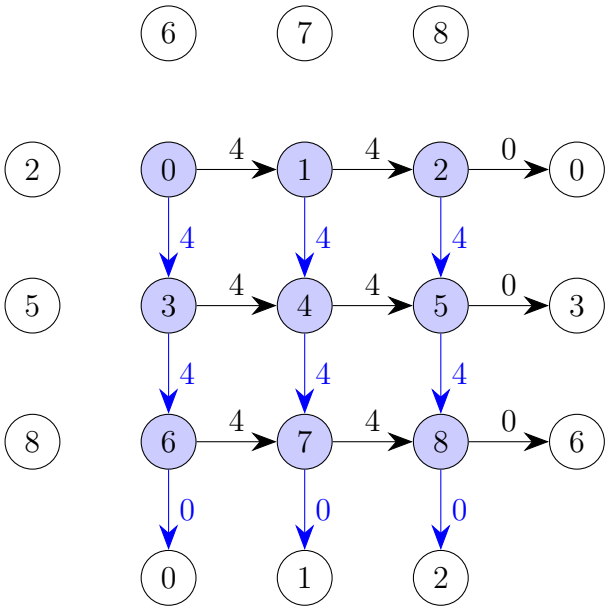

FIGURE 3.8: Illustration of the node representation (blue nodes) and inner edge and capacity building of a graph corresponding to a square 3×3 RFIM with periodic boundary conditions for $J = 1$. White nodes denote the neighbours of the vertices at the borders of the lattice. Black and blue edges correspond to `add_edge` for the right and down directions, respectively.

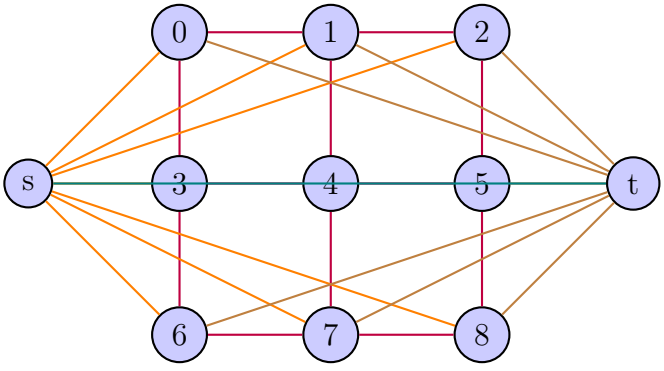

FIGURE 3.9: Example of all possible edges of a square RFIM for $L = 3$. We will add the inner, source, sink and offset edges and their corresponding capacities.

out $(i \to j)$	in $(j \to i)$
$C_r = 4J$	$C_r = 0$
$C_d = 4J$	$C_d = 0$
$C_l = 0$	$C_l = 4J$
$C_{up} = 0$	$C_{up} = 4J$

TABLE 3.1: Capacities of the edges moving from node i to its neighbours j (out) and from the neighbours to the node (in). For our case, $J = 1$.

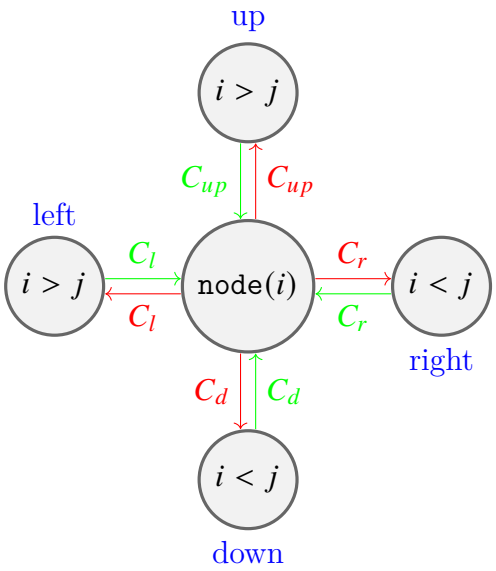

FIGURE 3.10: A single node's out and in inner capacities.

We are now left with the task of building the edges and plugging in the capacities, in order to complete our graph generation. Let us look at Fig. 3.9 and start with the inner edges, whose capacities are given by Eq. (3.10), where $J_{ij} = 1$. Hence, in our code we can define for each neighbour j of a single node i (see Fig. 3.10) the capacities of the edges going from the node to its neighbours and the capacities of the edges going from the neighbours to the node, according to Table 3.1.

Thus, for the inner capacities, we have:

```
Crightout=4*J;
Crightin=0;
Cleftout=0;
Cleftin=4*J;
Cdownout=4*J;
Cdownin=0;
Cupout=0;
```

```
Cupin=4*J;
```

As for the lattice borders, the capacities read:

```
if (right % L == 0 ){
    Crightout=0;
    Crightin=4*J;
}
if (node % L == 0){
    Cleftout=4*J;
    Cleftin=0;
}
if (up < 0){
Cupout=4*J;
Cupin=0;
}
if (down > (N-1)){
Cdownout=0;
Cdownin=4*J;
}
```

Finally, when building the edges and allocating their capacities, we choose the positive capacities to be the ones linking each site to its right and to its down neighbour:

```
g -> add_edge(node, right, Crightout, Crightin);
g -> add_edge(node, down, Cdownout, Cdownin);
```

Now as far as it concerns the network's terminals, we need to work out the weight w_i in order to define the capacities of the edges connecting them to the inner nodes. According to Eq. (3.13):

$$\begin{aligned} w_i &\equiv -2h_i - \frac{1}{2}\sum_j (c_{ij} - c_{ji}) \\ &\equiv -2 * h[\text{node}] - \frac{1}{2}\left(C_r + C_d + C_l + C_{up} - C_r - C_d - C_l - C_{up}\right). \end{aligned} \tag{3.27}$$

Hence, we have:

```
w=-2*h[node] - (Crightout+Cleftout+Cdownout+Cupout-Crightin
-Cleftin-Cdownin-Cupin)/2;
```

Now the source and sink capacities sC and tC are calculated according to Eqs. 3.11 and 3.12 and then they are added to the graph:

```
if(w > 0) {
    sC=0;
    tC=w;
}
else {
    sC=-w;
    tC=0;
}

g -> add_tweights(node, sC, tC );
```

The final step for plugging in the capacities is to define the offset edge with capacity oC, that is the one defined by Eq. (3.14). Note that after some manipulations and according to the code's notation (see Appx. Sec. A.4), Eq. (3.14) may read:

$$c_{0,n+1} \equiv -\frac{1}{4}\left(4 \cdot N_l\right) - \frac{1}{2}k = -N_l - \frac{1}{2}k, \tag{3.28}$$

where

$$k = \sum_{i=0}^{i<N} \Big(sC[\text{node}] + tC[\text{node}] \Big) \tag{3.29}$$

and N_l is the total number of the lattice's edges (see Subsec. 3.1.1). Therefore, the code writes:

```
for(int node=0; node<N; node++) {
k += sC+tC;
}
double oC=-N*z/2-k/2;
```

Once all capacities are defined and the network is constructed, its minimum cut is figured out by calling the `maxflow()` function, which separates the network into the source set S and the sink set $\overline{S}$ and determines the spin configuration of the ground state according to Eq. (3.9):

$$\text{spin}[\text{node}] = \begin{cases} 1, & \text{node} \in S \\ -1, & \text{node} \in \overline{S} \end{cases}. \tag{3.30}$$

As for the correct value of the maximum flow, which is equal to the GS energy of the system, recall that the offset edge capacity oC should be taken into account:

```
flow += oC;
```

By implementing the BK algorithm with the code given in Appx. Sec. A.4, we have calculated the unique GS energy of the $2D$ $L \times L$ square RFIM with Gaussian random-field distribution, for various choices of L. We have figured out precisely its corresponding spin configuration, for one specific realization of the quenched disorder, where $J = 1$ and the disorder strength is $h = 1$. What is more, we have determined various thermodynamic quantities of our interest. Our results are presented in Table 3.2 and in Fig. 3.11. We have used python in order to plot the spin configurations of the ground states and the relevant code is included in Appx. Sec. B.1.

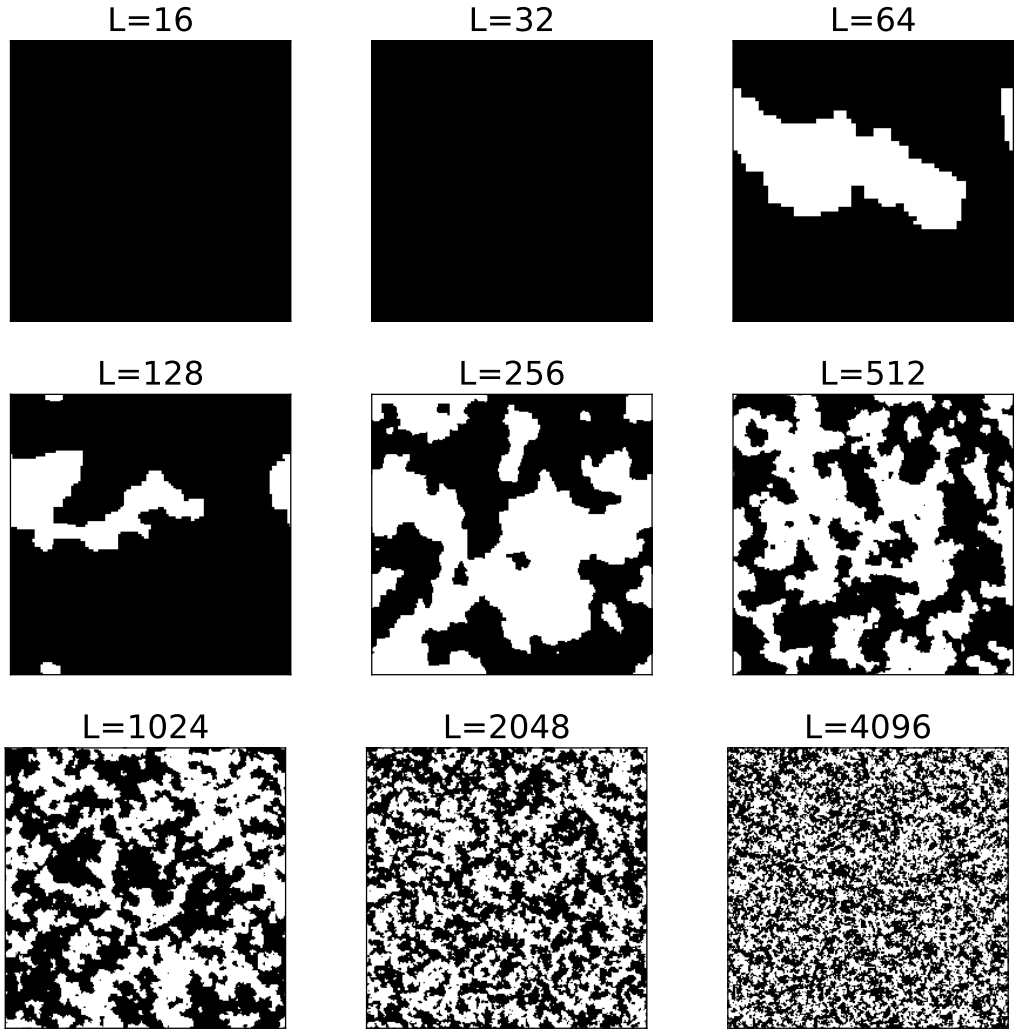

FIGURE 3.11: Spin configurations of $L \times L$ grids, for various square lattice sizes, for $J = 1$ and disorder strength $h = 1$. The seed for the random number generator is the same $\forall L$. White and black points represent $s_i = 1$ and $s_i = -1$, respectively. We can observe disordered ground states for lattice sizes $L \geq 64$.

L	$E_{\text{total\_N}}$	$E_{\text{field\_N}}$	$E_{\text{bond\_N}}$	m	Computational time
4	-2.13421	-0.13421	-2	1	0.00s
8	-2.07318	-0.0731845	-2	1	0.00s
16	-2.06862	-0.0686185	-2	-1	0.00s
32	-2.02403	-0.0240342	-2	-1	0.00s
64	-2.01583	-0.115439	-1.90039	-0.558594	0.01s
128	-2.01471	-0.0708667	-1.94385	-0.735474	0.05s
256	-2.0182	-0.107866	-1.91034	-0.0134583	0.13s
512	-2.01873	-0.111794	-1.90691	-0.0520706	0.49s
1024	-2.01862	-0.103278	-1.91534	-0.0487671	1.95s
2048	-2.01832	-0.104013	-1.91431	-0.0342841	8.15s
4096	-2.01825	-0.104776	-1.91348	-0.0058893	35.76s

TABLE 3.2: Table for many different square lattice sizes L, for $h = 1$. The first signs of disorder appear at $L = 64$, as seen by the bond energy not being -2 and the absolute value of the magnetization not being 1. Same seed is used for all cases of L.

Triangular lattice

Let us now turn our attention to the triangular $2D$ RFIM, where the random field follows again the Gaussian distribution. Our aim is the same as the one regarding the square RFIM, that is, work out the GS energy and spin configuration for a given realization of the disorder and estimate the expectation values of thermodynamic quantities of our interest. Here, one more time, we have a lattice of $N = L \times L$ spins and hence 2^N possible spin configurations.

The procedure we follow for implementing the BK algorithm in order to achieve the above goal is similar to the one we applied for the square lattice. Therefore, we are only going to focus on the differences between the two geometries. The code for a single realization of the disorder for the triangular $2D$ $L \times L$ RFIM is given in Appx. Sec. A.5.

The differences occurring in the implementation of the BK code between the two lattice geometries are related to the number of neighbours a single node has. Thus, let us start with the coordination number z, which is $z = 6$ for the triangular case. This does not affect our code, since the only change that needs to be made is the allocation of this value to the variable `z`. Considering that now every node has six neighbours, as one can see in Fig. 3.13, the two extra neighbours `upright` and `downleft` of a single node are defined in the code according to Fig. 3.12:

```
for(int node=0; node<N; node++) {
```

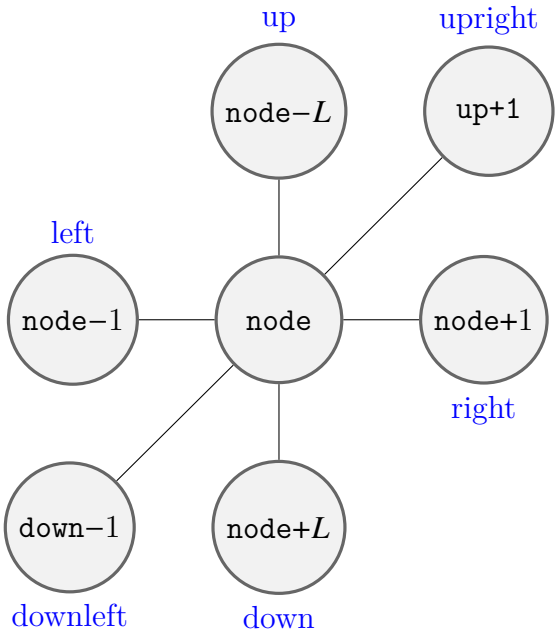

FIGURE 3.12: Triangular lattice: Expressions for a single node's six nearest neighbours ($z = 6$).

```
    int upright = up + 1;
    int downleft = down - 1;
}
```

As for the weighted edges (see Fig. 3.14), we start again with the inner ones, with capacities given again by Eq. (3.10), for $J_{ij} = 1$. This time, the capacities for the two extra neighbours should be taken into consideration (see Fig. 3.15 and Table 3.3). Therefore, we have the following extra definitions in the code:

```
Cuprightout=0;
Cuprightin=4*J;
Cdownleftout=4*J;
Cdownleftin=0;
```

What is more, the following lines must be modified in order to take into account the changes that occur at the lattice borders:

```
if (right % L == 0 ){
    right -= L;
    upright=right-L;
    if (up < 0){
        upright+=N;
    }
```

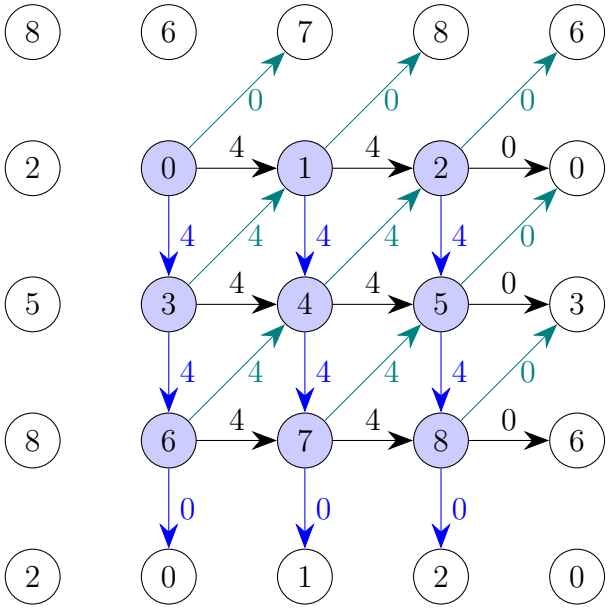

FIGURE 3.13: Node representation and inner edge and capacity building of a graph corresponding to a triangular 3×3 RFIM with periodic boundary conditions for $J = 1$. Black, blue and dark green edges correspond to `add_edge` for the right, down and upright directions, respectively.

```
    Crightout=0;
    Crightin=4*J;
}
if (node % L == 0){ //left border
    left +=L;
    downleft=left+L;
    if (down > (N-1)){  //special case for down left corner
        downleft-=N;
    }
    Cleftout=4*J;
    Cleftin=0;
}
if (up < 0){
    up +=N;
    upright=up+1;
    if (right % L == 0 ){  //special case for top right corner
        upright-=L;
    }
    Cupout=4*J;
```

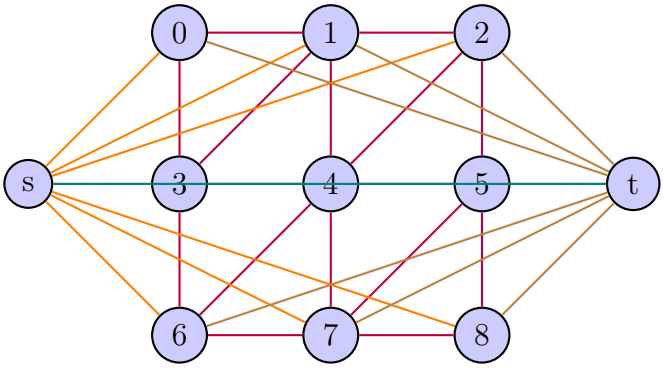

FIGURE 3.14: Example of all possible edges of a triangular RFIM for $L = 3$. Like the square case, the inner, source, sink and offset edges and their corresponding capacities will be added with the use of the graph constructor.

```
    Cupin=0;
    Cuprightout=4*J;
    Cuprightin=0;
}
if (down > (N-1)){ //down border
    down -=N;
    downleft = down -1;
    if (node % L == 0){  //special case for down left corner
        downleft+=L;
    }
    Cdownout=0;
    Cdownin=4*J;
    Cdownleftout=0;
    Cdownleftin=4*J;
}
```

We now choose between the two extra neighbours the `upright` to be the ones which are linked to each site with edges of positive capacities and we add the additional weighted edges to the graph:

```
g -> add_edge(node, upright, Cuprightout, Cuprightin);
```

After building the inner edges, we have to modify the weight w_i so that we can define the terminal capacities. Hence, Eq. (3.27) now becomes:

$$w_i \equiv -2 * h[\text{node}] - \frac{1}{2}(C_r + C_d + C_{dl} + C_l + C_{up} + C_{upr} - C_r - C_d - C_{dl} - C_l - C_{up} - C_{upr}) \tag{3.31}$$

out $(i \to j)$	in $(j \to i)$
$C_r = 4J$	$C_r = 0$
$C_d = 4J$	$C_d = 0$
$C_{dl} = 4J$	$C_{dl} = 0$
$C_l = 0$	$C_l = 4J$
$C_{up} = 0$	$C_{up} = 4J$
$C_{upr} = 0$	$C_{upr} = 4J$

TABLE 3.3: In and out capacities for the triangular case, for $J = 1$.

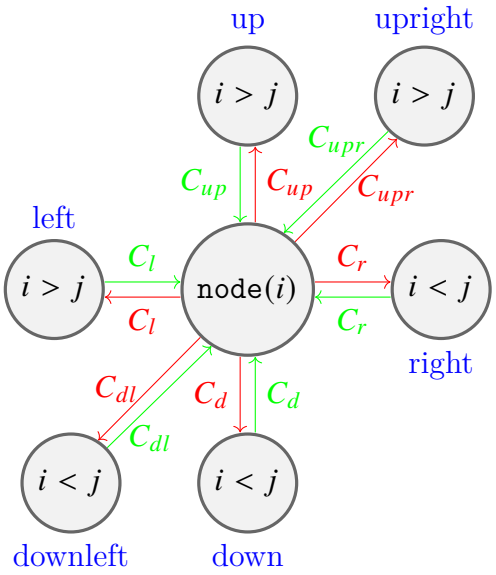

FIGURE 3.15: A single node's out and in inner capacities for the triangular lattice.

L	$E_{\text{total\_N}}$	$E_{\text{field\_N}}$	$E_{\text{bond\_N}}$	m	Computational time
4	-3.13421	-0.13421	-3	1	0.00s
8	-3.07318	-0.0731845	-3	1	0.00s
16	-3.06862	-0.0686185	-3	-1	0.00s
32	-3.02403	-0.0240342	-3	-1	0.00s
64	-3.01416	-0.0141627	-3	-1	0.01s
128	-3.01113	-0.0111254	-3	-1	0.04s
256	-3.00139	-0.00138895	-3	-1	0.22s
512	-3.00088	-0.000877803	-3	1	1.34s
1024	-3.00062	-0.00061612	-3	-1	10.96s
2048	-3.0004	-0.00352145	-2.99688	-0.802304	119.12s
4096	-3.00051	-0.00613277	-2.99438	0.0484595	565.79s

TABLE 3.4: Table for many different triangular lattice sizes L, for $h = 1$. Here we can note that disorder appears for first time at $L = 2048$. Same seed is used for all cases of L.

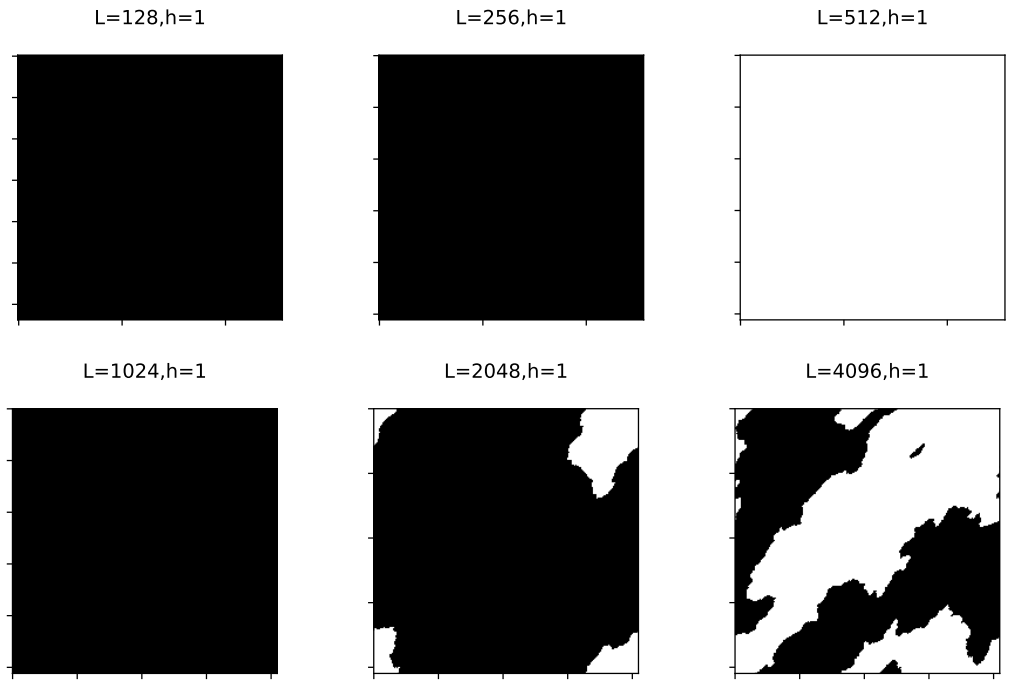

FIGURE 3.16: Spin configurations of $L \times L$ grids, for different triangular lattice sizes, for $J = 1$ and disorder strength $h = 1$. The seed for the RNG is the same $\forall L$. Disorder is detected for lattice sizes $L \geq 2048$.

and therefore the source, sink and offset capacities, sC, tC and oC are added to the graph in a similar way to the square lattice case. The remaining steps of the code for figuring out the maximum flow of the network and hence the spin configuration of the ground state and the thermodynamic quantities, remain the same, apart from the calculation of the bond energy, E_{bond}, where the two extra neighbours of a single node should be taken into account.

We are now ready to proceed to the implementation of the BK algorithm, this time for the triangular lattice geometry (see Appx. Sec. A.5). The results of the GS spin configurations for $J = 1$ and $h = 1$ are presented in Fig. 3.16. The python code for plotting the triangular lattice configurations is given in Appx. Sec. B.2. Note that for the same disorder strength, we don't observe disordered states for the same lattice sizes as with the square case. In fact, disorder starts at larger lattice sizes, for $L \geq 2048$. This is an expected behaviour, since the two extra bonds between the nearest neighbours in the case of the triangular lattice contribute to the resistance of breaking the GS order. The rest of our results regarding the GS energy and other thermodynamic quantities are included in Table 3.4.

Algorithm	Method style	Time complexity
General push-relabel (Goldberg and Tarjan, 1988)	push-relabel	$O(n^2m)$
H_PRF (Cherkassky and Goldberg, 1997)	push-relabel	$O(n^2\sqrt{m})$
Q_PRF (Cherkassky and Goldberg, 1997)	push-relabel	$O(n^3)$
Ford-Fulkerson (Ford and Fulkerson, 1956)	augmenting path	$O(m\|C\|)$
Edmonds-Karp (Edmonds and Karp, 1972)	augmenting path	$O(nm^2)$
Dinic (Dinic, 1970)	augmenting path	$O(n^2m)$
BK (Boykov and Kolmogorov, 2004)	augmenting path	$O(n^2m\|C\|)$

TABLE 3.5: Theoretical worst case running time complexities of minimum cut-maximum flow algorithms. n = #nodes, m = #edges, $|C|$ = cost of a minimum cut.

3.2.5 Computational time complexity of BK

Following the successful implementation of our code in order to figure out several GS properties of the $2D$ RFIM, we now carry on with investigating the computational time complexity of our code. Let us start by looking at Table 3.5, which includes a summary of the theoretical time complexities of the minimum cut-maximum flow algorithms discussed in Subsec. 3.2.1. It has been shown that H_PRF and Q_PRF outrun many other minimum cut-maximum flow algorithms in real-size experiments (Cherkassky and Goldberg, 1997). However, experimental tests in $2D$ for up to $L = 511$ indicate that BK is faster than these two algorithms (Boykov and Kolmogorov, 2004).

For the purpose of validating the above argument, we shall generate multiple realizations of the quenched disorder for some specific disorder strength h and work out the computational time of the BK code for each one of them. Afterwards, the calculation of the mean value of those computational times will give us an estimate of the average computational time needed for working out the GS spin configuration and the corresponding thermodynamic quantities for a specific disorder strength. Consequently, we shall compare our results to the ones we obtain by implementing the Q_PRF-style algorithm described in Subsec. 3.2.1 (Fytas and Martín-Mayor, 2016) for determining exactly the same critical properties of the square RFIM, for the same realizations and for the same disorder strength.

The implementation of the BK code for creating multiple realizations of the quenched disorder can be carried out in two different ways, that is, by either building the graph from scratch for every single disorder sample or building the

graph once and reallocating the graph's edges' capacities for every disorder realization. We expect that the latter constitutes an optimized version of the code, since the graph needs to be constructed outside of the loop for the many samples of the disorder. The two methods can be summarized as follows:

- **Rebuilding the graph.** Steps ∀ realization:
  1. Generate the graph.
  2. Allocate inner, source and sink capacities.
  3. Define the offset capacity and hence calculate the correct maximum flow.
  4. Determine the spin configuration of the GS and therefore work out E_{field}, E_{bond}, E_{total}, m.
  5. Delete graph.
  6. Go back to step one for the next realization.
- **Building the graph once.** First, the graph is generated and all capacities (inner and terminal) are initially set to zero. Then, the maximum flow function, `maxflow()` is called. Afterwards, we have the steps ∀ realization:
  1. Reallocate inner, source and sink capacities.
     Here, one may recall that the graph's inner edge capacities remain invariant under any changes of the random fields h_i. However, for every different realization of the disorder, `maxflow()` needs to be called in order to determine the spin configuration of the ground state. Calling it causes the graph's capacities to change and this is why all graph's edges' capacities need to be reallocated.
  2. Call `maxflow()` and hence determine the spin configuration of the GS, E_{field}, E_{bond}, E_{total} and m.
  3. Go back to step one for the next realization.
     Note that the offset capacity is not defined here, therefore calling the `maxflow()` function does not yield the correct result for the maximum flow, which is nevertheless equal to E_{total}. However, the function still needs to be called for determining the GS spin configuration.

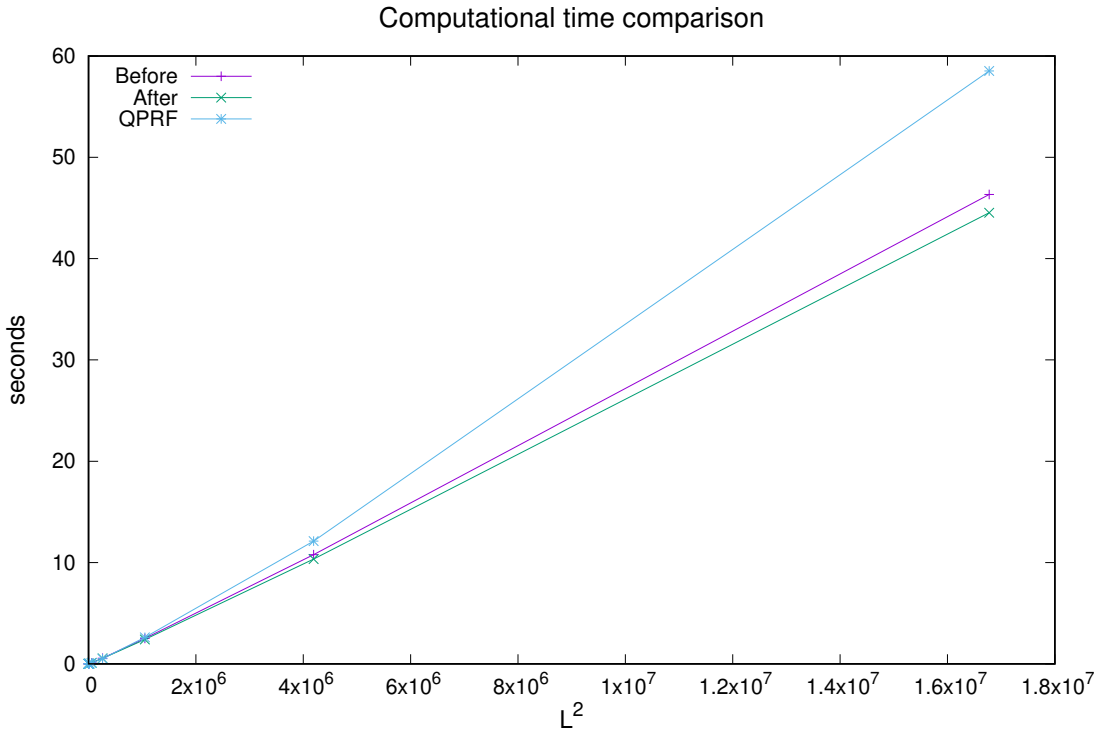

FIGURE 3.17: Plot of the computational times of the Q_PRF-style algorithm described by Fytas and Martín-Mayor (2016) and of the BK algorithm before and after optimization. Each point represents the computational time, divided by 100, of 1 run of the code for one specific lattice size, $h = 1$ and for 100 samples, for calculating $\frac{E_j}{N}$, $\frac{E_h}{N}$, $\frac{E}{N}$, $\frac{|m|}{N}$, U_4 and the GS's spin configuration for each sample. Each job was run solely on a cluster node with 8 CPUs. The 7.1.0 `gcc` version was used.

After we are done with all the steps for either of the above cases and for several lattice sizes of up to $L = 4096$, we sum all our results from the different realizations of the disorder strength for each thermodynamic quantity and divide them by the total amount of different samples in order to obtain the disorder averages. We also calculate the fourth order cumulant of the order parameter, given by Eq. (1.32).

The codes for working out the GS properties of the square RFIM for many realizations of the quenched disorder for a given disorder strength h are presented in Appx. Secs. A.6 and A.7, for the cases of rebuilding the graph and building the graph just one time, respectively. The code for building the graph only once was developed with the help of instructions and functions (`set_rcap()`, `set_trcap()`)

found in the `graph.h` file of the software provided by the developers (Boykov and Kolmogorov, 2004).

Our results regarding the computational time of the BK and Q_PRF-style algorithms are presented in Fig. 3.17. As we have suspected, the method of building the graph just once outruns the method of building the graph for every realization of the disorder and therefore it shall be preferred for our simulations from now on, as an optimized version of our code. It is clear however, that both versions of the BK algorithm are significantly faster than the Q_PRF-style algorithm, a fact that verifies the claim of Boykov and Kolmogorov (2004), namely, that BK outruns many of the most efficient minimum cut-maximum flow algorithms for lattice sizes of experimental interest.

Finally, one may notice that the difference between the two versions of our code implementation is relatively small, regarding the computational time. This is because the biggest portion of the computational time is used when calling the `maxflow()` function, i.e. for the actual calculation of the maximum flow rather than for setting up the graph structure. In fact, this can be confirmed by employing the `clock()` function for a test run of our code. For example, running the code for the $2D$ square RFIM for $h = 1$, $L = 4096$ for a single realization of the disorder takes roughly 33.7 seconds, with around 30 of them being spent for working out the maximum flow of the network and only 3.7 needed for building the graph and setting up the rest of the problem.

Chapter 4

Breakup Length in 2D

4.1 Background and methodology

The final chapter of the current thesis is devoted to the breakup length scale, $\ell_{\mathrm{B}}(h)$ (Binder, 1983; Bray and Moore, 1985a; Seppälä, Petäjä et al., 1998; Kumar et al., 2018) of the 2-dimensional RFIM; we study above what length scale does the system break its ferromagnetic ordering, at $T = 0$.

Recall that a crossing of the phase boundary of the RFIM at $T > 0$, $T < +\infty$ leads to the same critical behaviour as a crossing at $T = 0$ by varying h does. Hence, by conducting the above study we are able to understand better the system's behaviour when it changes from ferromagnetic (FM) to paramagnetic (PM), by using this alternative method. As we shall shortly discuss, previous results regarding the breakup length scale are not consistent and there is no concrete study of the breakup length scale for big lattice sizes. We therefore attempt to clarify once and for all the RFIM's behaviour in terms of the breakup length scale and answer the open questions regarding the problem.

4.1.1 Previous results

Results so far suggest an exponential scaling of the breakup length $\ell_{\mathrm{B}}(h)$ and they all refer to square lattices. Let us start with the analytical approach performed by Binder (1983), who attempted, in 2 and 3 dimensions and at 0 temperature, a detailed understanding of the width $w(L, h)$ of the interface between states of positive and negative magnetization of the RFIM, which is formed due to boundary conditions.

For the 2-dimensional case, Binder started by considering a large but finite Ising square $L \times L$ lattice and applied fixed-spin boundary conditions such that the GS for $h = 0$ has one interface of length L, separating the regions of spins

up and spins down, which is perfectly flat for $h = 0$ and $T = 0$. In turn, he performed various uniform displacements of the interface by different distances w and examined how the interfacial excess energy varied. He concluded that there exists a length scale ℓ_B for which the total interfacial energy becomes negative:

$$\ell_B \propto e^{A(\frac{J}{h})^2}, \tag{4.1}$$

a fact that implies that ℓ_B is the minimum length scale at which the system can be spontaneously broken into domains (Binder, 1983).

Following Binder's work, Bray and Moore (1985a) used RG techniques in order to calculate the contribution to the wall energy from the roughening of the domain wall induced by the random field. They ended up with a similar exponential law, suggesting that the contribution to the wall energy is analogous to $-L\frac{h^2}{J}\ln L$.

The first numerical evaluation of the breakup length scale was performed by Seppäla, Petäjä et al. (1998). Their calculations were based on the definition of the ratio r of the number of the ferromagnetic ground states over the overall number of realizations of the disorder which were under consideration:

$$r = \frac{\#\texttt{FMGS}}{\#\texttt{samples}} = 0.5. \tag{4.2}$$

They investigated the breakup length scale of the $2D$ RFIM for both the Gaussian and the bimodal distributions and for lattice sizes up to $L = 1000$ via exact combinatorial optimization techniques and in particular, with the use of an efficient push-relabel preflow-type algorithm (Goldberg and Tarjan, 1988). Their approach led to the same scaling law as the one derived by Binder, given from Eq. (4.1) for $J = 1$, for both distributions of the random fields.

Another numerical approach to the problem, this time with the use of graph cut methods described by Boykov, Veksler et al. (2001), was conducted by Kumar et al. (2018), this time for the $2D$ random-field Potts model (RFPM). Square lattices of size up to $L = 512$ were considered and, following the work of Seppäla, Petäjä et al. (1998), the definition given by Eq. (4.2) was employed. Their calculations led to the following exponential law regarding the breakup length scale:

$$\ell_B \propto e^{\frac{A}{h}}. \tag{4.3}$$

The above result is not consistent with the scaling law derived by Binder (1983) and Seppäla, Petäjä et al. (1998) regarding the RFIM, but it is in agreement with the outcome of the numerical simulations of the RFIM performed by Shrivastav, Banerjee et al. (2014).

4.1.2 Our approach

In order to address the breakup length scale problem, we employ graph cut methods at $T = 0$ and in particular, we make use of the BK algorithm for our simulations. We consider the Gaussian distribution for the random fields h_i and various lattice sizes of up to $L = 2048$.

More specifically, $\forall L \in \{8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$ we look at 10^5 samples (realizations of the quenched disorder) and we figure out the GS of each one of them. We are therefore able to calculate the ratio:

$$r = \frac{\#\texttt{FMGS}}{10^5}, \tag{4.4}$$

for different disorder strengths h. Afterwards, we perform Jackknife analysis (Young, 2012) to estimate the breakup field $h_{\mathrm{B}}(L)$, i.e. for what disorder strength does r satisfy our definition of choice for the ratio r. Hence, we can examine how the breakup length scales.

A more detailed description of how we conduct the study follows. However, before that, let us point out the four main points of focus of our research:

1. Which law governs the scaling of the breakup length $\ell_{\mathrm{B}}(h)$?
2. Does this law depend on the definition used for the ratio?
   $\longrightarrow$ Examine the case where $r = 0.01$.
3. Does this law depend on the lattice geometry? (Hayden et al., 2019)
   $\longrightarrow$ Examine the case where the lattice is *non-bipartite*, e.g. the triangular lattice.
4. What about the scaling of the correlation length ξ?

Let us now focus on the first point above and illustrate our method in more detail. In order to select the appropriate disorder strength values for which we shall perform our large-scale simulations, we start with fewer samples, namely, 10^4 realizations of the disorder. We focus on the square lattice case. We use the optimized version of the BK algorithm to figure out the ground states of our samples and then calculate the ratio $r = \frac{\#\texttt{FMGS}}{10^4}$, for various disorder strength values. Thus, we can produce a plot of our results, as seen in Fig. 4.1. The code for calculating the ratio r is given in Appx. Sec. A.8.

It is clear from Fig. 4.1 that for all lattice sizes, as the disorder strength values increase, there holds that $r \to 0$, which implies that the disorder is more prevalent. Now as L increases, we notice a similar behaviour for r and what is more, the curves are getting steeper. The latter fact denotes that bigger lattice

sizes exhibit a more localized breakup field, i.e. disorder is susceptible to very small variations of the disorder strength. Finally, one may notice that the widths between the curves are getting smaller as L grows. This is an expected behaviour for the ratio r as we approach the asymptotic limit, analogous to the one observed with the pseudo-critical temperatures.

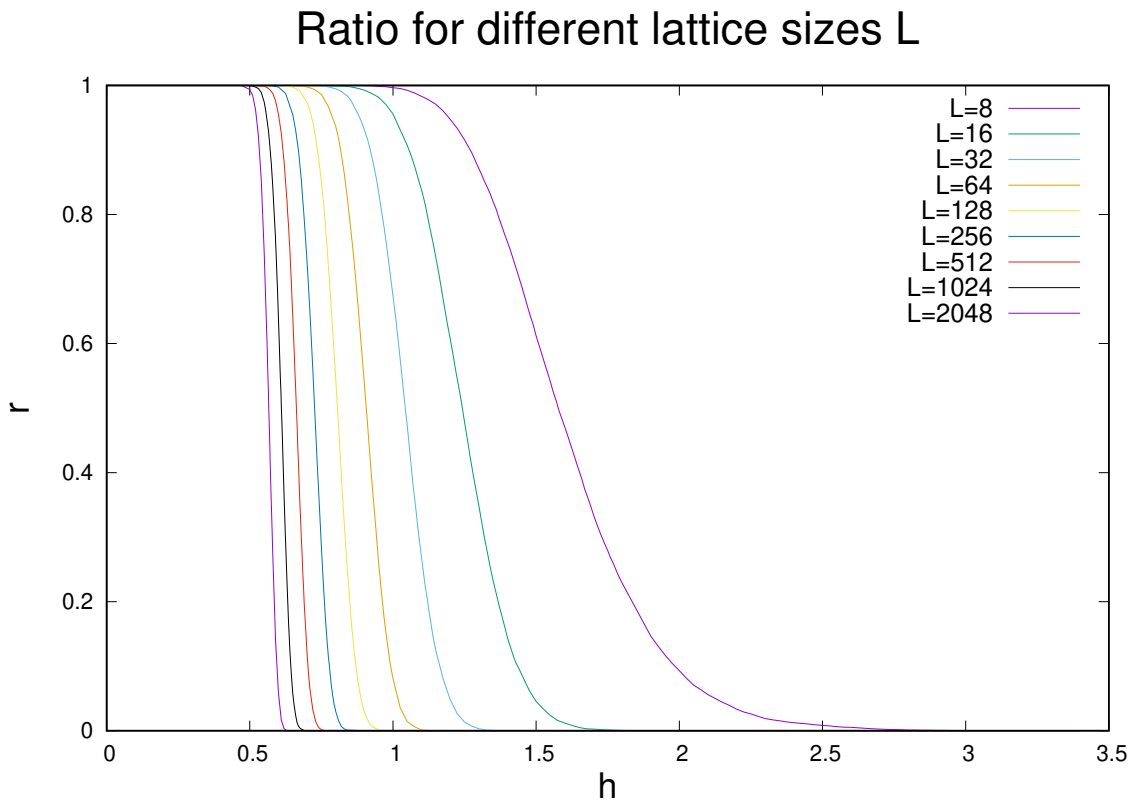

FIGURE 4.1: Plots of the ratios $r = \frac{\texttt{FM GS}}{\texttt{ALL GS}}$ as functions of the disorder strength h, for various system sizes L. For this plot, #ALL GS = 10^4.

Now let us illustrate how Fig. 4.1 allows us to pick the appropriate h points for our large scale simulations (#ALL GS = 10^5) and Jackknife analysis. From our results for the various disorder strength values, we choose an interval around $h_{\mathrm{B}}(L)$ that corresponds to an interval for the ratios $r \in (0.4, 0.6)$. Inside this interval, we rerun the code with a small step for h, in order to obtain more points and perform a first fit. We choose polynomial fits of 2^{nd} and 3^{rd} degrees and we therefore obtain a first, rough estimation for $h_{\mathrm{B}}(L)$, by calculating the mean value of $h_{\mathrm{B}}(L)$, $\overline{h^*_{\mathrm{B}}(L)}$ derived from the first (2^{nd} degree) and second (3^{rd} degree) fit, $h^*_{\mathrm{B}}(L)_2$ and $h^*_{\mathrm{B}}(L)_3$, respectively:

$$\overline{h^*_{\mathrm{B}}(L)} = \frac{h^*_{\mathrm{B}}(L)_2 + h^*_{\mathrm{B}}(L)_3}{2}. \tag{4.5}$$

L	$h_{\mathrm{B}}^{*}(L)_2$	$h_{\mathrm{B}}^{*}(L)_3$	$\overline{h_{\mathrm{B}}^{*}(L)}$
8	1.5775	1.57828	1.57789
16	1.2398	1.24128	1.24054
32	1.0417	1.04268	1.04219
64	0.907259	0.906194	0.9067265
128	0.80687	0.805343	0.8061065
256	0.72630	0.726406	0.726353
512	0.6627	0.662553	0.6626265
1024	0.61036	0.577719	0.5940395
2048	0.566308	0.566285	0.566295

TABLE 4.1: A first estimation of the breakup field $h_{\mathrm{B}}(L)$, as the mean $\overline{h_{\mathrm{B}}^{*}(L)}$ of the results of two different fits performed inside the disorder strength interval corresponding to ratio values belonging to (0.4, 0.6).

The results of the fits are presented in Table 4.1. The above results constitute only a rough estimation of the breakup field and they can only be used as a guiding point in order to pick the appropriate h values for the productive runs. This can be done by choosing 5 values at and around the breakup field $\overline{h_{\mathrm{B}}^{*}(L)}$ and performing simulations for 10^5 samples at these disorder strength points. We ensure that our productive runs will offer results with high statistics, not only by increasing the number of samples, but also by feeding our random number generator with different seeds for every realization of the disorder. What is more, as can be seen in the discussion following in Subsec. 4.2, we shall not use a simple fit, but a Jackknife analysis of the results.

At this moment let us point out that the process described above for the estimation of the breakup field using the definition $r = 0.5$ for the ratio is the same when one chooses to concentrate on the second or third focal spots of our research, namely, the $r = 0.01$ definition and/or the triangular lattice geometry, respectively.

Before proceeding to the description of our different approach using the correlation length ξ (fourth focus point of our study), let us outline the Jackknife analysis we use for our results. Up till now, as we have discussed, for each lattice size L we examine each one of 10^4 ground states and decide whether it is ordered or disordered. Thus, we calculate the ratio of the ordered states over the overall samples. Doing so for different disorder strengths allows us to obtain the plots for the various square lattice sizes presented in Fig. 4.1.

Now as far as it concerns the productive runs, we take into consideration

10^5 realizations of the quenched disorder and 5 disorder strength points for each lattice size. However, for each size L, instead of the above procedure, we separate the samples into 100 bins, where each bin contains 1000 ground states. For each of the 5 disorder strength values, we calculate again the ratio r, excluding one of the 100 bins and therefore produce a new plot. We repeat this process 100 times (once for each bin) and therefore obtain 100 plots $\forall L$. The code for this first step of the analysis of our results is provided in Appx. Sec. C.1.

Afterwards, for each of the 100 above plots, we estimate the breakup field $h_{\mathrm{B}}(L)$ with a linear fit, that is, by solving the below equation:

$$r = a h_{\mathrm{B}_i}(L) + b \tag{4.6}$$

100 times and therefore obtain 100 estimations of the breakup field, one for each bin $i \in 0, 1, ..., 99$. The code for performing the 100 simultaneous fits is given in Appx. Sec. C.2 and the code that uses the parameter values a, b yielded from each fit in order to calculate the breakup field for each bin is given in Appx. Sec. C.3. Thus, we are ready to carry out our Jackknife analysis to these 100 values derived from solving Eq. (4.6) 100 times.

For the above purpose, we need a code that performs the following steps:

- Reads the data file including the 100 solutions to the linear fits:

$$h_{\mathrm{B}_i}(L) = \overline{x_i}, \tag{4.7}$$

  where $i = 1, ..., N$ and $N = 100$ bins.

- Calculates the overall Jackknife estimate, which is their mean (Young, 2012):

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\overline{x_i} = \frac{1}{100}\sum_{i=1}^{100} h_{\mathrm{B}_i}(L), \tag{4.8}$$

  which is the final estimation of the breakup field $h_{\mathrm{B}}(L)$.

- Works out the variance:

$$s^2 = \frac{1}{N}\sum_{i=1}^{N}(\overline{x_i})^2 - \mu^2 = \frac{1}{100}\sum_{i=1}^{100}\left(h_{\mathrm{B}_i}(L)\right)^2 - \mu^2, \tag{4.9}$$

  which yields the error (σ) bar for the final estimation of the breakup field, via the equation:

$$\sigma = s\sqrt{N-1} = s\sqrt{99}. \tag{4.10}$$

The code for the Jackknife analysis of the 100 values of the breakup field is provided in Appx. Sec. C.4.

At this point we proceed with the description of an alternative approach in the study of the zero-temperature RFIM. In particular for the first time in the literature we scrutinize the scaling behaviour of the correlation length of the model using the methods developed by Fytas and Martín-Mayor (2016).

On the technical side, for this part of our study we made use of the push-relabel algorithm (Goldberg and Tarjan, 1988; Papadimitriou, 1994). As we mentioned and briefly described in Subsec. 3.2.1, our implementation involves a modification proposed by Middleton and collaborators (Middleton, 2001; Middleton and Fisher, 2002).

Let us now focus on the physical problem of the actual computation of the correlation length. Let us consider an instance of the random fields $\{h_x\}$. In the following, thermal mean values are denoted as $\langle \cdots \rangle$, while the subsequent average over samples is indicated by an over-line. As it is well-known and already mentioned previously a crucial feature of the RFIM is that we have to deal with two different correlation functions, namely the *connected* and *disconnected* propagators.

The disconnected propagator is straightforward to compute both in real, $G_{xy}^{(\mathrm{dis})}$ and Fourier space, $\chi_k^{(\mathrm{dis})}$:

$$G_{xy}^{(\mathrm{dis})} = \overline{\langle s_x s_y \rangle}\,, \quad \chi_k^{(\mathrm{dis})} = L^D \overline{\langle |m_k|^2 \rangle}_k\,, \tag{4.11}$$

where

$$m_k = \frac{1}{L^D} \sum_x \mathrm{e}^{\mathrm{i}k\cdot x} s_x\,. \tag{4.12}$$

In particular, special notations are standard for vanishing wavevector: $m_{k=(0,0)} = m$ (i.e. the order-parameter density) and $\chi_{k=(0,0)}^{(\mathrm{dis})} = \chi^{(\mathrm{dis})}$ (i.e. the disconnected susceptibility).

On the other hand, we have the connected propagator:

$$G_{xy} = \overline{\frac{\partial \langle s_x \rangle}{\partial h_y}}\,. \tag{4.13}$$

At finite temperature, one could compute G_{xy} from the Fluctuation-Dissipation theorem:

$$G_{xy} = \frac{1}{T} \overline{\langle s_x s_y \rangle - \langle s_x \rangle \langle S_y \rangle}\,. \tag{4.14}$$

Square lattice											
First law						Second law					
$r = 0.5$			$r = 0.01$			$r = 0.5$			$r = 0.01$		
a_1	b_1	χ^2	c_1	d_1	χ^2	a_2	b_2	χ^2	c_2	b_2	χ^2
0.188992	0.328021	626.003	0.194177	0.150389	144.008	0.566961	−1.23428	4945.31	0.525496	−1.39076	1655
Triangular lattice											
q_1	r_1	χ^2				q_2	r_2	χ^2			
0.110506	0.210464	55.5393				0.189357	−0.365636	2354.85			

TABLE 4.2: Parameter and fit quality values for the linear fits performed to the data obtained for the breakup length scale behaviour of the $2D$ RFIM. Two different lattice geometries, that is, square and triangular, two different definitions for the ratios, i.e., $r = 0.5$ and $r = 0.01$ and two different laws, namely $\frac{1}{h}$ and $\frac{1}{h^2}$ are investigated.

However, we work directly at $T = 0$. Therefore, Eq. (4.14) is clearly unsuitable for us and to overcome this barrier we made use of fluctuation-dissipation formulas of Fytas and Martín-Mayor (2016) (see also Schwartz and Soffer, 1985).

Now, the connected propagator in Fourier space is:

$$\chi_k = \frac{1}{L^D} \sum_{x,y} e^{ik\cdot(x-y)} \frac{G_{xy} + G_{yx}}{2} . \tag{4.15}$$

Again, the case of vanishing wavevector deserves a special naming: $\chi_{k=(0,0)} = \chi$ is the connected susceptibility.

From both propagators, we computed the connected, $\xi^{(\mathrm{con})}$ and disconnected, $\xi^{(\mathrm{dis})}$, second-moment correlation lengths (Amit and Martín-Mayor, 2005; Cooper et al., 1982) for both square and triangular lattice geometries via

$$\xi^{\#} = \frac{1}{2\sin(\pi/L)} \sqrt{\frac{\chi^{\#}}{\chi^{\#}_{k_{\min}}} - 1} , \tag{4.16}$$

where the superscript $^{\#}$ stands both for the connected or the disconnected case and as usual $k_{\min} = (2\pi/L, 0)$.

4.2 Results

In this final section of the current thesis, we present the results of our work regarding the breakup length scale problem of the $2D$ RFIM. Recall that we take into consideration 10^5 realizations of the disorder and lattice sizes of up to $L = 2048$ for our simulations and analysis.

We initially turn our attention to the law that governs the breakup length $\ell_{\mathrm{B}}(h)$ for the two different definitions of the ratio r (see Eq. (4.4)), namely $r = 0.5$ and

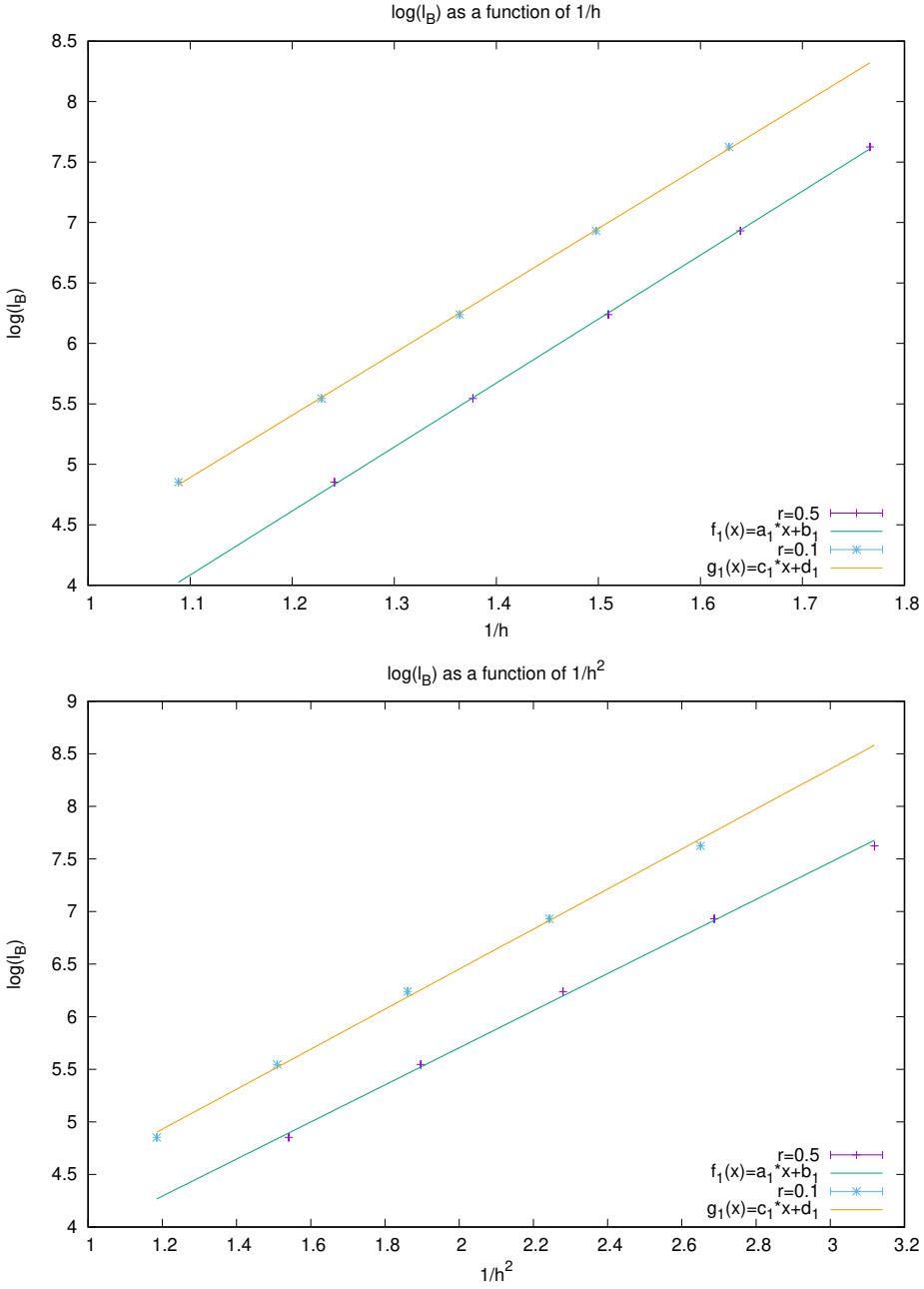

FIGURE 4.2: Examine if the break-up length scales according to the exponential law $\ell_B \propto e^{A\frac{1}{h}}$ (up) or the exponential law $\ell_B \propto e^{A(\frac{1}{h})^2}$ (down). The results are obtained $\forall L \in \{8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$, from Jackknife analysis on 10^5 samples. The first 4 points are excluded when performing the linear fits.

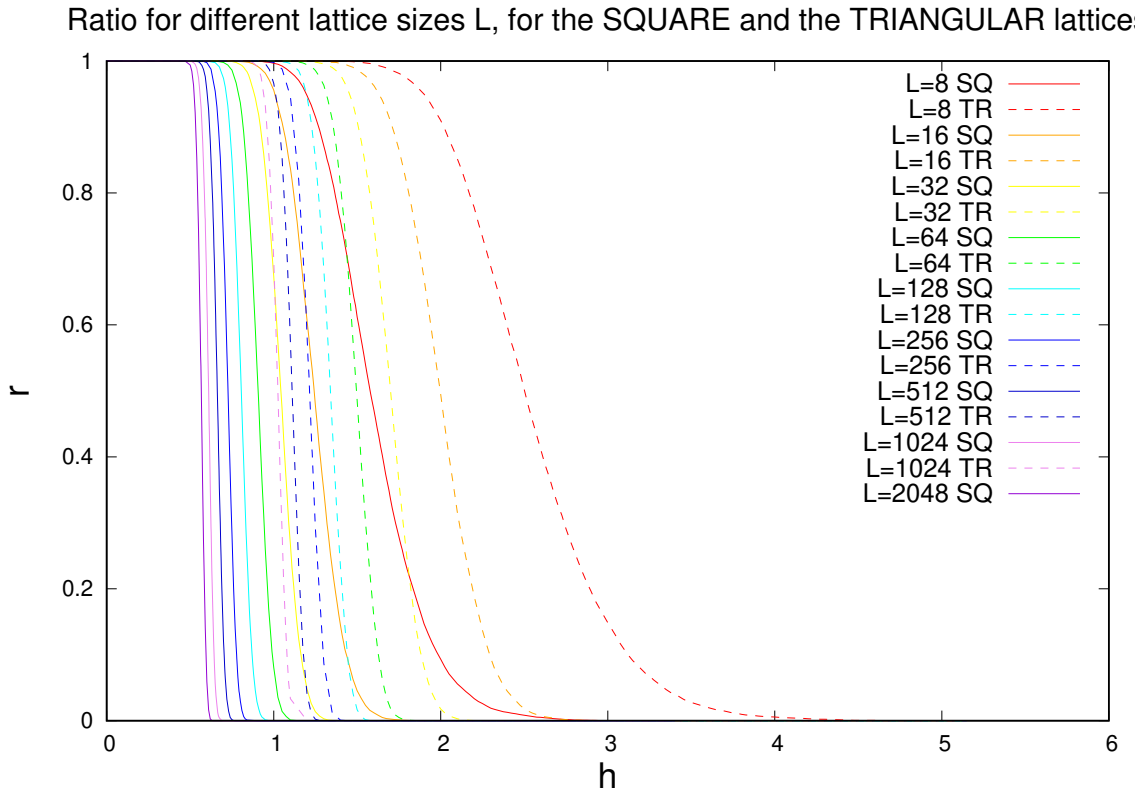

FIGURE 4.3: Plots of ratios $r = \frac{\texttt{FM GS}}{\texttt{ALL GS}}$ as functions of the disorder strength h, for square and triangular lattice geometries (solid and dashed lines, respectively). Lines of same colour indicate same lattice size, L. $\#\texttt{ALL GS} = 10^4$.

$r = 0.01$. We examine the square lattice first. As we saw in Subsec. 4.1.2, we choose 5 disorder strength values for every lattice size L for our productive runs and Jackknife analysis. In Fig. 4.2 we present our results for the two different ratio definitions and we perform a comparison between the two different exponential laws (see Eq. (4.1) and Eq. (4.3)).

For the above purpose we perform linear fits to our results, as can be seen in Fig. 4.2, with the values of the fit parameters and of χ^2 being provided in Table 4.2. It is obvious from Fig. 4.2 but also from the quality of the fits as seen in Table 4.2 that our results are in favour of the exponential law given by Eq. (4.3), for both ratio definitions, although no definite conclusion may be drawn. It is also implied that the use of a different definition for the ratio, namely, $r = 0.01$ does not affect the length scale behaviour of the RFIM. Finally, Fig. 4.2 indicates the presence of FSS corrections and therefore a more delicate analysis for the determination of the critical behaviour of the RFIM is needed. Thus, we shall

shortly apply our approach regarding the correlation length ξ, as described in Subsec. 4.1.2.

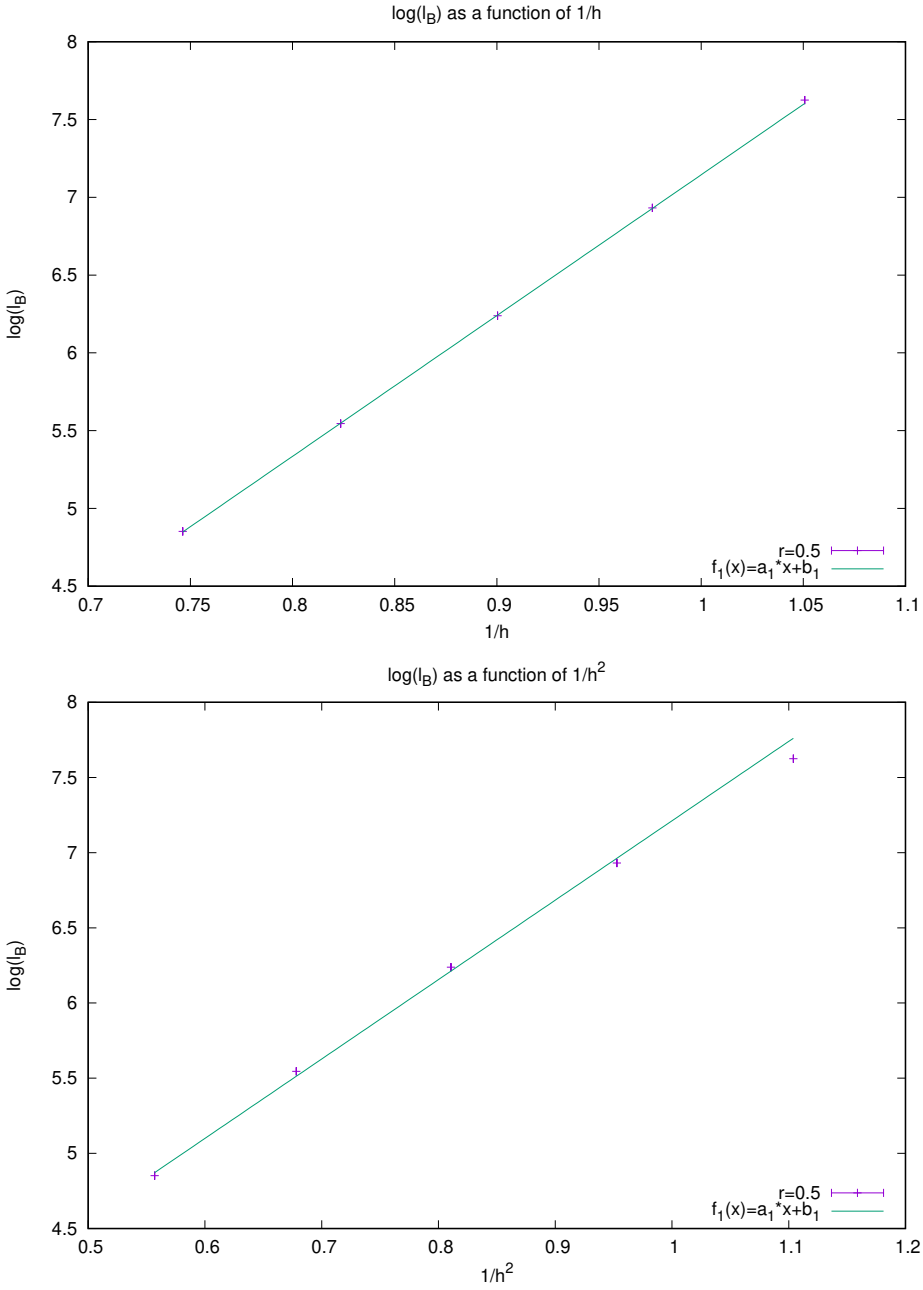

FIGURE 4.4: Examine if the break-up length scales according to the exponential law $\ell_B \propto e^{A\frac{1}{h}}$ (up) or the exponential law $\ell_B \propto e^{A(\frac{1}{h})^2}$ (down). The results refer to the triangular lattice and they are derived from Jackknife analysis on 10^5 samples, $\forall L \in \{8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$. The first 4 points are not taken into account for the fits.

Now let us turn our focus on the triangular lattice geometry. We implement

the BK algorithm for the triangular RFIM for the first time. As seen in Subsec. 4.1.2, we take into consideration 10^4 samples for our initial estimation of the length scale behaviour of the triangular RFIM. In Fig. 4.3 we plot the ratios as presented in Fig. 4.1, along with our new results regarding the triangular lattice. It is clear that the breakup length follows a similar behaviour for both lattice geometries. One may also observe that higher disorder strengths are required for the triangular case in order to break the ferromagnetic ordering. This is expected, since the extra bonds between the neighbours make the ferromagnetic ordering more resistant to the external magnetic field.

Let us now proceed to the productive runs and examine if the above indications are correct. In Fig. 4.4 we perform linear fits to the results we derive from the productive runs performed for 10^5 samples. The fit parameters and χ^2 values are presented in Table 4.2. Similarly to the square lattice case, the results for the triangular RFIM are in agreement to the exponential law given by Eq. (4.3). This fact implies that a change to the lattice geometry did not affect the length scale behaviour of the RFIM. Note that the alternative definition for the ratio, namely, $r = 0.01$ is not used this time, since we have already concluded that it plays no role in the system's behaviour.

We now proceed to our alternative approach that involves the scaling of the correlation length ξ computed form ground-state simulations and the fluctuation-dissipation formalism of Fytas and Martín-Mayor (2016). This is the first computation of ξ in the 2D random-field problem in the literature and as will be seen below clears out the current ambiguous situation with respect to the scaling behavior. Again we simulated square and triangular lattices with linear sizes in the range from $L_{\min} = 128$ to $L_{\max} = 2048$ and periodic boundary conditions. For each set of $\{L, h\}$ we average over 10^5 samples outperforming previous relevant studies. For comparison: $L_{\max} = 1000$ and 5×10^3 samples in Seppäla, Petäjä et al. (1998).

An illustrative overview of our results is given in Fig. 4.5. In particular, in panel (a) we show the disconnected correlation length ξ extracted from the fluctuation-dissipation formalism for lattices of sizes $L = 128$ up to $L = 2048$ as a function of h compared to the breakup length ℓ_b defined from the value of h where $r = 0.5$. In panels (b) and (c) we present the compatibility of the scaling of ξ and ℓ_b with the functional form $e^{\frac{-A}{h}}$ and $e^{\frac{-A}{h^2}}$, respectively. It is clear that the numerical data favour the scaling law $\xi \sim e^{\frac{-A}{h^2}}$ in panel (c), as it is also manifested by the fitting quality documented by the χ^2/d.o.f. criterion. Similar results and conclusions have been drawn also for the case of the connected correlation

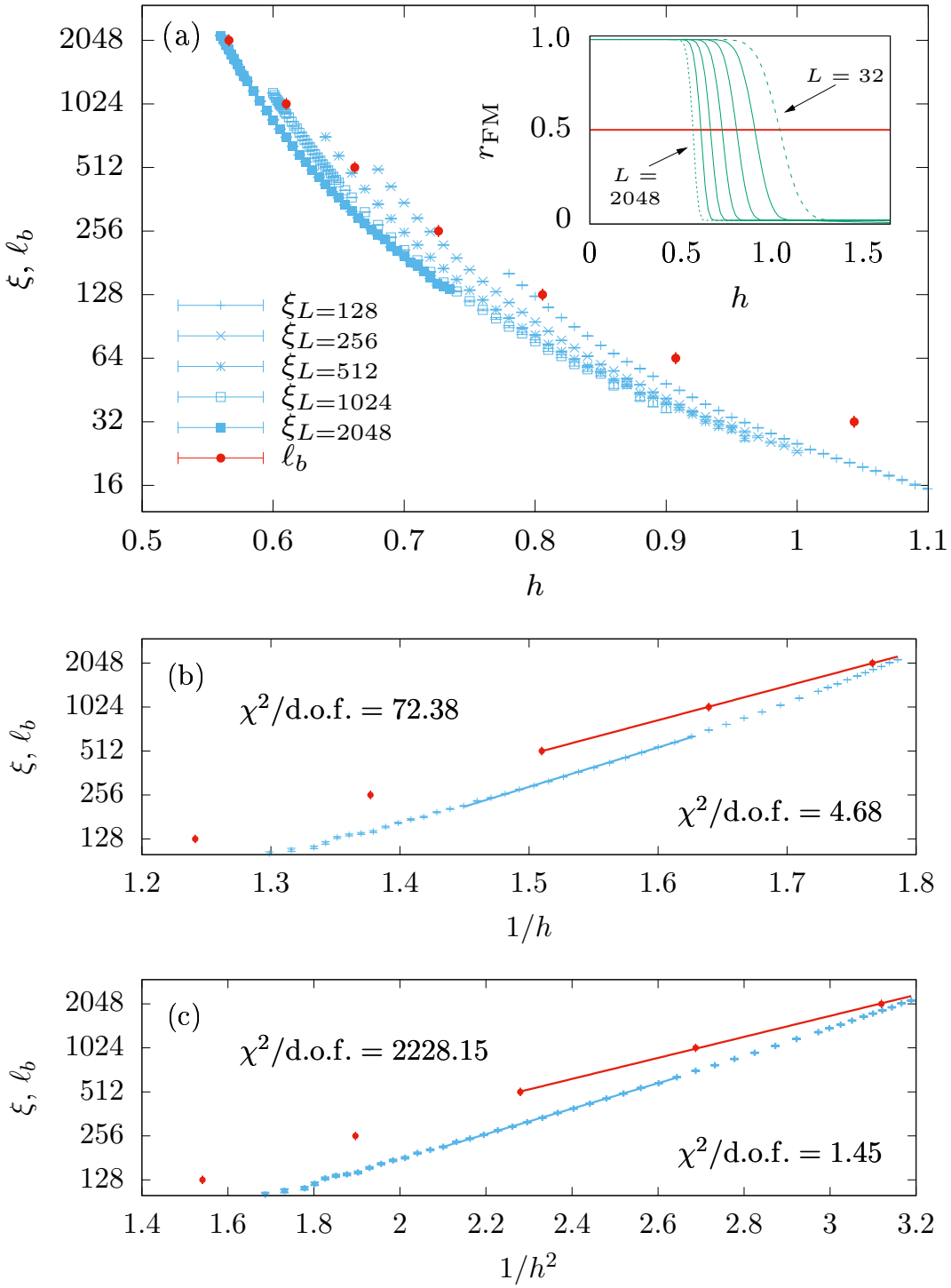

FIGURE 4.5: Correlation length ξ and breakup length ℓ_b of the $2D$ RFIM from exact GS calculations. (a) Disconnected correlation length ξ extracted from the fluctuation-dissipation formalism for lattices of sizes $L = 128$ up to $L = 2048$ as a function of h compared to the breakup length ℓ_b defined from the value of h where $r = 0.5$. (b) Compatibility of the scaling of ξ and ℓ_b with the functional form $e^{\frac{-A}{h}}$. (c) Compatibility of the scaling of ξ and ℓ_b with the functional form $e^{\frac{-A}{h^2}}$.

length and for the triangular lattice as well (Fytas, Mainou and Weigel, 2021). A summary of effective exponents x for fits of the form $\xi \sim e^{\frac{-A}{h^x}}$ varying the cutoff value of the random field $h_{\max}$ used in the fits is given in Fig. 4.6 for both lattice geometries and correlation lengths which again agrees nicely with the results presented in Fig. 4.5.

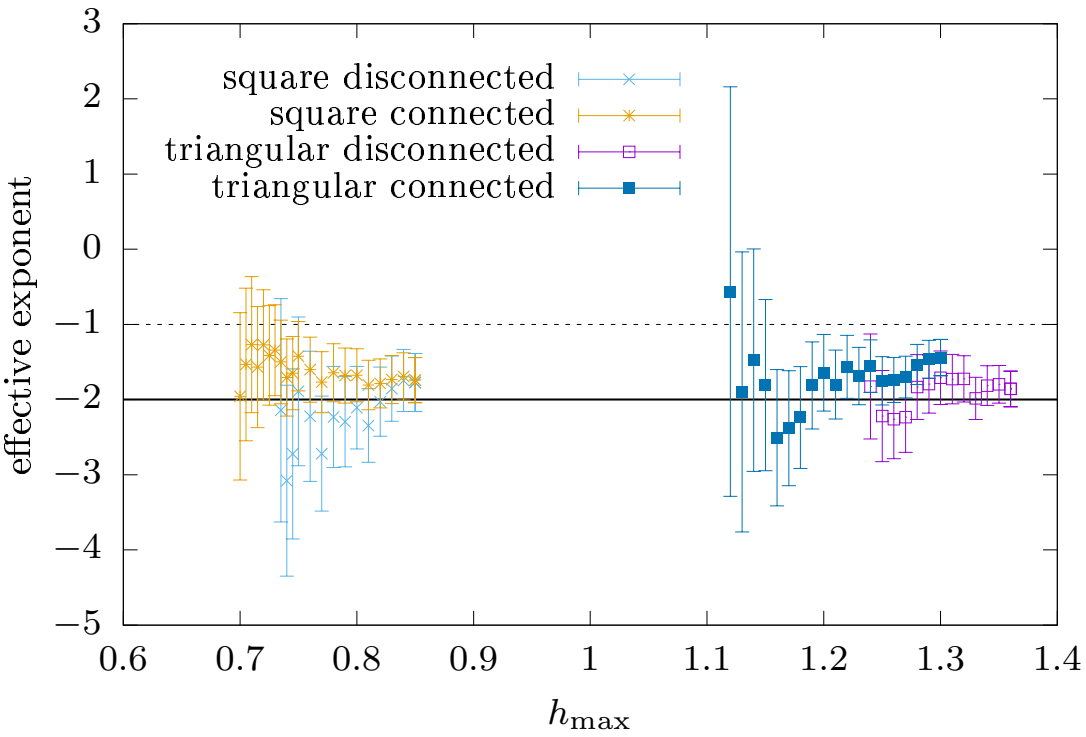

FIGURE 4.6: A summary of effective exponents x for fits of the form $\xi \sim e^{\frac{-A}{h^x}}$ varying the cutoff value of the random field $h_{\max}$ used in the fits for both lattice geometries and the two types of correlation lengths.

As an overall, our results indicate that while the breakup length scales according to the exponential law given by Eq. (4.3), the same does not hold for the correlation length ξ, which clearly scales according to the exponential law $\xi \sim e^{\frac{-A}{h^2}}$. Nonetheless, we observe that for both cases, a change to the lattice geometry of the RFIM does not affect its critical behaviour.

Conclusions

To summarize, the present thesis deals with several aspects of phase transitions and critical phenomena of pure and disordered magnetic systems. This is an area of active research in condensed matter physics with numerous applications that goes well beyond physics itself.

In the first part of the thesis, Chap. 1, we introduced several basic concepts of the theoretical background of statistical physics. In the scope of understanding some of the open problems in the theory of critical phenomena of pure and disordered systems, two models were studied: the two-dimensional Blume-Capel and random-field Ising models.

In Chap. 2, the standard computational methods in the research of complex systems were presented, namely Monte Carlo techniques such as the Metropolis and the Wolff algorithms, along with their limitations. An application of these methods was illustrated in the study of the interfacial adsorption of the Blume-Capel model. In particular, we studied the finite-size scaling behavior of the interfacial adsorption of the pure model at both its first- and second-order transition regimes, as well as at the vicinity of the tricritical point. Our analysis benefited from the currently existing quite accurate estimates of the relevant (tri)critical-point locations. In all studied cases, the numerical results verified to a level of high accuracy the expected scenarios derived from analytic free-energy scaling arguments. We also investigated the size dependence of the interfacial adsorption under the presence of quenched bond randomness at the originally first-order transition regime (disorder-induced continuous transition) and the relevant self-averaging properties of the system. For this ex-first-order regime, where strong transient effects are shown to be present, our findings support the scenario of a non-divergent scaling, similar to that found in the original second-order transition regime of the pure model.

Urged by the mentioned limitations on the above most popular Monte Carlo numerical techniques and by the fact that a crossing of the phase boundary of the random-field Ising model at positive, finite temperature leads to the same critical behaviour as crossing at zero temperature and varying the disorder strength

h does, in Chap. 3, the graph cut methods used for the study of the critical behaviour of the RFIM were described and an appropriate algorithm in order to implement such methods was introduced, that is, the Boykov-Kolmogorov algorithm. The effectiveness of the algorithm was demonstrated through several examples and an optimization of the code was developed. It was therefore shown that the algorithm outruns many of the most popular minimum cut-maximum flow algorithms when implemented for linear lattice sizes up to $L = 4096$.

Finally, in Chap. 4, the breakup length scale problem of the random-field Ising model was introduced, which constitutes the main application of our research on graph cut methods. Previous results regarding above what length scale does the random-field Ising model break its ferromagnetic ordering are inconsistent and an attempt to answer this question once and for all was made. Implementing the optimized version of the Boykov-Kolmogorov algorithm for different definitions of the ratio $r = \frac{\#\texttt{FMGS}}{\#\texttt{samples}}$ and for square and triangular lattices, questions such as which law governs the scaling of the breakup length of the random-field Ising model and whether this law depends on the definition used for the ratio or on the lattice geometry were addressed. An interesting result of our analysis is that the widely used break-up length scale of the system appears to be afflicted by very strong scaling corrections and hence constitutes a rather less useful quantity. On the other hand, an alternative approach that involves the scaling analysis of the second-moment correlation length ξ of the model, as obtained from a recently developed fluctuation-dissipation formalism, allowed us to present compelling numerical evidence for both lattice geometries that without doubt support the form $\xi \sim e^{\frac{A}{h^2}}$. This result is in line with early theoretical work but at variance with some more recent numerical and analytical results.

Appendix A

BK Codes

Appx. A includes the content of the `main.cpp` files created for working out the maximum flow of various graph examples, using the software given by Boykov and Kolmogorov (2004).

A.1 Code for the example in the 3.01 version of the BK software

```
#include <stdio.h>
#include "graph.h"

int main()
{
typedef Graph<int,int,int> GraphType;
GraphType *g = new GraphType(2, 6);

g -> add_node();
g -> add_node();

    g -> add_tweights(0, 2, 6);
g -> add_tweights(1, 1, 5);
g -> add_edge(0, 1, 4, 3);

int flow = g -> maxflow();
    printf("Flow = %d\n", flow);

printf("Minimum cut configuration:\n");
if (g->what_segment(0) == GraphType::SOURCE)
```

```
printf("node0 is in the SOURCE set\n");
else
printf("node0 is in the SINK set\n");
if (g->what_segment(1) == GraphType::SOURCE)
printf("node1 is in the SOURCE set\n");
else
printf("node1 is in the SINK set\n");

delete g;

return 0;
}
```

A.2 Code for a 2×2 RFIM without magnetic field

```
#include <stdio.h>
#include "graph.h"

int main()
{
typedef Graph<int,int,int> GraphType;

GraphType *g = new GraphType(4, 10);

g -> add_node();
g -> add_node();
    g -> add_node();
g -> add_node();

g -> add_tweights(0, 4, 0);
g -> add_tweights(3, 0, 4);

        g -> add_edge(0, 1, 4, 0);
        g -> add_edge(0, 2, 4, 0);
        g -> add_edge(1, 3, 4, 0);
        g -> add_edge(2, 3, 4, 0);
```

```
int flow = g -> maxflow();
    int oC = -8;
    flow += oC;
printf("Flow = %d\n", flow);

printf("Minimum cut configuration:\n");

        for(int i=0; i<4; i++) {
                if (g->what_segment(i) == GraphType::SOURCE)
printf("node %d is in the SOURCE set\n", i);
        else
printf("node %d is in the SINK set\n", i);
        }

delete g;

return 0;
}
```

A.3 Code for a 2×2 RFIM with magnetic field

```
#include <stdio.h>
#include "graph.h"

int main()
{
typedef Graph<int,int,int> GraphType;
GraphType *g = new GraphType(4, 10);

g -> add_node();
g -> add_node();
    g -> add_node();
g -> add_node();

g -> add_tweights(0, 4, 0);
g -> add_tweights(1, 0, 4);
    g -> add_tweights(2, 4, 0);
```

```
g -> add_tweights(3, 0, 4);

        g -> add_edge(0, 1, 4, 0);
        g -> add_edge(0, 2, 4, 0);
        g -> add_edge(1, 3, 4, 0);
        g -> add_edge(2, 3, 4, 0);

int flow = g -> maxflow();
    int oC = -12;
    flow += oC;
printf("Flow = %d\n", flow);

printf("Minimum cut configuration:\n");
        for(int i=0; i<4; i++) {
                if (g->what_segment(i) == GraphType::SOURCE)
printf("node %d is in the SOURCE set\n", i);
        else
printf("node %d is in the SINK set\n", i);
        }

delete g;

return 0;
}
```

A.4 Code for one realization of the quenched disorder, for a square $L \times L$ RFIM with Gaussian random-field distribution, where $L = 4$ and $h = 1$

```
#include <stdio.h>
#include "graph.h"
#include <gsl/gsl_randist.h>
#include <math.h>
#define L 4
#define N (L*L)
```

```
#define z 4
#define J 1
#define D 2


using namespace std;

static gsl_rng *rng;


void Init_RNG(int seed) {
    rng=gsl_rng_alloc(gsl_rng_mt19937);
    gsl_rng_set(rng, seed);
}


int main(int argc, char *argv[]){

        int seed=12345;
        Init_RNG(seed);

        double *h;
        h = (double *)malloc(sizeof(double)*N);
        if (h == NULL) {
            printf("Memory not allocated\n");
            exit(0);
        }
        else {
            printf("Memory successfully allocated for the random fields\n");
        }

        int node_RF;

        for(int i=0; i<L; i++) {
            for(int j=0;j<L;j++) {
                node_RF=j+L*i;
                double r=gsl_ran_gaussian(rng, 1);
                h[node_RF]=r;
```

```
    }
}

typedef Graph<double,double,double> GraphType;

GraphType *g = new GraphType(N+2, 2*N+N*z/2+1);
for(int i=0; i<N; i++)  g -> add_node();

double w=0,k=0;
double sC=0,tC=0;
double Crightout=0,Crightin=0,Cleftout=0,Cleftin=0;
double Cdownout=0,Cdownin=0,Cupout=0,Cupin=0;

for(int node=0; node<N; node++) {

    int right = node + 1;
    int left = node - 1;
    int up = node - L;
    int down = node + L;

    Crightout=4*J;
    Crightin=0;
    Cleftout=0;
    Cleftin=4*J;
    Cdownout=4*J;
    Cdownin=0;
    Cupout=0;
    Cupin=4*J;

    if (right % L == 0 ){
        right -= L;
        Crightout=0;
        Crightin=4*J;
    }
    if (node % L == 0){
        left +=L;
        Cleftout=4*J;
        Cleftin=0;
```

```
    }
    if (up < 0){
        up +=N;
        Cupout=4*J;
        Cupin=0;
    }
    if (down > (N-1)){
        down -=N;
        Cdownout=0;
        Cdownin=4*J;
    }

    w=-2*h[node] - (Crightout+Cleftout+Cdownout+Cupout-Crightin
    -Cleftin-Cdownin-Cupin)/2;

    if(w > 0) {
      sC=0;
      tC=w;
      }
    else {
      sC=-w;
      tC=0;
      }

    k += sC+tC;

    g -> add_tweights(node, sC, tC );
    g -> add_edge(node, right, Crightout, Crightin);
    g -> add_edge(node, down, Cdownout, Cdownin);

    }

double oC=-N*z/2-k/2;

double flow = g -> maxflow();
flow += oC;

double Efield=0;
```

```
int Ebond=0;
double Etotal=0;
double mag = 0;

double *spin;
spin = (double *)malloc(sizeof(double)*N);
if (spin == NULL) {
    printf("Memory not allocated\n");
    exit(0);
}
else {
    printf("Memory successfully allocated for the spins\n");
}

printf("Spin configuration:\n");
for(int node=0; node<N; node++) {

    if (g->what_segment(node) == GraphType::SOURCE){
        spin[node]=1;
        printf("node %d is in the SOURCE set and
        the corresponding spin is %g\n", node, spin[node]);
    }
    else{
        spin[node]=-1;
        printf("node %d is in the SINK set and
        the corresponding spin is %g\n", node, spin[node]);
    }

    Efield = Efield -h[node] * spin[node];
}


for(int node=0; node<N; node++) {
    int right = node + 1;
    int down = node + L;

    if (right % L == 0 ) right -= L;
    if (down > (N-1)) down -=N;
```

```
            Ebond = Ebond -J* (spin[node]*spin[right] + spin[node]*spin[down]);

            Etotal=Ebond+Efield;

            mag = mag + spin[node];
        }

        free(h);
        free(spin);


        printf("Total flow = %g\n", flow);
        double flow_spin = flow/N;
        printf("Total flow per spin = %g\n", flow_spin);

        printf("Field energy = %g\n", Efield);
        double Efield_spin = Efield/N;
        printf("Field energy per spin = %g\n", Efield_spin);

        printf("Bond energy = %g\n", Ebond);
        double Ebond_spin = (double) Ebond/N;
        printf("Bond energy per spin = %g\n", Ebond_spin);

        printf("Total energy = %g\n", Etotal);
        double Etotal_spin = Etotal/N;
        printf("Total energy per spin = %g\n", Etotal_spin);

        printf("Magnetization = %d\n", mag);
        double mag_spin = mag/pow(L,D);
        printf("Magnetization per spin = %g\n", mag_spin);


        delete g; //delete the graph
        return 0;
}
```

A.5 Code for one realization of the quenched disorder, for a triangular $L \times L$ RFIM with Gaussian random-field distribution, where $L = 4$ and $h = 1$

```
#include <stdio.h>
#include "graph.h"
#include <gsl/gsl_randist.h>
#include <math.h>
#define L 4
#define N (L*L)
#define z 6
#define J 1
#define D 2

using namespace std;

static gsl_rng *rng;


void Init_RNG(int seed) {
    rng=gsl_rng_alloc(gsl_rng_mt19937);
    gsl_rng_set(rng, seed);
}


int main(int argc, char *argv[]){

        int seed=12345;
        Init_RNG(seed);

        double *h;
        h = (double *)malloc(sizeof(double)*N);
        if (h == NULL) {
            printf("Memory not allocated\n");
            exit(0);
        }
```

```
else {
    printf("Memory successfully allocated for the random fields\n");
}

int node_RF;

for(int i=0; i<L; i++) {
    for(int j=0;j<L;j++) {
        node_RF=j+L*i;
        double r=gsl_ran_gaussian(rng, 1);
        h[node_RF]=r;
    }
}

typedef Graph<double,double,double> GraphType;
GraphType *g = new GraphType(N+2, 2*N+N*z/2+1);
for(int i=0; i<N; i++)  g -> add_node();

double w=0,k=0;
double sC=0,tC=0;
double Crightout=0,Crightin=0,Cleftout=0,Cleftin=0,Cdownout=0;
double Cdownin=0,Cupout=0,Cupin=0,Cuprightout=0,Cuprightin=0;
double Cdownleftout=0,Cdownleftin=0;

for(int node=0; node<N; node++) {

    int right = node + 1;
    int left = node - 1;
    int up = node - L;
    int down = node + L;
    int upright = up +1;
    int downleft = down -1;

    Crightout=4*J;
    Crightin=0;
    Cleftout=0;
    Cleftin=4*J;
    Cdownout=4*J;
```

```
Cdownin=0;
Cupout=0;
Cupin=4*J;
Cuprightout=0;
Cuprightin=4*J;
Cdownleftout=4*J;
Cdownleftin=0;

if (right % L == 0 ){
    right -= L;
    upright=right-L;
    if (up < 0){
       upright+=N;
    }
    Crightout=0;
    Crightin=4*J;
}
if (node % L == 0){
    left +=L;
    downleft=left+L;
    if (down > (N-1)){
       downleft-=N;
    }
    Cleftout=4*J;
    Cleftin=0;
}
if (up < 0){
    up +=N;
    upright=up+1;
    if (right % L == 0 ){
       upright-=L;
    }
    Cupout=4*J;
    Cupin=0;
    Cuprightout=4*J;
    Cuprightin=0;
}
if (down > (N-1)){
```

```
            down -=N;
            downleft = down -1;
            if (node % L == 0){
               downleft+=L;
            }
            Cdownout=0;
            Cdownin=4*J;
            Cdownleftout=0;
            Cdownleftin=4*J;
        }

        w=-2*h[node] - (Crightout+Cleftout+Cdownout+Cupout+Cuprightout
        +Cdownleftout-Crightin-Cleftin-Cdownin-Cupin-Cuprightin
        -Cdownleftin)/2;

        if(w > 0) {
          sC=0;
          tC=w;
          }
        else {
          sC=-w;
          tC=0;
          }

        k += sC+tC;

        g -> add_tweights(node, sC, tC );

        g -> add_edge(node, right, Crightout, Crightin);
        g -> add_edge(node, down, Cdownout, Cdownin);
        g -> add_edge(node, upright, Cuprightout, Cuprightin);

        }

    double oC=-N*z/2-k/2;

    double flow = g -> maxflow();
    flow += oC;
```

```
double Efield=0;
double Ebond=0;
double mag = 0;
double Etotal = 0;

double *spin;
spin = (double *)malloc(sizeof(double)*N);
if (spin == NULL) {
    printf("Memory not allocated\n");
    exit(0);
}
else {
    printf("Memory successfully allocated for the spins\n");
}

for(int node=0; node<N; node++) {

    if (g->what_segment(node) == GraphType::SOURCE){
        spin[node]=1;
    }
    else{
        spin[node]=-1;
    }


    Efield = Efield -h[node] * spin[node];
}

for(int node=0; node<N; node++) {

    int right = node + 1;
    int up = node - L;
    int down = node + L;
    int upright = up +1;

    if (right % L == 0 ){
```

```
            right -= L;
            upright=right-L;
            if (up < 0){
               upright+=N;
            }
        }
        if (up < 0){
            up +=N;
            upright=up+1;
            if (right % L == 0 ){
               upright-=L;
            }
        }
        if (down > (N-1)){
            down -=N;
        }


        Ebond = Ebond -J* (spin[node]*spin[right] + spin[node]*spin[down]
        +spin[node]*spin[upright]);

        Etotal=Ebond+Efield;

        mag = mag + spin[node];
    }


    free(h);
    free(spin);


    printf("Total flow = %g\n", flow);
    double flow_spin = flow/N;
    printf("Total flow per spin = %g\n", flow_spin);

    printf("Field energy = %g\n", Efield);
    double Efield_spin = Efield/N;
    printf("Field energy per spin = %g\n", Efield_spin);
```

```
        printf("Bond energy = %g\n", Ebond);
        double Ebond_spin = (double) Ebond/N;
        printf("Bond energy per spin = %g\n", Ebond_spin);

        printf("Total energy = %g\n", Etotal);
        double Etotal_spin = Etotal/N;
        printf("Total energy per spin = %g\n", Etotal_spin);

        printf("Magnetization = %d\n", mag);
        double mag_spin = mag/pow(L,D);
        printf("Magnetization per spin = %g\n", mag_spin);

        delete g;
        return 0;
}
```

A.6 Code for multiple realizations of the quenched disorder, where the graph is rebuilt for each one of them, for a square $L \times L$ RFIM with Gaussian random-field distribution, where $L = 4$

```
#include <stdio.h>
#include "graph.h"
#include <gsl/gsl_randist.h>
#include <math.h>
#define L 4
#define N (L*L)
#define z 4
#define J 1
#define D 2  /
#define statesNo 10000
#define delta 1.0

using namespace std;
```

```
static gsl_rng *rng;


void Init_RNG(int seed) {
    rng=gsl_rng_alloc(gsl_rng_mt19937);
    gsl_rng_set(rng, seed);
}


int main(int argc, char *argv[]){

        double *h;
        h = (double *)malloc(sizeof(double)*N);

        double *spin;
        spin = (double *)malloc(sizeof(double)*N);

        FILE *Thermodynamic_quantities;
        char filename1[64];
        sprintf(filename1,"results/L_%d_h_%g_statesNo_%d_
        Thermodynamic_quantities.dat", L, delta, statesNo);

        Thermodynamic_quantities = fopen(filename1, "w");

        if (Thermodynamic_quantities == NULL) {
            printf("I couldn't open L_%d_h_%g_statesNo_%d_Thermodynamic
            _quantities.dat for writing.\n");
            exit(0);
        }

        FILE *Disorder_average;
        char filename2[64];
        sprintf(filename2, "results/L_%d_h_%g_statesNo_%d_Disorder_average.dat",
        L, delta, statesNo);

        Disorder_average = fopen(filename2, "w");

        if (Disorder_average == NULL) {
```

```
   printf("I couldn't open L_%d_h_%g_statesNo_%d_Disorder_average.dat
   for writing.\n");
   exit(0);
}

int seed=12345;
Init_RNG(seed);

double abs_mag_spin_AVE=0;
double mag2_spin_AVE=0;
double mag4_spin_AVE=0;
double Efield_spin_AVE=0;
double Ebond_spin_AVE=0;
double flow_spin_AVE=0;

for (int i=0; i<statesNo; i++){

        for(int i=0; i<L; i++) {
            for(int j=0;j<L;j++) {
                int node_RF=j+L*i;
                double r=gsl_ran_gaussian(rng, delta);
                h[node_RF]=r;
            }
        }


        typedef Graph<double,double,double> GraphType;
        GraphType *g = new GraphType(N+2, 2*N+N*z/2+1);
        for(int i=0; i<N; i++)  g -> add_node();


        double w=0,k=0;
        double sC=0,tC=0;
        double Crightout=0,Crightin=0,Cleftout=0,Cleftin=0,Cdownout=0;
        double Cdownin=0,Cupout=0,Cupin=0;

        for(int node=0; node<N; node++) {
```

```
int right = node + 1;
int left = node - 1;
int up = node - L;
int down = node + L;

Crightout=4*J;
Crightin=0;
Cleftout=0;
Cleftin=4*J;
Cdownout=4*J;
Cdownin=0;
Cupout=0;
Cupin=4*J;

if (right % L == 0 ){
   right -= L;
   Crightout=0;
   Crightin=4*J;
}
if (node % L == 0){
   left +=L;
   Cleftout=4*J;
   Cleftin=0;
}
if (up < 0){
   up +=N;
   Cupout=4*J;
   Cupin=0;
}
if (down > (N-1)){
   down -=N;
   Cdownout=0;
   Cdownin=4*J;
}

w=-2*h[node] - (Crightout+Cleftout+Cdownout+Cupout
-Crightin-Cleftin-Cdownin-Cupin)/2;
```

```
   if(w > 0) {
     sC=0;
     tC=w;
   }
   else {
     sC=-w;
     tC=0;
   }

   k += sC+tC;

   g -> add_tweights(node, sC, tC );


   g -> add_edge(node, right, Crightout, Crightin);

   g -> add_edge(node, down, Cdownout, Cdownin);

   }

double oC=-N*z/2-k/2;


double flow = g -> maxflow();
flow += oC;


double Efield=0;
int Ebond=0;
double Etotal=0;
double mag = 0;

for(int node=0; node<N; node++) {

   if (g->what_segment(node) == GraphType::SOURCE){
      spin[node]=1;
   }
   else{
```

```
      spin[node]=-1;
   }


   Efield = Efield -h[node] * spin[node];
}

for(int node=0; node<N; node++) {
   int right = node + 1;
   int down = node + L;

   if (right % L == 0 ) right -= L;
   if (down > (N-1)) down -=N;


   Ebond = Ebond -J* (spin[node]*spin[right]
   + spin[node]*spin[down]);

   Etotal=Ebond+Efield;

   mag = mag + spin[node];
}


double flow_spin = flow/N;
printf("Total flow per spin = %g\n", flow_spin);

double Efield_spin = Efield/N;
printf("Field energy per spin = %g\n", Efield_spin);

double Ebond_spin = (double) Ebond/N;
printf("Bond energy per spin = %g\n", Ebond_spin);

double Etotal_spin = Etotal/N;
printf("Total energy per spin = %g\n", Etotal_spin);

double mag_spin = mag/pow(L,D);
printf("Magnetization per spin = %g\n", mag_spin);
```

```
    double abs_mag_spin = fabs (mag_spin);
    printf("Absolute value of magnetization per spin = %g\n",
    abs_mag_spin);

    double mag2_spin =pow(mag_spin,2);
    printf("2nd magnetic moment per spin = %f\n", mag2_spin);
    double mag4_spin = pow(mag2_spin,2);
    printf("4th magnetic moment per spin = %f\n", mag4_spin);

    fprintf(Thermodynamic_quantities, "%d, %g, %g, %g, %g, %g, %g\n",
    i, abs_mag_spin, mag2_spin, mag4_spin, Efield_spin, Ebond_spin,
    flow_spin);

    abs_mag_spin_AVE = abs_mag_spin_AVE + abs_mag_spin/statesNo;
    mag2_spin_AVE = mag2_spin_AVE + mag2_spin/statesNo;
    mag4_spin_AVE = mag4_spin_AVE + mag4_spin/statesNo;
    Efield_spin_AVE = Efield_spin_AVE + Efield_spin/statesNo;
    Ebond_spin_AVE = Ebond_spin_AVE + Ebond_spin/statesNo;
    flow_spin_AVE = flow_spin_AVE + flow_spin/statesNo;



    delete g; //delete the graph

}


fclose(Thermodynamic_quantities);

if (spin == NULL) {
   printf("Memory not allocated\n");
   exit(0);
}
else {
   printf("Memory successfully allocated for the spins\n");
}
if (h == NULL) {
   printf("Memory not allocated\n");
```

```
            exit(0);
         }
         else {
            printf("Memory successfully allocated for the random fields\n");
         }

         free(h);
         free(spin);
         double U_4_spin_AVE = 1 - (mag4_spin_AVE/3*pow(mag2_spin_AVE,2));

         fprintf(Disorder_average, "%g, %g, %g, %g, %g, %g, %g, %g\n", delta,
         abs_mag_spin_AVE, mag2_spin_AVE, mag4_spin_AVE, Efield_spin_AVE,
         Ebond_spin_AVE, flow_spin_AVE, U_4_spin_AVE);

         fclose(Disorder_average);


      return 0;
}
```

A.7 Code for multiple realizations of the quenched disorder, where the graph is built only once, for a square $L \times L$ RFIM with Gaussian random-field distribution, where $L = 4$

```
#include <stdio.h>
#include "graph.h"
#include <gsl/gsl_randist.h>
#include <math.h>
#define L 4
#define N (L*L)
#define z 4
#define J 1
#define D 2
#define statesNo 10000
#define delta 1.0
```

```
using namespace std;
static gsl_rng *rng;


void Init_RNG(int seed) {
    rng=gsl_rng_alloc(gsl_rng_mt19937);
    gsl_rng_set(rng, seed);
}


int main(int argc, char *argv[]){

        typedef Graph<double,double,double> GraphType;
        GraphType *g = new GraphType(N+2, 2*N+N*z/2+1);

        for(int i=0; i<N; i++)  g -> add_node();

        for(int node=0; node<N; node++) {

            int right = node + 1;
            int left = node - 1;
            int up = node - L;
            int down = node + L;

            if (right % L == 0 ){
                right -= L;
            }
            if (node % L == 0){
                left +=L;
            }
            if (up < 0){
                up +=N;
            }
            if (down > (N-1)){
                down -=N;
            }
```

```
    g -> add_tweights(node, 0, 0);

    g -> add_edge(node, right, 0, 0);
    g -> add_edge(node, down, 0, 0);

    }

    typename GraphType::arc_id arc = g->get_first_arc();


double flow = g -> maxflow();


FILE *Thermodynamic_quantities;
char filename1[64];
sprintf(filename1, "results/L_%d_h_%g_statesNo_%d_
Thermodynamic_quantities.dat", L, delta, statesNo);

Thermodynamic_quantities = fopen(filename1, "w");

if (Thermodynamic_quantities == NULL) {
    printf("I couldn't open L_%d_h_%g_statesNo_%d_
    Thermodynamic_quantities.dat for writing.\n");
    exit(0);
}

FILE *Disorder_average;
char filename2[64];
sprintf(filename2, "results/L_%d_h_%g_statesNo_%d_
Disorder_average.dat", L, delta, statesNo);

Disorder_average = fopen(filename2, "w");

if (Disorder_average == NULL) {
   printf("I couldn't open L_%d_h_%g_statesNo_%d_
   Disorder_average.dat for writing.\n");
   exit(0);
}
```

```
int seed=12345;
Init_RNG(seed);

h = (double *)malloc(sizeof(double)*N);
double w=0,k=0;
double sC=0,tC=0;
double Crightout=0,Crightin=0,Cleftout=0,Cleftin=0,Cdownout=0;
double Cdownin=0,Cupout=0,Cupin=0;

double *spin;
spin = (double *)malloc(sizeof(double)*N);

double mag = 0, Efield=0,Etotal=0, mag_spin = 0, mag2_spin =0;
double mag4_spin =0, Efield_spin=0, Ebond_spin=0, Etotal_spin=0;
double abs_mag_spin=0;
int Ebond=0;

double abs_mag_spin_AVE=0, mag2_spin_AVE=0, mag4_spin_AVE=0;
double Efield_spin_AVE=0, Ebond_spin_AVE=0, Etotal_spin_AVE=0;


for (int i=0; i<statesNo; i++){

        for(int i=0; i<L; i++) {
            for(int j=0;j<L;j++) {
                int node_RF=j+L*i;
                double r=gsl_ran_gaussian(rng, delta);
                h[node_RF]=r;
            }
        }


        arc = g->get_first_arc();

        for(int node=0; node<N; node++) {

           int right = node + 1;
```

```
int left = node - 1;
int up = node - L;
int down = node + L;

Crightout=4*J;
Crightin=0;
Cleftout=0;
Cleftin=4*J;
Cdownout=4*J;
Cdownin=0;
Cupout=0;
Cupin=4*J;

if (right % L == 0 ){
   right -= L;
   Crightout=0;
   Crightin=4*J;
}
if (node % L == 0){
   left +=L;
   Cleftout=4*J;
   Cleftin=0;
}
if (up < 0){
   up +=N;
   Cupout=4*J;
   Cupin=0;
}
if (down > (N-1)){
   down -=N;
   Cdownout=0;
   Cdownin=4*J;
}

w=-2*h[node] - (Crightout+Cleftout+Cdownout+Cupout-Crightin
-Cleftin-Cdownin-Cupin)/2;

if(w > 0) {
```

```
      sC=0;
      tC=w;
    }
    else {
      sC=-w;
      tC=0;
    }

    g -> set_trcap(node, sC-tC);


    g->set_rcap(arc, Crightout);

    arc = g->get_next_arc( arc );
    g->set_rcap(arc, Crightin);

    arc = g->get_next_arc( arc );
    g->set_rcap(arc, Cdownout);

    arc = g->get_next_arc( arc );
    g->set_rcap(arc, Cdownin);
    arc = g->get_next_arc( arc );

}

flow = g -> maxflow();

Efield=0; //initialize field energy

for(int node=0; node<N; node++) {

    if (g->what_segment(node) == GraphType::SOURCE){
       spin[node]=1;
    }
    else{
       spin[node]=-1;
    }
```

```
   Efield = Efield -h[node] * spin[node];
}

Ebond=0;
mag = 0;

for(int node=0; node<N; node++) {
   int right = node + 1;
   int down = node + L;

   if (right % L == 0 ) right -= L;
   if (down > (N-1)) down -=N;


   Ebond = Ebond -J* (spin[node]*spin[right] +
   spin[node]*spin[down]);

   Etotal=Ebond+Efield;

   mag = mag + spin[node];
}



Efield_spin = Efield/N;
printf("Field energy per spin = %g\n", Efield_spin);


Ebond_spin = (double) Ebond/N;
printf("Bond energy per spin = %g\n", Ebond_spin);

Etotal_spin = Etotal/N;
printf("Total energy per spin = %g\n", Etotal_spin);


mag_spin = mag/pow(L,D);
printf("Magnetization per spin = %g\n", mag_spin);
```

```
            abs_mag_spin = fabs (mag_spin);
            printf("Absolute value of magnetization per spin = %g\n",
            abs_mag_spin);


            double mag2_spin =pow(mag_spin,2);
            printf("2nd magnetic moment per spin = %f\n", mag2_spin);
            double mag4_spin = pow(mag2_spin,2);
            printf("4th magnetic moment per spin = %f\n", mag4_spin);


            fprintf(Thermodynamic_quantities, "%d, %g, %g, %g, %g, %g, %g\n",
            i, abs_mag_spin, mag2_spin, mag4_spin, Efield_spin, Ebond_spin,
            Etotal_spin);


            abs_mag_spin_AVE = abs_mag_spin_AVE + abs_mag_spin/statesNo;
            mag2_spin_AVE = mag2_spin_AVE + mag2_spin/statesNo;
            mag4_spin_AVE = mag4_spin_AVE + mag4_spin/statesNo;
            Efield_spin_AVE = Efield_spin_AVE + Efield_spin/statesNo;
            Ebond_spin_AVE = Ebond_spin_AVE + Ebond_spin/statesNo;
            Etotal_spin_AVE = Etotal_spin_AVE + Etotal_spin/statesNo;


        }



        fclose(Thermodynamic_quantities);


        if (spin == NULL) {
            printf("Memory not allocated\n");
            exit(0);
        }
        else {
            printf("Memory successfully allocated for the spins\n");
```

```
        }
        if (h == NULL) {
            printf("Memory not allocated\n");
            exit(0);
        }
        else {
            printf("Memory successfully allocated for the random fields\n");
        }


        free(h);
        free(spin);

        double U_4_spin_AVE = 1 - (mag4_spin_AVE/3*pow(mag2_spin_AVE,2));


        fprintf(Disorder_average, "%g, %g, %g, %g, %g, %g, %g, %g\n",
        delta, abs_mag_spin_AVE, mag2_spin_AVE, mag4_spin_AVE, Efield_spin_AVE,
        Ebond_spin_AVE, Etotal_spin_AVE, U_4_spin_AVE);

        fclose(Disorder_average);


        return 0;
}
```

A.8 Code for calculating the ratio r, for a square $L \times L$ RFIM with Gaussian random-field distribution, where $L = 4$, for various disorder strengths h

```
#include <stdio.h>
#include "graph.h"
#include <gsl/gsl_randist.h>
#include <math.h>
//#define L 4
```

```
#define N (L*L)
#define z 4
#define J 1
#define D 2

using namespace std;
static gsl_rng *rng;


void Init_RNG(int seed) {
    rng=gsl_rng_alloc(gsl_rng_mt19937);
    gsl_rng_set(rng, seed);
}

int main(int argc, char *argv[]){

        int L = atoi(argv[1]);
        int statesNo = atoi(argv[2]);

        typedef Graph<double,double,double> GraphType;
        GraphType *g = new GraphType(N+2, 2*N+N*z/2+1);

        for(int i=0; i<N; i++)  g -> add_node();

        for(int node=0; node<N; node++) {

            int right = node + 1;
            int left = node - 1;
            int up = node - L;
            int down = node + L;

            if (right % L == 0 ){
                right -= L;
            }
            if (node % L == 0){
                left +=L;
            }
            if (up < 0){
```

```
            up +=N;
        }
        if (down > (N-1)){
            down -=N;
        }

        g -> add_tweights(node, 0, 0);

        g -> add_edge(node, right, 0, 0);
        g -> add_edge(node, down, 0, 0);

        }

        typename GraphType::arc_id arc = g->get_first_arc();

    double flow = g -> maxflow();

    double a = 1.0;
    double b = 2.0;
    double step = 0.2;

    FILE *Breakup;
    char filename2[64];
    sprintf(filename2, "results/L%d_%dsamples_h[%g,%g]_step%g_Breakup.dat",
    L, statesNo, a, b, step);

    Breakup = fopen(filename2, "w");

    if (Breakup == NULL) {
       printf("I couldn't open L%d_%dsamples_h[%g,%g]_step%g_Breakup.dat
       for writing.\n");
       exit(0);
    }

    for (a; a<= b; a += step){
        double delta = a;

        int seed=12345;
```

```
Init_RNG(seed);

double *h;
h = (double *)malloc(sizeof(double)*N);

double *spin;
spin = (double *)malloc(sizeof(double)*N);

double w=0,k=0;
double sC=0,tC=0;
double Crightout=0,Crightin=0,Cleftout=0,Cleftin=0,Cdownout=0,
Cdownin=0,Cupout=0,Cupin=0;

int count=0;

for (int i=0; i<statesNo; i++){

    for(int i=0; i<L; i++) {
        for(int j=0;j<L;j++) {
            int node_RF=j+L*i;
            double r=gsl_ran_gaussian(rng, delta);
            h[node_RF]=r;
        }
    }

    arc = g->get_first_arc();

    for(int node=0; node<N; node++) {

       int right = node + 1;
       int left = node - 1;
       int up = node - L;
       int down = node + L;

       Crightout=4*J;
       Crightin=0;
       Cleftout=0;
       Cleftin=4*J;
```

```
Cdownout=4*J;
Cdownin=0;
Cupout=0;
Cupin=4*J;

if (right % L == 0 ){
   right -= L;
   Crightout=0;
   Crightin=4*J;
}
if (node % L == 0){
   left +=L;
   Cleftout=4*J;
   Cleftin=0;
}
if (up < 0){
   up +=N;
   Cupout=4*J;
   Cupin=0;
}
if (down > (N-1)){
   down -=N;
   Cdownout=0;
   Cdownin=4*J;
}

w=-2*h[node] - (Crightout+Cleftout+Cdownout+Cupout-Crightin
-Cleftin-Cdownin-Cupin)/2;

if(w > 0) {
  sC=0;
  tC=w;
}
else {
  sC=-w;
  tC=0;
}
```

```
   g -> set_trcap(node, sC-tC);

   g->set_rcap(arc, Crightout);

   arc = g->get_next_arc( arc );
   g->set_rcap(arc, Crightin);

   arc = g->get_next_arc( arc );
   g->set_rcap(arc, Cdownout);

   arc = g->get_next_arc( arc );
   g->set_rcap(arc, Cdownin);
   arc = g->get_next_arc( arc );

}

flow = g -> maxflow();

for(int node=0; node<N; node++) {

   if (g->what_segment(node) == GraphType::SOURCE){
      spin[node]=1;
   }
   else{
      spin[node]=-1;
   }

}


double mag = 0;
for(int node=0; node<N; node++) {
   mag = mag + spin[node];
}

double mag_spin = 0, abs_mag_spin=0;

mag_spin = mag/pow(L,D);
```

```
                abs_mag_spin = fabs (mag_spin);

                if (abs_mag_spin != 1){
                    ++count;
                }



            }

            double ratio= (statesNo - double (count))/ statesNo;
            fprintf(Breakup, "%g, %g\n", delta, ratio);

            if (spin == NULL) {
               printf("Memory not allocated\n");
               exit(0);
            }
            else {
               printf("Memory successfully allocated for the spins\n");
            }
            if (h == NULL) {
               printf("Memory not allocated\n");
               exit(0);
            }
            else { // Memory has been successfully allocated
               printf("Memory successfully allocated for the random fields\n");
            }

            free(h);
            free(spin);

        }

        fclose(Breakup);
        return 0;
}
```

Appendix B

Python Codes

Appx. B includes the content of the `python` codes that we developed for plotting spin configurations of the ground states of many different RFIM examples. The codes read appropriate data files where all the spins (± 1) of a particular ground state for a specific lattice size L, disorder strength h and RNG seed are stored vertically.

B.1 Code for a square $L \times L$ RFIM

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm

FigureName=['2048','1', '12345']

data = np.loadtxt('SQUARE_GS_spins_L_{0}_h_{1}_seed_{2}_HOR.dat'
.format(FigureName[0],FigureName[1],FigureName[2]))

Nx,Ny = np.shape(data)
x = np.arange(0,Nx,1)

fig, axes = plt.subplots(figsize=(3,3))

axes.imshow(data,vmin=-1,vmax=1,cmap=cm.Greys_r)

plt.title('SQUARE \n L={0},h={1},seed={2}'.format(FigureName[0]
,FigureName[1],FigureName[2]),y=1.08)

axes.set_yticklabels([])
```

```
axes.set_xticklabels([])

fig.tight_layout()

plt.savefig('SQUARE_CONF_L{0}_h{1}_seed{2}.pdf'.format(FigureName[0],
FigureName[1],FigureName[2]))
```

B.2 Code for a triangular $L \times L$ RFIM

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import cm

FigureName=['4096','1', '12345']

data = np.loadtxt('TRIANGULAR_GS_spins_L_{0}_h_{1}_seed_{2}_HOR.dat'
.format(FigureName[0],FigureName[1],FigureName[2]))

Nx,Ny = np.shape(data)
x = np.arange(0,Nx,1)

fig, axes = plt.subplots(figsize=(3,3))

axes.imshow(data,vmin=-1,vmax=1,cmap=cm.Greys_r)

.title('TRIANGULAR \n L={0},h={1},seed={2}'.format(FigureName[0],FigureName[1]
,FigureName[2]),y=1.08)

axes.set_yticklabels([])
axes.set_xticklabels([])

#axes.set_xticks(x)
#axes.set_yticks(x)

fig.tight_layout()

plt.savefig('TRIANGULAR_CONF_L{0}_h{1}_seed{2}.pdf'.format(FigureName[0]
,FigureName[1],FigureName[2]))
```

Appendix C

Jackknife Codes

In Appx. C, the codes for performing the Jackknife analysis of our results from our r definition approach regarding the breakup length problem of the square and triangular $2D$ RFIM are included. The codes read appropriate data files which contain the results of our simulations and are named according to specific lattice sizes, disorder strengths, number of samples and data types.

C.1 Code for calculating the ratio r for 100 different bins, for $L = 8$

```
#include <stdio.h>
#include <stdlib.h>
#define N 100000
#define binsNo 100
#define L 8
#define delta 2.375

int main() {

int binsSize = N/binsNo;

FILE * ALLSTATES;
ALLSTATES = fopen("ALLstates_2.375_L8.dat","r");

if (ALLSTATES == NULL) {
    printf("Can't open/find file\n");
    return 1;
}
```

```
FILE *HUNDREDPLOTS;
char filename3[64];
sprintf(filename3, "Hundred_plots/L%d_h%g_states%d_100plots.dat", L, N, delta);

HUNDREDPLOTS = fopen(filename3, "w");
if (HUNDREDPLOTS == NULL) {
    printf("I couldn't open L%d_h%g_states%d_100plots.dat for writing.\n");
    exit(0);
}

fprintf(HUNDREDPLOTS, "%f\t", delta);  //print delta to the file

double *magnet= (double *) malloc(sizeof(double)*N);
double *E_bond= (double *) malloc(sizeof(double)*N);
double *E_field= (double *) malloc(sizeof(double)*N);
double *FM= (double *) malloc(sizeof(double)*N);

double sumALL=0;

for (int i = 0; i<N ; i++) {
fscanf(ALLSTATES, "%lf, %lf, %lf, %lf", &magnet[i], &E_bond[i],
&E_field[i], &FM[i] );

sumALL = sumALL + FM[i];

}

fclose(ALLSTATES);

double ratioALL=0;
ratioALL = sumALL/N;
printf("\nSum of ALL ordered states is %g out of %d and the ratio is %f\n\n",
sumALL , N, ratioALL);

double ratioBIN = 0;

int b1=0;
```

```
int b2 = binsSize - 1;

for (int bin=0; bin < binsNo; bin++) {

    double sumBIN = sumALL;

    for (int i=0; i<N; i++){
        if (i>=b1 && i<=b2){
            sumBIN = sumBIN - FM[i];
        }
    }

    ratioBIN= sumBIN/ (N - binsSize);


    if (bin < binsNo -1) fprintf(HUNDREDPLOTS, "%f\t", ratioBIN);


    b1 = b1 + binsSize;
    b2 = b2 + binsSize;

}

fprintf(HUNDREDPLOTS, "%f\n", ratioBIN);


if (FM == NULL) {
    printf("Memory not allocated\n");
    exit(0);
}
else {
    printf("Memory successfully allocated for the FM states\n");
}


free(FM);

fclose(HUNDREDPLOTS);
```

```
return 0;
}
```

C.2 Code for performing 100 simultaneous fits for each of the 100 plots corresponding to a bin, for $L = 8$

```
gnuplot
N=101

do for [i=2:N] {
   a=1;b=1;
   f(x) = a*x + b
   fit f(x) 'ALLdeltas_L8_100plots.dat' u 1:i via a,b

   set print "parameters.txt" append
   print a,b
   reset
}
```

C.3 Code for calculating the 100 solutions occurring from the 100 fits performed in our results, for $L = 8$

```
#include <stdio.h>
#include <stdlib.h>
#define N 100
#define L 8

int main() {

FILE * PARAMETERS;
PARAMETERS = fopen("parameters.txt","r");

if (PARAMETERS == NULL) {
```

```
    printf("Can't open/find file\n");
    return 1;
}


FILE *HUNDREDSOLUTIONS;
char filename3[64];
sprintf(filename3, "L%d_100solutions_linearfit.dat", L);

HUNDREDSOLUTIONS = fopen(filename3, "w");
if (HUNDREDSOLUTIONS == NULL) {
    printf("I couldn't open L%d_100solutions_linearfit.dat for writing.\n");
    exit(0);
}


double *alpha= (double *) malloc(sizeof(double)*N);
double *beta= (double *) malloc(sizeof(double)*N);


for (int i = 0; i<N ; i++) {
fscanf(PARAMETERS, "%lf %lf", &alpha[i], &beta[i]);

}

fclose(PARAMETERS);

double x=0;
for (int i=0; i<N; i++){
    x = (0.5 - beta[i])/alpha[i];
    fprintf(HUNDREDSOLUTIONS, "%f\n", x);
    x=0;
    }

if (alpha == NULL) {
    printf("Memory not allocated\n");
    exit(0);
}
```

```
else {
    printf("Memory successfully allocated for the a parameter\n");
}
if (beta == NULL) {
    printf("Memory not allocated\n");
    exit(0);
}
else {
    printf("Memory successfully allocated for the b parameter\n");
}


free(alpha);
free(beta);


fclose(HUNDREDSOLUTIONS);

return 0;
}
```

C.4 Code for the Jackknife analysis of the 100 solutions for the breakup field $h_{\mathrm{B}}(L)$, for $L = 8$

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#define N 100
#define L 8


int main() {

FILE * SOLUTIONS;
SOLUTIONS = fopen("L8_100solutions_linearfit.dat","r");
```

```
if (SOLUTIONS == NULL) {
    printf("Can't open/find file\n");
    return 1;
}

FILE *FINALRESULT;
char filename3[64];
sprintf(filename3, "L%d_JKresults_linearfit.dat", L);

FINALRESULT = fopen(filename3, "w");
if (FINALRESULT == NULL) {
    printf("I couldn't open L%d_JKresults_linearfit.dat for writing.\n");
    exit(0);
}

double *HundredSolutions= (double *) malloc(sizeof(double)*N);

for (int i = 0; i<N ; i++) {
fscanf(SOLUTIONS, "%lf", &HundredSolutions[i]);
//printf("%f\n", HundredSolutions[i]);
}

fclose(SOLUTIONS);

fprintf(FINALRESULT, "%d\t", L);

double JKsum, JKvarsum;
double JKmean=0;
double JKvar, JKerror;


JKsum = HundredSolutions[0];
JKvarsum = HundredSolutions[0]*HundredSolutions[0];
for(int i=1; i<N; i++) {
    JKsum = JKsum + HundredSolutions[i];
    JKvarsum = JKvarsum + HundredSolutions[i]*HundredSolutions[i];
}
```

```
JKmean= JKsum/N;
printf("The final JK estimation is %f\n", JKmean);
fprintf(FINALRESULT, "%f\t", JKmean);

printf("The sum of the squares of the 100 solutions is %f\n", JKvarsum);
JKvar = JKvarsum/N - pow(JKmean,2);



JKerror = sqrt(JKvar)*sqrt(N-1);
printf("The final JK error is %f\n", JKerror);
fprintf(FINALRESULT, "%f\n", JKerror);


if (HundredSolutions == NULL) {
    printf("Memory not allocated\n");
    exit(0);
}
else {
    printf("Memory successfully allocated for the 100 solutions\n");
}

free(HundredSolutions);


fclose(FINALRESULT);

return 0;
}
```

Bibliography

Abraham, D. B. and E. R. Smith (1982). In: *Phys. Rev. B* 26, p. 1480 (cit. on p. 39).

Abraham, D. B. (1986). *Phase Transitions and Critical Phenomena*. Edited by C. Domb and J. L. Lebowitz. (Academic Press, New York), Vol. 10 (cit. on p. 32).

Aharony, A. and M. E. Fisher (1983). In: *Phys. Rev. B* 27, p. 4394 (cit. on p. 11).

Aharony, A. (1978). In: *Phys. Rev. B* 18, p. 3318 (cit. on pp. 19, 20).

Aharony, A., Y. Imry and S. .-k. Ma (1976). In: *Phys. Rev. Lett.* 37, p. 1364 (cit. on p. 17).

Ahrens, B. and A. Hartmann (2011). In: *Physical Review B* 83, p. 014205 (cit. on pp. 20, 59).

Alava, M. J., P. M. Duxbury, C. F. Moukarzel and H. Rieger (2001). In: *Phase Transitions and Critical Phenomena*. Ed. by C. Domb and J. L. Lebowitz (cit. on pp. 58, 59).

Albano, E. V. and K. Binder (2012). In: *Phys. Rev. E* 85, p. 061601 (cit. on p. 33).

— (2014). In: *J. Stat. Phys.* 157, p. 436 (cit. on p. 33).

Amit, D. J. and V. Martín-Mayor (2005). *Field Theory, the Renormalization Group and Critical Phenomena*. third. Singapore: World Scientific. DOI: 10.1142/9789812775313_bmatter. URL: http://www.worldscientific.com/worldscibooks/10.1142/5715 (cit. on p. 94).

Anglés d' Auriac, J. .-C. and N. Sourlas (1997). In: *Europhys. Lett.* 39, p. 473 (cit. on pp. 20, 58).

Anglés d'Auriac, J. .-C. (1986). PhD thesis. Centre de Recherches sur les Très Basses Tempèratures, Grenoble, France (cit. on p. 58).

Anglés d'Auriac, J. .-C., M. Preissmann and R. Rammal (1985). In: *J. Physique Lett.* 46, p. 173 (cit. on p. 58).

Annunziata, M. A. and A. Pelissetto (2012). In: *Phys. Rev. E* 86, p. 041804 (cit. on p. 17).

Barber, M. N. (1983). In: *Phase Transitions and critical Phenomena*. Ed. by C. Domb and J. L. Lebowitz (cit. on pp. 2, 8, 10, 11).

Barber, W. and D. Belanger (2001). In: *J. Magn. Magn. Mater.* 226-230, p. 545 (cit. on p. 17).

Bastea, S. and P. M. Duxbury (1998). In: *Phys. Rev. E* 58, p. 4261 (cit. on p. 58).

Belanger, D. and A. Young (1991). In: *J. Magn. Magn. Matter* 100, p. 272 (cit. on pp. 16, 17).

Belanger, D. P., A. R. King, V. Jaccarino and J. L. Cardy (1983). In: *Phys. Rev. B* 28, p. 2522 (cit. on p. 17).

Belanger, D. P. and T. Nattermann (1998). In: *Spin Glasses and Random Fields.* Ed. by A. P. Young (cit. on pp. 1, 2, 13, 16, 17).

Beretti, A. (1985). In: *Phys. Rev. Lett.* 55, p. 2499 (cit. on p. 18).

Berker, A. N. and S. R. McKay (1986). In: *Phys. Rev. B* 33, p. 4712 (cit. on p. 19).

Binder, K. (1981a). In: *Z. Phys. B* 43, p. 119 (cit. on p. 12).

— (1981b). In: *Phys. Rev. Lett.* 47, p. 693 (cit. on p. 12).

— (1983). In: *Z. Phys. B Condensed Matter* 50, pp. 343–352 (cit. on pp. 87, 88).

Blöte, H. W. J., E. Luijten and J. R. Heringa (1995). In: *J. Phys. A: Math. Gen.* 28, p. 6289 (cit. on p. 37).

Blume, M. (1966). In: *Phys. Rev.* 141, p. 517 (cit. on p. 35).

Bonn, D., J. Eggers, J. Indekeu, J. Meunier and E. Rolley (2009). In: *Rev. Mod. Phys.* 81, p. 739 (cit. on p. 32).

Boykov, Y. and V. Kolmogorov (2001). In: International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR). Sophia Antipolis, France: Springer-Verlag, pp. 359–374 (cit. on p. 61).

— (2004). In: *IEEE Transactions on PAMI* 26, pp. 1124–1137 (cit. on pp. 61–64, 82, 85, 103).

Boykov, Y., O. Veksler and R. Zabih (2001). In: *IEEE T. Pattern Anal.* 23, p. 1222 (cit. on p. 88).

Bray, A. J. and M. A. Moore (1985a). In: *J. Phys. C* 18, p. L927 (cit. on pp. 19, 87, 88).

— (1985b). In: *Phys. Rev. B* 31, p. 631 (cit. on p. 58).

Brener, A. (2010). PhD thesis. RWTH Aachen University (cit. on p. 33).

Bricmont, J. and A. Kupiainen (1987). In: *Phys. Rev. Lett.* 59, p. 1829 (cit. on p. 21).

Bricmont, J. and J. L. Lebowitz (1987). In: *J. Stat. Phys.* 46, p. 1015 (cit. on pp. 18, 19, 33).

Burgy, J., M. Mayr, V. Martin-Mayor, A. Moreo and E. Dagotto (2001). In: *Phys. Rev. Lett.* 87, p. 277202 (cit. on p. 17).

Capel, H. W. (1966). In: *Physica (Amsterdam)* 32, p. 966 (cit. on p. 35).

Cardy, J. (1985). In: *Phys. Rev. B* 29, p. 505 (cit. on p. 17).

— (1996). *Scaling and Renormalization in Statistical Physics*. Cambridge Univ. Press, Cambridge (cit. on pp. 1, 3–5).

— (2000). In: *Nucl. Phys. B* 565, p. 506 (cit. on p. 33).

Cardy, J. and L. Jacobsen (1997). In: *Phys. Rev. Lett.* 79, p. 4063 (cit. on p. 17).

Cardy, J. L. (1984). In: *Phys. Rev. B* 29, p. 505 (cit. on p. 17).

Carlon, E., F. Igloi, W. Selke and F. Szalma (1999). In: *J. Stat. Phys.* 96, p. 531 (cit. on p. 33).

Chayes, J. T., L. Chayes, D. S. Fisher and T. Spencer (1986). In: *Phys. Rev. Lett.* 57, p. 2999 (cit. on pp. 15, 16).

Cherkassky, B. V. and A. V. Goldberg (1997). In: *Algorithmica* 19, pp. 390–410 (cit. on pp. 60, 82).

Coddington, P. D. and C. F. Ballie (1992). In: *Phys. Rev. Lett.* 68, p. 962 (cit. on p. 31).

Cooper, F., B. Freedman and D. Preston (1982). 'Solving $\phi^4_{1,2}$ field theory with Monte Carlo'. In: *Nucl. Phys. B* 210, p. 210. DOI: 10.1016/0550-3213(82)90240-1 (cit. on p. 94).

Cormen, T. H., C. E. Leiserson and R. L. Rivest (1990). *Introduction To Algorithms*. 1st ed. MIT Press, Cambridge (cit. on pp. 58, 60).

Dagotto, E. (2005). In: *Science* 309, p. 257 (cit. on p. 17).

Dayan, I., M. Schwartz and A. P. Young (1992). In: *J. Phys. A* 26, p. 3093 (cit. on p. 20).

Delfino, G. (2016). In: *J. High Energy Phys.* 05, p. 032 (cit. on p. 33).

Delfino, G., W. Selke and A. Squarcini (2018). In: *J. Stat. Mech.*, p. 053203 (cit. on p. 33).

Delfino, G. and A. Squarcini (2013). In: *J. Stat. Mech.*, P05010 (cit. on p. 33).

— (2014a). In: *Phys. Rev. Lett.* 113, p. 066101 (cit. on p. 33).

— (2014b). In: *Ann. Phys. (N. Y.)* 342, p. 171 (cit. on p. 33).

— (2015). In: *Nucl. Phys. B* 901, p. 430 (cit. on p. 33).

— (2016). In: *J. High Energy Phys.* 11, p. 119 (cit. on p. 33).

Delfino, G. and J. Viti (2012). In: *J. Stat. Mech.*, P10009 (cit. on p. 33).

Dietrich, S. (1988). *Phase Transitions and Critical Phenomena*. Edited by C. Domb and J. L. Lebowitz. (Academic Press, New York), Vol. 12 (cit. on p. 32).

Dinic, E. A. (1970). In: *Soviet Math. Dokl.* 11, p. 1277 (cit. on pp. 60, 82).

Dotsenko, V. S. (2007). In: *J. Stat. Mech.*, P09005 (cit. on p. 17).

Dukovski, I. and J. Machta (2003). In: *Phys. Rev. B* 67, p. 014413 (cit. on p. 58).

Duxbury, P. M. and J. H. Meinke (2001). In: *Phys. Rev. E* 64, p. 036112 (cit. on p. 58).

Edmonds, J. and R. M. Karp (1972). In: *J. ACM* 19, p. 248 (cit. on pp. 60, 82).

Esser, J., U. Nowak and K. D. Usadel (1997). In: *Phys. Rev. B* 55, p. 5866 (cit. on p. 17).

Fernández, L. A., A. Gordillo-Guerrero, V. Martin-Mayor and J. J. Ruiz-Lorenzo (2008a). In: *Phys. Rev. B* 86, p. 184428 (cit. on p. 17).

— (2008b). In: *Phys. Rev. Lett.* 100, p. 057201 (cit. on p. 17).

Ferrenberg, A. M. and D. P. Landau (1991). In: *Phys. Rev. B* 44, p. 5081 (cit. on pp. 11, 12).

Fisher, D. S. (1986). In: *Phys. Rev. Lett.* 56, p. 416 (cit. on p. 19).

— (1995). In: *Phys. Rev. B* 51, p. 6411 (cit. on p. 16).

Fisher, D. S., G. M. Grinstein and A. Kuharana (1988). 'Theory of Random Magnets'. In: *Physics Today* 41, p. 56 (cit. on pp. 1, 13, 16).

Fisher, D. S. and D. A. Huse (1986). In: *Phys. Rev. Lett.* 56, p. 1601 (cit. on p. 58).

Fisher, M. E. (1984). In: *J. Stat. Phys.* 34, p. 667 (cit. on pp. 32, 39).

Fishman, S. and A. Aharony (1979). In: *J. Phys. C* 12, p. 729 (cit. on p. 17).

Ford, L. R. and D. R. Fulkerson (1956). In: *Canadian J. Math* 8, pp. 399–404 (cit. on pp. 47, 59, 82).

— (1957). In: *Canadian J. Math* 9, pp. 210–218 (cit. on p. 47).

Fytas, N. G., A. Mainou, P. E. Theodorakis and A. Malakis (2019). In: *Phys. Rev. E* 99, p. 012111 (cit. on p. 32).

Fytas, N. G., A. Mainou and M. Weigel (2021). In preparation (cit. on p. 100).

Fytas, N. G. and A. Malakis (2008). In: *Eur. Phys. J. B* 61, p. 111 (cit. on p. 19).

Fytas, N. G., A. Malakis, W. Selke and L. N. Shchur (2015). In: *Eur. Phys. J B* 88, p. 204 (cit. on p. 33).

Fytas, N. G. and V. Martín-Mayor (2013a). In: *Phys. Rev. E* 93, p. 063308 (cit. on p. 21).

— (2013b). In: *Phys. Rev. Lett.* 110, p. 22720 (cit. on pp. 21, 58).

— (2016). In: *Phys. Rev. E* 93, p. 06330 (cit. on pp. 17, 19, 60, 82, 84, 93, 94, 98).

Fytas, N. G., V. Martín-Mayor, G. Parisi, M. Picco and N. Sourlas (2019). In: *Phys. Rev. Lett.* 122, p. 240603 (cit. on p. 21).

Fytas, N. G., V. Martín-Mayor, M. Picco and N. Sourlas (2016). In: *Phys. Rev. Lett.* 116, p. 227201 (cit. on p. 21).

— (2017a). In: *J. Stat. Mech.*, p. 033302 (cit. on p. 21).

— (2017b). In: *Physical Review E* 95, p. 042117 (cit. on p. 21).

Fytas, N. G. and W. Selke (2013). In: *Eur. Phys. J. B* 86, p. 365 (cit. on p. 33).

Fytas, N. G., P. E. Theodorakis and A. Malakis (2017). In: *Phys. Rev. E* 95, p. 032126 (cit. on pp. 33, 40, 42).

Fytas, N. G., J. Zierenberg et al. (2018). In: *Phys. Rev. E* 97, 040102(R) (cit. on pp. 34, 42).

Gennes, P. G. de (1984). In: *J. Chem. Phys.* 88, p. 6469 (cit. on p. 17).

Gofman, M., J. Adler, A. Aharony, A. B. Harris and M. Schwartz (1992). In: *Phys. Rev. Lett.* 71, p. 1569 (cit. on p. 17).

Goldberg, A. V. and R. E. Tarjan (1988). In: *J. ACM* 35, p. 921 (cit. on pp. 60, 82, 88, 93).

Goldenfeld, N. (1995). *Lectures on Phase Transitions and the Renormalization Group*. Addison-Wesley, Reading, Massachussets (cit. on pp. 1, 4, 5, 10).

Graham, J. T., M. Maliepaard, J. H. Page, S. R. P. Smith and D. R. Taylor (1987). In: *Phys. Rev. B* 35, p. 2098 (cit. on p. 17).

Greig, D. M., B. T. Porteous and A. H. Seheult (1989). In: *JRSS* 51, pp. 271–279 (cit. on p. 47).

Grinstein, G. (1976). In: *Phys. Rev. Lett.* 37, p. 1976 (cit. on p. 18).

Grinstein, G. and S.-K. Ma (1983). In: *Phys. Rev. B* 28, p. 2588 (cit. on p. 18).

Gros, C. (2009). *Complex and Adaptive Dynamical Systems: A Primer*. Springer Publishing Company, Incorporated (cit. on p. 49).

Guggenheim, E. A. (1945). 'The Principle of Corresponding States'. In: *J. Chem. Phys.* 13, p. 253 (cit. on p. 1).

Hammer, P. L. (1965). In: *Operations Research* 13, pp. 388–399 (cit. on pp. 48, 50).

Hammer, P. L., I. Rosenberg and S. Rudeanu (1968). In: *Stud. Cerc. Mat.* 14, pp. 359–364 (cit. on p. 48).

Harris, A. B. (1974). In: *J. Phys. C* 7, p. 1671 (cit. on p. 14).

Hartmann, A. and U. Nowak (1999). In: *Eur. Phys. J. B* 7, p. 105 (cit. on pp. 20, 58).

Hartmann, A. and H. Rieger (2004). *Optimization Algorithms in Physics*. John wiley & Sons (cit. on pp. 2, 52, 53).

Hartmann, A. and K. Usadel (1995). In: *Physica A: Statistical Mechanics and its Applications* 214, p. 141 (cit. on p. 58).

Hartmann, A. and A. P. Young (2001). In: *Phys. Rev. B* 64, p. 214419 (cit. on p. 58).

Hartmann, A. K. (2004). *Random-field systems, spin glasses and vertex covers: on the relation of statistical physics and combinatorial optimization*. Kumulative

Habilitationsschrift, Institut für Theoretische Physic, Universität Göttingen (cit. on pp. 13, 14).

Hasenbusch, M. (2010). In: *Phys. Rev. B* 82, p. 174433 (cit. on p. 37).

Hayden, L. X., A. Raju and J. P. Sethna (Oct. 2019). 'Unusual scaling for two-dimensional avalanches: Curing the faceting and scaling in the lower critical dimension'. In: *Phys. Rev. Research* 1 (3), p. 033060 (cit. on p. 89).

Heller, P. and G. B. Benedek (1962). 'Nuclear Magnetic Resonance in MnF_2 Near the Critical Point'. In: *Phys. Rev. Lett.* 8 (11), pp. 428–432 (cit. on p. 1).

Hernández, L. and H. Ceva (2008). In: *Physica A* 387, p. 2793 (cit. on p. 20).

Hohenberg, P. C. and B. Halperin (1977). In: *Rev. Mod. Phys.* 49, p. 435 (cit. on p. 44).

Huse, D. A. and D. S. Fisher (1987). In: *Phys. Rev. B* 35, p. 6841 (cit. on p. 16).

Imbrie, J. Z. (1984). In: *Phys. Rev. Lett.* 53, p. 1747 (cit. on pp. 17, 18).

Imry, Y. and S.-K. Ma (1975). 'Random-Field Instability of the Ordered State of Continuous Symmetry'. In: *Phys. Rev. Lett.* 35, pp. 1399–1401 (cit. on pp. 1, 17–19).

Itakura, M. (2001). In: *Phys. Rev. B* 64, p. 012415 (cit. on p. 19).

Kinzel, W. and E. Domany (1981). In: *Phys. Rev. B* 23, p. 3421 (cit. on p. 33).

Kumar, M. et al. (2018). In: *Phys. Rev. E* 97 (cit. on pp. 87, 88).

Kwak, W., J. Jeong, J. Lee and D. .-H. Kim (2015). In: *Phys. Rev. E* 92, p. 022134 (cit. on pp. 33, 36–38).

Landau, D. P. and K. Binder (2000). *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge Univ. Press, Cambridge (cit. on pp. 1, 2).

Lawler, E. (1976). *Combinatorial optimization - networks and matroids*. Holt, Rinehart and Winston (cit. on p. 49).

Lawrie, I. D. and S. Sarbach (1984). In: *Phase Transitions and Critical Phenomena* 9. Ed. by C. Domb and J. L. Lebowitz (cit. on p. 35).

Lipowsky, R., D. M. Kroll and R. Zia (1983). In: *Phys. Rev. B* 27, p. 4499 (cit. on pp. 38, 39).

Machta, J., M. E. J. Newman and L. B. Chayes (2000). In: *Phys. Rev. E* 62, p. 8782 (cit. on p. 19).

Malakis, A., A. N. Berker, N. G. Fytas and T. Papakonstantinou (2012). In: *Phys. Rev. E* 85, p. 061106 (cit. on p. 37).

Malakis, A., A. N. Berker, I. A. Hadjiagapiou and N. G. Fytas (2009). In: *Phys. Rev. E* 79, p. 011125 (cit. on p. 33).

— (2010). In: *Phys. Rev. E* 81, p. 041113 (cit. on pp. 33, 37, 41).

Matsumoto, M. and T. Nishimura (1998). In: *ACM Trans. Model. Comput. Simul.* 8, pp. 3–30 (cit. on p. 68).

Messager, A., S. Miracle-Sole, J. Ruiz and S. Shlosman (1991). In: *Commun. Math. Phys.* 140, p. 275 (cit. on p. 33).

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller (1953). In: *J. Chem. Phys.* 21, p. 1087 (cit. on p. 27).

Middleton, A. A. (2001). In: *Phys. Rev. Lett.* 88, p. 017202 (cit. on pp. 58, 60, 93).

— (2002). 'Scaling, domains, and states in the four-dimensional random field Ising magnet'. In: *arXiv: Disordered Systems and Neural Networks* (cit. on pp. 60, 61).

Middleton, A. A. and D. S. Fisher (2002). In: *Phys. Rev. B* 65, p. 134411 (cit. on pp. 17, 58, 60, 61, 93).

Moldover, M. R. and J. W. Cahn (1980). In: *Science* 207, p. 1073 (cit. on p. 32).

Monthus, C. and T. Garel (2008). In: *Phys. Rev. B* 77, p. 134416 (cit. on p. 33).

Mosquera, D. Y. (2011). PhD thesis. Complutense University of Madrid (cit. on p. 45).

Nattermann, T. (1990). In: *Ferroelectricts* 104, p. 171 (cit. on p. 17).

— (1997). In: *Spin Glasses and Random Fields*. Ed. by A. P. Young (cit. on pp. 18, 19).

Nattermann, T. and J. Villain (1988). In: *Phase Transitions* 11, p. 5 (cit. on pp. 1, 17).

Newman, M. E. J. and G. T. Barkema (1996). In: *Phys. Rev. E* 53, p. 393 (cit. on p. 19).

— (1999). *Monte Carlo Simulations in Statistical Physics*. Claredon Press, Oxford (cit. on pp. 1, 2, 25, 28, 31, 44).

Newman, M. E. J., B. W. Roberts, G. T. Barkema and J. P. Sethna (1992). In: *Phys. Rev. B* 48, p. 16533 (cit. on p. 19).

Nightingale, M. P. and H. W. J. Blöte (1996). In: *Phys. Rev. Lett.* 76, p. 4548 (cit. on p. 29).

Ogielski, A. T. (1986). In: *Phys. Rev. Lett.* 57, p. 1251 (cit. on p. 58).

Papadimitriou, C. H. (1994). *Computational Complexity*. 1st ed. Addison-Wesley, Reading, MA (cit. on pp. 58, 60, 93).

Parisi, G. (1979). In: *Phys. Rev. Lett.* 43, p. 1754 (cit. on p. 17).

Parisi, G. and N. Sourlas (1979). In: *Phys. Rev. Lett.* 43, p. 744 (cit. on p. 21).

— (2002). In: *Phys. Rev. Lett.* 89, p. 257204 (cit. on p. 20).

Perković, O., K. A. Dahmen and J. P. Sethna (1999). In: *Phys. Rev. B* 59, p. 6106 (cit. on p. 17).

Picard, J. C. and H. D. Ratliff (1973). In: *Operations Research* 21, pp. 261–269 (cit. on p. 51).

Picard, J. C. and H. D. Ratliff (1975). In: *Networks* 5, p. 357 (cit. on pp. 48, 51).

Privman, V. (1990). In: *Finite-Size Scaling and Numerical Simulation of Statistical Systems*. Ed. by V. Pivman (cit. on pp. 2, 7–9).

Ralston, J., M. Popescu and R. Sedev (2008). In: *Annu. Rev. Mater. Res.* 38, p. 23 (cit. on p. 32).

Riedel, E. K. (1972). In: *Phys. Rev. Lett.* 28, p. 675 (cit. on p. 40).

Rieger, H. (1995a). In: *Annual Reviews of Computational Physics II*. Ed. by D. Stauffer (cit. on p. 17).

— (1995b). In: *Phys. Rev. B* 52, p. 6659 (cit. on p. 16).

— (1998). *Lectures Notes in Physics*. Springer-Verlag, Heidelberg (cit. on pp. 13, 14, 17).

Schneider, T. and E. Pytte (1977). In: *Phys. Rev. B* 15, p. 1519 (cit. on p. 20).

Schwartz, M. A. and A. Soffer (1985). In: *J. Stat. Phys.* 38, p. 483 (cit. on pp. 17, 94).

Selke, W. and D. A. Huse (1983). In: *Z. Physik B* 50, p. 113 (cit. on pp. 32, 33, 40).

Selke, W., D. A. Huse and D. M. Kroll (1984). In: *J. Phys. A: Math. Gen.* 17, p. 3019 (cit. on pp. 33, 34, 36–40, 43).

Selke, W. and W. Pesch (1982). In: *Z. Physik B* 47, p. 335 (cit. on pp. 32, 40, 41).

Selke, W. and J. M. Yeomans (1983). In: *J. Phys. A: Math. Gen.* 16, p. 2789 (cit. on pp. 33, 34, 36, 38–41).

Seppäla, E. T. and M. J. Alava (2000). In: *Phys. Rev. Lett.* 84, p. 3982 (cit. on p. 48).

— (2001). In: *Phys. Rev. E* 63, p. 066109 (cit. on p. 58).

Seppäla, E. T., M. J. Alava and P. M. Duxbury (2001). In: *Phys. Rev. E* 63, p. 036126 (cit. on p. 48).

Seppäla, E. T., V. Petäjä and M. Alava (1998). In: *Phys. Rev. E* 58 (cit. on pp. 87, 88, 98).

Seppäla, E. T., V. I. Raisanen and M. J. Alava (2000). In: *Phys. Rev. E* 61, p. 6312 (cit. on p. 48).

Sethna, J. P. et al. (1992). In: *Phys. Rev. Lett.* 70, p. 3347 (cit. on p. 17).

Shrivastav, G. P., V. Banerjee and S. Puri (2014). In: *Eur. Phys. J. E* 37, p. 98 (cit. on p. 88).

Shrivastav, G. P., S. Krishnamoorthy, V. Banerjee and S. Puri (2011). In: *EPL (Europhysics Letters)* 96, p. 36003 (cit. on p. 58).

Silevitch, D. M., G. Aeppli and T. F. Rosenbaum (2010). In: *Proc. Natl. Acad. Sci. USA* 107, p. 2797 (cit. on p. 17).

Silva, C. J., A. A. Caparica and J. A. Plascak (2006). In: *Phys. Rev. E* 73, p. 036702 (cit. on p. 33).

Sourlas, N. (1996). In: *Comp. Phys. Commun.* 121, p. 183 (cit. on p. 20).

— ed. (1998). *Computer Physics Communications*. Vol. 121. 183. Europhysics Conference on Computational Physics (cit. on p. 58).

Stanley, H. E. (1971). *Introduction to Phase Transitions and Critical Phenomena*. Oxford Univ. Press, Oxford (cit. on pp. 1, 4, 7–9).

Stevenson, J. D. and M. Weigel (2011). In: *EPL (Europhysics Letters)* 95, p. 40001 (cit. on p. 59).

Swendsen, R. H. and J. .-S. Wang (1987). In: *Phys. Rev. Lett.* 58, p. 86 (cit. on p. 28).

Swift, M. R., A. J. Bray, A. Maritan, M. Cieplak and J. R. Banavar (1997). In: *EPL (Europhysics Letters)* 38, p. 273 (cit. on p. 58).

Swift et al. (1996). (preprint) (cit. on p. 20).

Tarjus, G., I. Balog and M. Tissier (2013). In: *EPL (Europhysics Letters)* 103, p. 61001 (cit. on p. 21).

Tissier, M. and G. Tarjus (2011). In: *Phys. Rev. Lett.* 107, p. 041601 (cit. on p. 21).

— (2012). In: *Phys. Rev. B* 85, p. 104203 (cit. on p. 21).

Trobo, M. L. and E. V. Albano (2014). In: *Eur. Phys. J. B* 87, p. 303 (cit. on p. 33).

Villain, J. (1984). In: *Phys. Rev. Lett.* 52, p. 1543 (cit. on pp. 19, 58).

Vink, R. L. C., K. Binder and H. Löwen (2006). In: *Phys. Rev. Lett.* 97, p. 230603 (cit. on p. 17).

Wiseman, S. and E. Domany (1998a). In: *Phys. Rev. E* 52, p. 3469 (cit. on p. 42).

— (1998b). In: *Phys. Rev. Lett.* 81, p. 22 (cit. on p. 42).

Wolff, U. (1989). In: *Phys. Rev. Lett.* 62, p. 361 (cit. on pp. 28, 30, 37).

Wu, Y. and J. Machta (2005). In: *Phys. Rev. Lett.* 95, p. 137208 (cit. on p. 58).

Yamagata, A. (1991). In: *Z. Phys. B* 84, p. 419 (cit. on p. 33).

Yamagata, A. and K. Kasono (1992). In: *Z. Phys. B* 87, p. 219 (cit. on p. 33).

Yeomans, J. M. (1992). *Statistical Mechanics of Phase Transitions*. Oxford Univ. Press, Oxford (cit. on pp. 1, 4).

Young, A. P. (1977). In: *J. Phys. C* 10, p. 257 (cit. on pp. 17, 21).

Young, P. (2012). *Everything You Wanted to Know About Data Analysis and Fitting but Were Afraid to Ask* (cit. on pp. 89, 92).

Zierenberg, J., N. G. Fytas, M. Weigel, W. Janke and A. Malakis (2017). In: *Eur. Phys. J. Special Topics* 226, p. 789 (cit. on pp. 33, 35).

Zinn-Justin, J. (2005). *Quantum Field Theory and Critical Phenomena*. Fourth edition (Clarendon Press, Oxford (cit. on p. 44).