Università degli Studi di Salerno

Facoltà di scienze matematiche, fisiche e naturali
Corso di Laurea in Fisica

Tesi di Laurea Magistrale

COARSENING AND PERCOLATION IN THE INHOMOGENEOUS ISING MODEL

Relatori:
Dott. Federico Corberi
Dott. Marco Picco
Ch.ma Prof.ssa Leticia Cugliandolo

Correlatore :
Ch.mo Prof. Gaetano Busiello

Candidato:
Ferdinando Insalata
Matricola 0522600038

Anno Accademico 2015/2016

Ad Angelina Moscato

Coarsening and Percolation in the Inhomogeneous Ising Model

Ferdinando Insalata

Indice

Introduction

Coarsening, the ordering of a system via domain growth after a quench from the homogeneous phase into one with broken symmetry, has attracted enormous interest in the last 50 years.

The simplest example is a ferromagnet rapidly cooled from very high temperature to well below the critical point. The system starts in an equilibrium disordered state. This configuration becomes unstable after the cooling and evolves towards one of the two possible low-temperature ordered configurations with opposite magnetizations. Relaxation toward the new equilibrium state occurs slowly (i.e. not exponentially fast) by the formation and growth (coarsening) of domains of aligned spins, of typical size $R(t)$. This problem is studied by the field of *phase ordering kinetics* and one theoretical approach is the kinetic Ising model, originally introduced by Glauber.

One of the reasons of interest in this class of phenomena is that they are found in, essentially, every area of science. Examples from Condensed Matter Physics include binary alloys or liquids, polymer blends and liquid crystals. Besides, coarsening is also believed to have influenced the formation of cosmic structures in the early universe. In social sciences it underlies, for instance, the spreading of opinions, while on the biological side it is associated to the growth of populations in an ecosystem. Many other manifestations of coarsening are also found in chemistry, finance and economy, medical sciences and other fields. This list of examples is far from being exhaustive and witnesses the broad scope of this field of research.

Perhaps, the most distinguishing aspect of coarsening is the scaling symmetry, a central aspect in many areas of Physics. In this context we deal with *dynamical scaling* : the structure of the growing domains looks statistically similar at all times upon measuring lengths in units of $R(t)$. One consequence is that space dependencies in correlation functions appear as ratios between distances and $R(t)$.

Dynamical scaling remains a hypothesis because its validity has been rigorously proven only in very simple models, like for instance the $1d$ Glauber-Ising Model. However, its success in describing experimental and numerical data is impressive. For example, the assumption of dynamical scaling allows

in some cases to predict the functional form of $R(t)$, a fundamental quantity in any problem of this kind.

In the last twenty years it has been shown that the evolution of coarsening systems is influenced by random percolative effects. Using the language of spin systems, we have *percolation* when there exists a cluster of aligned spins crossing the lattice. Random percolation theory, which describes the properties of such clusters in a system of non-interacting spins, is a purely geometrical problem. Therefore, its connection to phase ordering kinetics, where interactions among spins play a prominent role, is surprising and interesting in its own. Moreover, a recent proof of dynamical scaling, valid for an important class of two-dimensional models, relies on the empirical observation that the domain morphology of a coarsening system in its early stages exhibits the topological properties of the random percolation clusters. This percolating structure appears after a relatively short transient, fully determining the later evolution and the final state of the system. Interestingly enough, this fact introduces another characteristic length in the system, requiring a generalization of the usual dynamical scaling framework formulated in terms of the sole $R(t)$.

The discussion above refers to pure systems, which can be modeled by the homogeneous Ising Model, for which coarsening has been widely studied and is quite well understood. Real systems, instead, are often inhomogeneous; in particular they are often characterized by the presence of quenched randomness, sources of inhomogeneity whose properties do not vary appreciably over the time scales of a typical experiment.

The general motivation behind this work has been to establish if the aforementioned correspondences between percolation and coarsening holds also for inhomogeneous models. This question is of paramount importance for a better understanding of the scaling properties of real systems characterized by structural disorder.

This Thesis is organized in two main parts.

The first part is an overview of the present understanding about the topics dealt with in the present study. Specifically, in Chapter 1 I review the theory of coarsening focusing on the class of systems considered in my work, namely with a scalar order parameter – e.g. the magnetization – that is not conserved during the evolution of the system. I also introduce the kinetic Glauber-Ising Model, which falls in this class and that I have adopted as an optimal playground to study coarsening.

Chapter 2 deals with a general description of percolation theory, together with refined mathematical instruments allowing to identify and quantify percolative effects on coarsening. I also touch upon the topic of fractals, explaining its connection to percolation.

In Chapter 3 the Ising Model is extended to include the most common sources of inhomogeneity, i.e. randomness as due to varying coupling constants and external fields, introducing the so called Random Field and Random Bond Ising Models. Then, I proceed to review what is known about the rich and complex issue of coarsening in these inhomogeneous systems. In particular I will discuss how even the form of $R(t)$ is not established, due to the scarcity of analytical methods and the huge numerical effort required by computer simulations to try to address this problem.

In the second part of the Thesis, I illustrate the methods and the original results of my research work. Chapter 4 contains an explanation of the numerical algorithms and protocols used. It is explained there how Glauber kinetic evolution can be implemented through a Monte Carlo algorithm specifically adapted to simulate efficiently quenches to a vanishing temperature. The realization of a numerical code based on this, the large-scale simulations performed with such routine, and the analysis of the data obtained in this way have been the main task of my Thesis.

Finally, in Chapter 5, I present and discuss a number of original results about the effects of percolation in the coarsening of inhomogeneous systems, thus far a completely unexplored topic. I conclude by discussing how disordered systems compare with homogeneous ones in terms of their scaling properties, examining if and to what extent the dynamical scaling hypothesis can be extended on the basis of our new results.

Parte I

Background

Capitolo 1

Coarsening phenomena

1.1 Introduction to coarsening

Let us consider a macroscopic system in thermodynamic equilibrium, whose state is described by a set of thermodynamic parameters. If we change one of the parameters or the enivironmental conditions, we observe a non-equilibrium transient, during which the system evolves toward a new equilibrium state. For example, imagine a cube of ice in a refrigerator. If we put it at room temperature, it will start to melt and will reach a new equilibrium state when it has completely transformed into water that is at the same temperature as the environement. Another example is the adiabatic expansion of a gas, which is left free to expand. It will spontaneously reach a new equilibrium state, characterized by a new set of parameters, notably with lower pressure.

We can make a distinction between the cases where the relaxation proceeds fast and equilibrium is reached in a relatively small time and the cases where it is very slow and takes place in a time diverging with the size of the system. Let us consider the familiar example of an uniaxial magnet in zero external field subject to a quench, i.e. a rapid, ideally instantanous decrease of the temperature. Denoting with T_C the Curie temperature, if the system is initially in the paramagnetic phase at $T_i > T_C$, and we bring it to a temperature T_f still above T_C, we will have a rapid relaxation to the new equilibrium state, with the system still remaining in the paramagnetic phase. We only observe a certain finite growth of the size of the correlated regions, but the average magnetization remains zero.

Similarly, if we quench from $T_i < T_C$ to $T_f < T_i$, the system starts and remains in the ferromagnetic phase, relaxing rapidly to the new equilibrium state, with a decrease in the fluctuations and an increase in the absolute value of the magnetization. The most interesting case is that of a quench from $T_i > T_C$ to $T_f < T_C$, where we drive the system through a second-order

phase transition. Immediately before the quench, the system is in an unstable disordered state corresponding to equilibrium at the initial temperature. At the final temperature there are two possible ordered configurations, with opposite values of the magnetization. The relaxation cannot take place instantanously, because the two possible final states are completely equivalent and there is no reason for the system to choose one of them[1], breaking this symmetry. Instead, the new equilibrium is reached by the formation and growth of domains of parallels spins, inside which the system is basically equilibrated in one of the two equilibrium phases, as depicted in Fig. 1.1. These domains compete in such a way that the smaller ones are eliminated, so that the typical size of the remaining ones increases. Therefore $R(t)$, the characteristic size of ordered regions, grows with time as the two different broken-simmetry phases compete to select the equilibrium state. This class of phenomena, exhibiting the gradual growth of a new phase – the ferromagnetic one in the last example – goes under the general name of *coarsening* and is studied by the theory of phase ordering kinetics [1]. One of the reasons of interest in the field is that in the thermodynamic limit final equilbrium is never achieved. Indeed, only when $R(t)$ becomes comparable to the system's size the comparison between the few remaining domains can make one to prevail. In an infinitely extended system this never occurs. Moreover, as we will see, also in a finite systems the lowest-energy state is not necessarily reached, as we will see.

Another important example of coarsening is *Ostwald Ripening*. This phenomenon is also common in emulsions, such as the oil-in-water one. For example, let us imagine to add a small quantity of oil to a glass of water and shake the mixture. At sufficiently low temperature we observe that smaller oil particles shrink, while the larger ones grow, and overall the average size will increase. As time tends to infinity, the entire population of oil particles becomes one large spherical particle. In this way, a more thermodynamically stable state is attained, the surface to volume ratio being minimized. Examples of similar phenomena taken from everyday life include beer foams, where smaller bubbles *coalesce* merging into larger ones, and re-crystallization of water in ice-cream. Other standard coarsening systems include binary alloys and liquids, polymer blends and liquid crystals.

Coarsening is not restricted to Physics in a strict sense, but occurs in a wealth of other non-physical branches of science as well. For example it is involved in the spatial spreading of opinions [2] or populations [3]. It also occurs in the field of ecology, when dealing with spatial conflict between, for instance, two species competing for territory along their mutual boundaries.

[1] As it would be the case, for example, if an external field were present. In that case all spins would rapidly align with the external field.

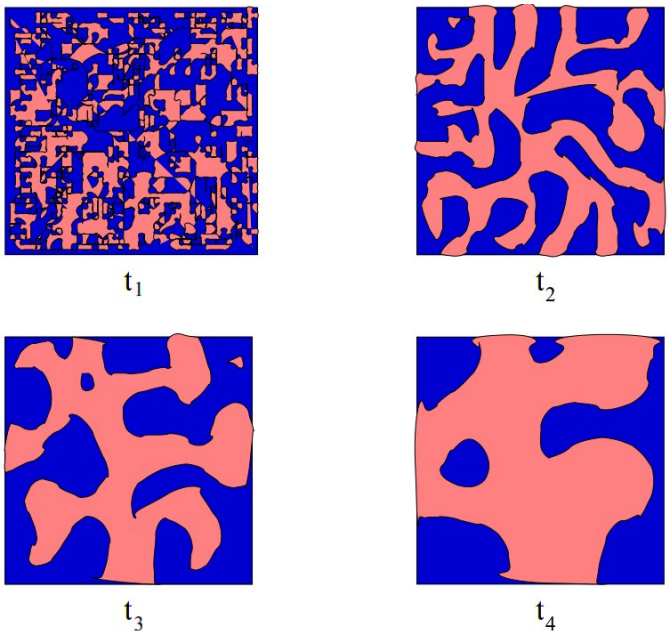

Figura 1.1: Pictorial representation of a coarsening process. Blue and pink regions are the growing domains of parallel spins. The four images correspond to snapshots of the system at subsequent times, with $t_1 < t_2 < t_3 < t_4$.

Many other examples could be made, witnessing the relevance of this field of research.

1.2 Ising Model and its kinetic evolution

In this work we focus on the domain growth in the two-dimensional ferromagnetic Ising Model after a temperature quench from a disordered phase at infinite temperature to a sub-critical one, in particular we will chose $T_f = 0$. We start therefore with a brief introduction to the model, initially presenting it in the context of equilibrium Statistical Mechanics.

The Ising Model was devised by the physicist Wilhelm Lenz in 1920, and it was assigned to his student Ernst Ising, after whom it is called, who published his doctoral thesis on it in 1925. Its aim was to describe phase transitions. Although the model was originally conceived for magnetic systems, it has been successfully applied to many physical situations, such as binary mixtures and alloys, and it is by far the most studied model of Statistical Mechanics. With some modifications, it is also widely used in biology, where it can model neural networks, flocking birds, and beating heart cells, or in sociology, where it has been used to describe, among other things, the spread and clustering of criminality. In general, it is useful to describe situa-

tions in which individual elements (e.g., atoms, animals, proteins, biological membrane, electors, criminals, customers, et cetera) modify their behaviour so as to conform to that of other agents in their vicinity.

Even though it is quite a crude model, it is of great interest because many refined models describing more accurately the critical phenomena which occur in real systems give the same results for a certain set of quantities. This fact goes under the name of *universality* and the quantities sharing this property are referred to as universal. Moreover, in some cases the Ising Model can be exactly solved through the methods of Statistical Mechanics, providing one of the few frameworks where the properties of phase transitions can be worked out with mathematical rigour.

The basic idea of the model is that atoms with a magnetic momentum are arranged in a lattice and a spin variable, whose simbol is s_i, is associated to each lattice site i. It is assumed that each of these elementary magnets may point along the two possible orientations of an *easy* axis, that will be referred to as *up* and *down*. Accordingly, the spin variable takes – apart from a trivial redefinition of units – the two values $s_i = \pm 1$, i.e. it is boolean. With these assumptions, the model is adequate to describe uniaxial magnets, i.e. systems with a preferred polarization axis. Generalizations to cases with more than one easy axis are possible, as well as for other types of anisotropies. We will focus on the uniaxal case. For a system made up of N magnetic moments, the set of spin variables $\{s\} \equiv s_1, \ldots, s_N$ specifies a configuration of the system, or a microstate ν in the Γ space, a N-dimensional discrete space consisting of 2^N points.

The model Hamiltonian is

$$H(\{s_i\}) = -J \sum_{\langle ij \rangle} s_i s_j - \sum_{i=1}^{N} H s_i \tag{1.1}$$

The first term on the r.h.s. mimics the interaction of strength J among couples ij of spins. The value of J is constant. $J > 0$ corresponds to ferromagnetism, favoring a common orientation $s_i s_j = 1$, and the choice $J < 0$ corresponds to antiferromagnetic materials. In the following we will always consider a ferromagnetic model. Moreover, nearest neighbours interactions will be considered, namely two spins s_i and s_j only interact when i and j are adjacent sites on the lattice. The notation $\langle ij \rangle$ signals in fact that the sum is extended only to nearest neighbor couples. Such sum consists of $\gamma \frac{N}{2}$ terms, where γ is the number of nearest neighbors of any given site. This, in turn, depends on the geometry of the system. For instance, $\gamma = 2$ for a one-dimensional lattice, i.e. a chain, $\gamma = 4$ for a two-dimensional square lattice, $\gamma = 6$ for a three-dimensional simple cubic lattice and so on. Finally, the second term on the r.h.s. of Eq. (1.1) is the energy associated to the

presence of a magnetic external constant field favouring the polarization of spins along H.

Let us notice that the parameters J and H in Eq. (1.1) are constant and therefore the model describes an *homogeneous* system for which any lattice site is *a priori* equivalent. In every real system some source of inhomogeneity is always present, and in some cases that can be modeled by introducing some site-dependent randomness in J or H, making the Ising model disordered. The properties of such a model will be discussed in Chap. 3. For the moment, however, we restrict the attention to the homogeneous case with Hamiltonian (1.1) which, when the external field is absent, reads

$$H(\{s_i\}) = -J \sum_{\langle ij \rangle} s_i s_j. \tag{1.2}$$

An important quantity is the magnetization, defined as

$$M(H, T, N) = \left\langle \sum_{i=1}^{N} s_i \right\rangle, \tag{1.3}$$

where N is the number of spins on the lattice and the square brackets stand for the ensemble average. $M(H = 0, T, N)$, the magnetization in the case of zero external field, is called *spontaneous magnetization*.

Thermodynamic properties, such as M, can be extracted in the canonical ensemble from the partition function Z, which for $H = 0$ reads

$$Z = \sum_{\{s_i\}} \exp \left[\beta J \sum_{\langle ij \rangle} s_i s_j \right], \tag{1.4}$$

where the sum is over all spin configurations and $\beta = 1/k_B T$, k_B being the Boltzmann constant.

In his doctoral thesis, Ising solved the one-dimensional model, proving the absence of phase transition. An exact solution in $d = 2$ in case of zero external field was provided by Onsager in 1944 [4]. He showed that in the thermodynamic limit the model presents a phase transition at the critical temperature $T_c \simeq 2.269 J/k_B$. The order parameter of this transition is the spontaneous magnetization, which is finite for $T < T_C$, behaves as $M(H = 0, T, N) \propto (T - T_c)^\beta$ with $\beta = 1/8$ for $T \to T_C^-$ and is zero for $T > T_C$, as shown in Fig. 1.2. At the critical temperature, the free energy has a non-analyticity.

As it is, the model does not posess a natural dynamics, since an evolution for the spin variables has not been provided. One theoretical approach to study phase ordering dynamics is the kinetic Ising Model, originally introduced by Glauber [5]. He devised a form of the model whose time evolution

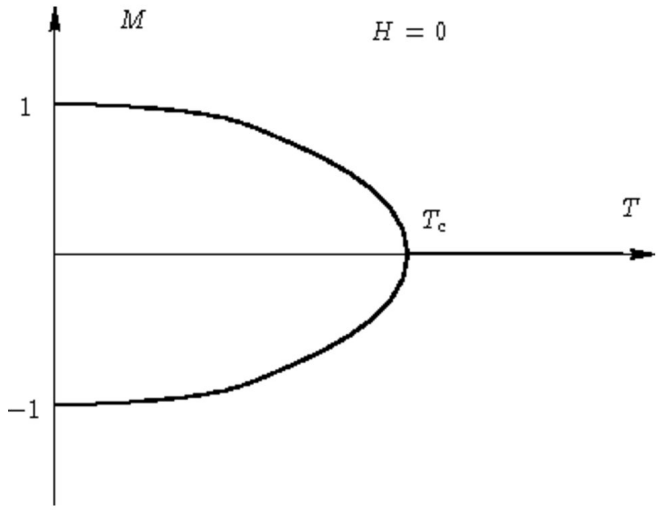

Figura 1.2: Behavior of the magnetization as a function of temperature in the $d = 2$ Ising Model for zero external field. Below T_C the system is found in one of the two possible ferromagnetic phases, with positive or negative value of the magnetization.

can be studied exactly, in statistical terms. Glauber introduced a stochastic dynamics for the spins, that can make random transitions between the values $+1$ and -1, according to a (discrete time) Markov chain. The distinctive feature of this stochastic process is that the probability that the system is in a certain state only depends on the state of the system in the previous instant (timestep), but not on the previous history. In this case the time variation of the probability P_ν of being in state ν is given by the so called *master equation* [8]

$$\frac{dP_\nu}{dt} = \sum_{\nu'} \left[P_{\nu'} w_{\nu'\nu} - P_\nu w_{\nu\nu'} \right], \tag{1.5}$$

which is the basic analytical tool to deal with the stochastic dynamics of discrete variables. In Eq. (1.5), $w_{\nu\nu'}$, the *transition rate*, is the conditional probability (per unit time) of going to a state ν' given that the current state is ν. This equation expresses the balance between a gain term and a loss term. The former – the first on the r.h.s. – represents an income probability flux, namely the probability per unit time that the system goes to the state ν from any other state ν'. The latter, – the second on the r.h.s. – is an outgoing probability flux, i.e. the probability that the system goes from the state ν to any other state ν'.

It is assumed that the transition rates only depend on the values of the neighbouring spins (apart possibly from the interaction with an external

field). We also require that a stationary state exist, imposing $\frac{dP_\nu}{dt} = 0$, and that it be described by the equilibrium canonical distribution. Therefore, setting $\frac{dP_\nu}{dt} = 0$ in Eq. (1.5) and replacing the probabilities P_ν and P'_ν with the canonical ones, one obtains

$$0 = \sum_{\nu'} \left[w_{\nu'\nu} \frac{e^{-\beta E_{\nu'}}}{Z} - w_{\nu\nu'} \frac{e^{-\beta E_\nu}}{Z} \right], \tag{1.6}$$

where Z is the canonical partition function. In order to satisfy this equation, one possibility is to set all the terms of the summation to zero. This leads to the *detailed balance* condition

$$w_{\nu\nu'} e^{-\beta E_\nu} = w_{\nu'\nu} e^{-\beta E_{\nu'}}, \tag{1.7}$$

that may be also written as

$$\frac{w_{\nu\nu'}}{w_{\nu'\nu}} = e^{-\beta(E_{\nu'} - E_\nu)}. \tag{1.8}$$

Notice that Eq. (1.8) does not fix the form of the transition rates, since there is one equation for two unknowns. Indeed, several choices of the w's are found in the literature. We observe that Eq. (1.7) encompasses one of the fundamental properties of equilibrium, namely the time reversal symmetry. Indeed this equality means that, if the measure is the canonical one, there is no net flow of probability between the states ν and ν'. This is because the l.h.s. represents the joint probability (per unit time) to be in state ν and to evolve to ν', while the r.h.s. is the probability of the reversed process. These two probability are set equal by Eq. (1.7), meaning that there are no net fluxes between the system's state or, equivalently, that the time reversal symmetry is at work. Clearly, all the above is only true when the system is at stationarity, namely when $\frac{dP_\nu}{dt} = 0$ in Eq. (1.5). If this is not the case, as we will see, the evolution occurs out of equilibrium, and both time-traslation invariance and time-reversal invariance are not necessarily obeyed.

In the original paper by Glauber, this method is specialized to the one-dimensional Ising Model, resulting in exactly solvable differential equations thanks to an *ad hoc* choice of the transition rates. In Sec. 4.3 we will explicitly report Glauber's choice verifying that condition 1.7 is satisfied. This kinetic evolution allows for a single-spin flip, which implies that the order parameter, the magnetization, can change over time. This is suited to the study of a ferromagnet, which acquires a non-zero magnetization after a quench from a paramagnetic phase. Therefore, Glauber dynamical evolution is a case of non-conserved order-parameter (NCOP) dynamics.

Needless to say, this method is coherent with equilibrium Statistical Mechanics, meaning that any static quantity can be recovered. It is also more powerful since :

- It is susceptible of several generalizations, by means of appropriate choices of the transition rates, provided that the condition (1.6) is satisfied. For instance, it allows one to describe a situation in which each spin is unequally influenced by its nearest neighbours [12] [13] [14]. Other examples, involving different model Hamiltonians, are the introduction of long range couplings, competing interactions, continuous variables and many other features.

- Even when an analytical solution of the equations for the chosen transition rates is not possible, Eq. (1.6) is a starting point to build numerical simulations, the so called *Montecarlo* method. Importantly enough, these computations often represent the only tool to investigate the equilibrium properties of a system.

- It not only allows one to recover static properties, but also discloses dynamical behaviors. For instance, Glauber not only recovered the absence of spontaneous magnetization in the Ising chain, but he also found its decay when the system is prepared in an initially magnetized state (for instance by applying a magnetic field). Even though the specific dynamical behavior is in principle dependent on the choice of the transition rates [2], it can be shown that basic properties do not depend sensibly on this option. Indeed, there exist a wealth of numerical results which agree with theoretical ones and actual experiments [6], irrespective of the choice of the transition rates. The Monte Carlo method is a powerful tool to study the dynamical evolution of Statistical Mechanical systems [7]. Specifically, it is the basic tool to investigate coarsening systems and, as it will be further discussed in Chapter 4, it will be adopted in this work to simulate the phase ordering kinetics of the two-dimensional Ising Model.

For completeness, we mention that there exist kinetic evolutions – COP dynamics – in which the order parameter is conserved. This is suited to describe binary mixtures, in which the two-states variable describe which type of particle – water or oil in the previous example about Ostwald ripening – occupies a given site. Evidently, the number of particles of each type has to remain constant, therefore variables can change their value only two by two. A stochastic dynamics used to describe mixtures must respect this constraint.

Finally, we observe that in the case of systems endowed with a stochastic evolution, the ensemble average of Statistical Mechanics appearing in Eq. (1.3) becomes a non equilibrium statistical average. This is an average taken

[2] Unlike the equilibrium properties, for which we have the same results provided that Eq. (1.8) is obeyed.

over all possible conditions compatible with the initial thermodynamic state (in our case, for instance, the infinite temperature equilibrium state of the Ising model) and *thermal histories*. This means that an average is taken also over the outcomes of all the possible dynamical trajectories generated by the transition rates, appropriately weighted by their occurrence probability. We will globally refer to it as *thermal* average.

1.3 Dynamical scaling

Scaling laws are central to Physics. In Statistical Mechanics, they occur most notably in the study of second order phase transitions. In phase ordering kinetics, a related concept emerges, namely *dynamical scaling*, a distinguishing feature of coarsening.

In order to exemplify it, let us consider again a ferromagnet, quenched to below T_C from a temperature T_i above it. The coarsening process is characterized by a patchwork of domains, as in Fig. 1.1, whose interior is essentially equilibrated, while the non-equilbrium behaviour occurs on the boundaries, which slowly move and tend to become smoother. The structure evolves with time and the ordered regions grow with a characteristic size $R(t)$. The dynamical scaling hypothesis states that the system is fully characterized by the single length-scale $R(t)$. Hence domain structures at different times are statistically equivalent if lengths are measured in terms of this characteristic length. From a pictorial point of view, this simmetry means that in Fig. 1.1. it is not possible to tell whether the fourth panel is a snapshot of the system at a time $t_4 > t_3$ or an enlargement of the configuration at time t_3, namely the one represented in the third panel.

A consequence of dynamical scaling is that all space dependencies in correlation functions appears as ratios between such distances and $R(t)$. For examples, referring to a spin system in its coarsening stage, let us consider the case of an equal-time spin-spin correlation function $C(r,t) = \langle s_i(t)s_j(t)\rangle$, where $s_i = \pm 1$ are spin variables on sites i and j of a regular lattice at a distance r. Notice that a correlation function should be properly defined as $C(r,t) = \langle s_i(t)s_j(t)\rangle - \langle s_i(t)\rangle\langle s_j(t)\rangle$, but the second term drops out because magnetization is not formed in the coarsening stage. When scaling holds, we have

$$C(r,t) = c\left[\frac{r}{R(t)}\right], \tag{1.9}$$

where $c(x)$ is a scaling function, with $x = r/R(t)$.

More generally, for a time-delayed spin-spin correlation function of the form $G(r,t,t_w) = \langle s_i(t) s_j(t_w) \rangle$, we have

$$G(r,t,t_w) = g\left[\frac{r}{R(t_w)}, \frac{R(t)}{R(t_w)}\right], \tag{1.10}$$

where $g(x,y)$ is a scaling function of two variables ($x = r/R(t)$, $y = R(t)/R(t_w)$). These scaling laws are, in principle, strictly obeyed only in a quench to a vanishing final temperature. In case of a quench to a finite temperature T_f, the behaviour of the correlation functions is modified at distances smaller than the equilibrium correlation length $\xi(T_f)$, which is small except in the critical region. In this work, we need not to be concerned by this since we will always work at $T_f \simeq 0$.

It is clear that the form of the function $R(t)$ is very important in the study of coarsening. For many homogeneous systems of interest, it can be determined analytically or it is accurately known by numerical simulations[9]. Usually the characteristic length grows algebraically in time, as

$$R(t) = t^{1/z}, \tag{1.11}$$

with z a dynamic growth exponent which depends on the dimension n of the order parameter – namely if it is a scalar with $n = 1$ as in the original Ising Model or a vector with n components – and the conservation law, namely if the order parameter is conserved or not. For scalar order parameter, we have $z = 2$ for NCOP dynamics and $z = 3$ for the COP case. These results are in excellent agreement with experimental and numerical data. The presence of inhomogeneities drastically modifies the growth law, as we will se in Sec. 3.3.

Finally, let us notice that for finite systems of size L, scaling may only hold in the time region where

$$\xi_{micro} \ll R(t) \ll L, \tag{1.12}$$

where ξ_{micro} is the characteristic microscopic scale of the system, usually the lattice spacing. Indeed, in order to have scaling domains must be well formed, with an interior and a boundary well separated, and this accounts for $R(t) \gg \xi_{micro}$. On the other hand, when the characteristic length is comparable to the system size, the scaling symmetry is spoiled by finite size effects.

1.4 Scaling and percolation

Even though there is indisputable evidence in favor of the dynamical scaling hypothesis from both numerical simulations and experiments, no general

proof is available. It can be shown to be valid analytically by the solution of some tractable models or in the framework of approximation schemes. Exact solutions are only possible in one dimension – e.g. it has been proved for the $1d$ Glauber-Ising Model [10] – or for vectorial spin systems in the limit of an infinite number of vector components.

For a scalar spin system in higher dimensions, analytical methods are scarcer and exact result are not available. One analytical approach [11] manages to prove dynamical scaling in $2d$ for NCOP dynamics with a scalar order parameter, enforcing the well known fact that the dynamics of interfaces is in this case driven by curvature. This means that the only driving force behind coarsening is the tendency of the domain walls to become smoother (due to surface tension), lowering the energy of the system. Given that, it is easy to compute the time needed for a domain of a certain area s immersed in a sea of spins of the opposite sign to shrink and disappear. A complete theory is possible if the distribution n_s of domains of area s is known at a given time. The theory builds on the empirical observation that such a distribution in the initial stages of the coarsening process, soon after the quench, is indistinguishable from the corresponding distribution obtained in the very different context of random percolation theory, a fact that lacks a rigorous proof.

The above derivation of dynamical scaling is for the moment limited to homogeneous models and one may ask if it may be generalized to inhomogeneous ones, for which no analytical result is available. The answers heavily depends on whether the aforementioned correspondence with percolation holds also for inhomogeneous models. Understanding that is one of the main motivations of our study.

Capitolo 2

Percolation

2.1 Introduction

Let us imagine a plane, like a very large table, on which we randomly throw some marbles. Some of them will end up touching each other, some will be isolated. The entire set may be partitioned into *clusters*, i.e. group of marbles that are connected by a "path" of contiguous ones. Similar systems are the object of study of *random percolation theory*, which investigates the properties of these clusters and the probability that there exists a *percolating* or spanning one, connecting two opposite boundaries of the system. If the marbles were made up of a conducting material, in the presence of a percolating cluster applying an electric potential difference between the ends of the system would result in an electrical current between the two connected boundaries.

Most real systems describable by a percolation model contain so many individual elements – the marbles of our example – that surface effects may be ignored and the system can be considered infinite. In this case, the percolating cluster is infinite too.

One of the main reasons of interest in percolation is that the appearance of the percolating cluster can be regarded as a phase transition. Indeed this is one of the simplest examples of second-order phase transition. At variance with other well-known transitions, such as the ferro-paramagnetic one discussed in the previous Chapter, this is a *non-thermal* transition, because temperature is not involved.

The first scientific formulation of this problem dates back to the 1940s and is found in Chemistry, as a model describing the process of polymerization by the formation of bonds between molecules [15] [16]. The mathematical formulation of percolation theory appeared in 1957 [17].

Many other phenomena are described in terms of percolation, witnessing the great theoretical and practical importance of this problem. For instance,

the formation of conducting clusters may cause the breakdown of a dielectric, the formation of a big cluster of trees in a forest leads to the possibility for a significant part of the forest to be destroyed by a fire, etc.

The purpose of this work is to discuss how percolation influences the phase ordering kinetics of an Ising ferromagnet after a sub-critical quench. Therefore, this chapter provides a concise description of the general theory of percolation, outlining the formulation of the problem in terms of phase transition and the connection to fractals. Moreover, we will introduce specific quantities to identify and quantify percolative effects on spin lattices.

2.2 The basics of percolation theory

2.2.1 First concepts and definitions

There are many types of percolation, but here we focus on the most common model which is that of random percolation, meaning in the previous example that marbles are randomly disposed on the table without correlation among them. Random percolation theory is defined on a lattice, and two variants are usually considered: *site* and *bond percolation*. To introduce random site percolation, let us consider a lattice whose sites are in one of two states, empty and occupied. Each site is either occupied with an *occupation probability* p, or empty with probability $1-p$. This happens independently of the state of any other site in the lattice. Therefore, we are dealing with *uncorrelated* random percolation; in the rest of this work we will sometimes refer to it simply as percolation. A cluster is a group of occupied sites which are either nearest neighboring (adjacent) or connected by a group of occupied nearest neighbours. A percolating cluster is one which spans the system from one side to the other. Examples of clusters are shown in Fig. 2.1. The yellow one, which crosses the system from top to bottom, is a percolating cluster.

Random percolation theory deals with the numbers and properties of the clusters formed in this way. Let us define the cluster number $n_s(p)$ as the number of clusters of size s (s-clusters) per lattice site. The (average) number of s-clusters in a hypercubic d-dimensional lattice of linear size L is $L^d n_s(p)$. Defining the cluster number per lattice site $n_s(p)$ – instead of the total number of s-clusters – provides us with a quantity which is independent of the lattice size L.

For finite lattices, $L < \infty$, it is clear that the probability of having a percolating cluster is very low if the occupation probability p is small and very high if p is close to 1. However, even for a very small $p \neq 0$ there is a finite probability of having a spanning cluster, and it is possible to have no spanning cluster even for p very close to one. Conversely, for infinite lattices, it is possible to define the *percolation threshold* p_c, i.e. the smallest

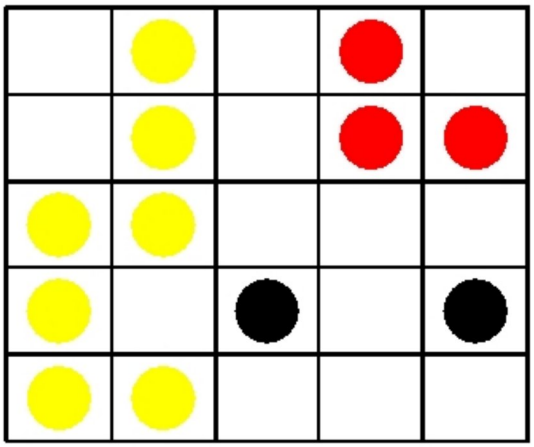

Figura 2.1: Site [ercolation in a $d = 2$ square lattice of linear size $L = 5$. We have one cluster of size $s = 7$ (yellow), one of size $s = 3$ (red), and two of size $s = 1$ (black).

occupation probability p at which a percolating cluster, which will be infinite, appears for the first time. This means that for $p < p_c$ a spanning cluster occurs in the infinite system with zero probability, whereas it is present with probability one for $p \geq p_c$.

The bond percolation problem is the counterpart of site percolation. Identifying sites with the vertices of the lattice [1] , bonds are the lines connecting them, as shown in Fig. 2.2. Each bond between neighbouring lattice sites can be occupied with probability p and empty with probability $1 - p$. A cluster is a group of occupied bonds which are either adjacent or connected by a path of occupied ones.

Table 2.1 lists p_c for site and bond percolation, for various lattices and dimensionalities.

Among systems described through bond percolation models we find porous materials. For example, when a rock containing petroleum oil possesses a system of pores connected one to another, the oil may flow through these pores to form oil clusters. People working in oil industry are interested in understanding these systems to maximize the quantity of oil extracted.

In the following we will be interested in site percolation, therefore we continue to refer to it for our presentation.

The existence for infinite lattices of a diverging quantity, namely the (average) cluster size, at a finite value of a parameter (p_c), and the qualitative change in behavior of the system after crossing this value tell us that one should look for critical behavior. In fact, near p_c several quantities exhibit

[1] Instead, they are usually located in the centres of the plaquettes when representing site percolation models, like in Fig. 2.1

Lattice	Coord. numb.	p_c site percolation	p_c bond percolation
1d	1	1	1
2d honeycomb	3	0.6962	0.65271
2d square	4	0.592746	0.5
2d triangular	6	0.5	0.34729
3d simple cubic	6	0.3116	0.2488
3d BCC	8	0.246	0.1803
3d FCC	12	0.198	0.199
4d hypercubic	8	0.197	0.1601

Tabella 2.1: Percolation threshold are given in column 3 for various lattices in various dimensions. Column 2 lists the coordination number, i.e. the number of nearest-neighbours. Within a given dimension, the percolation threshold decreases with increasing number of nearest-neighbours.

power-law behavior, and there are scaling laws relating the different critical exponents, as in ordinary critical phenomena. The concepts and tools used to study phase transitions and critical phenomena can be applied to percolation. In this probabilistic systems, the relevant parameter akin to temperature is p, the occupation probability for site percolation and the bond density for bond percolation. The critical occupation p_c plays the role of the critical temperature T_c in a magnet.

2.2.2 Percolation in one dimension

Although limited in scope, the one-dimensional percolation is useful to illustrate the main features of this problem, as it can be solved analytically. Imagine an infinitely long chain, made up of equally spaced sites occupied according to the rule discussed above. We can deduce the percolation threshold. We are looking for a cluster spanning the whole system. For any $p < 1$ there are empty sites, and therefore there is no continuous chain of occupied sites spanning the system. On the other hand, for $p = 1$, all sites are occupied. Thus, for the one dimensional system

$$p_c = 1. \tag{2.1}$$

Let us consider the clusters formed in the $1d$ lattice, in the limit $L \to \infty$, which allows us to ignore the effect of the boundary sites. Since a cluster requires the existence of two empty sites, one at each side of the cluster, the probability of an arbitrary site being, say, the left (or right) hand side of an s-cluster[2] is

$$n_s(p) = (1-p)p^s(1-p) = (1-p)^2p^s. \tag{2.2}$$

[2]The same as the average number of s-clusters per lattice site, i.e. n_s.

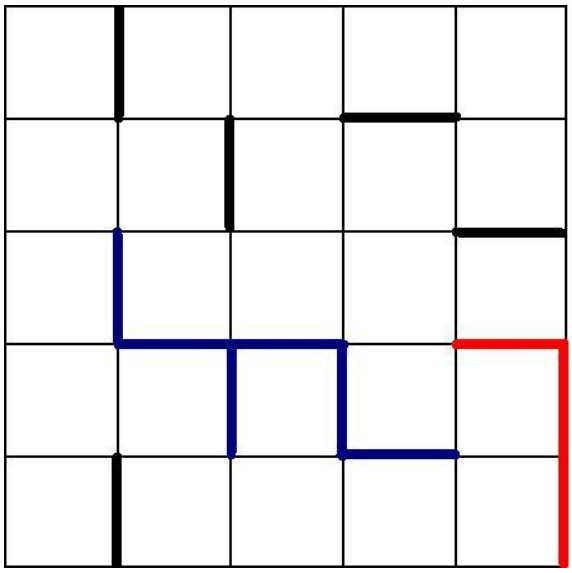

Figura 2.2: In this lattice, sites are the vertices and bonds are the lines connecting them. Occupied bonds are highlighted with thick lines. Blue and red bonds form clusters of size 6 and 3, respectively.

From $n_s(p)$ it is easy to infer the probability that a site belongs to an s-cluster, which is $sn_s(p)$, since there are s occupied sites in such a cluster.

For $p < p_c$, the probability that an arbitrary site belongs to any (finite) cluster is simply the probability p of it being occupied. We can verify that as a simple exercise by direct calculation :

$$\begin{aligned} \sum_{s=1}^{\infty} sn_s(p) &= \sum_{s=1}^{\infty} s(1-p)^2 p^s = (1-p)^2 \sum_{s=1}^{\infty} p\frac{d(p^s)}{dp} = \\ &= (1-p)^2 p\frac{d}{dp}\sum_{s=1}^{\infty} p^s = (1-p)^2 p\frac{d}{dp}\left(\frac{p}{1-p}\right) = p. \end{aligned} \tag{2.3}$$

where we have used the formula to sum a geometric series.

We can now calculate the mean cluster size. The probability w_s that the cluster to which an occupied sites belongs contains s sites is

$$w_s(p) = \frac{sn_s(p)}{p}, \tag{2.4}$$

which is a conditional probability. Therefore, the mean or *average* cluster size $S(p)$ is given by [18]

$$\begin{aligned} S(p) &= \sum_{s=1}^{\infty} s w_s = p^{-1}(1-p)^2 \sum_{s=1}^{\infty} s^2 p^s = \\ &= p^{-1}(1-p)^2 \left(p\frac{d}{dp}\right)\left(p\frac{d}{dp}\right)\sum_{s=1}^{\infty} p^s \\ &= \frac{p_c + p}{p_c - p}, \end{aligned} \tag{2.5}$$

where we have used $p_c = 1$.

Eq. (2.6) shows that the mean cluster size diverges for $p \to p_c$ as a power-law in the distance from the critical occupation probability p_c, namely

$$S(p) \propto |p_c - p|^{-1}. \tag{2.6}$$

As this property is also valid in higher dimensions, but with a different numerical exponent, it is natural to define a symbol for the exponent. The *critical* exponent γ is defined by

$$S(p) \propto |p_c - p|^{-\gamma}, \tag{2.7}$$

and in the $1d$ lattice $\gamma = 1$. Notice that in a one-dimensional lattice where $p_c = 1$ the only possibility is to approach it from below. This will not be the case in higher dimensions, as we will see.

Another important quantity is the *pair connectivity* $\mathrm{g}(\vec{r})$, sometimes also called (improperly) *correlation function*, namely the probability that a site at distance $\vec{r}$ from an occupied site belongs to the same finite cluster [3]. This definition excludes the contribution from the infinite cluster, so it is valid for $p < p_c = 1$ in $1d$. Let $r = |\vec{r}|$. Due to space homogeneity the correlation function depends only on r. Clearly, $g(r = 0) = 1$, since the site is occupied by definition. In $1d$, for a site at position r to be occupied and belong to the same cluster, this site and the $(r-1)$ intermediate sites must be occupied, leaving

$$g(r) = p^r = e^{r \ln p} = e^{-\frac{r}{\xi}}, \tag{2.8}$$

where the quantity $\xi = -\frac{1}{\ln(p)}$ is called the *correlation length*. Near the percolation threshold, where $p \to p_c = 1$, we can rewrite it as

$$\xi = \frac{-1}{\ln(p)} = \frac{-1}{\ln[p_c - (p_c - p)]} \simeq \frac{1}{p_c - p}, \tag{2.9}$$

where we have used the fact that $\ln(1-x) \simeq -x$ for $x \to 0$. It diverges for $p \to p_c$, here as in higher dimensions. Therefore, we have another critical exponent ν defined by

$$\xi \propto |p_c - p|^{-\nu}, \tag{2.10}$$

[3] We use $\vec{r}$ since this definition is valid for arbitrary dimension.

being $\nu = 1$ for the one-dimensional case.

Through the analytical solution of the $1d$ percolation problem, we have seen how the mean cluster size $S(p)$ and the correlation length ξ diverge at the percolation threshold.[4] The divergence can be described by simple power laws of the distance from the critical occupation probability $|p_c - p|$, namely

$$S(p) \propto |p_c - p|^{-\gamma}, \qquad \xi \propto |p_c - p|^{-\nu}. \tag{2.11}$$

Analogous equations, with different critical exponents, are found in higher dimensions, although analytical solutions are not available.

2.2.3 Self-similarity and fractal dimension

In higher dimensions, the percolating cluster at $p = p_c$ is an example of a *random fractal*. Before discussing this fact, we give a conceptual introduction to the concept of fractality and self-similarity, which are also of general interest in Physics and Mathematics.

For the sake of simplicity, we will exemplify them referring to the *Sierpinski gasket*, which is not a random fractal but a *deterministic* one, namely a figure obtained by the repetition of a deterministic algorithm. This fractal is built by taking the midpoints of each side of an equilateral triangle, connecting them and emptying, among the four new smaller triangles obtained, the central one. Ideally, this procedure should be repeated an infinite number of times, for each equilateral triangle obtained in this way. Fig. 2.3 presents the four initial steps of this process.

Self-similarity means that any neighbourhood of any point in the figure contains a copy of the entire figure. Indeed, let us imagine to iterate the construction of Fig. 2.3 infinitely many times. Then we can choose any point of the figure, and draw a circle of any radius around it. If we magnify the encircled portion, we will see again the whole figure, including the internal structure. Equivalently, this means that the figure can be decomposed into some number of disjoint pieces, each of which is an exact copy of the entire figure.

A topological indicator known as *fractal dimension* is associated to any fractal. There are several different definitions of it. One can be introduced similarly to the classical definition of the space-dimensionality in standard geometry. Let us consider a plane figure. If its linear dimension L is doubled, the area S increases by a factor of 4. In general one has that if the linear dimension increases by a factor X_L the area grows by a factor X_S given by

$$X_S = X_L^D, \tag{2.12}$$

[4] There are other quantities, not discussed here, which exhibit a similar behavior.

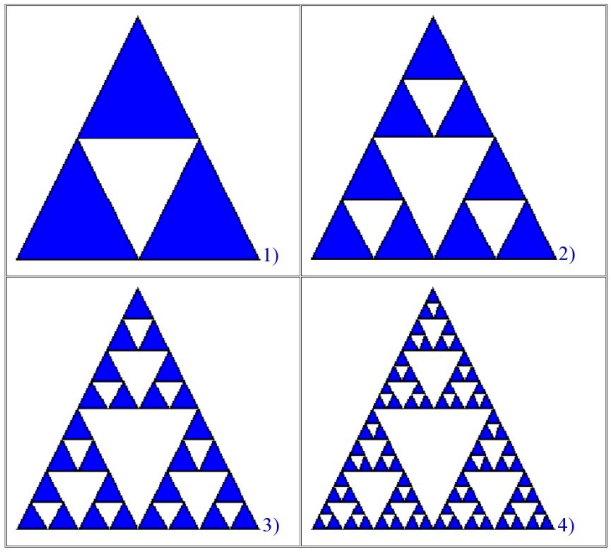

Figura 2.3: The first four steps of the construction of the Sierpinski gasket. In each step, we connect the midpoints of each equilateral triangle in the figure. The area of the fractal is the blue region.

where D is the space dimension. We can rearrange the previous equation into

$$D = \frac{\log X_S}{\log X_L}, \tag{2.13}$$

where $D = 2$ for a plane figure. This is a way to define the dimension of a geometrical shape. Clearly, we speak of a surface since we are considering $D = 2$, we should speak of a volume for $D = 3$ and so on, but this is only a matter of terminology. Looking at Fig. 2.3, we can notice that if the linear dimension of the basis triangle is doubled ($X_L = 2$), then the area of whole fractal (blue triangles) increases by a factor of three ($X_S = 3$). Proceeding as above, we can say the fractal dimension of the Sierpinski gasket is

$$D_{SG} = \frac{\log 3}{\log 2} \simeq 1.585. \tag{2.14}$$

Eq. (2.12) is equivalent to

$$S \propto L^D. \tag{2.15}$$

In general, the occurrence of this kind of power law with a non-integer D signals that we are dealing with a fractal.

The difference between a deterministic fractal, like the Sierpinski gasket, and a random fractals is that the former is identical at all scales while the

latter exhibits a statistical self-similarity, repeating a pattern stochastically. The concept of non-integer dimension applies to both cases.

2.2.4 Cluster structure

We now consider the geometry of clusters. All the following concepts apply in any dimension.

Let us start by considering the mean square distance $\mathcal{R}_s$ between two sites of an s-cluster

$$\mathcal{R}_s^2 = \frac{1}{s}\sum_{i=1}^{s} |\vec{r}_i - \vec{r}_{cm}|^2 \tag{2.16}$$

From that we can define the (squared) correlation length as

$$\xi^2(p) = \frac{\sum_s \mathcal{R}_s^2(p) s^2 n_s(p)}{\sum_s s^2 n_s(p)}. \tag{2.17}$$

The numerator of the R.H.S. of this formula is the average of $\mathcal{R}_s^2$, over all cluster sizes, with weights given by $s^2 n_s(p)$, the probability that an occupied site belongs to an s-cluster multiplied by the number of sites it is then connected to. The denominator acts as a normalization. Therefore, $\xi(p)$ is the average of the distance between two sites belonging to the same cluster. As the most important contribution is given by the largest clusters, it is also the typical radius of the largest finite cluster, both below and above p_c, the percolating cluster being excluded when $p \geq p_c$. As p approaches p_c, we expect it to diverge as

$$\xi \propto |p_c - p|^{-\nu}, \tag{2.18}$$

analogously to the 1d case, but with a different exponent. For $2d$ percolation, methods of Conformal Field Theory [20] give $\nu = 4/3$, in excellent agreement with numerical results. In 3d, ν is somewhat smaller than 0.9.

We can now discuss the behavior of the mass $M(L)$ of the percolating cluster, namely the number of sites it is made up of. If the percolating cluster were a compact figure, we would have $M(L) \propto L^d$, with d the geometric dimension of the lattice. Instead, as already mentioned, the percolating cluster is a fractal for $p \to p_c$, when the correlation length diverges. This can be verified as follows. Imagine to have several lattices of different sizes L, large enough that finite size effects can be neglected, and to bring all of them to the percolation threshold by randomly occupying sites with probability p_c. Let us then plot the masses of the percolating cluster which has appeared in each lattice against the respective sizes. Numerical data obtained simulating this procedure are shown in Fig. 2.4 and they are well described by

$$M(L) \propto L^D, \tag{2.19}$$

with D non-integer, being the fractal dimension of Eq. (2.12). For the $2d$ case, we have $D = \frac{91}{48} < 2$ from numerical data [21] and this result has also been proved in the context of Conformal Field Theory [22]. We recall that this holds when $p = p_c$ and consequently $L \ll \xi \to \infty$.

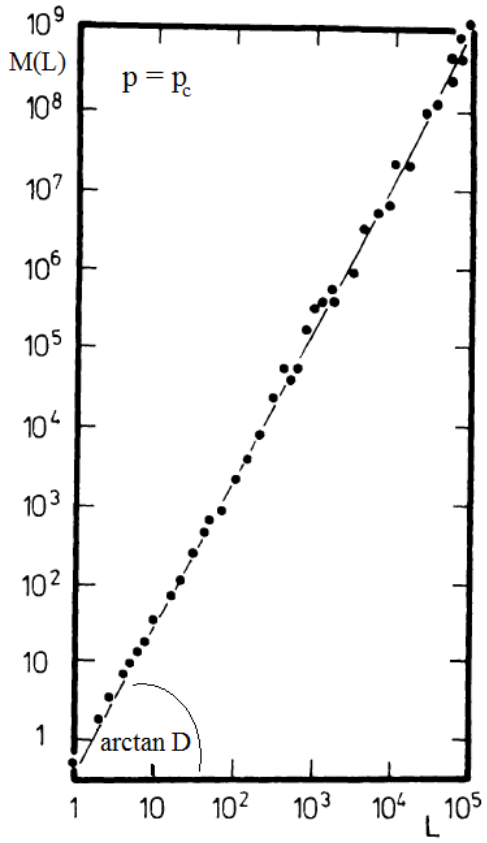

Figura 2.4: Numerical data for the size of the largest cluster at $p = p_c$ as a function of the lattice size L on a double-logarithmic plot. The power law $M(L) \propto L^D$, with $D = 91/48$ signals the fractal behaviour typical of critical percolation.

In order to understand what happens when the occupation probability becomes larger than the critical value, let us remember that the correlation length $\xi(p)$ constitutes the characteristic linear size of the largest finite cluster for any $p \neq p_c$. Since for $p > p_c$ finite clusters reside inside the holes of the percolating cluster, in this case the correlation length can be identified as the typical radius of the largest holes in the percolating cluster [5]. If p continues to increase after having exceeded p_c, more and more sites are occupied and the average hole size decreases. Therefore ξ decreases, and for

[5]Indeed, when ξ diverges at $p = p_c$ there are holes of all sizes, consistently with the property of self-similarity of fractals.

$p \gg p_c$ we will have $\xi \ll L$. In this case we can divide the system into boxes of linear size ξ, inside which the percolating cluster has fractal properties. In d dimensions, the total volume L^d will be divided into $(L/\xi)^d$ boxes. Since the cluster inside each of these boxes of size ξ^d has a mass of order ξ^D, the total mass of the cluster is given by

$$M(L) = \left(\frac{L}{\xi}\right)^d \xi^D = \xi^{D-d} L^d. \tag{2.20}$$

We can summarize the behaviour of the total mass of the cluster, putting it in scaling form, writing

$$M(L,p) = L^D m\left(\frac{L}{\xi(p)}\right), \tag{2.21}$$

where $m(x)$ is a scaling function whose behaviour is

$$m(x) = \begin{cases} \text{const.} & \text{for} \quad x \ll 1 \\ x^{d-D} & \text{for} \quad x \gg 1. \end{cases} \tag{2.22}$$

The scaling function describes a crossover from fractal to uniform behavior at length scales L much larger than the correlation length. In Nature, there are many examples of this type of behavior. For example, let us consider the surface of a table. On length scales much larger than its size, the surface is smooth so that it can be considered a regular two-dimensional plane figure (usually a rectangle or a circle) and $m(x)$ in Eq. (2.21) is a constant. Instead, going to very small scales, the surface will become very rough and probably fractal. Taking the surface of the little bumps and pits into account, the area of the entire figure increases more than fourfold if the linear size is doubled. This is expressed by a fractal dimension $2 < D < 3$.

Finally, let us notice that we have not specified the geometry of the lattice, i.e. square, triangular, kagome or other. That is why the results presented in this section are *universal* : D and the critical exponent ν have the same values for all lattices of a given dimension. This is true also for the other critical exponents, like γ in Eq. (2.7), and is reminiscent of the Statistical Mechanical theory of phase transitions.

2.3 Classification of spanning clusters at p_c

Let us now focus on the $2d$ square lattice with periodic boundary conditions (PBC) at the percolation threshold. It is possible to classify the various types of spanning cluster according to some of their topological properties. Concretely, we can assign a label (m, n) to such clusters according to their

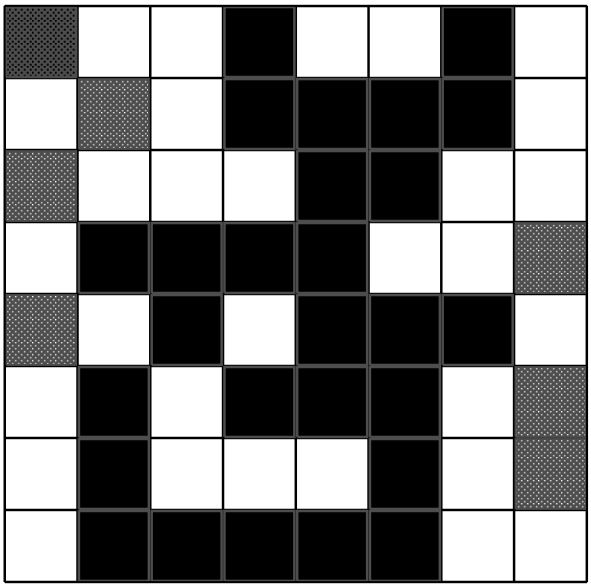

Figura 2.5: An example of $(0, 1)$ percolating cluster, crossing vertically, on a square lattice of linear size $L = 8$. White sites are empty, gray and black sites are occupied, with the black ones forming the spanning cluster.

winding numbers, i.e. how many times they wind in the horizontal (n) and vertical (m) direction. There are several possibilities, as the PBC lattice is a torus and clusters can wind in various ways around it. We illustrate the three most probable cases, which are :

- A percolating cluster crossing the system either horizontally or vertically, the latter is shown in Fig 2.5. Unfolding the torus into the square, such a cluster connects two opposite boundaries of the lattice. Horizontal clusters are labeled $(1, 0)$ and vertical ones $(0, 1)$.
- A percolating cluster conceivable as the union of a $(1, 0)$ and a $(0, 1)$ structure, therefore spanning the system in the two directions as seen in Fig. 2.6. We convene to label it $(0, 0)$.
- A percolating cluster that wraps around the lattice once in the vertical direction and once in the horizontal direction, giving a diagonal crossing

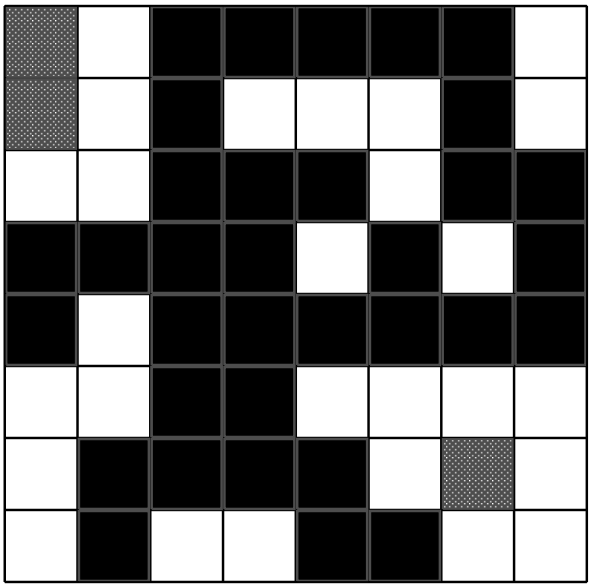

Figura 2.6: An example of $(0,0)$ percolating cluster crossing both horizontally and vertically, on a square lattice of linear size $L = 8$. White sites are empty, gray and black sites are occupied, with the black ones forming the spanning cluster.

when the torus is unrolled into the square. This is labeled $(1,1)$ if the crossing is from top left to bottom right, while $(1,-1)$ if it is from bottom left to top right. The latter possibility is exemplified in Fig. 2.7.

We denote the probability of having an (n,m) configuration by $\pi_{n,m}$. In the context of random percolation theory, it can be exactly proven [23] that

$$\pi_{0,1} + \pi_{1,0} \simeq 0.3388,$$

$$\pi_{0,0} = 0.61908,$$

$$\pi_{1,-1} + \pi_{1,1} \simeq 0.04196,$$

with $\pi_{0,1} = \pi_{1,0}$ and $\pi_{1,-1} = \pi_{1,1}$. There exist configurations with higher winding numbers occurring with exceedingly small probabilities, that we will not consider here. Table 2.2 summarizes the probabilities discussed insofar, it will be useful for future reference.

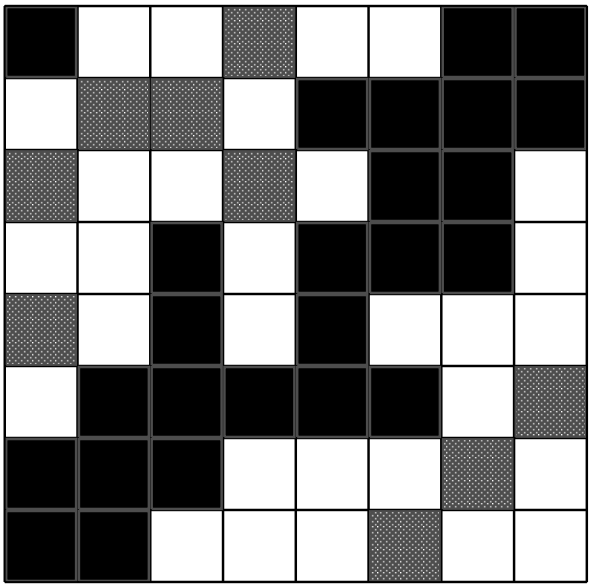

Figura 2.7: An example of $(-1, 1)$ percolating cluster, crossing diagonally, on a square lattice of linear size $L = 8$ with PBC. White sites are empty, gray and black sites are occupied, with the black ones forming the spanning cluster.

Type of crossing	Probability
Horizontal and vertical	$\pi_{0,0} = 0.61908$
Straight, horizontal	$\pi_{1,0} = 0.1694$
Straight, vertical	$\pi_{0,1} = 0.1694$
From left bottom to right top	$\pi_{1,-1} = 0.0209$
From left top to right bottom	$\pi_{1,1} = 0.0209$

Tabella 2.2: Summary of the three types of crossings considered, for a cluster at percolation threshold on a square lattice with PBC, with respective probability of occurrence. The values of the probabilities are approximated.

2.4 Winding angle for hulls

We now proceed to introduce some concepts and results which will allow us to evaluate the effects of percolation in the phase ordering kinetics of the quenched ferromagnetic Ising Model.

In our presentation of percolation we have considered empty and occupied lattice sites. This corresponds to the gas lattice variable description of Statistical Mechanics. We can go from that to spin variables simply identifying an empty site with a down spin and an occupied sites with an 'up' one, or vice versa. From now on, we will use the spin variable description in which a spin cluster or domain is a connected set of aligned spins.

A *broken bond* is a link of the lattice between two unaligned neighboring spins. This is exemplified in Fig. 2.8, where we use square lattice for simplicity, but the extension to other kind of lattices is immediate.

The *domain wall* of a spin cluster is its external and internal contour, constructed as follows. One first generates a dual lattice by placing a site at the center of each plaquette of the original lattice. Next, the links on the dual lattice that cross broken bonds on the original lattice are joined together. In this way, one finds closed loops on the dual lattice that run along the external and, if present, internal boundaries of a spin cluster. The *hull* of a domain is the external part of the countour. We can associate a length to these contours by counting the number of broken bonds they cross, therefore taking the lattice spacing as the unity. Simple examples are a cluster composed of a unique spin, with an hull of length 4, and a two-spin cluster, whose hull has length 6 [6]. An example of hull of length 30 is shown in Fig. 2.9.

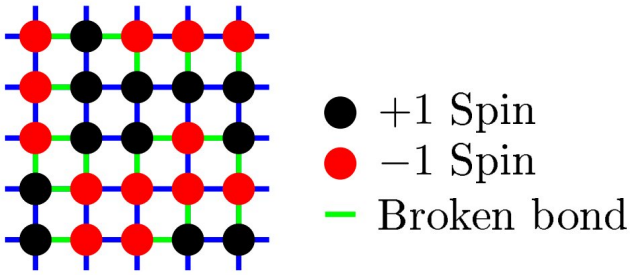

Figura 2.8: Sketch of an Ising spin configuration with the boolean spin variables represented by black and red dots on the square lattice sites. Broken bonds are drawn with green lines on the edges of the lattice

Let us consider two points P and Q at a distance x along the hull of a domain, and trace the tangent to the hull in these two points. The angle between the two tangents, measured counterclockwise in radiants units, is the *winding angle* $\theta(P,Q)$. By fixing x, one can calculate the average winding angle for two points separated by this distance, $\langle\theta(x)\rangle$.

[6] In these cases the domain walls coincides with the hulls

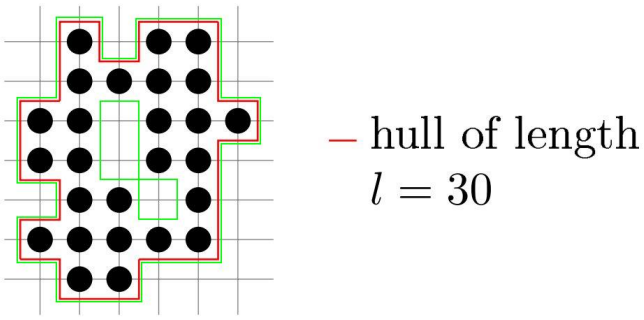

Figura 2.9: Example of a domain, its domain wall (in green) and hull (in red). Unaligned spins have been removed.

On a lattice at the percolation threshold, it can be exactly proven through methods of Conformal Field Theory that for each hull the average squared winding angle $\langle \theta^2(x) \rangle$ satisfies [24]

$$\langle \theta^2(x) \rangle = a + \frac{4k}{8+k} \ln x, \tag{2.23}$$

where a is a constant and $k = 6$. This equation holds when $x \gg 1$, the unity of distance being the lattice spacing, therefore in the continuous limit, when the microscopic structure of the lattice can be ignored.

Let us mention that Eq. (2.23) has been derived also for other critical phenomena, with a different value of k. For example, one can perform the same analysis for the thermal phase transition of the $2d$ ferromagnetic Ising Model. After preparing the system at the critical temperature T_C, one can also measure the squared winding angles of the correlated clusters' hull, finding the same equation but with a value of k as different as $k = 3$.

We stress that these are purely geometrical properties which can be measured on any lattice configuration. Indeed Eq. (2.23), and the value of k in particular, will be used in Chapter 5 as a tool to highlight the presence of a cluster with the same topological properties of the ones found in random percolation, in the a priori very different problem of the phase-ordering of the quenched Ising model. Upon measuring $\langle \theta^2(x) \rangle$ along the coarsening clusters of spins we will find Eq. (2.23) obeyed with a value $k = 6$ with excellent precision.

Capitolo 3

Inhomogeneous Systems

3.1 Introduction

In Sec. 1.2, we said that Eq. (1.1) is the Hamiltonian of the homogeneous Ising Model (IM) because the parameters in it do not vary in space. In such a model, in the thermodynamic limit all lattice sites are equivalent. This is true *a priori*, in the sense that although a single realization of the system can be inhomogeneous, local observable do not depend on space after the thermal averaging. For example consider the set $\{s_i\}$ of spins variable : in a realization we may have $s_j = +1$ and $s_k = -1$ (see for instance Fig. 1.1), but for the thermal averages we have $\langle s_i \rangle = \langle s_j \rangle$. An inhomogeneous model might be simply one in which we fix the value of one spin, say $s_i = 1$, or we adopt fixed boundary conditions. Each of these modifications would spoil space-translation invariance.

One possible source of inhomogeneity is *quenched disorder*. A system is disordered if some parameters describing it are not deterministically fixed, but extracted from a probability distribution. The original part of this work will focus on the phase ordering kinetics of two types of disordered Ising Model, the Random Bond lsing Model, in which we introduce randomness in the coupling constants, and the Random Field Ising Model, in which there is a random external field acting on each spin. We will focus on the case in which such features are fixed for a given realization of the system and do not vary during the coarsening process, acting as external constraints. We call this type of disorder *quenched* – i.e. frozen – or *quenched randomness*. If disorder acts locally, as in the two models mentioned above, it spoils the homogeneity of the system.

This chapter serves as an introduction to disordered systems. We will formally describe the two aforementioned models and we will review some general results about the phase ordering kinetics in presence of quenched

randomness. We will focus on the case with NCOP dynamics, on which our research is exclusively focused.

3.2 Models

Imagine we want to describe a real material, like a magnetic solid, using the IM with the usual nearest-neighbor interaction. An improvement can be done considering that in real systems the coupling between particles is not constant, but varies in space. This can be induced by different effects, such as lattice deformations or other defects. To account for that, the Ising Hamiltonian for zero external field can be written more properly as

$$H(\{s\}) = -\sum_{\langle ij \rangle} J_{ij} s_i s_j. \tag{3.1}$$

It is practically impossible to know which is the precise spatial configurations of the couplings J_{ij}. A useful simplified model assumes that they are stochastic variables with a certain distribution, namely

$$J_{ij} = J_0 + \theta_{ij}, \tag{3.2}$$

where J_0 is the coupling constant for the pure model and the θ's are random numbers.

If $J_0 + \theta_{ij} > 0 \; \forall (i,j)$ one has always $J_{ij} > 0$, i.e. all the couplings remain positive. The ferromagnetic phase is preserved and we have a disordered ferromagnet. In this case disorder only influences some quantitative properties of the system, such as the value of the critical temperature, but the gross features of the model are retained. For this reason this kind of disorder is sometimes denoted as *weak* (although the same term is sometimes used in disordered systems with a different meaning).

The situation changes if for some (i,j) it is $J_{ij} < 0$. The fact that some couplings are negative may have consequences on the existence of the ferromagnetic phase. Indeed in this case a new ingredient, *frustration*, comes on the scene, drastically modifying the properties of the system, as it is well known in the case of spin-glasses. In this case we speak of *strong* disorder.

Finally, if for some (i,j) it is $J_{ij} = 0$, it means that some bonds are removed [1]. This kind of disorder is weak or strong depending on the fraction of missing bonds, which in turn depends on the distribution of the θ's.

The θ's are usually uncorrelated and extracted from a density described by a single parameter $\mathcal{J}$, such that $\overline{\theta_{ij}\theta_{kl}} = \mathcal{J}^2 \delta_{ij}\delta_{kl}$. [Here and in the

[1] A related model, with *bond dilution*, is obtained by removing some bonds while leaving unaltered the values of the other coupling constant. This can also be obtained by the RBIM choosing $\theta_{ij} = \pm J_0$, each value taken with probability d.

following we will denote by $\overline{\cdots}$ an average over the possible values of the quenched-disorder variable (here θ_{ij}). This in order to distinguish it from the thermal average $\langle \dots \rangle$, which, as explained at the end of Sec. 1.2, is taken over the distribution of the random events – such as the spin-flips in the Ising model – occurring in the stochastic evolution of the system for any given realization of the quenched disorder (if present).] For example, one can choose a bimodal distribution with $\theta_{ij} = \pm\mathcal{J}$, or a Gaussian with zero mean and variance $\mathcal{J}^2$. Another possibility is to extract the θ's from a uniform distribution with support $[-\mathcal{J}, \mathcal{J}]$, therefore with zero average. In our numerical experiments we always make this choice, and we always restrict to the case $J_0 > \mathcal{J}$ without frustration. However, the general results we will present in the next sections are expected to be valid also for the RBIM with different choices of the noise distribution, provided that the disorder is weak.

The Random Field Ising Model (RFIM) attempts to describe the fact that any material experiences some external field, which in many cases is unknown and position-dependent. We can describe that by adding to the homogeneous Ising Hamiltonian a site-dependent external field H_i, namely

$$H(\{s_i\}) = -J \sum_{\langle ij \rangle} s_i s_j - \sum_{i=1}^{N} H_i s_i. \tag{3.3}$$

As for the coupling constants in the RBIM, the fields are also usually taken to be uncorrelated random variables for which various choices of the distribution can be made. In our study we have adopted a bimodal distribution with $H_i = \pm h$, each value taken with probability 1/2.

We ask now if the RFIM admits a ferromagnetic (ordered) phase, namely if random fields introduce a weak or strong disorder. To answer, we follow an argument due to Imry and Ma [25]. Let us start by considering a pure system at temperature well below T_c and suppose that there is a positive magnetization. If one tries to destroy the magnetized state by reversing the spins in a bubble of linear size r inside an ordered domain – a process referred to as *nucleation* – one has to pay an energy of order $\Delta E \sim J r^{d-1}$. Indeed there is an energetic cost $2J$ for every pair of unaligned spins, and the number of such couples is proportional to the surface of the droplet which, in turn, is proportional to r^{d-1}. At low temperature this corresponds to a comparable increase of the free energy, since the entropic gain is negligible in that situation. That is why the formation of such droplets is suppressed in a pure system. Let us now introduce a random field with bimodal distribution. In a bubble where there is a majority of $H_i = -h$, the formation of the droplet can be advantageous because now the alignment of spins with the external field is accompanied by an energy gain. According to the central

limit theorem, the sum of the $N \propto r^d$ random fields inside the bubble is a stochastic variable with zero average and standard deviation $\sigma = h\sqrt{N} \sim H_0 r^{d/2}$. From that we can estimate the total energy variation by adding this contribution to the surface term computed before

$$\Delta E = Jr^{d-1} - H_0 r^{d/2}. \tag{3.4}$$

If this quantity is non-positive the droplet will be flipped and others will do the same, thus destroying the ferromagnetic state. For $d/2 > d - 1$, namely $d < 2$, the second term in Eq. (3.4) always prevails over the first for sufficiently large bubble size r, regardless of the value of J and H_0. In this case, there exist a bubble of size r_{IM} (an Imry-Ma domain) which can be advantageously reversed, triggering the destruction of the ferromagnetic phase. Notice that r_{IM} increases when the strength of the disorder H_0 decreases. Therefore, for $d < 2$ the ferromagnetic phase of the RFIM is unstable against *any* random field. Since the reasoning is based on energetic considerations (not free-energy considerations) it is true only at $T = 0$. This simple heuristic argument can be supported by a more rigorous proof, which shows that also for $d = 2$ the ferromagnetic phase is not sustained at any temperature. At finite temperature the situation is more complicated, and will not be considered here.

3.3 Coarsening in inhomogeneous systems

In a homogeneous system, as for instance one described by the homogeneous IM, coarsening occurs at any final temperature T_f, provided it is below T_C, including $T_f = 0$. Indeed, as said in Sec. 1.4, in this case the domain growth proceeds by smoothening of the interfaces, which tend to become flatter to lower the energy of the system. This process is regulated by their curvature [11] and does not need thermal activation.

Quenched disorder usually introduces preferred positions where interfaces get *pinned* – stuck – in local energy minima. The dynamics can then proceed only by means of thermal activation. Take, for instance, the RFIM and consider two neighboring domains, one in which the majority of the spins are down and the field is predominantly negative and the other in which the reversed situation occurs. This corresponds to the existence of an energy barriers ΔE associated to the domain wall between the two regions,. Indeed, for one domain to grow (say the first), some spins of the other domain, which were previously aligned with the random field, have to disalign with it, resulting in an energy increase. This barrier pins the interface which can move only by means of thermal fluctuations. A similar situation occurs in the RBIM.

Pinning barriers slow down down the kinetic with respect to the homogeneous case. In the rest of the Section we will present a simple argument, which shows how the growth of the characteristic length $R(t)$ is modified by the presence of quenched disorder.

Assuming the dynamical scaling property, the typical ΔE is expected to depend on the configuration only through the growing length, namely

$$\Delta E = f[R(t,\epsilon),\epsilon], \tag{3.5}$$

where ϵ generally represents the parameter [2] denoting the strength of the disorder and $R(t,\epsilon)$ is the growing length in presence of disorder. (Here and in the following, $R(t)$ – as well as all thermodynamic quantities – is defined as averaged both over the thermal history and the disorder realizations). The time t_{esc} needed to escape an energy barrier ΔE is given by the Arrhenius expression

$$t_{esc} \propto e^{\beta \Delta E}. \tag{3.6}$$

Making the simplifying assumption that the slowing down of the evolution can be taken into account by a simple rescaling of time ($t \to t/t_{esc}$) in the expression (1.11) of $R(t)$ (with $z = 2$ for NCOP dynamics) one has

$$R(t,\epsilon) = a t^{1/2} e^{-\frac{\beta}{2}\Delta E}. \tag{3.7}$$

The next point is to find the form of $f[R(t,\epsilon),\epsilon]$ in Eq. (3.5), in order to substitute into Eq. (3.7). Although this cannot be done in general, most of the systems can be divided into three [26] classes, according to the way ΔE depends asymptotically on $R(t)$. We detail below these cases, using $R(t)$ to mean $R(t,\epsilon)$ in order to simplify the notation.

1. ΔE approaches a constant value, i.e.

$$\lim_{R(t)\to\infty} f[R(t),\epsilon] \leq c(\epsilon),$$

   where $c(\epsilon)$ is usually an increasing function of disorder, but it is constant with respect to $R(t)$. This implies that there is an upper limit to the height of the barriers. Accordingly, from Eq. (3.7) one finds that for long times the same growth-law of a homogeneous system is obeyed, although with a smaller pre-factor, of order $ae^{-\frac{\beta}{2}c(\epsilon)}$, which depends strongly on temperature and disorder. A prototypical example of this behavior is the RBIM in $d = 1$.

2. ΔE diverges logarithmically with the growing length, that is

$$\lim_{R(t)\to\infty} f[R(t),\epsilon] = z(\epsilon) \ln R(t),$$

[2] Or the parameters, if more than one is needed to describe the disorder.

with $z(\epsilon)$ constant in time. Substituting into Eq. (3.7), one finds the large-time behavior

$$R(t) \simeq t^{1/\zeta(\epsilon)}, \tag{3.8}$$

with an exponent $\zeta(\epsilon) = 2 + \beta z(\epsilon)$ which depends on temperature and disorder. This behavior, as we will see, is observed pre-asymptotically both in the RBIM and in the RFIM in $d > 1$.

3. ΔE diverges algebraically with the growing length, namely

$$\lim_{R(t)\to\infty} f[R(t), \epsilon] = b^{-1}(\epsilon) R(t)^{\psi},$$

where $\psi > 0$ is an exponent and $b(\epsilon)$ is another constant. This leads to a logarithmic growth

$$R(t) \simeq [b(\epsilon)\beta^{-1} \ln t]^{1/\psi}. \tag{3.9}$$

This is the asymptotic behavior observed in most disordered magnets.

3.4 Crossover in the growth law

The next step is to establish to which class the RBIM and RFIM belong. This can be more easily understood in one-dimensional systems, for which the task is simplified by the fact that interfaces are point-like. As an example of the type of reasoning involved, let us consider the RFIM for $d = 1$, always assuming a bimodal distribution with $h_i = \pm h$ like we did in our research work [3]. Let us imagine to have a single interface in the system originally located between sites i and $i + 1$ (namely $s_i s_{i+1} < 0$) as represented in Fig. 3.1 by the continuous blue line.
Let us suppose that at a later time the interface moves a distance r away from the original position and is located between sites $i + r$ and $i + 1 + r$. According to the central limit theorem, the energy variation (the barrier) is a stochastic variable with zero average and standard deviation $\sigma = hr^{1/2}$, so one has

$$\Delta E_{i,i+r} = -\sum_{j=1}^{i+r} h_j \sim hr^{1/2}. \tag{3.10}$$

This equation indicates that the pinning energy increases algebraically with the typical distance $x(t)$ traveled by the interface. Since it can be shown that (as in the clean case) the average size of domains $R(t)$ grows proportional to $x(t)$ [27], we conclude that the $1d$ RFIM belongs to the third class of Sec. 3.3. Indeed, in the large time limit one has $R(t) \propto (\ln t)^2$ [28].

[3] Other possible choices do not make significant differences.

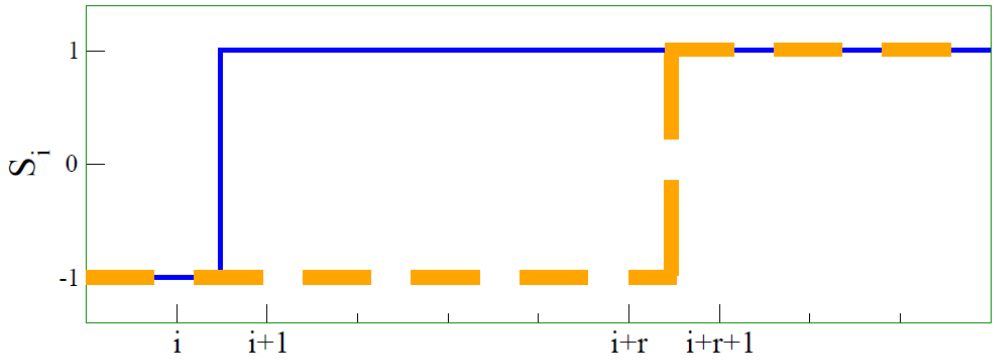

Figura 3.1: An interface initially located between sites i and $i+1$ (plotted in the continuous blue line) moves to a new position r sites away (dashed orange line).

As ΔE is a monotonously increasing function of $R(t)$, it will prevail over the thermal energy scale $k_B T_f$ if $R(t)$ is larger than a given value which depends on h and T_f. Hence, there exist a length λ such that for $R(t) \sim \lambda$ it is $\Delta E \sim k_B T_f$. Substituting Eq. (3.10) into the last expression, we find

$$\lambda(h/T_f) \sim (h/T_f)^{-2}. \tag{3.11}$$

This *crossover length* is important because it separates the ordering process into two regimes. In the first one, corresponding to $R(t) \ll \lambda$, domain growth is dominated by thermal fluctuations and is therefore the same as in the pure system with $R(t) = t^{1/2}$. In the second regime, when $R(t) \gg \lambda$, the kinetics is strongly affected by disorder and one has $R(t) \propto (\ln t)^2$.

As a basis for the original work presented in Chapter 5, we want to know if these results transfer to higher dimensions, for the RFIM as well as the RBIM. However, understanding coarsening in inhomogeneous systems in $d > 1$ is much harder. On one hand this is due to the fact that analytical approaches are much more difficult, and on the other hand because physical intuition is less straightforward when interfaces are lines or surfaces that in the presence of the pinning centers can bend and stretch. Moreover, numerical simulations – crucial in the absence of analytical tools – are very demanding due to the slow growth of $R(t)$.

For the RFIM, the pattern of behaviors exhibited in higher dimensions is richer tha in the one-dimensional case. First, a wealth of numerical data show that also in $d = 2$ [29] [30] [31] and $d = 3$ [29] [32] the asymptotic growth of $R(t)$ is logarithmic as in Eq. (3.9), although the value of the exponent ψ cannot be precisely determined numerically, due to the very slow increase of $R(t)$. Therefore, the pinning barriers encountered at late times should have a similar nature to the one-dimensional ones, as it is also suggested by

the fact that for the crossover length one has $\lambda(h/T_f) \sim (h/T_f)^{-2}$ as for $d = 1$ [Eq. (3.11]. Experiments on real systems [33] confirm the logarithmic growth.

The behavior of the model in the pre-asymptotic stage is, instead, richer and more debated. In [32] a pure-like early stage with $R(t) = t^{1/2}$ was found. However, in [34], by taking the limit $T_f \to 0$ with $\epsilon = h/T_f$ fixed, a power law of the type of Eq. (3.8) was observed in the initial stage with the h/T_f dependent exponent ζ.

It was proposed [35] that these apparently contrasting observations can be reconciled in a *double crossover* scenario. Accordingly, by opportunely tuning the parameters, it should be possible to observe two crossovers in a single quench history. A first one from the homogeneous-like growth to the disorder-dependent power law of Eq. (3.8), and a second one after which one recovers the asymptotic logarithmic behavior of Eq. (3.9). However, this has never been reported thus far and the question remains unsettled.

For the RBIM, roughening theory applied to interfaces predicts [36] that in $d \geq 2$ the growth law is logarithmic with $\psi = 1/4$. The correctness of this prediction remained controversial until recently, when large-scale numerical simulations for $d = 2$ [37] have produced sufficient evidence for the existence, after a long-lasting algebraic regime of the type of Eq. (3.8), of a crossover to an asymptotic logarithmic growth. It was also found that the crossover length behaves as in the RFIM, namely $\lambda(\mathcal{J}/T_f) \sim (\mathcal{J}/T_f)^{-2}$. Nonetheless, a precise numerical determination of the exponent ψ is impossible also in this case. Experiments on real two-dimensional systems find a logarithmic growth with $\psi = 4$ [38] [39]. The nature of the pre-asymptotic algebraic power-law is still unclear.

3.5 Zero-temperature quenches

From the discussion of Sec. 3.3 it seems that it is not possible to study zero-temperature quenches in the RFIM and RBIM. Indeed, we know that the presence of quenched randomness gives rise to energy barriers which can be overcome only by means of thermal activations, as exemplified in Sec. 3.4 for the $1d$ RFIM. Therefore, at $T = 0$ the dynamics is frozen.

Nonetheless, there is a way to study – *de facto* – a zero temperature quench numerically. It is indeed possible to quench to a temperature T_f high enough to allow for thermal activation, but low enough to prevent the destruction of the ordered phase by thermal fluctuations. For the RBIM, it is sufficient to choose a temperature well below T_C but which allows to overcome barriers, in order to be able to observe a congruous growth of the domains in a reasonable time. If, in addition, T_f is so low that its effect only

amounts to a facilitation in crossing energy barriers, quenching to such a T_f basically reproduces the main features of the true dynamics in the $T_f \to 0$ limit.

For the RFIM, instead, we know from Sec. 3.2 that for $d = 2$ the ferromagnetic phase is not sustained at any temperature, including $T = 0$. However, it is still possible to observe coarsening by quenching at very low temperature if the strength of disorder h is sufficiently small. Indeed, we have seen that nucleation – which destroys the ordered phase – can only occur for a droplet of size r_{IM}. Therefore, after a quench to a low temperature the system will order as long as $R(t) \ll r_{IM}$ and, limiting the simulations to a range of times such that this condition is fulfilled, we can avoid nucleation and observe an effective ferromagnetic ordering of the system. Let us recall that r_{IM} diverges in the limit $h \to 0$. Therefore, by choosing sufficiently small values of h, the coarsening regime can extend to very long times.

Mutatis mutandis, the same happens for the homogeneous one-dimensional Ising Model. The critical point is $T = 0$ and therefore, strictly speaking, if $T \neq 0$ there is no phase transition and the system remains globally disordered. However, if we quench to a very low temperature, the model actually orders as at $T = 0$ until $R(t)$ is much smaller than the equilibrium correlation length $\xi(T_f)$. Since $\xi(T_f) \simeq e^{2J/k_B T_f}$, it can be made arbitrarily large by controlling the quenching temperature. Hence, one can observe the coarsening process for an arbitrarily long time.

In our simulations, we will enforce the two limits where both the temperature and the strength of the disorder vanish. Specifically, we will take $T \to 0$ and $\mathcal{J} \to 0$ or $h \to 0$, for the RBIM and the RFIM respectively, keeping the ratios $\epsilon_{RB} = \mathcal{J}/T$ and $\epsilon_{RF} = h/T$ constant. In so doing, the system is described by a single parameter, which makes the results more easily interpretable (more details on the numerical algorithm will be given in Sec. 4.3 and 4.4). Another important advantage of a small temperature limit is to reduce thermal fluctuations and hence the noise affecting the data.

3.6 Numerical results for growing lengths

The knowledge of the growing lentgh $R(t)$ is fundamental in every coarsening problem. Therefore we report preliminarly here our results for $R(t)$, for the various systems we have simulated.

As already mentioned, previous studies on the RFIM report a crossover from a disorder-dependent power law to a logarithmic growth, in the limit of $T_f \to 0$ and ϵ fixed we are interested in. In Fig. 3.2, we plot our numerical determination of the growing length, for different values of ϵ_{RF}. Let us first observe that the curves are indistinguishable until $R(t) \simeq 2$, meaning

that disorder is not effective when the configuration is still very close to the initial condition. Roughly speaking this means that the first pinning center has not been encountered yet in the evolution. Then, when domains start to establish, we observe the disorder-dependent power law, with an exponent which decreases as ϵ_{RF} increases. This is clearly observed in the double-logarithmic plot of Fig. 3.2 as a straight behavior of the curves in a certain time-domain. This is true until a crossover time is met, when the curves start to deviate from the power law behavior, with a gradual downward bending, signaling the beginning of the crossover to the logarithmic growth (which is not fully entered in our simulations). Notice that the crossover time is smaller for larger ϵ, as expected since $\lambda(\epsilon) \sim \epsilon^{-2}$, as mentioned in Sec. 3.4.

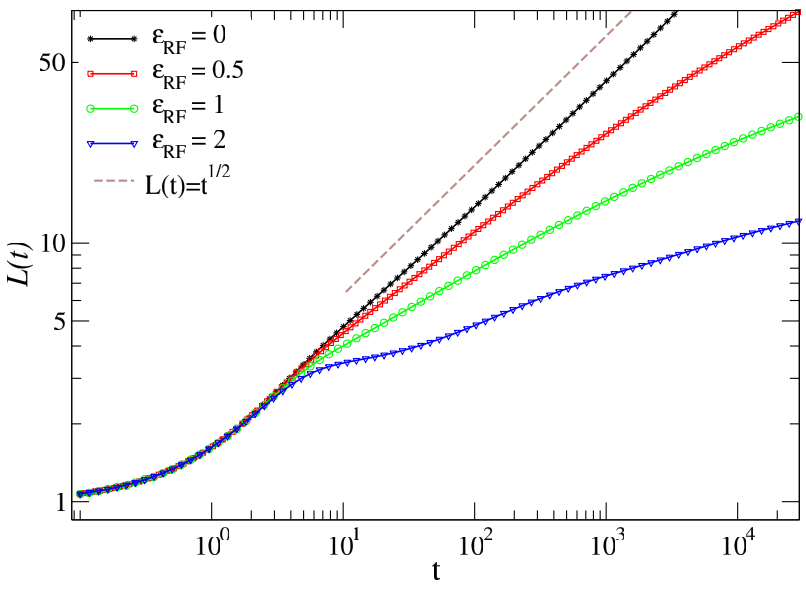

Figura 3.2: Growth law for the $2d$ RFIM with various disorder strengths ϵ_{RF} (increasing from top to bottom, see key), for a lattice of size $L = 200$. Data have been averaged over $10^5 - 10^6$ samples. The dashed line is the $t^{1/2}$ law.

Also for the RBIM, we know from Ref. [37] that the logarithmic growth law is preceded by a very long-lasting pre-asymptotic regime in which $R(t)$ obeys a disorder-dependent power law. In Fig. 3.3 we observe precisely this power law. In Ref. [37] it is also shown that the crossover to the asymptotic regime where $R(t)$ grows logarithmically occurs at very long times, of order

$10^6 - 10^7$ MCS at least. This explains why we see no sign of such a crossover in the time region considered.

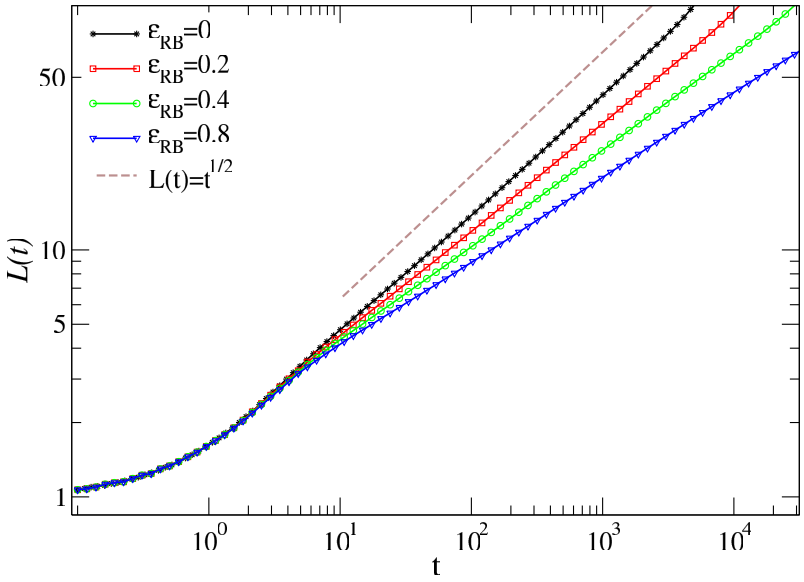

Figura 3.3: Growth law for the $2d$ RBIM with various disorder strength ϵ_{RB} (increasing from top to bottom), for a lattice of size $L = 200$. Data have been averaged over 10^6 samples. The dashed line is the $t^{1/2}$ law.

Accessing the regime of logarithmic growth for the two models requires a huge numerical effort, not only because of the pinning of the dynamics but also because enormous system sizes are needed to obtain a reasonable time-window of domain growth before we encounter finite-size effects. However, the time region here depicted is enough for our purposes. Therefore the above plots serve as a reference to understand the original results presented in Chapter 5.

Parte II

Percolation and Coarsening in Disordered Systems

Capitolo 4

Numerical Algorithms and Protocols

4.1 Introduction

In this work, we have used large-scale computer simulations to study the coarsening dynamics of the homogeneous and inhomogeneous Ising Model. Therefore we start with a brief generic introduction to numerical simulations to put this in a more general context.

The invention of computers has had strong effects on Statistical Mechanics and Physics in general. They are used to make the content of the theory explicit, performing tasks and calculations that are too long and tedious to be carried out by people. They have led to the development of a new method of scientific research, *simulation*, which is sometimes called the third branch of science, because it is at the same level as theory and experiment.

In a simulation of a physical system we define the equations which describe it, so that the computer can explicitly solve them. It can be considered an experiment, sometimes referred to as a *numerical experiment*.

Results of a numerical experiment can be directly compared to real experiments in order to determine the validity of the model, and to analytical solutions to judge the validity of the various approximations. Furthermore, it is possible to consider situations which are not accessible in real experiments, checking the theoretical predictions. Moreover, numerical experiments give access to microscopical details, such as position and velocity of each particle at any time. For example, one could numerically solve the equation of motion for the single molecules in a gas obtaining a complete dynamical description of the system at a microscopic level. Finally, we have a total control over a numerical experiment, which is impossible in real experiments. For example, keeping a system at a constant temperature while performing measurements on it is not an easy task. Likewise, it is practically impossible

to obtain a perfectly pure sample of a material. Instead, these ideal conditions are naturally set up in the numerical experiment while modeling the system.

4.2 Monte Carlo methods

There are two general classes of simulations: *molecular dynamics* and the *Monte Carlo methods*.

In molecular dynamics one simulates the evolution of a mechanical system composed of a large number of particles numerically solving its deterministic equations of motion. A clear advantage is that this procedure relies on the *true* kinetics of the constituents and therefore provides information on dynamical properties, such as viscosity, thermal conductivity, besides the static equilibrium ones, such as, for example, the equation of state. A serious disadvantage, however, is the complexity of the calculations, which require great computational power even for simple and small systems. This makes it impossible to simulate large-size systems and therefore to reach the thermodynamic limit.

Instead, as we have seen in Sec.1.2, the Monte Carlo methods allows one to simulate systems whose dynamics is not defined (as the Ising model). This is done relying on a stochastic dynamics – an example is Glauber kinetic evolution – whose equations often cannot be solved analytically. Therefore, these simulations are often the only available tool. The origin of the name has to do with the exploitation of random number sequences, which are encountered when playing roulette in the city of Monte Carlo as well as in this type of algorithms.

An important aspect of this procedure is that it not only allows one to simulate the dynamical evolution of a system, but also to investigate equilibrium properties. Indeed analytical estimations of the partition function are available only in very few ideal cases, and a direct summation of all the terms involved in its calculation is impractical even for a small system. For example, for 20x20 = 400 spins a computer should enumerate $2^{400} > 10^{100}$ configurations and therefore as many terms in the partition function of Eq. (1.4). Instead, by the Monte Carlo method one manages to sample phase space in the most statistically convenient way. This is the same situation pollsters face when they interview people to find out their preferences or opinions. In that case, the interviewers try to devise methods to obtain estimates from a small but statistically significant fraction of the population. Monte Carlo calculations are analogous. A sample of representative states is created by performing a biased random walk through the configuration space in which the relative frequency of visitations is consistent with the equilibrium en-

semble distribution. For example, Monte Carlo procedures treat the above 20x20 spin system successfully sampling only 10^6 configurations. That is because these schemes are devised so as to probe mainly those states that are statistically more important in the sum (1.4), performing what is called *importance sampling*. For example, for $T \to 0$ the vast majority of the 2^{400} configurations are of such a high energy that they have negligible weight $e^{-\beta E}$ in the Boltzmann distribution and sampling them is not needed.

In this work, we have used the Monte Carlo method to simulate the phase ordering kinetics of the $2d$ ferromagnetic Ising Model, implementing Glauber kinetic evolution. The rationale behind this method has been discussed at the end of Sec. 1.2. We have written a numerical code in FORTRAN which simulates the relaxation to equilibrium of the model after a quench from the infinite-temperature equilibrium state to a very low temperature, allowing us to analyze the dynamical process of coarsening. The code is reported in Appendix A and we proceed to explain its working principles in the next section.

4.3 Dynamical evolution

Our algorithm, specialized to the $2d$ case, starts by generating a configuration of the system in an initial state ν, in which the orientation of each spin is completely random. On average, we will have one half of the spins for each orientation, and this completely homogeneous configuration can be described as an infinite-temperature one, meaning that thermal fluctuations dominate the system and there is no correlation between spins. We choose such a $T_i = \infty$ initial state because it is easily prepared, but it can be shown that the effects of a finite T_i do not spoil the basic results obtained for $T_i = \infty$.

After preparing the initial condition we randomly pick out one of the spins in the lattice, indexed by i, with the random choice made with the aid of a pseudo-random number generator. Let us call ν' the configuration which would be generated from ν by flipping the spins s_i.

Then comes the core of the Monte Carlo method for the Ising Model. We introduce a stochastic dynamics specifying the transition probability $w_{\nu\nu'}$ from the configuration ν to ν'. This is done in accordance with the detailed balance condition (1.8) which we rewrite as

$$\frac{w_{\nu\nu'}}{w_{\nu'\nu}} = e^{-\beta \Delta E_{\nu\nu'}}, \tag{4.1}$$

using

$$\Delta E_{\nu\nu'} = E_{\nu'} - E_{\nu}. \tag{4.2}$$

As explained in Sec. 1.2, this condition describes a system for which a stationary state exists and amounts to equilibrium. Therefore, choosing transition rates that satisfy Eq. (4.1) allows us to simulate the relaxation to the new low-temperature equilibrium state of the system after the initial condition has become unstable due to the quench. In our algorithm we have adopted

$$w_{\nu\nu'} = \frac{1}{2}\left[1 - \tanh\frac{\beta}{2}\Delta E_{\nu\nu'}\right], \tag{4.3}$$

a generalization of the transition rates Glauber introduced in the original 1963 paper [1].

First of all, let us explicitly verify that this choice satisfies the detailed balance condition by substitution into Eq. (4.1)

$$\begin{aligned}\frac{w_{\nu\nu'}}{w_{\nu'\nu}} &= \frac{1-\tanh\frac{\beta}{2}\Delta E_{\nu\nu'}}{1-\tanh\frac{\beta}{2}(-\Delta E_{\nu\nu'})} = \frac{1-\tanh\frac{\beta}{2}\Delta E_{\nu\nu'}}{1+\tanh\frac{\beta}{2}\Delta E_{\nu\nu'}} = \\ &= \frac{\cosh\frac{\beta}{2}\Delta E_{\nu\nu'} - \sinh\frac{\beta}{2}\Delta E_{\nu\nu'}}{\cosh\frac{\beta}{2}\Delta E_{\nu\nu'} + \sinh\frac{\beta}{2}\Delta E_{\nu\nu'}} = \\ &= \frac{2e^{\left[-\frac{\beta}{2}\Delta E_{\nu\nu'}\right]}}{2e^{\left[\frac{\beta}{2}\Delta E_{\nu\nu'}\right]}} = e^{-\beta\Delta E_{\nu\nu'}}.\end{aligned} \tag{4.4}$$

After establishing whether the spin is flipped or not, the procedure is iterated by randomly choosing another one. A Monte Carlo step (MCS), the time unit, corresponds to L^2 single flip attempts.

Our aim is to study the coarsening process which takes place in a ferromagnet when it is cooled from $T_i > T_C$ to $T_f < T_C$. In order to do that, it is sufficient, after having prepared the initial state as discussed above, to evolve the spin system by means of the Montecarlo procedure using T_f in Eq. (4.3).

In our work we choose the limit $T_f \to 0$, for the reason that will be discussed in the next sections. This limit is built in the numerical routine by assigning a very small value, like 0.01, to the temperature T appearing in the transition rates (4.3) through [2] β.

4.4 Accelerated dynamics

In order to better understand the $T \to 0$ dynamics and its advantages, let us start from the simpler case of the pure system. For notational convenience,

[1] It is easily verified that this form, suited to any dimensionality and also to inhomogeneous systems, reduces to the simpler original Glauber's form in the homogeneous one-dimensional case.

[2] We set $k_B = 1$ in the code. Together with $|J| = 1$, this amounts to measuring the temperature in unit of J/k_B.

in the following we will use both the notation s_k or $s_{i,j}$ for the k-th spin located on a two-dimensional lattice site of cohordinates i, j. Let us consider the variation of energy $\Delta E^{pure}_{\nu\nu'}$, given in Eq. (4.2), resulting from the flip of such spin. In this case, the local field h^{loc}_{ij} is the sum of the values of the nearest neighboring spins:

$$h^{loc}_{ij} = J(s_{i,j+1} + s_{i,j-1} + s_{i+1,j} + s_{i-1,j}), \tag{4.5}$$

with J is the coupling constant, which we take equal to unity. This enters in the expression used to calculate $\Delta E^{pure}_{\nu\nu'}$:

$$\Delta E^{pure}_{\nu\nu'} = -2h^{loc}_{ij} s_{ij}, \tag{4.6}$$

a result which can be easily inferred directly from the Hamiltonian (1.2). If the local field is zero, flipping the spin causes no change in the energy of the system, $\Delta E^{pure}_{\nu\nu'} = 0$. Eq. (4.3) becomes $w_{\nu\nu'} = 1/2$. Such a move is therefore accepted with probability 1/2.

If the local field is not zero and the spin is unaligned with it, a flip causes the energy to decrease, namely $\Delta E^{pure}_{\nu\nu'} < 0$. In this case, Eq. (4.3) in the limit $T \to 0$ gives $w_{\nu\nu'} = 1$. In other words, energy-decreasing spin flips are always accepted.

Finally, if the local field is not zero and the spin is aligned with it, we would have $\Delta E^{pure}_{\nu\nu'} > 0$. This brings $w_{\nu\nu'} = 0$, namely energy-increasing spin flips are always rejected.

In all our simulations, the value of the temperature is always finite, however small. Therefore, strictly speaking the transition rates for the two last cases above – namely when the energy increases or decreases after the move, respectively – tend to the extreme values 0 and 1. For example, when trying to update a spin aligned with its nearest neighbors – one in the bulk of a domain – this would imply an exceedingly small probability of accepting this energy-increasing flip, of order $e^{-\Delta E/T}$ with $\Delta E = 8$ (since the coordination number is 4 on the square lattice considered here). For, say, $T = 0.01$ this probability is so small that no such a flip is observed during any reasonable computer time, therefore we do not even include attempts to flip these spins in the dynamical evolution of the systems. In other words, the algorithm considers the spins in the bulk not updatable and does not try to flip them at all, greatly increasing the efficiency of the simulation. Indeed, when domains start to form after the first MCSs, almost all the updatable spins are those located on the corners of interfaces, as shown in Fig. 4.1, and their number in the late stages of the evolution is a small fraction of the total. We call this accelerated dynamics *no-bulk-flip*: it improves the quality of the results and is exact in the limit $T \to 0$ we are considering, as shown in a number of studies [40] [41].

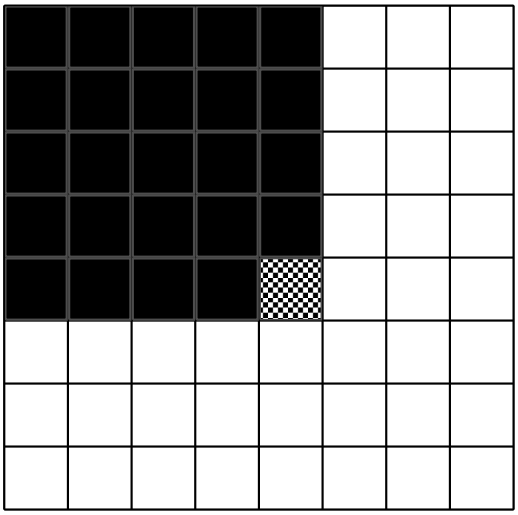

Figura 4.1: Enlargement of a portion of an interface. White spins are up, while black spins (including the checkered one) are down. All spins are aligned with the majority of their nearest neighbors, therefore are not updatable at $T_f = 0$, except the checkered one on the corner which is aligned with half of its nearest neighbors. Our algorithm flips such spins with probability 1/2.

In presence of quenched randomness, we use the same transition rates (4.3). What changes is the local field acting on a single spin. In the RFIM we must add to the energy variation of Eq. (4.6) the effect of the external random field $h_{ij}^{ext} = \epsilon_{RF}$ [3] acting on the spin. In this way, the energy variation $\Delta E_{\nu\nu'}^{RF}$ for the RFIM becomes

$$\Delta E_{\nu\nu'}^{RF} = \Delta E_{\nu\nu'}^{pure} + 2h_{ij}^{ext} s_{ij}. \tag{4.7}$$

Since we will always consider the limit $T \to 0$ with ϵ fixed, it is always $2h_{ij}^{ext} s_{ij} \ll \Delta E_{\nu\nu'}^{pure}$. Therefore, the random field significantly affects only the spins for which $\Delta E^{pure} = 0$, namely those with a zero pure local field (4.5). Let us call them paramagnetic spins and focus on a particular one, s_k, which is aligned to the external random field so that $s_k h_k^{loc} > 0$. According to Eq. (4.7), its flip would bring an energy variation $\Delta E_k = +2h_{amp}$, hence it is a disadvantageous energy-raising move. From Eq. (4.3), such a spin will be

[3] As seen at the end of Sec. 3.5

flipped with probability w_k equal to

$$w_k = \frac{1}{2}\left[1 - \tanh\frac{2h_{amp}}{2T}\right] = \frac{1}{2}\left(1 - \tanh\epsilon_{RF}\right). \tag{4.8}$$

For example, if $\epsilon_{RF} = 0.5$, we have $w_k \simeq 0.25$, meaning that on average 4 attempts will be needed to flip the spin. Following the same train of thoughts, if we need an average of 8 attempts if $\epsilon_{RF} = 1$ and 55 attempts if $\epsilon_{RF} = 2$. In pure systems, coarsening proceeds by smoothening of the domain interfaces through the update of paramagnetic spins. Therefore, it is evident that a sufficiently large random external field slows down the phase ordering kinetics of the system, as the presence of spins like the above s_k will tend to block the evolution of the interfaces, this mechanism being known as *pinning*. When this occurs, the system remains in a local energy minimum until an energy-raising fluctuation takes place, with a probability given by Eq. (4.8). These fluctuations are possible because the value of the temperature is always finite, although very small. Let us stress that the only relevant fluctuations are those for which $\Delta E^{pure} = 0$, namely exerted on paramagnetic spins, because $e^{\Delta E^{pure}/K_B T}$ is huge. Therefore, in this disordered case the dynamics can proceed only by means of thermal activation. At the same time, from the numerical examples after Eq. (4.8) it is clear that pinning effects soon become prohibitive as the strength of disorder increases. For this reason we limited ourselves to $\epsilon_{RF} \leq 2$.

In the RBIM, the energy variation in flipping a spin is the same as in Eq. (4.6), but the local field (4.5) must be modified using a different coupling constant for each nearest neighbor. Spins aligned with half of their nearest neighbors have an actual local field whose only contribution comes from the random components θ's of the couplings of Eq. (3.2). The θ's are extracted from a uniform distribution with support $[-\epsilon_{RB}T, +\epsilon_{RB}T]$ and so the sum of four of them is typically also of order $\pm\epsilon_{RB}T$. Spins for which $\Delta E^{RB}_{\nu\nu'} \simeq +\epsilon_{RB}T$ originate the pinning phenomenon, similarly to what observed in the RFIM.

4.5 Growing length

The growing length $R(t)$, the typical size of domains, is of primary importance in the study of coarsening as it is evident from Secs. 1.1, 1.3 and 3.3. A related quantity is the number N_{def} of *defects*, namely the couples of unaligned spins in the system.

Defects are located on domains' boundaries, therefore N_{def} is proportional to the length of the typical domain interface. This is in turn proportional to the number N_{dom} of domains multiplied by their typical size $R(t)$. Therefore

we arrive at

$$N_{def} \propto N_{dom} \cdot R(t). \tag{4.9}$$

The area of a typical domain is proportional to $[R(t)]^2$, therefore the number of domain is proportional to the total number N of spins – the surface of the lattice – divided by $[R(t)]^2$, namely

$$N_{dom} \propto \frac{N}{[R(t)]^2}. \tag{4.10}$$

Inserting Eq. (4.10) in Eq. (4.9) gives

$$R(t) \propto \frac{N}{N_{def}}, \tag{4.11}$$

The algorithm uses this argument as follows: it counts the number N_{def} of defects, scanning the lattice, and then calculates $R(t) = N/N_{def}$. The multiplicative constant is irrelevant as we are interested in how $R(t)$ grows with time.

We can confirm that $R(t)$ obtained from Eq. (4.11) has the meaning of typical domain size through some examples. At $t = 0$, in the completely disordered state, $N_{def} \simeq N$, so $R(t) \simeq 1$. In a configuration with only two domains of spins separated by two interfaces of length L , we would have

$$R(t) = \frac{N}{2L} = \frac{L}{2},$$

in accordance with the fact that each domain occupies on average half of the available space

Let us notice that values of $R(t)$ are meaningful until $R(t) \leq L$, as of course the typical dimension of a part of the system cannot exceed the system size. Therefore, the values $R(t) > L$ obtained through Eq. (4.11) must be discarded. For example, in a completely magnetized configuration with all the spins pointing in the same direction, $N_{def} = 0$ and the growing length would diverge according to the above definition. Finite size effects start affecting the growth when $R(t)$ starts to be comparable to L.

4.6 Properties of clusters

In order to identify and quantify percolative effects in the phase ordering kinetics of the system, we need to measure a number of properties of the clusters present in the system (see Secs. 2.4 and 2.3).

First, the algorithm counts the clusters and measures their masses. To do so, an array V is introduced composed of N entries, the value of each entry recording if the site has yet been visited in the search (see below). Then the algorithm proceeds as follows :

- An initial site is selected. This is classified as visited, namely the corresponding entry of V changes from 0 to 1
- The algorithm checks how many of the nearest neighbors of the initial site belong to the same cluster, i.e. their spins are aligned with the initial one.
- The same procedure is repeated starting from each of the sites recognized as belonging to the cluster, provided they were not visited before, namely if the corresponding V entry is still 0.
- When no more such sites are found, all the elements of the cluster have been counted.
- Another unvisited site is then selected and the procedure is iterated with the difference that, when a new cluster, say the n-th is explored the entries of V change from 0 to n. In this way we also record, for each site, the cluster to which it belongs.

Recognizing all the clusters allows one to compute the pair connectivity $g(r)$, defined in Sec. 2.2.2. The algorithm stops when all the sites have been visited.

By following the interface of a cluster, the algorithm measures it and check if the cluster crosses the boundaries of the system. In such case, the winding numbers, introduced in Sec. 2.3, are also determined.

The largest cluster is singled out as its hull is best suited to the measure of the winding angle (Sec. 2.4), since a larger interval of distances x is represented. The winding angle for each distance x is measured cumulatively by following the contour.

Capitolo 5

Coarsening and Percolation

5.1 Introduction

In this chapter, we report and explain a number of results about the influence of percolation in the coarsening of the quenched ferromagnetic Ising Model. These results concern pure and inhomogeneous $2d$ models. For the former they partly build on previous findings, nonetheless confirming them independently, while they are completely original for the latter.

We will start by what was already known for the homogeneous system, underlining a recent result [46] about the time scale over which percolation starts to affect the phase ordering kinetics. Then, we will present numerical data for the various quantities defined in Secs. 2.4, 2.3 and 4.5 which allow one to identify and quantify the aforementioned effects. We will see that these tools are particularly apt to extend the analysis to disordered ferromagnets. Indeed, we will explicitly compare results for the homogeneous and inhomogeneous systems, touching upon the concept of *superuniversality*, which is relevant in the study of disordered systems.

In the following we will always consider the coarsening of the $2d$ ferromagnetic Ising Model on the square lattice with PBC, after a quench from infinite to a very small temperature. Data are obtained by means of Monte Carlo simulations which implement Glauber NCOP dynamics, as discussed in Sec. 4.3. We will make large use of the concept of spin cluster, as defined in Sec. 2.4, which is a percolation-inspired way of defining a domain of aligned spins.

5.2 An unexplained connection

Let us consider the coarsening system without disorder, characterized by a patchwork of domains whose typical linear size $R(t)$ grows as

$$R(t) = t^{1/2}, \tag{5.1}$$

as explained in Sec. 1.3.

At some point, $R(t)$ will equal the lattice size L. This happens after a time t_{eq} which we can estimate by imposing $R(t_{eq}) \simeq L$, leading to

$$t_{eq} \simeq L^2. \tag{5.2}$$

t_{eq} represents the *minimum* time scale over which the system equilibrates – we will see shortly why. There is at least one *correlated* spanning cluster in the system after this time, where correlated stands for a cluster which has reached the boundaries by gradual growth of its size and which is internally equilibrated, like that in Fig. 5.1.

It is possible to classify the spanning clusters appearing at t_{eq} according to their topological properties, namely the number of times they wrap around the torus in the vertical and horizontal direction [42]. We will label a cluster winding n times in the horizontal direction and m times in the vertical one by its winding numbers (n, m). The three most probable configurations are :

- A straight stripe spanning the system, of which an example is given in Fig. 5.1 for vertical crossing. Focusing on the interface of this stripe, we see that once formed this configuration persists indefinitely, in the sense that its winding numbers do not change although the domain wall becomes smoother. Indeed, when the system equilibrates, the entire interface will be completely straight. Since flipping a spin on a straight interface is energetically disadvantageous, this is impossible at $T_f = 0$. As shown in Fig. 5.2, when this cluster form, the system is stuck in such an infinitely lived metastable state and does not reach the completely ordered ground state (GS). We refer to a vertical stripe as $(0, 1)$ and to horizontal one by $(1, 0)$.

- A cluster which can be seen as the union of a vertical and a horizontal stripe, namely crossing the system in the two directions. As time goes on, this kind of cluster engulfs the entire system that rapidly reaches the GS. Therefore, we can also consider it stable. As it surely leads to a situation in which all spins are aligned, we convene to label it by $(0, 0)$.

- A diagonal crossing, illustrated in Fig. 5.3. This kind of stripe is not stable, because the spins on its interface (see Fig. 5.4) can be flipped with no variation in the energy of the system; we have called them paramagnetic spins in Sec. 4.4. On the interface, after a spin is flipped others will become updatable. Therefore this configuration decays to

the GS. However, the process of flipping paramagnetic spins is slow, and this structure survives quite a long time. Attainment of the GS occurs in fact over a time scale $t_{diag} \simeq L^3$ [43]. This explains why we said that t_{eq} is the smallest time scale needed to reach equilibrium: the final configuration is attained in a time of order t_{eq} for the previous cluster of type $(0,0)$, but if the system ends up with the present ones the time needed is larger. As our simulations run up to final times $t \ll t_{eq} \ll t_{diag}$, we will not observe this decay. We label a diagonal crossing $(-1,1)$ if the stripe spans the system from left bottom to top right and $(1,-1)$ if spans from left top to bottom right.

These are the most probable types of wrapping cluster one can have at t_{eq}. More exotic types of crossings are possible, but since their probabilities of occurrence are exceedingly small, they are more difficult to access by means of numerical simulations. Denoting with $\pi_{n,m}$ the probability of finding an (n,m) spanning cluster, it is found numerically that for a coarsening process in the conditions we have adopted [42]

$$\pi_{0,1} = \pi_{1,0},$$

$$\pi_{1,-1} = \pi_{1,1}.$$

We limit ourselves to these three types, for which we report in table 2.2 the relevant information, including the probability of occurrence.

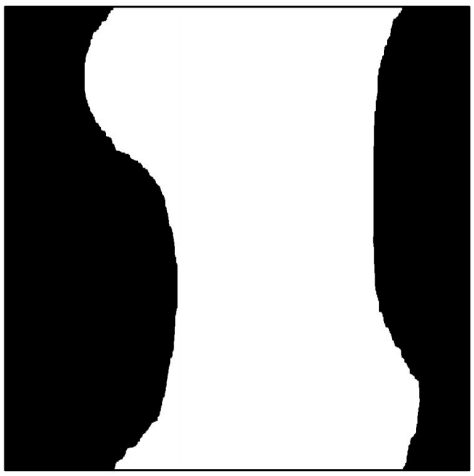

Figura 5.1: Snapshot of a homogeneous $2d$ IM, coarsening after a zero-temperature quench, which has reached a vertical straight stripe state, labeled by $(0,1)$. Up spins are white and down spins are black.

In Fig. 5.5 we have plotted the probabilities $\pi(n,m)$ of the above types of crossings against time obtained from the simulation of a homogeneous IM

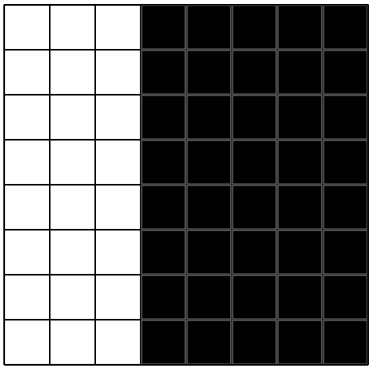

Figura 5.2: A straight portion of the interface in Fig. 5.1. Up spins are white and down spins are black. No spin can be flipped at $T_f = 0$, because they are all aligned with three nearest neighbors.

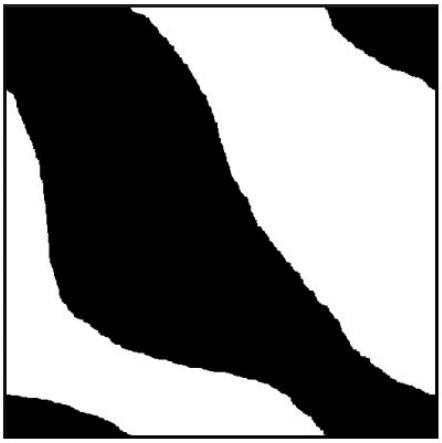

Figura 5.3: Snapshot of a homogeneous $2d$ IM, coarsening after a zero-temperature quench, which has reached a diagonal stripe state, labeled by $(1, 1)$. Up spins are white and down spins are black.

of size $L = 1280$. We see how the curves reach the corresponding values listed in Table 5.1. These values, here as in all the following, are marked by dashed lines representing, from the upper to the lower one, $\pi_{0,0}$, $\pi_{1,0} + \pi_{0,1}$ and $\pi_{1,-1} + \pi_{1,1}$.

Now, let us notice one of the fundamental facts which has inspired this work. The values reached by the curves in Fig. 5.5 are exactly those of Table 2.2 of Sec. 2.3, relative to a lattice at percolation threshold p_c [23].

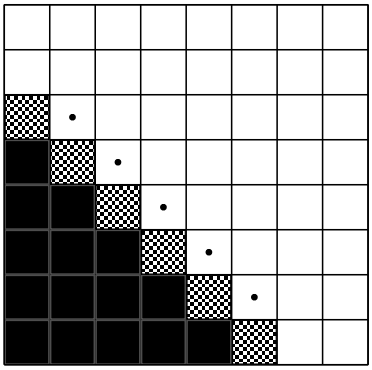

Figura 5.4: Enlargement of a portion of the interface for a diagonal stripe in Fig. 5.3. White spins (including the dotted ones) are up and black spins (including the checkered ones) are down. All checkered spins on the interface of the black domain can be flipped, as well as the dotted spins on the interface of the white domain. This explains why diagonal stripes are unstable.

Type of crossing	Probability	Stability
Horizontal and vertical	$\pi_{0,0} = 0.61908$	Stable, leading to GS
Straight, horizontal	$\pi_{1,0} = 0.1694$	Stable , GS not reached
Straight, vertical	$\pi_{0,1} = 0.1694$	Stable , GS not reached
From left bottom to right top	$\pi_{1,-1} = 0.0209$	Decaying to GS over t_{diag}
From left top to right bottom	$\pi_{1,1} = 0.0209$	Decaying to GS over t_{diag}

Tabella 5.1: Summary of the three types of crossings considered for a pure $2d$ IM on a square lattice with PBC, with probability of occurrence and stability properties. $t_{diag} \simeq t^3$ is the time scale needed for diagonal stripes to decay.

This means that the probability of having in the coarsening system at t_{eq} a spanning cluster with winding numbers (n, m) is the same of having a percolating cluster with the same winding numbers on a lattice at the critical occupation probability p_c. For percolation, mathematical proofs are available for the values of the crossing probabilities [23] [42]. As for coarsening, they are indisputably confirmed by numerical evidence (as our data reported in Fig. 5.5), but to date no progress has been made toward framing these results in a mathematical theory, neither this connection to the radically different problem of random percolation is explained.

In the coarsening problem, the initial fraction of spins of a given sign approaches $1/2$ in the large-size limit and does not change significantly over t_{eq}. As the critical occupation probability on the square lattice is $p_c \simeq 0.59$, the

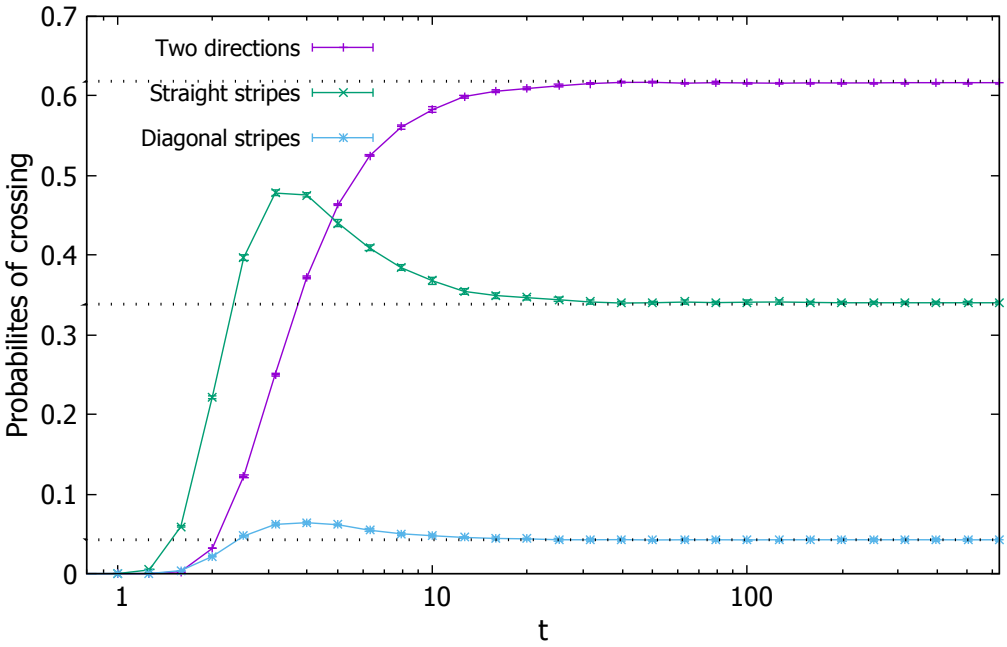

Figura 5.5: Numerical data for the probabilities $\pi(n, m)$ of the three different types of crossings of table 5.1 for a homogeneous $2d$ IM on a square lattice with $L = 1280$ and PBC, quenched from $T_i \to \infty$ to $T = 0$. The three values do not sum to one because other types of crossings occur with small probabilities.

above correspondence is not trivially due to a spin configuration that can be mapped onto the critical point of percolation by switching from spin variables to gas lattice variables.

That being said, this quantitatively precise connection comes as completely unexpected. Indeed, coarsening is a non-equilibrium statistical mechanical problem whose dynamical evolution is driven by spin interactions, while percolation is a purely geometrical problem where sites are uncorrelated.

In the rest of this work, we will further explore this surprising result and its consequences, in particular extending the analysis to inhomogeneous systems for the first time.

5.3 A quickly sealed fate

Given that structures akin to those of random percolation are present at t_{eq} under the form of correlated clusters which have reached the boundaries of the lattice, one may ask if it is possible to pinpoint at what time they have appeared in the system for the first time. This question has been the subject

of a recent study [46], concerning the homogeneous model. We resume the essential facts here.

According to the analysis in [46], there exists a time $t_p \ll t_{eq}$ which divides the evolution of the system into two stages. For $0 < t < t_p$, spanning clusters are either absent or are instable. They may appear, but then they are cut, disappearing until another one forms. This happens typically $10-20$ times before t_p. Then, at t_p a stable percolating cluster establishes.

The situation described insofar can be traced back by looking at Figs. 5.6, 5.7, 5.8, 5.9, where snapshots of the evolution of a homogeneous $2d$ IM on a square lattice, coarsening after a quench from infinite to zero temperature, are shown.

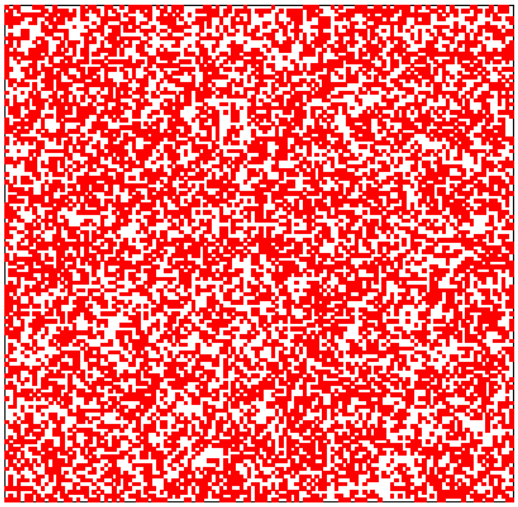

Figura 5.6: Snapshot of the evolution of a $2d$ IM on a square lattice with $L = 128$, coarsening after a quench at $t = 0$ from infinite to zero temperature. Up spins are shown in red, down spins in white. At t=0, no percolating cluster is present.

At the time of the quench $t = 0$ a percolating cluster typically does not exist, since the initial occupation probability for the two spin orientation is $p = 1/2$, which is different from the critical occupation probability $p_c \simeq 0.59$ for the square lattice. This can be seen in Fig. 5.6.

A first percolating cluster appears very soon, in this example at $t_f = 0.57 \ll t_{eq} \simeq L^2$, see Fig. 5.7. However, it quickly disappears at $t \simeq 1$ (Fig. 5.8). It is replaced by another percolating cluster which also disappears, and so on until at time $t_p \simeq 7 \ll t_{eq}$ a stable percolating cluster of type $(0, 0)$ (see Table 5.1) establishes. This is shown in Fig. 5.9. The winding numbers of this cluster will remain unchanged, even though the holes inside it will fill, so that at t_{eq} it will constitute a stripe of type $(0, 0)$

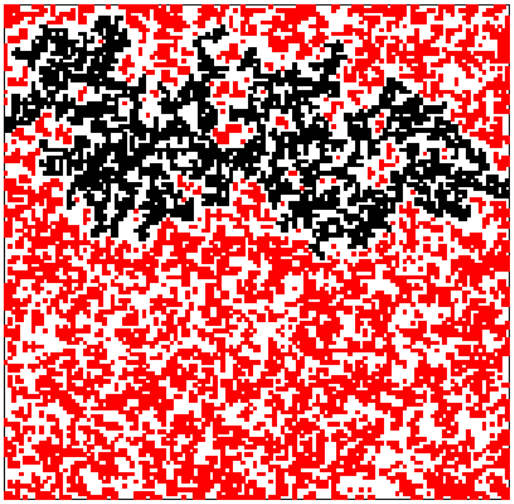

Figura 5.7: Same as Fig. 5.6, but for $t_f = 0.57$. A percolating cluster is shown in black.

Here, there are four important points to make:

1. The cluster formed at t_p will persist indefinitely in the evolution. It will not be cut, as those appearing and disappearing before t_p.

2. This spanning cluster will already have one of the couples of winding numbers of Table 5.1, and these numbers will remain either unchanged or will change over the time scale $t_{diag} \gg t_{eq}$ if the crossing is diagonal.

3. Even though the winding numbers of these structures are the same of those observed at t_{eq}, other geometrical features may be very different, as it is obvious since $t_p \ll t_{eq}$ and correspondingly $R(t_{eq}) \ll R(t_{eq} \simeq L$, meaning that domains change a lot in going from t_p to t_{eq}. In particular, the objects found at t_p are not ordered domains internally equilibrated, with the non-equilibrium behavior taking place only on the interface, as it is at t_{eq}. Instead, as shown in the snapshots , this percolating cluster is a highly-ramified structure with many holes, i.e. internal regions of unaligned spins, which in fact exhibits the morphological properties of critical percolation [1]. What matters is that the cluster can only fill in : the inner holes will disappear over the time scale t_{eq}, but it will not be cut and the winding numbers will remain the same up to t_{eq} at least.

4. Perhaps most importantly, t_p scales with the system size as

$$t_p \simeq L^{\alpha_p}, \tag{5.3}$$

[1] This point will be quantitatively expressed and verified in Secs. 5.5 and 5.6

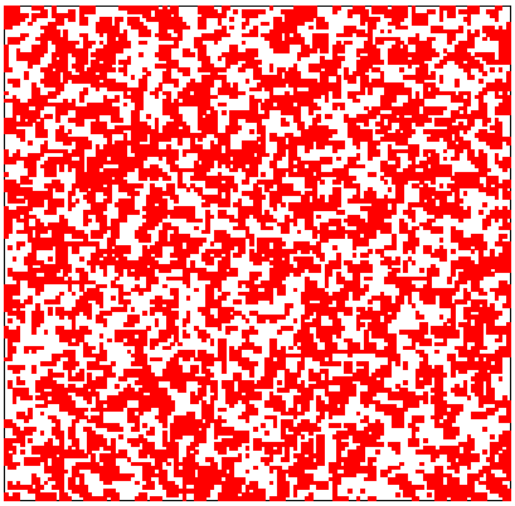

Figura 5.8: Same as Fig. 5.6, but for $t_f \simeq 1$. The percolating cluster present in Fig. 5.7 has disappeared.

where the constant prefactor is of order 1 and the exponent α_p is dependent on the geometry of the lattice. For our purposes, it suffices to know that α_p is very close to 0.5 for the square lattice and comparable $(0.33 - 0.5)$ for other geometries considered in [46], like the kagome, triangular and bow-tie lattice.The result (5.3) shows that t_p is not a microscopic time, since it grows unbounded with the size L of the system. Moreover, as $\alpha_p < 2$, it is also $t_p \ll t_{eq}$, hence the formation of the percolating structure occurs – so to say – in the middle of the coarsening stage. As we will discuss later in Sec. 5.3.2 this has important consequences.

5.3.1 Evidence for the existence of t_p

There are different techniques whereby it is possible to confirm Eq. (5.3). Given the importance of this result, we explain here one of these methods, which also informs us on the value of the exponent α_p. The basic idea is that if a percolating cluster appears at t_p and persists later in the evolution of the system, then two systems with an identical spin configuration at t_p evolving under different thermal histories should be strongly correlated in some sense.
One quantity which can measure this correlation and give it a precise meaning is the *overlap*. This is defined as follows : at $t = t_w$ one makes two copies of the configuration $s_i(t_w) = \sigma_i(t_w)$, and lets them evolve with different thermal noises. This means that flips of spins are attempted with different random numbers. The overlap between the two clones at time

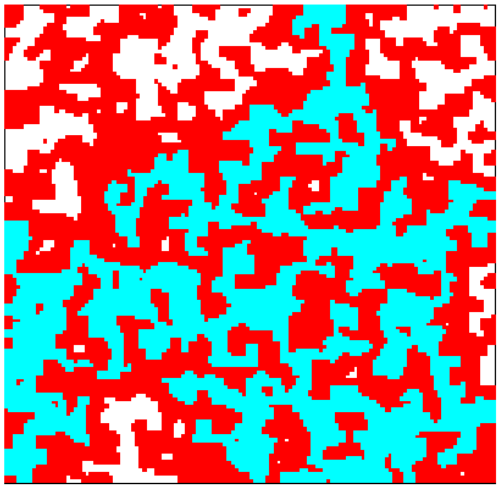

Figura 5.9: Same as Fig. 5.6, but for $t_p \simeq 7$. At time t_p, a stable percolating cluster, shown in blue, establishes. The winding numbers of this cluster will remain unchanged.

$t \geq t_w$ is

$$q_(t_w, t, L) = \frac{1}{N} \sum_i \langle s_i(t) \sigma_i(t) \rangle, \tag{5.4}$$

where the angular brackets indicate an average over different realizations of the evolution of the two replicated systems. If we make the copy of the system at $t_w < t_p$, there is no reason why the two different thermal histories will take the clones in a correlated state. Therefore, the value of the overlap will decay to zero over time. Instead, by letting t_w go beyond t_p, we create the two clones when the percolating structure has already established. As such a cluster spans the system and is stable (in the sense discussed above), we expect the two copies to be strongly correlated for all subsequent times. Accordingly, the value of the overlap will stabilize to a finite value for $t \to \infty$.

This argument can be checked by computing the asymptotic overlap $\lim_{t\to\infty} q(t_w, t, L)$ for various t_w and for different sizes. Selecting the values of t_w which keep the overlap constant as L varies provides the relation between t_p and L, because only $t_w = t_p$ can give rise to the same correlation over varying system sizes. For the square lattice, if $t_w = L^{0.5}$ the overlap decreases with L, while it increases if $t_w = L^{0.55}$, as shown in Fig. (5.10). As these variations are very small, we conclude that Eq. (5.3) holds with $0.5 < \alpha_p < 0.55$.

5.3.2 Consequences of the existence of t_p

There are many reasons of interest in the existence of a new characteristic timescale t_p expressed by Eq. (5.3). In the first place, t_p diverges in the

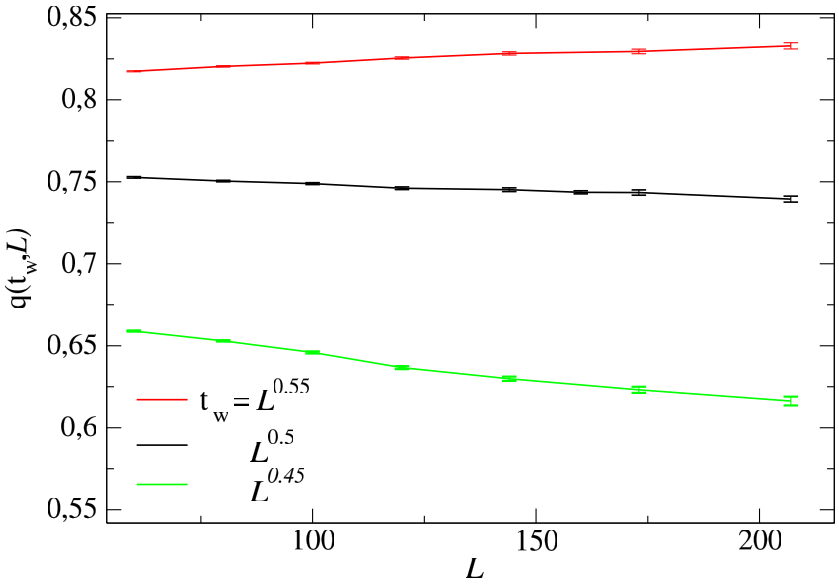

Figura 5.10: The L dependence of the asymptotic overlap $\lim_{t\to\infty} q_(t_w, t, L)$ in the $2d$ IM on a square lattice with PBC, quenched from infinite to zero temperature. For each curve, t_w is chosen as reported in the key. The value of the overlap has been measured at $t \simeq 4 \cdot 10^6$ and data have been averaged over $10^5 - 10^6$ samples. The asymptotic overlap remains approximately constant for an exponent α_p very close to 0.5.

thermodynamic limit, although slowly. Therefore, we are not dealing with an irrelevant detail which only affects the dynamics of the system on a microscopic time scale. At the same time, comparing Eqs. (5.2) and (5.3), we see that for sufficiently large sizes – in particular in the thermodynamic limit – $t_p \ll t_{eq}$. In other words, the stable percolating cluster appears in the early stages of the coarsening process. The presence of a such a structure, occupying a significant portion of the lattice, will characterize the large scale properties in the scaling regime and will determine if the system reaches the GS or a frozen stripe state. The fate of the system is *sealed* very soon in its evolution, right at t_p.

Another important consequence concerns dynamical scaling. Going back to the discussion of Sec. 1.3, we started from the fact that there is only one relevant length in the coarsening process, given by the growing length $R(t)$. Indeed, this has been the commonly shared idea during more than fifty years since the birth of this field of study. As a matter of fact, the *extra* time scale t_p introduces a new typical length L_p, namely the typical

size of the domains when the first stable percolating structure forms. This new length L_p is given by

$$L_p = R(t_p) = L^{\alpha_p/z}, \tag{5.5}$$

where z is the exponent of Eq. (1.11). For the case at hand, we have $z = 2$ and $\alpha_p \simeq 1/2$, therefore $L_p = L^{1/4}$. We will mainly refer to Eq. (5.5) in the following, since it is better to substitute time with the growing length, meaning that we take the latter as the parameter describing the evolution of the system. This is licit since t_p and L_p are in a one-to-one correspondence. It is also very convenient, because the central question we will address is if and how this result can be transferred to the inhomogeneous models of Chapter 3, whose stochastic dynamics is subject to pinning. Hence it does not make sense to compare a pure and a disordered magnet at some time t, because the kinetics of the latter is much slower. We must do it for equal values of the typical domain size, i.e. when they have reached the same stage in the phase ordering process. Accordingly, from now on we will always reason in terms of $R(t, \epsilon)$.

The existence of L_p requires a generalization of the dynamical scaling hypothesis, because in equations like (1.10) L_p plays a companion role to $R(t)$. Dynamical scaling is arguably the distinguishing feature of coarsening. Therefore, the presence of another characteristic length associated to percolation is a highly remarkable fact, previously ignored, which must be taken into account in the analysis. We discuss this point considering the correlator $C(r, t) = \langle s_i(t) s_j(t) \rangle$ with $r = |i - j|$, for which dynamical scaling implies Eq. (1.9) that we rewrite here for simplicity

$$C(r, t, L) = c\left[\frac{r}{R(t)}\right], \tag{5.6}$$

with $c(x)$ a scaling function.
However, as we have said, the presence of t_p introduces the new characteristic length L_p. Therefore, the scaling form for $C(r, t, L)$ should be upgraded to a two-parameter scaling function as follows

$$C(r, t, L) = f\left[\frac{r}{R(t)}, \frac{L_p}{R(t)}\right], \tag{5.7}$$

with a new two-variable scaling function $f(x, y)$. One way to check which equation is correct is to collapse data for $C(r, t, L)$ at different measurement times t.

If Eq. (5.6) is correct, one obtains a perfect collapse by simply plotting the curves against $r/R(t)$, according to the usual procedure.

Instead, if the correct scaling form is Eq. (5.7), in order to obtain the collapse we have to follow a different strategy. Indeed, plotting against

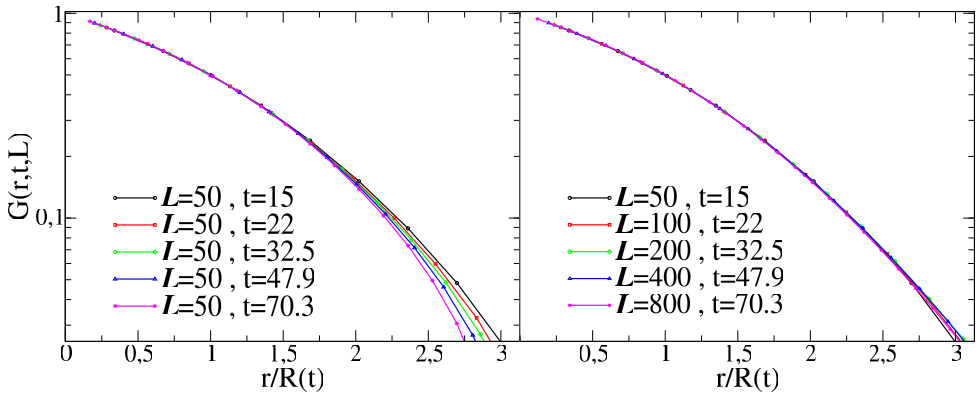

Figura 5.11: $G(r,t,L)$ agains $r/R(t)$ in the $2d$ IM quenched from infinite to zero temperature. The system sizes and measuring times are given in the keys. The collapse is much better taking into account the length L_p associated to percolation, according to Eq. (5.7). Data have been averaged over $2 \cdot 10^4$ samples.

$r/R(t)$ is not enough in this case, but we have to keep the value of the second entry of the scaling function f constant. This means that, for two different measurement times t_i and t_j we have to use two different system sizes L_i and L_j, so that

$$\frac{L_{pi}}{R(t_i)} = \frac{L_{pj}}{R(t_j)},$$

which using the relation $L_p = L^{1/4}$ valid for the square lattice becomes

$$\frac{L_i^{1/4}}{R(t_i)} = \frac{L_j^{1/4}}{R(t_j)}.$$

In Fig. 5.11, reported after [46], we compare the results of the two procedures. In the first panel (a) the collapse has been tested following Eq. (5.6), by simply plotting against $r/R(t)$ ignoring the additional length L_p. We observe a systematic downward spreading of the curves, signaling that the one-parameter scaling of Eq. (5.6) is not correct.

In panel (b), where system sizes and times have been chosen as indicated in the key, namely as to keep the second entry of the scaling function f in Eq. (5.7) constant, the quality of the collapse is much better.

We precise that in both cases times are such that we are in the scaling regime, namely when domains are well formed but small as compared to the system size.

Let us remark that the measurement times in Fig. 5.11 have been chosen specifically to highlight the difference between Eq. (5.6) and (5.7). Indeed,

when $R(t) \gg L_p$ the validity of Eq. (5.6) gets progressively restored since one expects for the two scaling functions

$$\lim_{y\to 0} f(x,y) \simeq f(x,0) = c(x).$$

Instead, the measurement times t_i and system sizes L_i are chosen so that $L_{pi} \gtrsim R(t_i)$

The result contained in Fig. 5.11 not only confirms the existence of another characteristic length L_p influencing the scaling properties of coarsening systems, but also confirms Eq. (5.5) independently. In the next sections, we will provide further evidences of this using other methods.

5.4 Probabilities of crossing

We now focus on the probabilities of the three types of crossings of table 5.1. In the previous section we have seen how the new characteristic length L_p influences the scaling functions. Another way to test it is to try to collapse the curves associated to the crossing probabilities $\pi(n,m)$ for different system sizes by plotting them against $R(t)/L_p$, being $L_p = L^{0.25}$ for the square lattice. Indeed, these probabilities depend on time until they reach their limit values and, if dynamical scaling holds we should find

$$\pi(n,m) = p_{nm}\left(\frac{R(t)}{L_p}\right), \tag{5.8}$$

where p_{nm} is another scaling function. This is the first time this method is used to check Eq. (5.5). The results are shown in Fig. 5.12 and we do not obtain the expected collapse. Conversely, plotting against $R(t)/L^{0.167}$ the collapse becomes very good, as it is evident in Fig. 5.13. However, when the size increases significantly, the collapse of the curves becomes progressively worse. For $L = 640$ we already see a deviation from $L = 160$ and including curves for larger sizes, the collapse is globally better for a slightly larger value of the exponent, namely $\simeq 0.175$ for $L > 10^3$ (not shown). Clearly, this method is not the best suited to measure the values of α_p. We are currently studying this point, to understand why the exponent changes with the sizes. The fact that by significantly increasing the size we obtain a good collapse for a somewhat larger value of the exponent might suggest that this quantity is very sensitive to finite size effects and that, were we able to go to huge sizes, we could perhaps recover the value 0.25. Moreover, preasymptotic effects might also play a major role, because when $x = R(t)/L^{0.25}$ is small in the plot in Fig. 5.12, the value of $R(t)$ is of the order of unity. We know that in this situation the law $R(t) = t^{1/2}$ is not obeyed, as shown in Figs. 3.2 and 3.3 in Sec. 3.6.

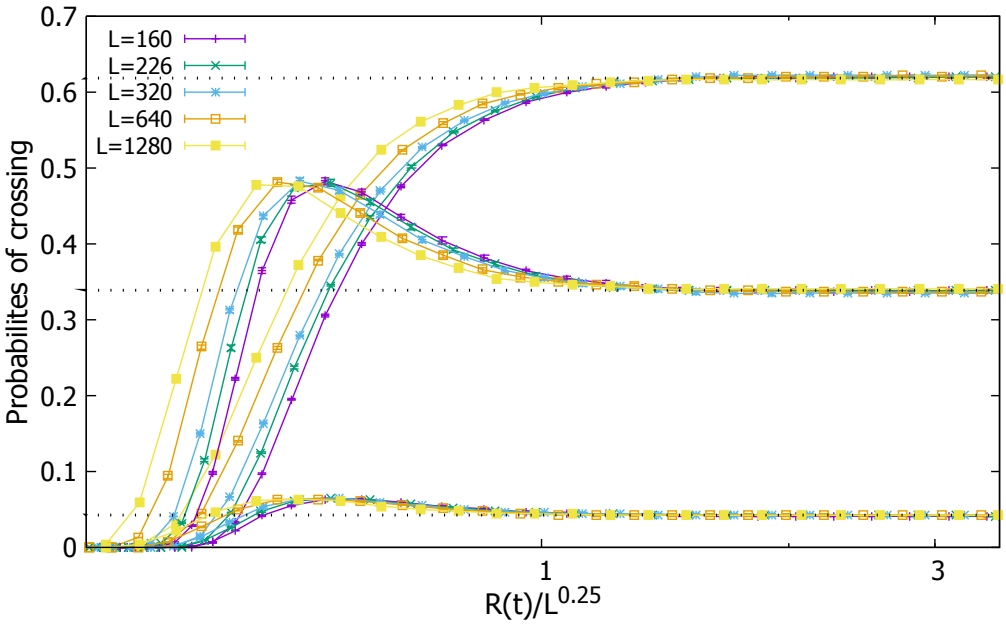

Figura 5.12: Probabilities of crossing against $R(t)/L_p$ for the homogeneous $2d$ IM on square lattice with PBC, quenched from infinite to zero temperature. Data have been averaged over 10^5 samples.

Despite all the above, we regard the very good collapse obtained in Fig. 5.13, as consistent with the scaling form in Eq.(5.8) with a value of α_p still depending on some finite-size, finite-time effects. Although, evidently, this method is not the best suited to measure the value of α_p, it turns out to be very efficient to extend the analysis to the disordered case. We have applied precisely the same line of reasoning to the RBIM and RFIM presented in Chapter 3 and in Fig. 5.14, 5.15 and 5.16 we report the same plot as in Fig. 5.13 for, respectively, $\epsilon_{RF} = 1$, $\epsilon_{RF} = 2$, $\epsilon_{RB} = 0.8$.

We see that the collapse is excellent by plotting against $R(t, \epsilon)/L^{0.16}$ for sizes up to $L = 1280$. The scaling form 5.8 seems to work exactly in the same way for the pure system and for the RFIM and RBIM.

This is a surprising result. As we have seen in Sec. 3.3, for these values of the parameter ϵ, disorder strongly affects the domain growth, drastically modifying the form of $R(t)$ as seen in Fig. 3.2 and 3.3 in Sec. 3.6. Nonetheless, percolation influences the coarsening process in the same way, meaning that its effects are very robust. Indeed, the correspondence with the values of table 5.1 remains equally impressive. As some kind of disorder is present in every real system, this result are expected to be relevant for future experimental and theoretical studies.

We have found analogous results, not reported, for smaller values of ϵ_{RF}

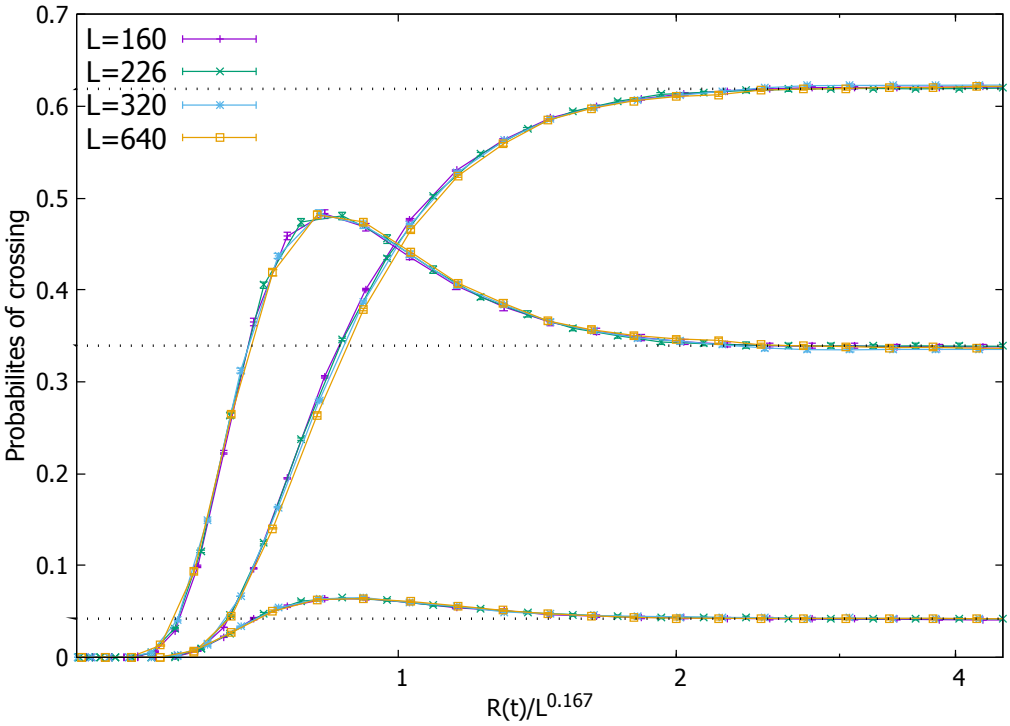

Figura 5.13: Same as Fig. 5.12, still for the homogeneous model but plotting against $R(t)/L^{0.167}$.

and for other values of ϵ_{RB}. Moreover, since for all the following quantities the results for the RBIM are precisely the same as those for the RFIM, we will usually report data for the latter.

We conclude by stressing the main result of this Section, namely that also in the presence of quenched randomness we find a typical length $L_p = L^{\alpha_p/z}$ separating an early stage in which there are no stable percolating structures from a late stage in which they exist. This new characteristic length has the same effects on the scaling function for $\pi_{nm}(t)$ in the homogeneous and in the disordered systems.

5.5 Winding angles of hulls

Recalling the discussion of Sec. 2.4, we infer that measuring the average squared winding angle for the hulls which form in the coarsening systems might confirm that it is actually critical random percolation which is at play in the early stages of the phase ordering process. As a side-result, this measurement, which had never been performed on disordered systems, will

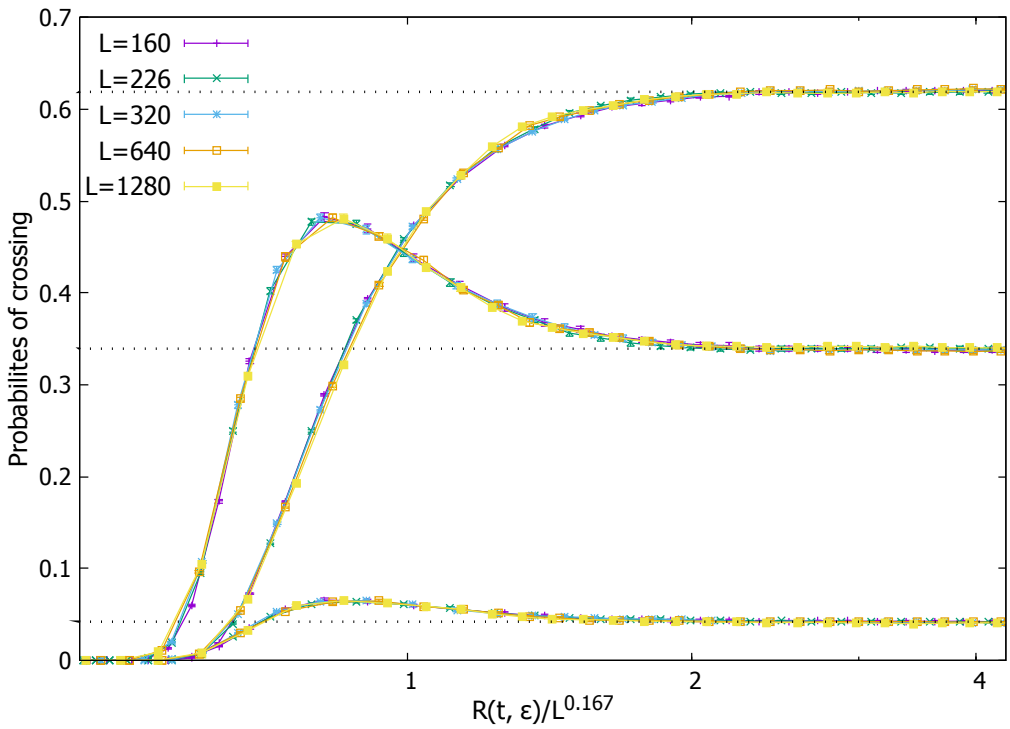

Figura 5.14: Same as Fig. 5.13, for RFIM with $\epsilon_{RF} = 1$ and substituting $R(t)$ by $R(t, \epsilon)$.

also provide a test of dynamical scaling which is very interesting in the light of the nature of the property considered, as we will explain.

We rewrite here Eq. (2.23) for reference :

$$\langle \theta^2(x) \rangle = a + \frac{4k}{8+k} \ln x, \tag{5.9}$$

recalling that x is the distance between two points of a hull. We know that $k = 6$ for percolation, therefore we want to measure the same parameter for coarsening and compare. For simplicity, in the following we will refer to the quantity $\langle \theta^2(x) \rangle$ as the winding angle. Measurements are performed on the largest cluster which is the best suited to ensure the continuum limit $x \gg 1$, necessary for Eq. (5.9) to hold, and to access large distances x.

In Fig. 5.17 we report results for the pure model of size $L = 640$, for various measurement times. The dashed blue line is the analytical curve, namely Eq. (5.9) with $k = 6$.

For each curve we observe two behaviors : an initial part (short distances along the hull) in which the analytical expression is not followed, while at large distances numerical data are very well described by Eq. (5.9). Indeed, we fitted the slope of the purple curve corresponding to $R(t) = 2.68$ and

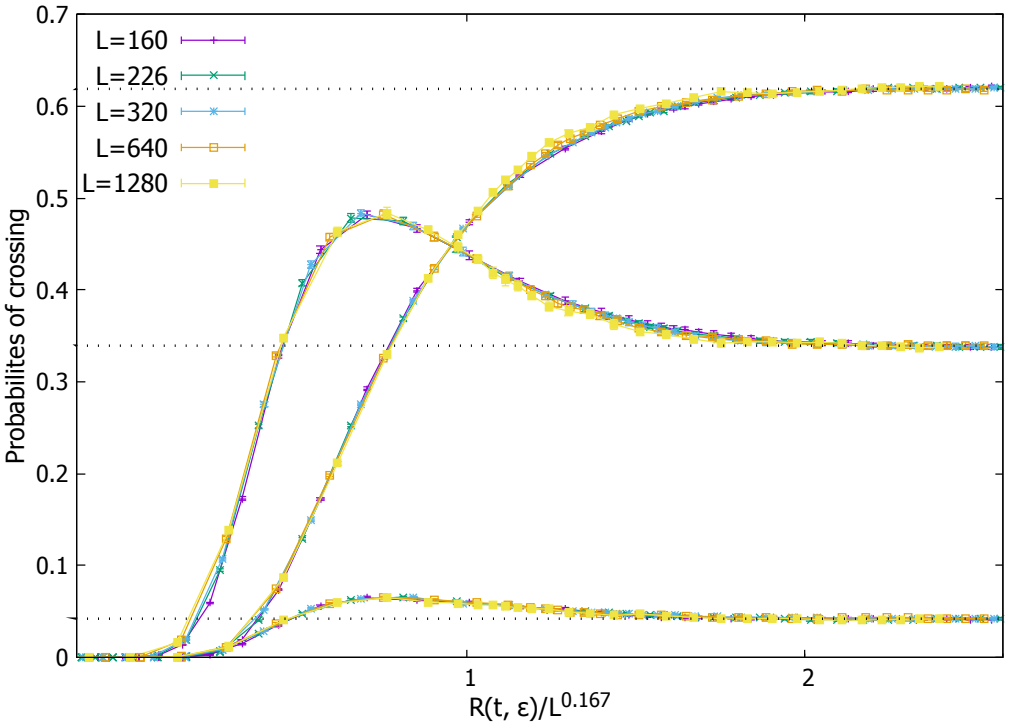

Figura 5.15: Same as Fig. 5.13, for RFIM with $\epsilon_{RF} = 2$ and substituting $R(t)$ by $R(t, \epsilon)$.

$t = 3.13$ in the range $4 < \ln(x) < 7$, and we obtained $k = 5.94$, in excellent agreement with the value $k = 6$ for critical percolation. This signals that over sufficiently large distances the largest cluster has the morphological properties of percolation. Moreover, that is true for all measurement times considered, is in agreement with the fact that percolating clusters appear very soon in coarsening process and are stable, as said in Sec. 5.3. Interestingly, we see that the crossover between the two behaviors takes place at a distance l_t which increases with the measurement time. We will provide an interpretation of this point in the following.

This situation is confirmed in the presence of quenched randomness. In Fig. 5.18, we report the same plot for $\epsilon_{RF} = 2$, the strongest value of disordered considered. The best fit is $k = 5.91$, obtained when $R(t, \epsilon) = 2.91$, which confirms the robustness of the percolative effects.

We regard this as compelling evidence that the clusters arising in the coarsening process *are* those of critical percolation, for both pure and disordered systems.

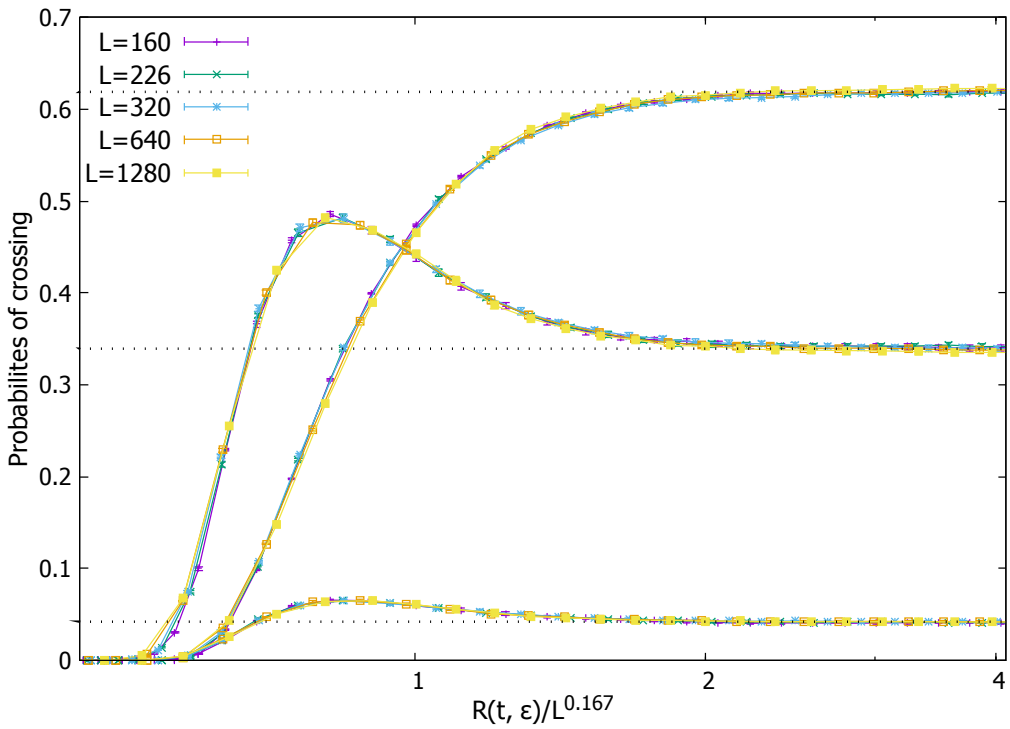

Figura 5.16: Same as Fig. 5.13, for RBIM with $\epsilon_{RB} = 0.8$ and substituting $R(t)$ by $R(t, \epsilon)$.

5.5.1 Scaling properties of the winding angle

We have seen that in Figs. 5.17 and 5.18 there is a length l_t increasing with time such that if we look at the hull over a scale $x \ll l_t$, it doest not have the typical morphological properties of critical percolation. Conversely, such properties are acquired for $x \gg l_t$, where Eq. (5.9) with $k \simeq 5.9 \simeq 6$ describes numerical data very well.

In a coarsening problem there is a length growing with time, i.e. $R(t)$. Hence, we expect the distance l_t to be identified with $R(t)$. In order to test it, we tried to collapse the curves in Fig.5.17 dividing x by the growing length $R(t)$. In Fig. 5.19 we see that the collapse is very good for the pure case; the same is true for the disordered systems, for all the values of ϵ considered (not shown).

This collapse means that for each measurement time the hulls acquire percolative properties when we look at them over a distance $x \gg R(t)$. Indeed, we see from Fig. 5.19 that the curve becomes a straight line when $x \simeq 10\, R(t)$. We can interpret this fact by recalling that in percolation, lattice sites are uncorrelated, whereas coarsening is a process driven by the

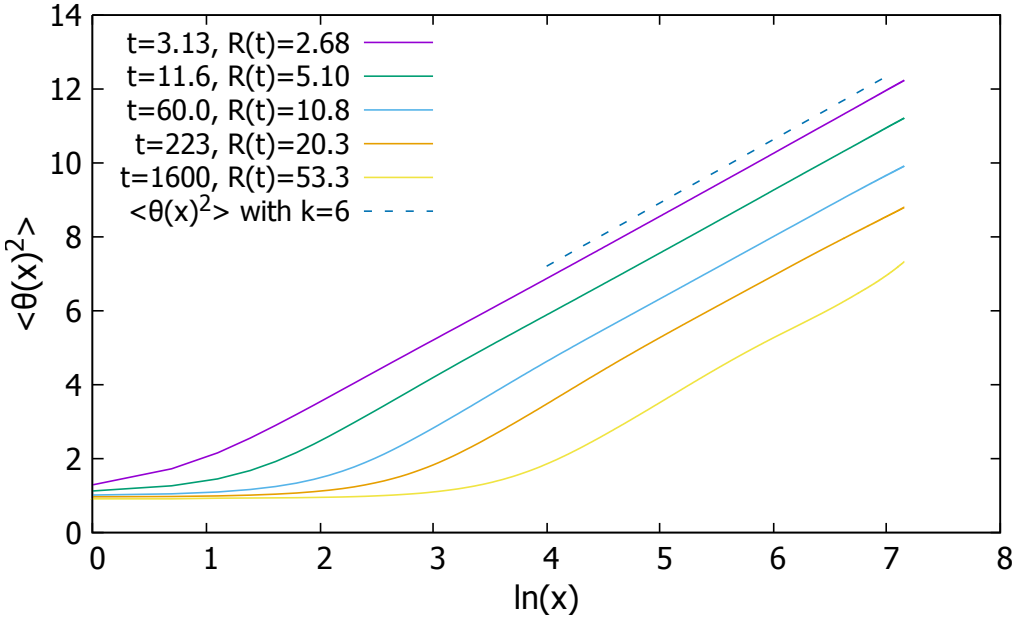

Figura 5.17: Average squared winding angle for the hull of the largest cluster of the pure $2d$ IM, plotted against $\ln x$, with x the distance along the hull. Results are relative to a lattice $L = 640$. In the key we report the measurement times and the associated values of $R(t)$. Data have been averaged over 10^5 samples and error bars are omitted as they are very small and not relevant for the analysis. The dashed lines is Eq. (5.9) with $k = 6$, the value of critical percolation.

correlation between spins. Such correlation extends over larger distances as time goes on and since $R(t)$ represents the typical size of the correlated regions, inside such regions we do not observe percolative effects. In other words, coarsening and percolation coexist on the same lattice, but are also mutually exclusive in the sense that they pertain to different length scales. It is therefore natural to expect the morphology of the hulls to be that of critical percolation *beyond* $R(t)$. This is precisely what we see in Fig. 5.19.

The main result presented in this section is a confirmation that the morphology of clusters in a system undergoing coarsening is very soon that of critical percolation. This analysis had never been performed on inhomogeneous models, and the results definitively confirm the connection between these two seemingly unrelated problems – phase-ordering and percolation – also in presence of quenched randomness.

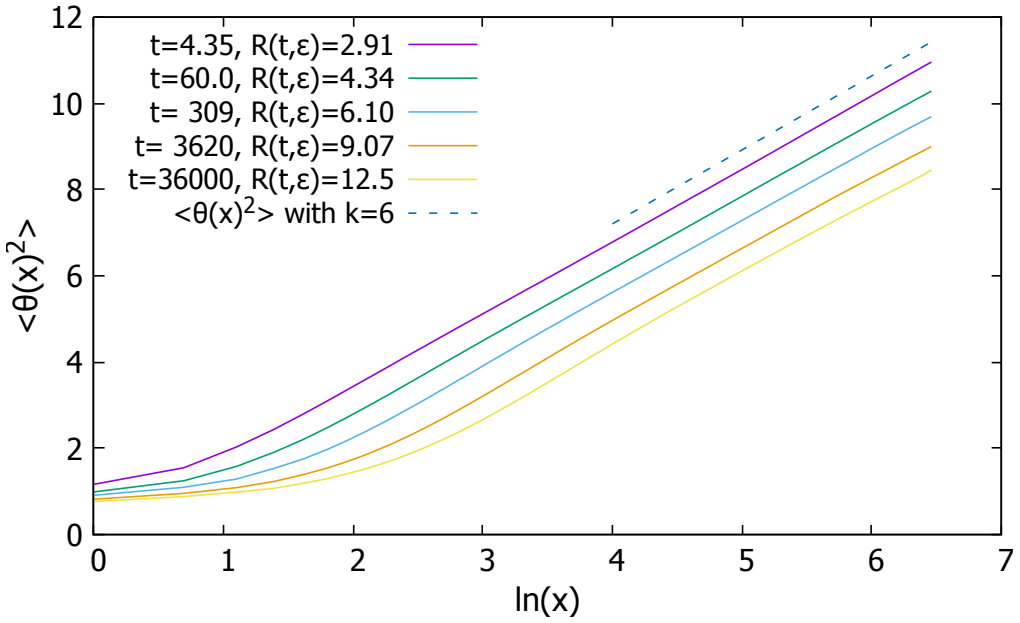

Figura 5.18: Same as Fig. 5.17, but for RFIM with $L = 320$ and $\epsilon_{RF} = 2$.

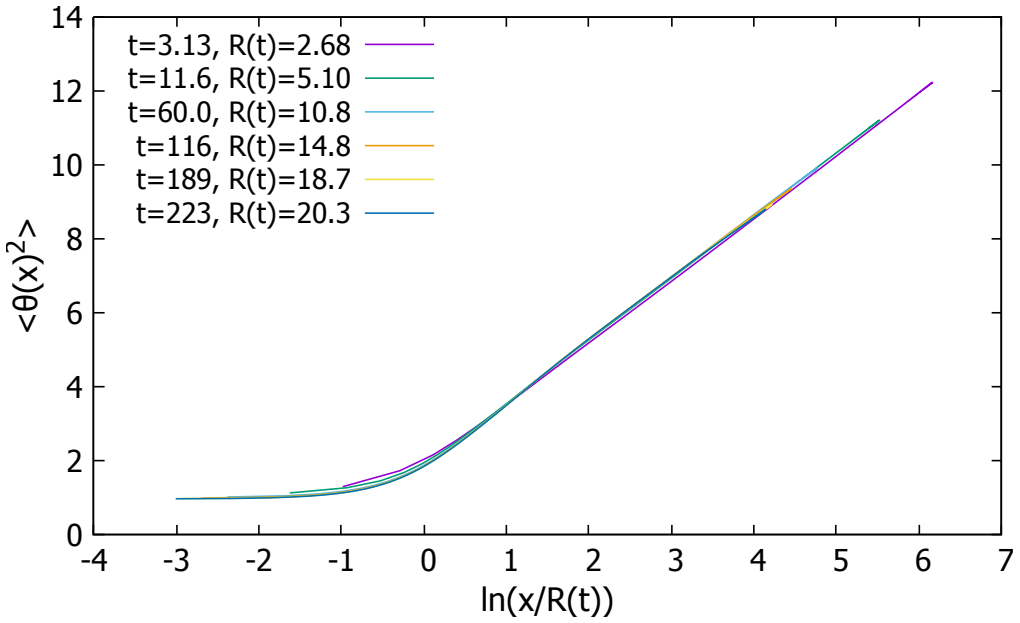

Figura 5.19: Same as Fig. 5.17, but plotted against $\ln[x/R(t)]$ to test the dynamical scaling hypothesis.

5.6 Pair connectivity

In Sec. 2.2.2, we have defined in $1d$ the pair connectivity $g(r)$ as the probability that two sites at distance r belong to the same cluster. The same defini-

tion applies to higher dimensions. It is known [20] that for a two-dimensional lattice[2] at percolation threshold $g(r)$ decays according to a power law

$$g(r) \simeq r^{-2\Delta}, \tag{5.10}$$

for $p \to p_c$, with $\Delta = 5/48$.

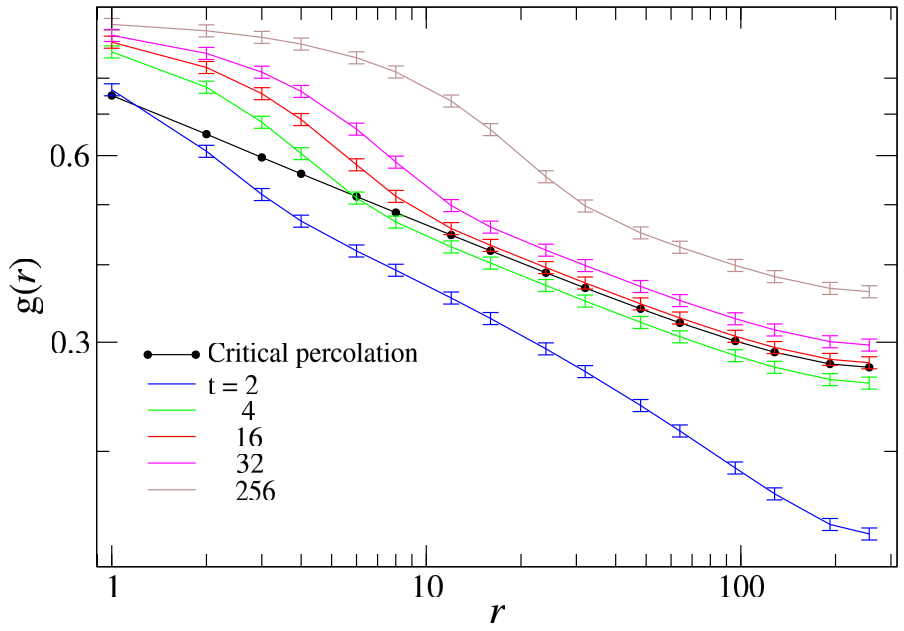

Figura 5.20: Pair connectivity against the distance r measured on a lattice at percolation threshold (black line) and for a coarsening homogeneous $2d$ IM (colored lines). Measurement times are given in the key. The lattice size is $L = 512$ in both cases.

Simply following its definition, $g(r)$ can be computed numerically on any lattice whose sites are in one of two (or more) possible states. We have done this for the clusters of positive and negative spins of the Ising Model under the usual conditions of our study, in order to check how the pair connectivity for a coarsening system compares with that of critical percolation. This is the first time this quantity is used to study a system undergoing phase ordering kinetics.

We performed the measurement of $g(r)$ at different times on a system of size $L = 512$. In Fig. 5.20 we report results for the pure coarsening system.

[2]For any lattice geometry, since it is a universal property.

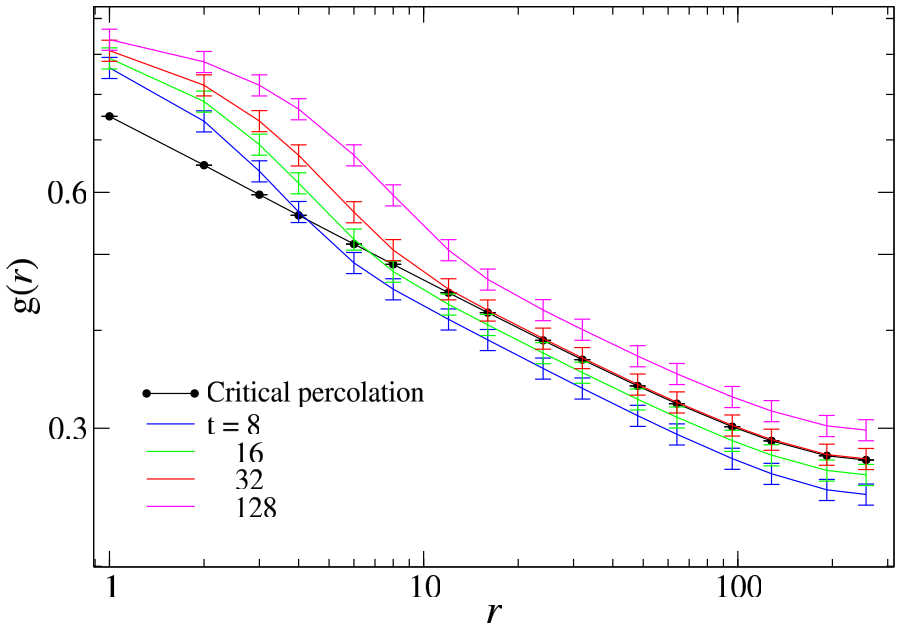

Figura 5.21: Same as Fig. 5.20 but for a RFIM with $\epsilon_{RF} = 1$

The black curve is the one obtained for random percolation at p_c which has the slope $-2\Delta = -10/48$ (at large distances this curve bends upward as due to finite size effects). Bearing in mind that this slope is the signature of percolation we observe the two following facts :

- For each curve except $t = 2$, after an initial distance which increases with time, curves for coarsening have the same slope as that for percolation. This supports the interpretation of Sec. 5.5.1, relative to the behavior of the winding angle : clusters acquire the morphological properties of percolation beyond a certain distance, over which the ordering system is still uncorrelated. Moreover, here we explicitly see that this is not true for very small times, when the system is still very close to the initial condition.

- We see that for $t = 16$ – corresponding to $R(t) \simeq 5.8$ – the pair connectivity for the IM overlaps with that of critical percolation. This is significant if we compare it with Fig. 5.21, where we report the same for a system with $\epsilon_{RF} = 1$ and $L = 512$. We see precisely the same struc-

ture, with the coarsening curve overlapping with that of percolation for $t = 32$, corresponding to $R(t, \epsilon) \simeq 5.6$. For $\epsilon_{RF} = 2$ (not shown) , the result is analogous and the two connectivities overlap when $R(t, \epsilon)$ is around 5.8. This, together with the results of Sec. 5.4, suggests that the effects of disorder can be fully accounted for by expressing everything in terms of the growing length, and no other features are added by the quenched randomness as regards the percolation structure.

Let us also notice that the curves are shifted toward the top for increasing times due to a renormalization factor in the definition of $g(r)$. We are currently looking for a way to obtain a full collapse.

Finally, we mention that the value of the critical exponent $\Delta = 5/48$ corresponds biunivocally [22] to the fractal dimension D of the percolating cluster at p_c, being $D = d - \Delta = 91/48$ (Sec. 2.2.4). Therefore, we have also checked that the clusters of the coarsening system and of critical percolation have the same fractal dimension.

These results further support the precise, quantitative correspondence between a coarsening system and a lattice at critical percolation.

5.7 The superuniversality hypothesis

From our results it seems that describing the coarsening process in terms of the growth of $R(t, \epsilon)$ allows for a unified description of percolative effects in models with different strengths ϵ of disorder, including the homogeneous case ($\epsilon = 0$). This surely true at a semi-quantitative, or approximate level. Our aim now is to check more precisely if this fact can be fully validated. In order to do that, we must before discuss the concept of *superuniversality* (SU) [49].

We introduce it by a simple example. Let us reconsider the equal-time spin-spin correlator $C(r, t)$ of Sec. 1.3 and the associated scaling law involving the scaling function $c(x)$:

$$C(r, t) = c\left[\frac{r}{R(t)}\right]. \tag{5.11}$$

According to a conjecture, the SU hypotheses, Eq. (5.11) should remains valid for a disordered system, provided that we replace $R(t)$ by the corresponding $R(t, \epsilon)$. In other words, we should have

$$C(r, t, \epsilon) = c\left[\frac{r}{R(t, \epsilon)}\right], \tag{5.12}$$

with the same scaling function $c(x)$, now being $x = r/R(t, \epsilon)$. Accordingly, it would be possible to collapse curves for different values of ϵ by plotting again-

st $r/R(t,\epsilon)$, i.e. keeping the variable x fixed (we neglect in this discussion the corrections introduced by L_p discussed in Sec. 5.3.2).

Instead, if SU does not hold, we expect a different scaling function which depends on the strength of disorder. To see how, let us recall that in Sec. 3.3 we have discussed how quenched randomness introduces a crossover length $\lambda(\epsilon)$ separating two different regimes in the growth of $R(t)$. In order to incorporate this in a scaling framework [35] one should have the form

$$C(r,t,\epsilon) = \mathcal{C}\left[\frac{r}{R(t,\epsilon)}, \frac{\lambda(\epsilon)}{R(t,\epsilon)}\right], \tag{5.13}$$

where $\mathcal{C}(x,y)$ is the generalized scaling function with two entries. This is analogous to what we did in Sec. 5.3.2 to introduce L_p in the scaling pattern.

Now we have a two-variable scaling function, therefore we cannot expect to have $\mathcal{C}(x,y) = c(x)$ identically for any value of the disorder ϵ. Therefore, we should not be able to collapse curves for $C(r,t,\epsilon)$ for different values of ϵ by plotting against $r/R(t,\epsilon)$. Indeed by doing that, despite the fact that we keep x fixed, y varies.

Although the previous discussion focused for simplicity on the correlator C, SU is expected to be property of the system, and therefore to be mirrored in any observable quantity, not only of correlators. Therefore, we can check if it holds in our system by considering, e.g., the crossing probabilites.

In order to do that, we refer to the scaling form in Eq. (5.8), which seems confirmed by the results of Sec. 5.4 also in presence of quenched randomness. Let us hypothesize that SU holds. We would have

$$\pi(n,m) = p_{nm}\left[\frac{R(t,\epsilon)}{L_p}\right]. \tag{5.14}$$

We limit the discussion to the RFIM and we choose $\epsilon_{RF} = 0, 0.5, 1, 2$. We fix the system size at $L = 1280$. Since the system size is fixed, the same is true also for L_p, hence we keep the entry of the scaling function in Eq. (5.14) fixed by simply plotting against $R(t,\epsilon)$, whereas we let the disorder strength vary. If we obtain a collapse, the SU hypothesis is actually correct.

We report the results of this procedure in Fig. 5.22. We observe increasing deviations from the homogeneous case as ϵ_{RF} increases. The fact that this deviations, although rather small, are systematic suggests that the actual scaling function might be of the same kind as $\mathcal{C}(x,y)$ in Eq. (5.13), with a second variable y depending on disorder. In that case, superuniversality would be in principle violated. As a consequence Eq. (5.14) would be wrong and could only be used to collapse curves for a single value of ϵ. This result would not be surprising since analogous small deviations were already found in the same models and using the same Monte Carlo algorithm [29]. Indeed,

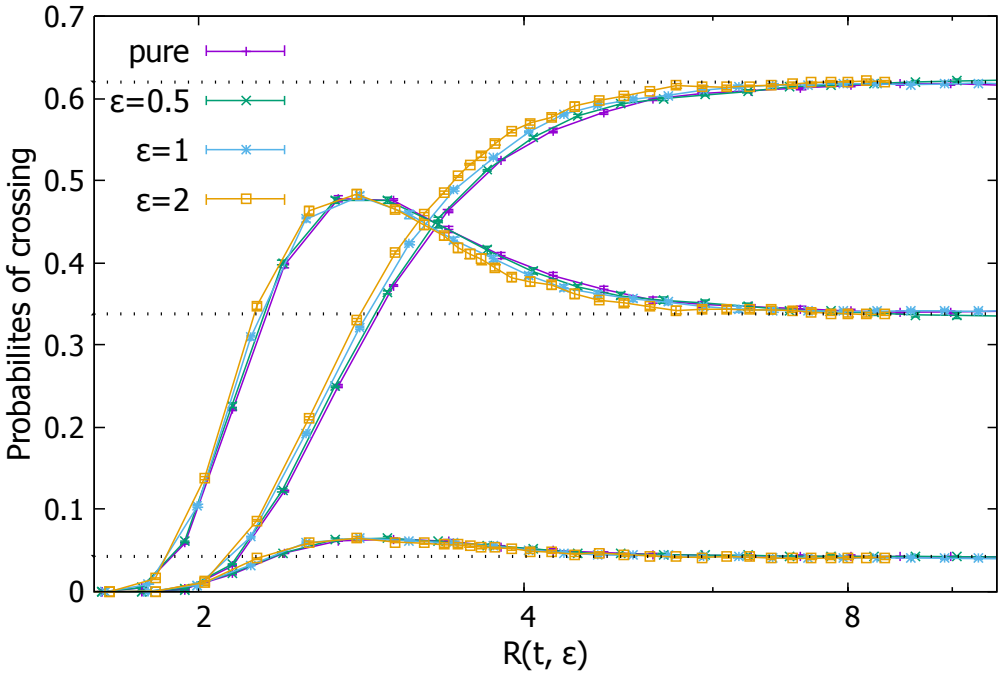

Figura 5.22: Probabilities of crossing against $R(t, \epsilon)$ for RFIM and pure IM. The lattice size is fixed to $L = 1280$ and the strength ϵ_{RF} of the external random field varies, as indicated in the key.

superuniversality remains a hypothesis which has received confirmation only in some specific cases and for some specific quantities [47] [32].

The data presented in Fig. 5.22 might suggest that weak violations of SU are observed. However this evidence is not compelling and such deviations, if present, are so small to make SU at least a valid approximation framework to describe the system. In fact, apart from fine effects, all the results we have presented point in the same direction : the presence of quenched randomness does not alter the essential fact that there exists a length L_p diverging with, but slower than, the size of the system. Considering also the results for the winding angle and for the pair connectivity, it is beyond doubt that this length is associated to critical percolation, both for pure and disordered models. Disorder does not alter these main features. Accordingly, we can conclude that for the purpose of our study the SU hypothesis holds at least at a semi-quantitative level for the strengths of disorder considered.

In order to disprove (or prove) the SU hypothesis, one usually need to embark on too demanding numerical simulations. For instance, in our case referring to Fig. 5.22, we expect increasing deviations for higher values of ϵ. This can be tested, but the curves for $\epsilon_{RF} = 2$ in Fig. 5.22 have already required 10^3 hours of CPU time and we know from Sec. 4.4 that this time

would increase very rapidly for larger ϵ.

Finally, in Sec. 3.4 we have discussed that, with our protocol, for the RBIM and RFIM we have a crossover from a pre-asymptotic algebraic growth law of $R(t, \epsilon)$ to an asymptotic algorithmic growth. In Figs. 3.2 and 3.3 we have shown that with our simulations we do not enter the logarithmic growth, whereas the most evident violations of SU have been reported in this asymptotic regime. In order to make a stringent test on SU, therefore, one should test Eq. (5.14) for values of L_p as large as to fall beyond the crossover length $\lambda(\epsilon)$. This would require much larger system sizes for larger values of ϵ, making the computational effort presently unaffordable.

Capitolo 6

Conclusions and Perspectives

In this work we have studied, through large-scale numerical simulations, the coarsening process of the homogeneous and inhomogeneous $2d$ IM evolving with Glauber dynamics after a quench from infinite to zero temperature. Our original results can be divided into two categories.

To the first one belongs new independent evidence of percolation on the coarsening process of the homogeneous IM. Presently this connection has only been observed in numerical simulations of specific observables, therefore its confirmation through the analysis of new quantities is important. The methods allows us to obtain impressive quantitative correspondences.

In particular, by measuring the slope $4k/(8+k)$ in Eq. (5.9) for the winding angle of the hull of the largest cluster, we found values of k close to the one $k = 6$ known for critical percolation (see Fig. 5.17) as $k \simeq 5.94$. To the best of our knowledge, this has been never been shown for a zero-temperature quench. Moreover, we computed for the first time the pair connectivity for a coarsening system and in Fig. 5.20 we showed that for sufficiently large distances it follows the characteristic behavior of critical percolation. These two results provide compelling evidence that the domain morphology of the ordering system in the early stages of its evolution is precisely that of critical percolation. This is a relevant fact, since its validity is assumed in one of the very few available proofs of dynamical scaling (Section 1.4 and [11]) for $2d$ models.

Concerning the scaling hypothesis, our results on the crossing probabilities independently confirm that a percolative cluster forms in the early stages of the evolution of the coarsening system. This implies the existence of a second characteristic length L_p which plays a role in the dynamical scaling framework beside $R(t)$.

The second category of results amounts to the extension of the above analysis to inhomogeneous models, in particular the RBIM and RFIM. This

was never attempted before. For these two types of quenched randomness, we found that the morphology of clusters is still that of critical percolation and that a cluster with percolative properties forms in the system when $R(t_p)$ is the same as in the pure case. Perhaps, the most important consequence is the robustness of the influence of the associated length L_p on dynamical scaling. All this is shown through the analysis of the crossing probabilities, the winding angles and the pair connectivity. This represents a first important step toward an extension of the proof of dynamical scaling to inhomogeneous systems, which would be welcome in this complex and debated field.

These results raise many questions which should be addressed in future research. First, it would be relevant to repeat the analysis for different lattice geometries, for example for kagome, hexagonal and bow-tie lattices. This would inform us on the universality character of the properties we have found studying the square lattice. Then, one might also consider to extend the study to higher dimensions.

Furthermore, different types of disorder might be included. In particular, one might consider the Site Diluted Ising Model in which a given fraction of spins is removed. Besides the relevance of this kind of disorder in its own, since dilution is commonly observed in most systems, this model could also help shed light on another point. Indeed, according to a conjecture proposed in [46], the exponent α_p of Eq. (5.5) depends on the geometry of the lattice considered, more precisely on its coordination number. Since site dilution lowers the average coordination number, determining the value of α_p for this model could help establish the validity of this hypothesis.

Finally, it would be interesting to repeat the analysis for higher strength of disorder, to shed more light on the validity of the SU hypothesis discussed in Sec. 5.7. In particular, referring to Fig. 5.22 one would like to see if the deviations between the curves for the pure and the inhomogeneous systems increase for larger values of ϵ. However, this requires a computational effort which is unaffordable with our algorithm, due to severe pinning effects which would slow down the dynamics to the point that huge simulation times would be necessary. One way out could be the use of an algorithm inspired to that introduced by Bortz, Kalos and Lebowitz [50], which could deal with these pinning effects more efficiently.

Appendice A

Appendix: Numerical code

We include the FORTRAN source code we have used to obtain all our numerical data.

```
      program relax_2d
!     ************************************************************
!     * MonteCarlo for Ising in d=2 with NCOP Glauber            *
!     * Random Field and Random Bond                             *
!     * Last revision  27 June 2016                              *
!     ************************************************************
!     (Now Adapted to measuring pair connectivity at many measurement times
!     to find when it starts to resemble the percolation curve.)

      implicit none
!     ---------------------------------------------------
!     First Variables Declaration
      integer L,realtot,maxnn,nprint, mL, n2
      real*8 amp,T,factime,Hamp,final_time,first_time
!.......input parameters.....................................

      parameter (L=512, n2=512*512)  !System size
      parameter ( nprint = 500)      !Number of realizations before printing
      parameter (mL=16 )         !Measurement distances for pair connectivity
      parameter(T=0.01)          ! Temperature
      parameter(first_time=1) ! First measurement time
      parameter(realtot=10000000) ! Number of realizations
      parameter(maxnn=80)       ! Max number of times it will write
      parameter(Hamp=0*T)      ! Amplitude of Random External Field
      parameter(amp=0.0*T)         ! Amplitude of Random Bonds
      parameter(final_time=50000) ! Total time
!.............................................................

!     Second Variables declaration
      integer hloc,dir,pbc(0:L+1),jjj,Ene
      integer i,j,k,ii,q,p,loop,counts,grnd,iv,jv
      integer sum1,inv_list(L,L),num_list,inv_listc(L,L),num_listc
      integer m,realiz,seed1,direz,chose_num,posiz
      integer L2,num_tempi,index
      real*8 tempo(0:maxnn)

      integer dist(1:24)
      real*8 ccor(1:30,1:mL)
      real*8 cor(1:30,1:mL)
      real*8 avccor(1:30,1:mL), savccor(1:30,1:mL)
      real*8 avcor(1:30,1:mL),  savcor(1:30,1:mL)
      real*8 rlz
      real ran2,defect(maxnn), avc
      integer s(L,L)
      integer  SS(0:n2-1)
      real JJ(0:L+1,0:L+1,4),DeltaE
      real Hfield(L,L)
      real*8 invL2
      integer nn_i(L,0:4),nn_j(L,0:4)
      integer kk,comp,conts,i_list((L+2)**2),j_list((L+2)**2)
      real*8 Boltz,time, dt
      integer mfile,ijk,ikj,nn,h
      integer nucr,ntime,hloc2,stored
      integer ppcr(1:maxnn,0:16),nncr
      integer n1(1:L*L),largest
      integer na(1:maxnn,1:L*L),nlargest(1:maxnn,1:L*L)
      integer flag_or
      integer n4,n0,nm4,al1,al2,lcluster
      real*8 alc(1:maxnn),an4(1:maxnn),an0(1:maxnn),anm4(1:maxnn)
      real*8 aal1(1:maxnn),aal2(1:maxnn)
      integer ntheta1(1:2*L),ntheta2(1:2*L)
      real*8 atheta1(1:2*L),atheta2(1:2*L)
      integer nntheta1(1:maxnn,1:2*L),nntheta2(1:maxnn,1:2*L)
      real*8 aatheta1(1:maxnn,1:2*L),aatheta2(1:maxnn,1:2*L),pi
c
      common/an/n4,n0,nm4,al1,al2,lcluster
```

```
      common/an2/atheta1,atheta2,ntheta1,ntheta2
      common/na/cor,ccor
c
!     ------------------------------------------------------
      seed1=-7763337
      L2=L**2
      pi=3.14159265358979323846264538328d0
      invL2=1./L2
      avc=0
c
!.......initial settings.....................................
           open(unit=1,file="2ptcor_G_L512")
           open(unit=2,file="Largest_G_L512")


           open (70,file='Lchar_L512.dat')
           open (90,file='PCR_L512.dat')
           open (91,file='NA_L512.dat')
           open (92,file='B_L512.dat')
           open (93,file='Ang0_L512.dat')
           open (94,file='Ang4_L512.dat')
           write (70,*) '#Temperature = ',T
           write (70,*) '#RF Amp = ',Hamp,'#RB Amp = ',amp
           write (70,*) '#Total number of realizations = ',realtot
           write (70,*) '#System size = ',L,' nprint = ',nprint
           write (90,*) '#Temperature = ',T
           write (90,*) '#RF Amp = ',Hamp,'#RB Amp = ',amp
           write (90,*) '#Total number of realizations = ',realtot
           write (90,*) '#System size = ',L,' nprint = ',nprint
           write (91,*) '#Temperature = ',T
           write (91,*) '#RF Amp = ',Hamp,'#RB Amp = ',amp
           write (91,*) '#Total number of realizations = ',realtot
           write (91,*) '#System size = ',L,' nprint = ',nprint
           write (92,*) '#Temperature = ',T
           write (92,*) '#RF Amp = ',Hamp,'#RB Amp = ',amp
           write (92,*) '#Total number of realizations = ',realtot
           write (92,*) '#System size = ',L,' nprint = ',nprint
           write (93,*) '#Temperature = ',T
           write (93,*) '#RF Amp = ',Hamp,'#RB Amp = ',amp
           write (93,*) '#Total number of realizations = ',realtot
           write (93,*) '#System size = ',L,' nprint = ',nprint
           write (94,*) '#Temperature = ',T
           write (94,*) '#RF Amp = ',Hamp,'#RB Amp = ',amp
           write (94,*) '#Total number of realizations = ',realtot
           write (94,*) '#System size = ',L,' nprint = ',nprint

         do i=1,24
          if(i.eq.1) dist(i)=1
          if(i.eq.2) dist(i)=2
          if(i.eq.3) dist(i)=3
          if(i.eq.4) dist(i)=4
          if(i.eq.5) dist(i)=6
          if(i.eq.6) dist(i)=8
          if(i.eq.7) dist(i)=12
          if(i.eq.8) dist(i)=16
          if(i.eq.9) dist(i)=24
          if(i.eq.10) dist(i)=32
          if(i.eq.11) dist(i)=48
          if(i.eq.12) dist(i)=64
          if(i.eq.13) dist(i)=96
          if(i.eq.14) dist(i)=128
          if(i.eq.15) dist(i)=192
          if(i.eq.16) dist(i)=256
          if(i.eq.17) dist(i)=384
          if(i.eq.18) dist(i)=512
          if(i.eq.19) dist(i)=768
```

```
         if(i.eq.20) dist(i)=1024
         if(i.eq.21) dist(i)=1536
         if(i.eq.22) dist(i)=2048
         if(i.eq.23) dist(i)=3072
         if(i.eq.24) dist(i)=4096

      enddo


      do i=1,maxnn
         alc(i)=0.d0
         an4(i)=0.d0
         an0(i)=0.d0
         anm4(i)=0.d0
         aal1(i)=0.d0
         aal2(i)=0.d0
         do j=0,16
            ppcr(i,j)=0
         enddo
         do j=1,2*L
            nntheta1(i,j)=0
            nntheta2(i,j)=0
            aatheta1(i,j)=0.d0
            aatheta2(i,j)=0.d0
         enddo
         do k=1,L*L
            na(i,k)=0
         enddo
         do k=1,L*L
            nlargest(i,k)=0
         enddo
      enddo
c
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
!
!          Construction of tempo arrays
      print*,final_time
      tempo(0)=0
      tempo(1)=first_time
      !factime=dexp(dlog(1d0*final_time/first_time)/(maxnn))
      h=0
      do m=2,maxnn
         tempo(m)=tempo(m-1)*1.1 !Change according to the quantity
         print*, tempo(m)
      enddo
      do i=1,L
         pbc(i)=i
      enddo
      pbc(L+1)=1
      pbc(0)=L
      do i=1,L
         do j=1,L
            nn_i(i,0)=i
            nn_j(j,0)=j
            nn_i(i,1)=pbc(i+1)
            nn_j(j,1)=j
            nn_i(i,2)=i
            nn_j(j,2)=pbc(j+1)
            nn_i(i,3)=pbc(i-1)
            nn_j(j,3)=j
            nn_i(i,4)=i
            nn_j(j,4)=pbc(j-1)
         enddo
      enddo
      do i=1,maxnn
         defect(i)=0.
```

```
      enddo

       do j=1,ml
         do i=1,30
            cor(i,j)=0.d0
            ccor(i,j)=0.d0
            avcor(i,j)=0.d0
            avccor(i,j)=0.d0
            savcor(i,j)=0.d0
            savccor(i,j)=0.d0
         enddo
      enddo


!     ----- Beginning of loop on realizations
      do realiz=1,realtot

       stored=0
         do i=1,L
            do j=1,L
               do dir=1,2
                  JJ(i,j,dir)=1-amp+2*amp*ran2(seed1)
               enddo
            enddo
         enddo
!     initial condition
         do i=1,L
            do j=1,L
               s(i,j)=sign(1.,ran2(seed1)-.5)
               Hfield(i,j)= Hamp*sign(1.,ran2(seed1)-.5)
            enddo
         enddo
!     .........create list of boundary sites ........................
         num_list=0
         do i=1,L
            do j=1,L
               inv_list(i,j)=0
               hloc=0
               do dir=1,4
                  hloc=hloc+s(nn_i(i,dir),nn_j(j,dir))
               enddo
               if (hloc*s(i,j).le.0)then
                  num_list=num_list+1
                  i_list(num_list)=i
                  j_list(num_list)=j
                  inv_list(i,j)=num_list
               endif
            enddo
         enddo
!     ------------------------------------------------------------
!initialization of variables


         flag_or=0
         time=0
         m=0
         ntime=1
         h=1
         rlz = 0

         dt=1D0/num_list
 121     continue

         if(time+dt.gt.tempo(ntime)) then
            do i=1,L
              do j=1,L
```

```
              SS((i-1)*L + j-1) = s(i,j)
            enddo
          enddo

           call analysisf (SS,ntime,largest)

      if (mod(realiz,nprint).eq.0) then

          do i=1,30
              if(cor(i,1).gt.0) then
                stored=i
                do j=1,mL
           if(cor(i,j).lt.0) print*, cor(i,j)
                 write(1,141)realiz,i,tempo(i),dist(j),cor(i,j)/nprint,
    c                    ccor(i,j)/nprint

                  avcor(i,j) = avcor(i,j) +cor(i,j)
                  avccor(i,j) = avccor(i,j) +ccor(i,j)
                  savcor(i,j) =savcor(i,j) + cor(i,j)*cor(i,j)
                  savccor(i,j) =savccor(i,j) + ccor(i,j)*ccor(i,j)

                   cor(i,j)=0.d0
                   ccor(i,j)=0.d0
                enddo
              endif
 141  format(i10,i8,f10.3,i8,2(f15.10))


          enddo
       endif
        call analysis0(s,nucr,n1,largest)
             if(nucr.eq.0) ppcr(ntime,0)=ppcr(ntime,0)+1
             if(nucr.eq.2) ppcr(ntime,1)=ppcr(ntime,1)+1
             if(nucr.eq.1) then
              call analysis1(s,nncr)
              ppcr(ntime,nncr)=ppcr(ntime,nncr)+1
             endif
             call analysis2(s)
            alc(ntime)=alc(ntime)+lcluster
            an4(ntime)=an4(ntime)+n4
            an0(ntime)=an0(ntime)+n0
            anm4(ntime)=anm4(ntime)+nm4
            aal1(ntime)=aal1(ntime)+al1
            aal2(ntime)=aal2(ntime)+al2
            do i=1,L*L
               na(ntime,i)=na(ntime,i)+n1(i)
            enddo
            nlargest(ntime,largest)=nlargest(ntime,largest)+1
            do k=1,2*L
               if(ntheta1(k).ne.0) then
                  nntheta1(ntime,k)=nntheta1(ntime,k)+ntheta1(k)
                  aatheta1(ntime,k)=aatheta1(ntime,k)+atheta1(k)
               endif
               if(ntheta2(k).ne.0) then
                  nntheta2(ntime,k)=nntheta2(ntime,k)+ntheta2(k)
                  aatheta2(ntime,k)=aatheta2(ntime,k)+atheta2(k)
               endif
            enddo
            Ene=0
            do j=1,L
               do i=1,L
                  Ene=Ene-(s(i,j)*s(nn_i(i,1),j)-1)
               enddo
            enddo
            defect(ntime)=defect(ntime)+Ene*invL2
```

```
            ntime=ntime+1
            if(ntime.gt.maxnn) goto 130
         endif
         chose_num=num_list*ran2(seed1)+1
         i=i_list(chose_num)
         j=j_list(chose_num)
         DeltaE=2*s(i,j)*(JJ(nn_i(i,3),j,1)*s(nn_i(i,3),j)+
     &        JJ(i,j,1)*s(nn_i(i,1),j)+
     &        JJ(i,nn_j(j,4),2)*s(i,nn_j(j,4))+
     &        JJ(i,j,2)*s(i,nn_j(j,2))+Hfield(i,j))
         Boltz=.5*(1.-tanh(DeltaE/2./T))
         if(ran2(seed1).le.Boltz)then
            s(i,j)=-s(i,j)



            do direz=0,4
               iv=nn_i(i,direz)
               jv=nn_j(j,direz)
               hloc=0
               do dir=1,4
                  hloc=hloc+s(nn_i(iv,dir),nn_j(jv,dir))
               enddo
               if (hloc*s(iv,jv).le.0)then
                  if(inv_list(iv,jv).eq.0)then
                     num_list=num_list+1
                     i_list(num_list)=iv
                     j_list(num_list)=jv
                     inv_list(iv,jv)=num_list
                  endif
               else
                  posiz=inv_list(iv,jv)
                  if(posiz.ne.0)then
                     i_list(posiz)=i_list(num_list)
                     j_list(posiz)=j_list(num_list)
                     inv_list(i_list(posiz),j_list(posiz))
     &                    =posiz
                     inv_list(iv,jv)=0
                     num_list=num_list-1
                  endif
               endif
            enddo
         endif
         time=time+dt
         dt=1D0/num_list
         goto 121
 130     continue


         if (mod(realiz,nprint).eq.0) then
           rlz = realiz
           open(unit=74,file="av_corr_L512")
           write (74,*) '#Temperature = ',T
           write (74,*) '#RF Amp = ',Hamp,'#RB Amp = ',amp
           write (74,*) '#Current number of realizations = ',realiz
           write (74,*) '#System size = ',L
              do i=1,stored
                 do j=1,mL
                  write(74,139)i,tempo(i),dist(j),avcor(i,j)/rlz,
     c                    avccor(i,j)/rlz,
     c          sqrt((savcor(i,j)/rlz-(avcor(i,j)/rlz)**2)/rlz),
     c          sqrt((savccor(i,j)/rlz-(avccor(i,j)/rlz)**2)/rlz)

                 enddo
 139   format(i10,f10.3,i8,4(f15.10) )
              enddo
```

```
          close(74)

           do i=1,maxnn
              if(defect(i).ne.0) then
                 write(70,*)tempo(i), 1d0/(defect(i)/nprint)
              endif
           enddo
c            close(70)
           do i=1,maxnn
              write(90,204)i,tempo(i),1d0/(defect(i)/nprint),
     c             (ppcr(i,j),j=0,16),nprint
              do j=0,16
                 ppcr(i,j)=0
              enddo
 204          format(i8,f15.5,f25.5,18(i8))
           enddo
c     close(90)
           do i=1,maxnn
              do j=1,L*L
                 if(na(i,j).ne.0) write(91,205)i,j,nprint,na(i,j),
     c                nlargest(i,j)
                 na(i,j)=0
                 nlargest(i,j)=0
              enddo
 205          format(i8,i8,i10,i12,i10)
           enddo
c     close(91)
           do i=1,maxnn
              write(92,206)i,tempo(i),1d0/(defect(i)/nprint),
     c             alc(i)/nprint,an4(i)/nprint,an0(i)/nprint,
     c             anm4(i)/nprint,aal1(i)/nprint,aal2(i)/nprint
              alc(i)=0.d0
              an4(i)=0.d0
              an0(i)=0.d0
              anm4(i)=0.d0
              aal1(i)=0.d0
              aal2(i)=0.d0
              defect(i)=0.d0
 206          format(i8,f25.5,f15.5,f12.4,f10.6,f10.6,f12.4,2(f15.4))
           enddo
c     close(92)
           do i=1,maxnn
              do k=1,2*L
                 if(nntheta1(i,k).ne.0) then
                    write(93,207)i,tempo(i),k,aatheta1(i,k)
     c                   /nntheta1(i,k)*pi*pi/4.d0,nntheta1(i,k)
                 endif
                 if(nntheta2(i,k).ne.0) then
                    write(94,207)i,tempo(i),k,aatheta2(i,k)
     c                   /nntheta2(i,k)*pi*pi/4.d0,nntheta2(i,k)
                 endif
                 nntheta1(i,k)=0
                 nntheta2(i,k)=0
                 aatheta1(i,k)=0.d0
                 aatheta2(i,k)=0.d0
              enddo
 207          format(i8,f25.5,i8,f15.6,i12)
           enddo
c     close(93)
c     close(94)
         endif
      enddo                     !loop over realizations
!     ----------------------------------------------------------
 444  stop
      end
```

```
      FUNCTION ran2(idum)
      INTEGER idum,IM1,IM2,IMM1,IA1,IA2,IQ1,IQ2,IR1,IR2,NTAB,NDIV
      REAL ran2,AM,EPS,RNMX
      PARAMETER (IM1=2147483563,IM2=2147483399,AM=1./IM1,IMM1=IM1-1,
     *IA1=40014,IA2=40692,IQ1=53668,IQ2=52774,IR1=12211,IR2=3791,
     *NTAB=32,NDIV=1+IMM1/NTAB,EPS=1.2e-7,RNMX=1.-EPS)
      INTEGER idum2,j,k,iv(NTAB),iy
      SAVE iv,iy,idum2
      DATA idum2/123456789/, iv/NTAB*0/, iy/0/
      if (idum.le.0) then
        idum=max(-idum,1)
        idum2=idum
        do 11 j=NTAB+8,1,-1
          k=idum/IQ1
          idum=IA1*(idum-k*IQ1)-k*IR1
          if (idum.lt.0) idum=idum+IM1
          if (j.le.NTAB) iv(j)=idum
11      continue
        iy=iv(1)
      endif
      k=idum/IQ1
      idum=IA1*(idum-k*IQ1)-k*IR1
      if (idum.lt.0) idum=idum+IM1
      k=idum2/IQ2
      idum2=IA2*(idum2-k*IQ2)-k*IR2
      if (idum2.lt.0) idum2=idum2+IM2
      j=1+iy/NDIV
      iy=iv(j)-idum2
      iv(j)=idum
      if(iy.lt.1)iy=iy+IMM1
      ran2=min(AM*iy,RNMX)
      return
      END


!cccccccccccccccccccccccccccccccccccccccccccc
!
!     SUBROUTINE analysis
!
!cccccccccccccccccccccccccccccccccccccccccccc
      subroutine analysis0(s,ncr,n1,largest)
      implicit none
      integer L,n2,k,indice_c
      parameter (L=512,n2=512*512)
      integer s(0:n2-1)
      integer touched(0:n2-1)
      integer i,ncr,nb,np,ncross,nc,dx,dy,dir,kdir
      integer touched2(0:n2-1),touched3(0:n2-1),touched4(0:n2-1)
      integer ncluster,v1,v2,v3,v4,c6,index,lpos
      integer cluster(1:n2),x1,x2,itime
      integer crossx(1:L),crossy(1:L),ncrossx,ncrossy
      integer largest
      integer pile(1:4*n2),pos
      integer n1(1:n2)

      largest=0
c
      nb=0
      ncr=-1
      do k=1,n2
         n1(k)=0
      enddo
      do k=0,n2-1
         v1=k+1
         if(mod(k+1,L).eq.0) v1=v1-L
         v2=k+L
```

```
         if(v2.ge.n2) v2=v2-n2
         if(S(k).eq.S(v1)) nb=nb+1
         if(S(k).eq.S(v2)) nb=nb+1
      enddo
!
!     Now compute the number of clusters
!
      do k=0,n2-1
         touched(k)=0
      enddo
      nc=0
      do k=0,n2-1
         if(touched(k).ne.0) goto 12
         pile(1)=k
         nc=nc+1
         touched(k)=1
         indice_c=1
         do index=1,n2
            if(index.gt.indice_c) goto 10
            lpos=pile(index)
            v1=lpos+1
            v2=lpos-1
            v3=lpos+L
            v4=lpos-L
            if(mod(lpos+1,L).eq.0) v1=v1-L
            if(mod(lpos,L).eq.0) v2=v2+L
            if(v3.ge.n2) v3=v3-n2
            if(v4.lt.0) v4=v4+n2
            if(touched(v1).eq.0) then
               if(S(v1).eq.S(lpos)) then
                  indice_c=indice_c+1
                  pile(indice_c)=v1
                  touched(v1)=1
               endif
            endif
            if(touched(v2).eq.0) then
               if(S(v2).eq.S(lpos)) then
                  indice_c=indice_c+1
                  pile(indice_c)=v2
                  touched(v2)=1
               endif
            endif
            if(touched(v3).eq.0) then
               if(S(v3).eq.S(lpos)) then
                  indice_c=indice_c+1
                  pile(indice_c)=v3
                  touched(v3)=1
               endif
            endif
            if(touched(v4).eq.0) then
               if(S(v4).eq.S(lpos)) then
                  indice_c=indice_c+1
                  pile(indice_c)=v4
                  touched(v4)=1
               endif
            endif
         enddo
 10      continue
         n1(indice_c)=n1(indice_c)+1
         if (indice_c.gt.largest) largest=indice_c
 12      continue
      enddo
c      nam(itime,largest)=nam(itime,largest)+1
!
!     Now we compute the number of polygons.
!
```

```
!        2     1
!           X
!        3     4
      ncross=0
      np=0
      do i=0,n2-1
         touched(i)=0
         touched2(i)=0
         touched3(i)=0
         touched4(i)=0
      enddo
      do i=0,n2-1
 50      continue
         dx=0
         dy=0
         if(touched(i).eq.0) then
            np=np+1
            kdir=1
            pos=i
            goto 110
         endif
         if(touched2(i).eq.0) then
            np=np+1
            kdir=2
            pos=i
            goto 120
         endif
         if(touched3(i).eq.0) then
            np=np+1
            kdir=3
            pos=i
            goto 130
         endif
         if(touched4(i).eq.0) then
            np=np+1
            kdir=4
            pos=i
            goto 140
         endif
         goto 200
 100     continue
         if((pos.eq.i).and.(dir.eq.kdir)) then
!     the loop is finished
            if(dx.ne.0) then
               ncross=ncross+1
            else
               if(dy.ne.0) ncross=ncross+1
            endif
            goto 50
         endif
         if(dir.eq.2) goto 120
         if(dir.eq.3) goto 130
         if(dir.eq.4) goto 140
 110     v1=pos+1
         if(mod(pos+1,L).eq.0) v1=v1-L
         if(S(v1).eq.S(pos)) then
            if(mod(pos+1,L).eq.0) dx=dx+1
            pos=v1
            dir=2
            touched2(v1)=1
            goto 100
         else
            dir=4
            touched4(pos)=1
            goto 100
         endif
```

```
 120     v1=pos+L
         if(v1.ge.n2) v1=v1-n2
         if(S(v1).eq.S(pos)) then
            if(pos+L.ge.n2) dy=dy+1
            pos=v1
            dir=3
            touched3(v1)=1
            goto 100
         else
            dir=1
            touched(pos)=1
            goto 100
         endif
 130     v1=pos-1
         if(mod(pos,L).eq.0) v1=v1+L
         if(S(v1).eq.S(pos)) then
            if(mod(pos,L).eq.0) dx=dx-1
            pos=v1
            dir=4
            touched4(v1)=1
            goto 100
         else
            dir=2
            touched2(pos)=1
            goto 100
         endif
 140     v1=pos-L
         if(v1.lt.0) v1=v1+n2
         if(S(v1).eq.S(pos)) then
            if(pos-L.lt.0) dy=dy-1
            pos=v1
            dir=1
            touched(v1)=1
            goto 100
         else
            dir=3
            touched3(pos)=1
            goto 100
         endif
 200     continue
      enddo

 565    format(i8,i4,i12,i12,i12,i12,i6)
      if(nb+2*nc-n2-np.eq.2) then
         ncr=2
      else
         if(nb+2*nc-n2-np.eq.0) then
            if(ncross.eq.0) then
               ncr=0
            else
               ncr=1
            endif
         else
            print*,"BUG",nb+2*nc-n2-np,nb,nc,n2,np,ncross
!            stop
         endif
      endif
      return
      end
c###############################################################
c
c     SUBROUTINE analysis1
c
      subroutine analysis1(S,nncr)
      implicit none
      integer L,n2
```

```
      parameter (L=512,n2=L*L)
      integer s(0:n2-1),ttouched(0:n2-1)
      integer touched(0:n2-1),s0
      integer ncluster
      integer pile(1:n2),pos,nc
      integer i,ii,ni,j,k,j0
      integer n,indice_c,lpos,index,v1,v2,v3,v4
      integer start,dir,dir0,length
      integer npile(1:n2)
      integer i1,i2,pos1,pos2,ki
      integer ncr(-5:5,-5:5),nncr,incr,jncr
c
      do i=-5,5
         do ii=-5,5
            ncr(i,ii)=0
         enddo
      enddo
      do k=0,n2-1
         touched(k)=0
      enddo
      ncluster=0
      do k=0,n2-1
         if(touched(k).eq.0) then
            ncluster=ncluster+1
            indice_c=1
            pile(1)=k
            touched(k)=ncluster
            do i=1,n2
               if(i.gt.indice_c) goto 99
               pos=pile(i)
               v1=pos+1
               v2=pos-1
               v3=pos+L
               v4=pos-L
               if(mod(pos+1,L).eq.0) v1=v1-L
               if(mod(pos,L).eq.0) v2=v2+L
               if(v3.ge.n2) v3=v3-n2
               if(v4.lt.0) v4=v4+n2
               if(s(v1).eq.s(k)) then
                  if(touched(v1).eq.0) then
                     touched(v1)=ncluster
                     indice_c=indice_c+1
                     pile(indice_c)=v1
                  endif
               endif
               if(s(v2).eq.s(k)) then
                  if(touched(v2).eq.0) then
                     touched(v2)=ncluster
                     indice_c=indice_c+1
                     pile(indice_c)=v2
                  endif
               endif
               if(s(v3).eq.s(k)) then
                  if(touched(v3).eq.0) then
                     touched(v3)=ncluster
                     indice_c=indice_c+1
                     pile(indice_c)=v3
                  endif
               endif
               if(s(v4).eq.s(k)) then
                  if(touched(v4).eq.0) then
                     touched(v4)=ncluster
                     indice_c=indice_c+1
                     pile(indice_c)=v4
                  endif
               endif
```

```
            enddo
 99         continue
         endif
      enddo
!     I put to zero all the spin which are not in the largest cluster and 1 otherwise
!     New version which computes all the loops.
!!!!!!!!!!!!!!!!
!!!  I first take a startig point such that   1 | 0
      k=0
      ki=0
      do i=0,n2-1
         ttouched(i)=1
      enddo
 100  continue
      do i=ki,n2-1
         if(ttouched(i).ne.0) then
            if(touched(i).gt.0) then
               s0=touched(i)
               j=i+1
               if(mod(j,L).eq.0) j=j-L
               if(touched(j).ne.s0) then
                  start=i
                  dir=1
                  goto 101
               endif
               j=i+L
               if(j.ge.n2) j=j-n2
               if(touched(j).ne.s0) then
                  start=i
                  dir=2
                  goto 101
               endif
               j=i-1
               if(mod(i,L).eq.0) j=j+L
               if(touched(j).ne.s0) then
                  start=i
                  dir=3
                  goto 101
               endif
               j=i-L
               if(j.lt.0) j=j+n2
               if(touched(j).ne.s0) then
                  start=i
                  dir=4
                  goto 101
               endif
            endif
         endif
      enddo
      goto 160
 101  continue
      length=0
      pos1=0
      pos2=0
      dir0=dir
      i=start
      ki=i+1
!!  I will start computing at start=i,dir=1
      goto 103
c
 102  continue
      if((i.eq.start).and.(dir.eq.dir0)) then
         if((pos1.ne.0).or.(pos2.ne.0)) then
            if(mod(pos1,L).ne.0) then
               print*,"Strange",pos1
               stop
```

```
            endif
            pos1=pos1/L
            if(mod(pos2,L).ne.0) then
               print*,"Strange",pos2
               stop
            endif
            pos2=pos2/L
            if(pos1.eq.0) then
               ncr(0,abs(pos2))=ncr(0,abs(pos2))+1
            else
               if(pos2.eq.0) then
                  ncr(abs(pos1),0)=ncr(abs(pos1),0)+1
               else
                  if(pos1*pos2.gt.0) then
                     ncr(abs(pos1),abs(pos2))=ncr(abs(pos1),abs(pos2))+1
                  else
                     ncr(abs(pos1),-abs(pos2))=
     c                    ncr(abs(pos1),-abs(pos2))+1
                  endif
               endif
            endif
         endif
c            do i2=9,0,-1
c               write(*,4421)(touched(i2*10+i1),i1=0,9)
c            enddo
c          endif
c 4421    format(10(i4))
         goto 150
       endif
 103   length=length+1
       npile(length)=i
       j0=j
       if(dir.eq.1) goto 110
       if(dir.eq.2) goto 120
       if(dir.eq.3) goto 130
       goto 140
c
c        Y  | Z
c        -    -
c        X  | 0
c
c
 110   j=i+L
       if(j.ge.n2) j=j-n2
       if(touched(j).ne.s0) then
          dir=2
       else
          i=j
          j=j+1
          if(mod(j,L).eq.0) j=j-L
          if(touched(j).eq.s0) then
             i=j
             j=j0
             dir=4
             pos1=pos1+1
             pos2=pos2+1
          else
             pos2=pos2+1
          endif
       endif
       goto 102
 120   j=i-1
       if(mod(i,L).eq.0) j=j+L
       if(touched(j).ne.s0) then
          dir=3
       else
```

```
         i=j
         j=j+L
         if(j.ge.n2) j=j-n2
         if(touched(j).eq.s0) then
            i=j
            j=j0
            dir=1
            pos1=pos1-1
            pos2=pos2+1
         else
            pos1=pos1-1
         endif
      endif
      goto 102
 130  j=i-L
      if(j.lt.0) j=j+n2
      if(touched(j).ne.s0) then
         dir=4
      else
         i=j
         j=j-1
         if(mod(i,L).eq.0) j=j+L
         if(touched(j).eq.s0) then
            i=j
            j=j0
            dir=2
            pos1=pos1-1
            pos2=pos2-1
         else
            pos2=pos2-1
         endif
      endif
      goto 102
 140  j=i+1
      if(mod(j,L).eq.0) j=j-L
      if(touched(j).ne.s0) then
         dir=1
      else
         i=j
         j=j-L
         if(j.lt.0) j=j+n2
         if(touched(j).eq.s0) then
            i=j
            j=j0
            dir=3
            pos1=pos1+1
            pos2=pos2-1
         else
            pos1=pos1+1
         endif
      endif
      goto 102
 150  continue
      do i=1,length
         ttouched(npile(i))=0
      enddo
      goto 100
 160  continue
!!!!!!!!!!!!!!!!!!
! Sanity check. Checking that only one type of crossing exists.
      nncr=0
      do i=-5,5
         do ii=-5,5
            if(ncr(i,ii).ne.0) then
               nncr=nncr+1
               incr=i
```

```
               jncr=ii
            endif
         enddo
      enddo
      if(nncr.gt.1) then
         print*,"Strange, more than one type of crossing"
         do i=-5,5
            do ii=-5,5
               if(ncr(i,ii).ne.0) print*,i,ii,ncr(i,ii)
            enddo
         enddo
         stop
      endif

!     Now I check the type of crossings.
!     Produces incr,jncr. (0,1) = one vertical , (0,2) = two verticals et cetera
!     One vertical means really one.
      nncr=0
      if(jncr.eq.0) then
         if(incr.eq.1) then
            if(ncr(incr,jncr).eq.2) then
               nncr=2
               goto 199
            endif
            if(ncr(incr,jncr).le.4) then
               nncr=4
               goto 199
            endif
            if(ncr(incr,jncr).ge.5) then
               nncr=6
               goto 199
            endif
            goto 200
         endif
         goto 200
      endif
      if(incr.eq.0) then
         if(jncr.eq.1) then
            if(ncr(incr,jncr).eq.2) then
               nncr=3
               goto 199
            endif
            if(ncr(incr,jncr).le.4) then
               nncr=5
               goto 199
            endif
            if(ncr(incr,jncr).ge.5) then
               nncr=7
               goto 199
            endif
            goto 200
         endif
         goto 200
      endif
      if((incr.eq.1).and.(jncr.eq.1)) then
         if(ncr(incr,jncr).eq.2) then
            nncr=8
            goto 199
         endif
         if(ncr(incr,jncr).ge.4) then
            nncr=10
            goto 199
         endif
         goto 200
      endif
      if((incr.eq.1).and.(jncr.eq.-1)) then
```

```
         if(ncr(incr,jncr).eq.2) then
            nncr=9
            goto 199
         endif
         if(ncr(incr,jncr).ge.4) then
            nncr=11
            goto 199
         endif
         goto 200
      endif
      if((incr.eq.1).and.(jncr.eq.2)) then
         nncr=12
         goto 199
      endif
      if((incr.eq.1).and.(jncr.eq.-2)) then
         nncr=13
         goto 199
      endif
      if((incr.eq.2).and.(jncr.eq.1)) then
         nncr=14
         goto 199
      endif
      if((incr.eq.2).and.(jncr.eq.-1)) then
         nncr=15
         goto 199
      endif
cccccc Only exotic cases remain.
      nncr=16
 199  continue
      goto 201
 200  continue
      print*,"Strange2",incr,jncr,ncr(incr,jncr)
      do i2=9,0,-1
         write(*,4421)(S(i2*10+i1),i1=0,9)
      enddo
 4421 format(10(i3))
      stop
 201  continue
      return
      end
c###############################################################
c
c     SUBROUTINE analysis2
c
      subroutine analysis2(S)
      implicit none
      integer L,n2
      parameter (L=512,n2=L*L)
      integer s(0:n2-1),theta(0:n2-1)
      integer touched(0:n2-1),itime
      integer it,ncluster,lcluster,pcluster
      integer pile(1:n2),maxs,pos,nc
      integer i,ii,ni,j,k,wrapping
      integer lh,n,indice_c,lpos,index,v1,v2,v3,v4
      integer start,dir,dir0,length,angle,j0,n0,n4,nm4
      integer store(1:20*L),storea(1:20*L),al1,al2
      integer ntheta1(1:2*L),ntheta2(1:2*L)
      real*8 atheta1(1:2*L),atheta2(1:2*L)
      integer i1,i2,pos1,pos2,ki
c
      common/an/n4,n0,nm4,al1,al2,lcluster
      common/an2/atheta1,atheta2,ntheta1,ntheta2

c
c     We will store the value of the largest cluster.
c     Next we will compute the winding of this cluster
```

```
      do i=1,2*L
         ntheta1(i)=0
         ntheta2(i)=0
         atheta1(i)=0.d0
         atheta2(i)=0.d0
      enddo
c
      do k=0,n2-1
         touched(k)=0
      enddo
      maxs=0
      ncluster=0
      lcluster=0
      do k=0,n2-1
         if(touched(k).eq.0) then
            ncluster=ncluster+1
            indice_c=1
            wrapping=0
            pile(1)=k
            touched(k)=ncluster
            do i=1,n2
               if(i.gt.indice_c) goto 99
               pos=pile(i)
               v1=pos+1
               v2=pos-1
               v3=pos+L
               v4=pos-L
               if(mod(pos+1,L).eq.0) v1=v1-L
               if(mod(pos,L).eq.0) v2=v2+L
               if(v3.ge.n2) v3=v3-n2
               if(v4.lt.0) v4=v4+n2
               if(s(v1).eq.s(k)) then
                  if(touched(v1).eq.0) then
                     touched(v1)=ncluster
                     indice_c=indice_c+1
                     pile(indice_c)=v1
                  endif
               endif
               if(s(v2).eq.s(k)) then
                  if(touched(v2).eq.0) then
                     touched(v2)=ncluster
                     indice_c=indice_c+1
                     pile(indice_c)=v2
                  endif
               endif
               if(s(v3).eq.s(k)) then
                  if(touched(v3).eq.0) then
                     touched(v3)=ncluster
                     indice_c=indice_c+1
                     pile(indice_c)=v3
                  endif
               endif
               if(s(v4).eq.s(k)) then
                  if(touched(v4).eq.0) then
                     touched(v4)=ncluster
                     indice_c=indice_c+1
                     pile(indice_c)=v4
                  endif
               endif
            enddo
 99         continue
            if(indice_c.gt.maxs) then
               lcluster=indice_c
               pcluster=k
            endif
            maxs=max(maxs,indice_c)
```

```
         endif
      enddo
c     I put to zero all the spin which are not in the largest cluster and 1 otherwise
      k=touched(pcluster)
      j=0
      do i=0,n2-1
         if(touched(i).eq.k) then
            touched(i)=1
            j=j+1
         else
            touched(i)=0
         endif
      enddo
ccccccccccccccccccccccccc
cccc  I first take a startig point such that   1 | 0
      k=0
      ki=0
 100  continue
      do i=ki,n2-1
         if(touched(i).eq.1) then
            j=i+1
            if(mod(j,L).eq.0) j=j-L
            if(touched(j).eq.0) then
               start=i
               dir=1
               goto 101
            endif
            j=i+L
            if(j.ge.n2) j=j-n2
            if(touched(j).eq.0) then
               start=i
               dir=2
               goto 101
            endif
            j=i-1
            if(mod(i,L).eq.0) j=j+L
            if(touched(j).eq.0) then
               start=i
               dir=3
               goto 101
            endif
            j=i-L
            if(j.lt.0) j=j+n2
            if(touched(j).eq.0) then
               start=i
               dir=4
               goto 101
            endif
         endif
      enddo
      goto 160
 101  continue
      k=k+1
      length=0
      angle=0
      theta(0)=0
      dir0=dir
      i=start
      ki=i+1
ccc   I will start computing at start=i,dir=1
      goto 103
c
 102  continue
      if((i.eq.start).and.(dir.eq.dir0)) then
         goto 150
      endif
```

```
 103  length=length+1
      theta(length)=angle
      touched(j)=-1
      j0=j
      if(dir.eq.1) goto 110
      if(dir.eq.2) goto 120
      if(dir.eq.3) goto 130
      goto 140
c
c        Y  | Z
c        -    -
c        X  | 0
c
c
 110  j=i+L
      if(j.ge.n2) j=j-n2
      if(touched(j).le.0) then
         dir=2
         angle=angle+1
      else
         i=j
         j=j+1
         if(mod(j,L).eq.0) j=j-L
         if(touched(j).eq.1) then
            i=j
            j=j0
            dir=4
            angle=angle-1
         endif
      endif
      goto 102
 120  j=i-1
      if(mod(i,L).eq.0) j=j+L
      if(touched(j).le.0) then
         dir=3
         angle=angle+1
      else
         i=j
         j=j+L
         if(j.ge.n2) j=j-n2
         if(touched(j).eq.1) then
            i=j
            j=j0
            dir=1
            angle=angle-1
         endif
      endif
      goto 102
 130  j=i-L
      if(j.lt.0) j=j+n2
      if(touched(j).le.0) then
         dir=4
         angle=angle+1
      else
         i=j
         j=j-1
         if(mod(i,L).eq.0) j=j+L
         if(touched(j).eq.1) then
            i=j
            j=j0
            dir=2
            angle=angle-1
         endif
      endif
      goto 102
 140  j=i+1
```

```
      if(mod(j,L).eq.0) j=j-L
      if(touched(j).le.0) then
         dir=1
         angle=angle+1
      else
         i=j
         j=j-L
         if(j.lt.0) j=j+n2
         if(touched(j).eq.1) then
            i=j
            j=j0
            dir=3
            angle=angle-1
         endif
      endif
      goto 102
 150  continue
c      print*,"loop is done and lenght = ",length,angle,s0,s00
      store(k)=length
      storea(k)=angle
      if(angle.eq.0) then
ccc   compute the average square angle for wrapping clusters
         do i=1,length
            do j=1,min(2*L,length/2)
               ii=i+j
               if(ii.gt.length) ii=ii-length
               atheta1(j)=atheta1(j)+(theta(i)
     c              -theta(ii))**2
               ntheta1(j)=ntheta1(j)+1
            enddo
         enddo
      endif
      if(angle.eq.4) then
ccc   compute the average square angle for wrapping clusters
         do i=1,length
            do j=1,min(2*L,length/2)
               ii=i+j
               if(ii.gt.length) then
                  ii=ii-length
                  atheta2(j)=atheta2(j)+(theta(i)-4
     c                 -theta(ii))**2
                  ntheta2(j)=ntheta2(j)+1
               else
                  atheta2(j)=atheta2(j)+(theta(i)
     c                 -theta(ii))**2
                  ntheta2(j)=ntheta2(j)+1
               endif
            enddo
         enddo
      endif
 159  continue
      goto 100
 160  continue
ccccccccccccccccccccc
ccc   We analyse the output
      n0=0
      n4=0
      nm4=0
      al1=0
      al2=0
      do i=1,k
         if(storea(i).eq.-4) then
            nm4=nm4+1
            al2=al2+store(i)
         else
            if(storea(i).eq.0) then
```

```
               n0=n0+1
               al1=al1+store(i)
            else
               if(storea(i).eq.4) then
                  n4=n4+1
                  al1=al1+store(i)
                  if(n4.gt.1) then
                     print*,"Second angle 4",n4
                     do j=1,k
                        print*,store(j),storea(j)
                     enddo
                     stop
                  endif
               else
                  print*,"Strange angle"
                  print*,(store(j),storea(j),j=1,k)
               endif
            endif
         endif
      enddo
      return
      end
ccccccccccccccccccccccccccccccccccccccccccccccccccc

      subroutine analysisf(s,itime,largest)
      implicit none
      integer L,n2,k,indice_c,mL
      parameter (L=512,n2=512*512)
      parameter (mL=16)
      integer s(0:n2-1)
      integer touched(0:n2-1)
      integer i,nc
      integer v1,v2,v3,v4,c1,c2,Index,lpos
      integer ix,iy,ixp,iyp,delta,itime
      integer largest,slargest,il,isl
      integer pile(1:n2)
      real*8 cor(1:30,1:mL)
      real*8 ccor(1:30,1:mL)
      common/na/cor,ccor

      largest=0
      slargest=0



!     Now compute the number of clusters
!
      do k=0,n2-1
         touched(k)=0
      enddo
      nc=0
      do k=0,n2-1
         if(touched(k).ne.0) goto 12
         pile(1)=k
         nc=nc+1
         touched(k)=1
         indice_c=1
         do index=1,n2
            if(index.gt.indice_c) goto 10
            lpos=pile(index)
            v1=lpos+1
            v2=lpos-1
            v3=lpos+L
            v4=lpos-L
            if(mod(lpos+1,L).eq.0) v1=v1-L
            if(mod(lpos,L).eq.0) v2=v2+L
```

```
            if(v3.ge.n2) v3=v3-n2
            if(v4.lt.0) v4=v4+n2
            if(touched(v1).eq.0) then
               if(S(v1).eq.S(lpos)) then
                  indice_c=indice_c+1
                  pile(indice_c)=v1
                  touched(v1)=nc
               endif
            endif
            if(touched(v2).eq.0) then
               if(S(v2).eq.S(lpos)) then
                  indice_c=indice_c+1
                  pile(indice_c)=v2
                  touched(v2)=nc
               endif
            endif
            if(touched(v3).eq.0) then
               if(S(v3).eq.S(lpos)) then
                  indice_c=indice_c+1
                  pile(indice_c)=v3
                  touched(v3)=nc
               endif
            endif
            if(touched(v4).eq.0) then
               if(S(v4).eq.S(lpos)) then
                  indice_c=indice_c+1
                  pile(indice_c)=v4
                  touched(v4)=nc
               endif
            endif
         enddo
  10     continue
         if (indice_c.gt.largest) then
            slargest=largest
            isl=il
            largest=indice_c
            il=nc
         else
            if(indice_c.gt.slargest) then
               slargest=indice_c
               isl=nc
            endif
         endif
 12      continue
      enddo
      write(2,13)itime,largest,il,slargest,isl,nc
 13   format(i8,5(i12))
ccccccccccccccc Compute the Cor. Functions ccccccccccccccccc
      do i=1,mL
         if(i.eq.1) delta=1
         if(i.eq.2) delta=2
         if(i.eq.3) delta=3
         if(i.eq.4) delta=4
         if(i.eq.5) delta=6
         if(i.eq.6) delta=8
         if(i.eq.7) delta=12
         if(i.eq.8) delta=16
         if(i.eq.9) delta=24
         if(i.eq.10) delta=32
         if(i.eq.11) delta=48
         if(i.eq.12) delta=64
         if(i.eq.13) delta=96
         if(i.eq.14) delta=128
         if(i.eq.15) delta=192
         if(i.eq.16) delta=256
         if(i.eq.17) delta=384
```

```
         if(i.eq.18) delta=512
         if(i.eq.19) delta=768
         if(i.eq.20) delta=1024
         if(i.eq.21) delta=1536
         if(i.eq.22) delta=2048
         if(i.eq.23) delta=3072
         if(i.eq.24) delta=4096
         c1=0
         c2=0
         do iy=0,L-1
            iyp=iy+delta
            if(iyp.ge.L) iyp=iyp-L
            do ix=0,L-1
               ixp=ix+delta
               if(ixp.ge.L) ixp=ixp-L
               if(touched(iy*L+ix).eq.touched(iyp*L+ix)) c1=c1+1
               if(touched(iy*L+ix).eq.touched(iy*L+ixp)) c1=c1+1
               c2=c2+S(iy*L+ix)*S(iyp*L+ix)
               c2=c2+S(iy*L+ix)*S(iy*L+ixp)
           !    print*, c2
            enddo
         enddo
         cor(itime,i)=cor(itime,i)+dfloat(c1)/2.d0/n2
         ccor(itime,i)=ccor(itime,i)+dfloat(c2)/2.d0/n2
      enddo

      return
      end
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
!cc   Prints the configuration needed
!ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine config(spin,field)
      parameter (L=512,n2=L*L)
      integer    s(0:L*L-1)
      real         rf(0:L*L-1),field(1:L,1:L)
      character*1  signe(0:L*L-1),signe2(0:L*L-1)
      integer       j,k,index
      integer       spin(1:L,1:L)

      do i=1,L
         do j=1,L
         s(i-1+(j-1)*L)=spin(i,j)
         rf(i-1+(j-1)*L)=field(i,j)
         enddo
      enddo
          do k=0,L-1
              do j=0,L-1
                 index=k*L+j
                 if(s(index).eq.1) then
                     signe(index)='+'
                     signe2(index)='+'
                     if(rf(index).gt.0.d0) signe2(index)='x'
                 else if (s(index).eq.-1) then
                     signe(index)='-'
                     signe2(index)='-'
                     if (rf(index).lt.0.d0) signe2(index)='='
                 else
                     signe(index)='o'
                 endif

              enddo
          enddo

          do k=0,L-1
              print*,signe(L*(L-1)-k*L:(L*L-1)-k*L),"           "
     c              ,signe2(L*(L-1)-k*L:(L*L-1)-k*L)
```

```
          enddo
          print*,"   "
      end
```

Bibliografia

[1] CUGLIANDOLO L., *Coarsening Phenomena*, C. R. Phys. 16, 257 (2015); doi : 10.1016/j.crhy.2015.02.005.

[2] CASTELLANO C., FORTUNATO S., LORETO V., *Statistical Physics of Social Dynamics*, Rev. Mod. Phys. 81, 591 (2009) doi : 10.1103/RevModPhys.81.591.

[3] TILMAN D., KAREIVA P., *Spatial Ecology - The Role of Space in Population Dynamics and Interspecific Interactions*, Princeton University Press, Princeton (1997)

[4] REICHL L. E., *A Modern Course in Statistical Physics*, Edward Arnold Publishers LTD, Usa, 1980.

[5] GLAUBER R. J., *Time-Dependent Statistics of the Ising Model*, J. Math. Phys. 4, 294 (1963); doi : 10.1063/1.1703954.

[6] SICILIA A., ARENZON J. J., DIERKING I., BRAY A. J., CUGLIANDOLO L. F., MARTINEZ-PEDIGUERO J., ALONSO I., PINTRE C. I., *Experimental test of curvature-driven dynamics in the phase ordering of a two dimensional liquid crystal*, Phys. Rev. Lett. 101, 197801 (2008); doi : 10.1103/PhysRevLett.101.197801

[7] BINDER K., MULLER-KRUMBHAAR H., *Dynamic Properties of the Monte Carlo Method in Statistical Mechanics*, J. Stat. Phys. 8, 1 (1973); doi : 10.1007/BF01008440.

[8] HUANG K., *Statistical Mechanics*, Wyley, New York, 1987.

[9] BRAY A. J., *Theory of Phase Ordering Kinetics*, Adv. Phys. 43, 357 (1994); doi : 10.1080/00018739400101505.

[10] BRAY A. J., *Universal Scaling Function for Domain Growth in the Glauber-Ising Chain*, J. Phys. A: Math. Gen. 23 L67; doi : http://dx.doi.org/10.1088/0305-4470/23/2/005.

[11] ARENZON J. J., BRAY A. J., CUGLIANDOLO L. F, SICILIA A., *Exact Results for Curvature-Driven Coarsening in Two Dimensions*, Phys. Rev. Lett. 98, 145701 (2007); doi : 10.1103/PhysRevLett.98.145701.

[12] GODRECHE C., *Dynamics of the Directed Ising Chain*, J. Stat. Mech. (2011); P04005 doi : 10.1088/1742-5468/2011/04/P04005.

[13] GODRECHE C., PLEIMLING M., *Dynamics of the Two-dimensional Directed Ising Model in the Paramagnetic Phase*, J. Stat. Mech. (2014); P05005 doi : 10.1088/1742-5468/2014/05/P05005.

[14] INSALATA F., *Ising Model : Statics and Dynamics of the Symmetric and Directed Chain*, undergraduate Thesis for bachelor degree in Physics, University of Salerno (2012).

[15] FLORY P. J., *Molecular Size Distribution in Three Dimensional Polymers. I. Gelation*, J. Am. Chem. Soc., 11, 63 (1941); doi : 10.1021/ja01856a061.

[16] STOCKMAYER W. H., *Theory of Molecular Size Distribution and Gel Formation in Branched-Chain Polymers*, J. Chem. Phys. 11, 45 (1943); doi : 10.1063/1.1723803.

[17] BROADBENT S. R., HAMMERSLEY J. M., *Percolation Processes*, Mathematical Proceedings of the Cambridge Philosophical Society, 3, 53 (1957); doi:10.1017/S0305004100032680.

[18] ABAIMOV S. G., *Statistical Physics of Non-Thermal Phase Transitions*, Springer, London, 2015

[19] KRISTENSEN K., MOLONEY N. R., *Complexity and Criticality*, Imperial College Press, London, 2005

[20] SMIRNOV S., WERNER W., *Critical Exponents for Two-Dimensional Percolation*, Math. Res. Lett., 8, 5-6 (2001).

[21] STAUFFER D., AHARONY A., *Introduction to Percolation Theory : Revised Second Edition*, Taylor and Francis, London, 1994.

[22] Lawler G. F., SCHRAMM O., WERNER W., *One-Arm Exponent for Critical 2D Percolation*, Electron. J. Probab. vol. 7, 1–13, (2002); doi : 10.1214/EJP.v7-101.

[23] OLEJARZ J., KRAPIVSKY P. L., REDNER S., *Fate of 2D Kinetic Ising Ferromagnets and Critical Percolation Crossing Probabilities*, Phys. Ref. Lett. 109, 195702 (2012); doi : 10.1103/PhysRevLett.109.195702.

[24] BLANCHARD T., CUGLIANDOLO L.F., PICCO M., *A Morphological Study of Cluster Dynamics Between Critical Points*, J. Stat. Mech. (2012) P05026; doi : 10.1088/1742-5468/2012/05/P05026.

[25] IMRY Y., MA S., *Random-Field Instability of the Ordered State of Continuous Symmetry*; Phys. Rev. Lett. 35, 1399 (1975); doi : http://dx.doi.org/10.1103/PhysRevLett.35.1399.

[26] LAI Z.W., MAZENKO G. F., VALLS O. T., *Classes for Growth Kinetics Problems at low Temperatures*, Phys. Rev. B 37, 9481 (1988); doi : http://dx.doi.org/10.1103/PhysRevB.37.9481.

[27] FISHER D.S., LE DOUSSAL P., MONTHUS C., *Nonequilibrium Dynamics of Random Field Ising Spin Chains: Exact Results via Real Space RG* Phys. Rev. E 64, 066107 (2001); doi : 10.1103/PhysRevE.64.066107.

[28] CORBERI F., CANDIA A., LIPPIELLO E., ZANNETTI M., *Off-Equilibrium Eesponse Function in the One-Dimensional Random-Field Ising model*, Phys. Rev. E 65, 046114 (2002); doi : http://dx.doi.org/10.1103/PhysRevE.65.046114.

[29] CORBERI F., LIPPIELLO E., MUKHERJEE A., PURI S., ZANNETTI M., *Crossover in Growth Law and Violation of Superuniversality in the Random Field Ising Model*, Phys. Rev. E 85, 021141 (2012); doi : 10.1103/PhysRevE.85.021141.F.

[30] PURI S., PAREKH N., *Non-Algebraic Domain Growth for Phase Ordering Dynamics in a Random Field*, J. Phys. A 26, 2777 (1993); doi : http://dx.doi.org/10.1088/0305-4470/26/12/018.

[31] RAO M., CHAKRABARTI A., *Conserved Dynamics of a Two-Dimensional Random Field Model*, Phys. Rev. E 48, R25(R) (1993); doi : http://dx.doi.org/10.1103/PhysRevE.48.R25.

[32] RAO M., CHAKRABARTI A., *Kinetics of Domain Growth in a Random-Field Model in Three Dimensions*, Phys. Rev. Lett. 71, 3501 (1993); doi : http://dx.doi.org/10.1103/PhysRevLett.71.3501.

[33] IKEDA H., ENDOH Y., ITOH S., *Ordering Kinetics in a Two-Dimensional Percolation Magnet*, Phys. Rev. Lett. 64, 1266 (1990); doi : http://dx.doi.org/10.1103/PhysRevLett.64.1266.

[34] SINHA S., MANDAL P. K., *Characterization of Kinetic Coarsening in a Random Field Ising Model*, Phys. Rev. E 89, 042144 (2014); doi : 10.1103/PhysRevE.89.042144.

[35] CORBERI F., *Coarsening in Inhomogeneous Systems*, Comptes rendus - Physique 16, 332 (2015); doi : 10.1016/j.crhy.2015.03.019.

[36] HUSE D. A., HENLEY C.L., *Pinning and Roughening of Domain Walls in Ising Systems Due to Random Impurities*, Phys. Rev. Lett. 54, 2708 (1985); doi : http://dx.doi.org/10.1103/PhysRevLett.54.2708.

[37] CORBERI F., LIPPIELLO E., MUKHERJEE A., PURI S., ZANNETTI M., *Growth Law and Superuniversality in the Coarsening of Disordered Ferromagnets*, J. Stat. Mech., P03016 (2011) ; doi : 10.1088/1742-5468/2011/03/P03016.

[38] SHENOY D.K., SELINGER J.V., GRUNENBERG K.A., NACIRI J., SHASHIDHAR R., *Coarsening of Reverse Tilt Domains in Liquid-Crystal Cells with Heterogeneous Alignment Layers*, Phys. Rev. Lett. 82, 1716 (1999);

[39] LIKODIMOS V., LABARDI M., ALLEGRINI M., *Kinetics of Ferroelectric Domains Investigated by Scanning Force Microscopy*, Phys. Rev. B 61, 14440 (2000); doi : http://dx.doi.org/10.1103/PhysRevB.61.14440.

[40] BURIONI R., CASSI D., CORBERI F., VEZZANI A., *Phase-Ordering Kinetics on Graphs*, Phys. Rev. E 75, 011113 (2007); doi : 10.1103/PhysRevE.75.011113.

[41] CORBERI F., LIPPIELLO E., ZANNETTI M., *Interface Fluctuations, Bulk Fluctuations and Fimensionality in the off Equilibrium Eesponse of Coarsening Systems*, Phys. Rev. E 63, 061506 (2001); doi : 10.1103/PhysRevE.63.061506

[42] BARROS K., KRAPIVSKY P. L., REDNER S., *Freezing into Stripe States in Two-Dimensional Ferromagnets and Crossing Probabilities in Critical Percolation*, Phys. Rev. E 80, 040101 (2009); doi : 10.1103/PhysRevE.80.040101.

[43] BLANCHARD T., PICCO M., *Frozen into Stripes : Fate of the Critical Ising Model after a Quench*, Phys. Rev. E 88, 032131 (2013); doi : 10.1103/PhysRevE.88.032131.

[44] BLANCHARD T., *Morphology of Domains in and out of Equilibrium* PhD Thesis, Pierre et Marie Curie University - Paris VI, 2014

[45] SPIRIN V., KRAPIVSKY P. L., REDNER S., *Fate of Zero-Temperature Ising Ferromagnets*, Phys. Rev. E 63, 036118 (2001); doi : http://dx.doi.org/10.1103/PhysRevE.63.036118.

[46] BLANCHARD T., CORBERI F., CUGLIANDOLO L. F., PICCO M., *How Soon after a Zero-Temperature Quench is the Fate of the Ising Model Sealed?*, Europhys. Lett. 106 66001 (2014); doi : 10.1209/0295-5075/106/66001.

[47] ARON C., CHAMON C., CUGLIANDOLO L. F., PICCO M., *Scaling and Super-Universality in the Coarsening Dynamics of the 3d Random Field Ising model*, J. Stat. Mech., P05016 (2008); doi : 10.1088/1742-5468/2008/05/P05016

[48] LIPPIELLO E., MUKHERJEE A., PURI S., ZANNETTI M., *Scaling Behavior of Response Functions in the Coarsening Dynamics of Disordered Ferromagnets*, Europhys. Lett. 90, 46006 (2010); doi : 10.1209/0295-5075/90/46006.

[49] HENKEL M., PLEIMLING M., *Superuniversality in Phase-Ordering Disordered Ferromagnets*, Phys. Rev. B78, 224419 (2008); doi : 10.1103/PhysRevB.78.224419.

[50] BORTZ A.B., KALOS M.H., LEBOWITZ J.L., *A new algorithm for Monte Carlo simulation of Ising spin systems*, Journal of Comp. Phys., 17, 1 (1975); doi : 10.1016/0021-9991(75)90060-1.

Acknowledgments

I thank my co-supervisors Professor Leticia Cugliandolo and Dr. Marco Picco for having accepted to work with me, giving me the unique opportunity to spend 5 months at Laboratoire de Physique Théorique et Hautes Énergies of Université Pierre et Marie Curie - Paris VI. I really appreciate all the time they have taken to share their knowledge and experience with me. I am grateful to Prof. Federico Corberi for his infinite patience in guiding me in every part of this experience, from the first discussion about the Erasmus Programme to the last piece of advice on this Thesis, without neglecting everything he taught me. His heroic attempts to teach me how to write properly about Physics succeeded (maybe) mostly thanks to his stoic attitude.

I also wish to thank Professor Gaetano Busiello, correlatore of this Thesis, for his kindness in the last hectic days of writing.

Finally, I will always remember all the professors of University of Salerno who have passionately and amicably guided me through our Science and/or Mathematics. Even when the number of my questions grew exponentially with time [1].

I find it very boring and superfluous to thank my family, they already know. As for my dearest friends, I want to share my picture with them.

Typical obsessive compulsive behavior of the subject while thinking about anything.

Last but not least, this work is dedicated to Angelina Moscato.

Ferdinando Gelsomino Insalata

[1] Often being in number of N, with N arbitrarily large.