SAPIENZA
UNIVERSITÀ DI ROMA

Out-of-equilibrium measure of critical parameters for second order phase transitions

Scuola di dottorato Vito Volterra

Dottorato di Ricerca in Fisica – XXV Ciclo

Candidate

Matteo Lulli
ID number 1105583

Thesis Advisors

Prof. Giorgio Parisi
Prof. Andrea Pelissetto

Co-Advisor

Prof. Massimo Bernaschi

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Physics

October 2014

Thesis not yet defended

Out-of-equilibrium measure of critical parameters for second order phase transitions
Ph.D. thesis. Sapienza – University of Rome
ISBN: 000000000-0

This thesis has been typeset by LaTeX and the Sapthesis class.

Version: December 7, 2014

Author's email: matteo.lulli@gmail.com

To all those that took a chance on me

Acknowledgements

The financial support for this thesis has been almost completely provided by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n° [247328]. I acknowledge the support of NVIDIA Corporation with the donation of the GTX Titan GPUs used for this research in the framework of the NVIDIA Academic Partnership. I would also wish to acknowledge the GREFI-MEFI network and the CNR-IAC for funding the 2012 Beg Rohu Summer School and 2014 Rome GPU conference respectively.

It is hard to convey my gratitude for Giorgio Parisi who gave me the chance to have my PhD in the field of statistical mechanics of disordered systems under his guidance. He trusted my abilities although I knew just a few of what was needed when I started, and allowed me to live an experience whose impotance from both a professional and a human points of view I am beginning to appreciate just now.

I wish to greatfully thank Andrea Pelissetto: I greatly benefitted from his profound understanding of the physics we worked on and I hope to retain many precious lessons I have learned from him. I am mostly thankful for the all the patience he had with me during these years of my PhD.

My work on the GPUs is based on that of Massimo Bernaschi whose approach deeply influenced my point of view on the subject. I wish to thank him for his efforts in giving me many opportunities to express and share my scientific production and for the crucial suggestions he gave me, helping me to choose my path.

Without the precious and reliable help of Andrea Maiorano it would not have been possible for me to face such a big computational effort as the one this thesis has required. His patience in teaching me the 'secrets' of Bash scripting and of system management has been mostly precious, along with the physical help he gave me in setting up 'piovra', the cluster I used for the simulations.

Federico Ricci-Tersenghi has opened the way for my PhD in this group at the very beginning. All I can say, even if it is probably not enough is: thank you for trusting me.

I wish to thank Lorenzo Rovigatti and Mario Spera since at the very beginning of my PhD they have been most important in getting me acquainted with the use of GPUs, sharing with and teaching me their experience on the subject.

I am deeply thankful to Andrea Parmeggiani and Avi Aminov who supported me in a delicate moment of my PhD. Talking with them truly helped me.

I also wish to thank all those friends that supported me in different times of my PhD, some of them even every day like Pietro Coletti, Matteo Figliuzzi, Marcello Carta, Fabrizio Antenucci and Andrea Maselli. The long lasting talks I had with Matteo Mitrano have been crucial in finding the right attitude in facing my PhD.

At the very basis, none of this would have been possible without the constant and lovely support of my family and of my neocathecumenal community who helped me, sometimes even ruoghly, keeping my feet on the ground during the good and the difficult times. Particularly, I would like to mention my aunt Patrizia and my uncle Bruno for the time they gave me, my mother for keeping giving me precious advices

even when I did not understand it and finally Paolo, Raffaele, Paola, Anna Lisa, Michele, Francesca and Maria for their constant presence, physically and in prayer.

Finally, I wish to express my gratitude to God who guided me all my life as a Father. That my work could be for his greater glory (Psalm 116:2) "For his loving kindness is great toward us and the Lord's faithfulness endures forever."

Abstract

In this thesis we discuss a new out-of-equilibrium finite-size scaling method which can be used to determine the critical parameters, *i.e.* the critical temperature and critical exponents, of any second order phase transition without reaching the thermal equilibrium. Such a method is based on the Renormalization Group framework just as equilibrium finite-size scaling is.

This technique is most interesting for those system exhibiting very slow relaxation dynamics. In most cases equilibrium methods are difficult to apply since thermalization is attained only for relatively small system sizes, often at a high computational cost.

As a representative case of this class of problems, we apply our new method to the three-dimensional bimodal Edwards-Anderson model describing the behaviour of an Ising spin-glass. The very determination of its critical temperature and critical exponents has been the subject of intense studies in the last thirty years, during which, different estimates of the critical parameters have been given. In order to obtain more precise estimates it has always been necessary to employ large amounts of computational resources, and, very often, dedicated machines have been used.

In this thesis we present the estimates of the critical parameters obtained using the developed out-of-equilibrium method. Results are compatible and comparable to those obtained most recently from equilibrium simulations executed on a dedicated machine. Our estimates are the second most precise so far.

In order to obtain these results we exploited the computational resources given by recent Graphics Processing Units (GPUs), which allowed us to exceed by, at least, one order of magnitude the performances of recent high-end CPUs. We wrote a highly tuned code implementing the Metropolis Monte Carlo (MC) dynamics using which we could obtain the necessary amount of statistics.

The thesis is organized as follows. In Chapter 1 we give a basic introduction to the Renormalization Group approach to second-order phase transitions, determining the scaling form in the relaxation out-of-equilibrium regime. We also describe some features of spin-glasses highlighting the problems posed by finite-dimensional models. In Chapter 2 we briefly review the basic concepts of GPU programming, stressing the importance of changing the point of view on data structures as usually implemented for serial execution on CPUs. In Chapter 3 we describe our implementation of the MC simulation for single- and multi-GPU systems reviewing the past literature and discussing the solutions we found for different problems. In Chapter 4 we discuss in some further detail the out-of-equilibrium framework as based on Renormalization Group scaling arguments, giving a detailed account of the past literature. We conclude by discussing our out-of-equilibrium finite-size scaling ansatz. In Chapter 5 we finally present our results: we begin by giving the out-of-equilibrium generalization of observables which are usually defined in the equilibrium regime, paying some attention to the characterization of their out-of-equilibrium scaling properties. We then continue by discussing the approximations we adopted and the various data analyses. Moreover, some consistency checks are shown. Further details are reported in the Appendices A and B. Finally, in Chapter 6 we summarize the results and we draw conclusions.

Contents

Chapter 1

Introduction

In this chapter we review the Renormalization Group (RG) approach as applied to **second-order**, or more precisely **continuous**, phase transitions in statistical mechanics models. Indeed, the RG is not an analytical tool allowing us to calculate detailed properties of a given model, but rather a method that predicts general scaling properties characterizing universal large-scale critical behaviour.

We show how in this context it is possible to employ RG to predict finite-size scaling relations that can be applied to finite-size systems Monte Carlo (MC) simulations, allowing us to study phase transitions whose critical behaviour cannot be determined by means of analytical methods.

Finally, we present the physical system we are interested in describing its phenomenology along with the Statistical Mechanics model describing it. In this presentation we follow the lines of [1, 2].

1.1 The Renormalization Group

1.1.1 Statistical Mechanics and Critical Systems

In equilibrium statistical mechanics the starting point for the description of the equilibrium thermodynamics of a given system is the Hamiltonian $\mathcal{H} = \mathcal{H}(\{\mathcal{C}\}, \{K_n\})$, which is a function of the set of N generalized dynamic coordinates $\{\mathcal{C}\}$, and M coupling constants $\mathbf{K} = \{K_n\}$. Generally speaking, the Hamiltonian can be written as

$$\mathcal{H}_N(\{\mathcal{C}\}, \mathbf{K}) = -\beta H_N(\{\mathcal{C}\}, \mathbf{K}) = \sum_{n=1}^{M} K_n \Theta_n(\{\mathcal{C}\}) \tag{1.1}$$

where $\beta = (k_B T)^{-1}$ is the inverse absolute temperature and Θ_n are functions of the coordinates $\{\mathcal{C}\}$. Next, one defines the **partition function**, or **partition sum**, Z

$$Z_N(\mathbf{K}) = \sum_{\{\mathcal{C}\}} \exp\left[\mathcal{H}_N(\{\mathcal{C}\}, \mathbf{K})\right], \tag{1.2}$$

where the sum $\sum_{\{\mathcal{C}\}}$ is over **all** possible values of the generalized coordinates.

The link with the equilibrium thermodynamics of the system is provided by the free-energy density

$$f(N, \beta, \mathbf{K}) = -\frac{1}{N\beta} \log Z_N(\mathbf{K}) = u - Ts, \tag{1.3}$$

where $u = U/N$ and $s = S/N$ are the internal energy and entropy densities, respectively. Thermodynamic observables, such as susceptibilities, magnetization, heat capacity and others, are obtained by taking the proper derivatives of (1.3).

Indeed, f is proportional to the logarithm of Z, which is a sum of exponential functions, which are analytic. Hence, for finite-size systems no singular behaviour can emerge.

However, when one takes the thermodynamic limit, *i.e.* $N \to \infty$, the partition sum Z might develop singularities, *e.g.* zeros: an infinite sum of analytic functions need not be analytic itself. It follows that phase transitions only occur in the thermodynamic limit. In fact, before the pioneering work of L. Onsager in 1944 [3] it was still doubted that phase transitions could be described by statistical mechanics models of finite-dimensional systems.

Observables show, in general, a singular behaviour at critical points, both for discontinuous and continuous transitions. Such a singular behaviour only occurs in the infinite-volume limit.

By using the RG approach, one can argue that the infinite-volume free-energy density can be written as a sum of a singular f_s and of a regular part f_b at the critical point in the thermodynamic limit, where the latter is usually referred to as **background**

$$f(\beta, \mathbf{K}) = \lim_{N\to\infty} f(N, \beta, \mathbf{K}) = f_s(\beta, \mathbf{K}) + f_b(\beta, \mathbf{K}). \tag{1.4}$$

Let us now focus on second-order or continuous phase transitions and define the two-point correlation function as $G = G(|\vec{x} - \vec{y}|) = G(r)$ for isotropic systems. One observes that for $\beta \neq \beta_c$, where β_c is the critical inverse temperature, the correlation length ξ takes on a finite value. The system is correlated at distances of order ξ and the correlation function exponentially decreases for $r \gg \xi$, which means that thermal fluctuations are statistically independent at large distances. More precisely, the correlation function in d dimensions can be generally written as

$$G(r) \propto \frac{\exp(-r/\xi)}{r^{d-2}}, \tag{1.5}$$

as long as ξ is finite.

On the other hand, at $\beta = \beta_c$ the correlation length diverges, which means that the correlation function behaves as a power-law

$$G(r) \propto \frac{1}{r^{d-2+\eta}}, \tag{1.6}$$

where η is the anomalous dimension. A diverging ξ implies that thermal fluctuations are statistically correlated along the entire (infinite in the thermodynamic limit) system. Differently from first-order transition, there is no phase coexistence in this case, so no latent heat is present for second-order phase transitions.

1.1.2 Universality and Scaling at the critical point

We give here a brief historical introduction of the concepts of *universality* and *scaling* focusing, as a test-ground, on magnetic systems in order to fix the notation. Above,

we mentioned that near a continuous phase transition the correlation length diverges. More precisely, ξ diverges as a power law near β_c, *i.e.*

$$\xi \sim \varepsilon^{-\nu}, \qquad \varepsilon = \frac{\beta - \beta_c}{\beta_c}, \tag{1.7}$$

where ν is called **critical exponent**. Other quantities behave as a power law near the critical point, defining an entire set of critical exponents.

Let c be the specific heat and χ the susceptibility of the order parameter, the magnetization m, with respect to a small change of an external field H. In principle, according to the direction of the critical temperature limit, *i.e.* $\varepsilon \to 0^{\pm}$, one may expect to have different values of the exponents. Taking this into account, we can write the following relations

$$\begin{cases} c & \sim \varepsilon^{-\alpha} \\ \chi & \sim \varepsilon^{-\gamma} \\ m & \sim \varepsilon^{\beta} \\ \xi & \sim \varepsilon^{-\nu} \end{cases} \quad \text{for} \quad \varepsilon > 0,\, H = 0 \qquad \begin{cases} c & \sim -\varepsilon^{-\alpha'} \\ \chi & \sim -\varepsilon^{-\gamma'} \\ m & \sim -\varepsilon^{\beta'} \\ \xi & \sim -\varepsilon^{-\nu'} \end{cases} \quad \text{for} \quad \varepsilon < 0,\, H = 0 \tag{1.8}$$

Moreover, at the critical temperature the following relations hold

$$\begin{aligned} &m \sim H^{1/\delta}, \quad \varepsilon = 0, \\ &G(r) \sim r^{2-d-\eta}, \quad \varepsilon = 0,\, H = 0. \end{aligned} \tag{1.9}$$

Experimentally it was found that the critical exponents defined above were not independent. First of all, the exponents appearing on both sides of the critical point, $\varepsilon \to 0^{\pm}$, are the same, which means we can drop the prime so that we end up with six exponents; note, however, that amplitudes differ. Second, there are only two independent exponents, because of the following **scaling relations**:

$$\begin{aligned} \alpha + 2\beta + \gamma &= 2, \\ \delta - 1 &= \frac{\gamma}{\beta}, \\ 2 - \alpha &= d\nu, \\ \nu(2 - \eta) &= \gamma. \end{aligned} \tag{1.10}$$

The first three relations are called Rushbrook [4, 5], Widom [6, 7] and Josephson [8, 9] identities, respectively. None of these relations has been rigorously proved. Only inequalities have been proved. These relations hold in the two-dimensional Ising model and in other systems for which an exact solution has been found. They have been also verified extensively in numerical studies and experiments. They will be justified below by using the RG approach. Having six exponents and four scaling relations, we conclude that there exist only two independent exponents, that, once evaluated, can be used to obtain the four remaining ones.

One of the most important results of the RG is the following: near the critical point the large-scale behaviour of dimensionless quantities is the same for a broad class of system whose description at the microscopic level might differ significantly. Such a class is defined by the dimensionality and the symmetry properties of the order

parameter, by the space dimensionality of the model itself and whether interactions are long- or short-range. Such a set of models defines a *static* **universality class**. We will see that the large-scale behaviour at criticality can be parametrized using critical exponents which must have the same value for every model belonging to a given universality class. This is the definition of **universality**.

The critical power-law behaviour of, *e.g.*, liquid and ferromagnetic systems was already explained by the van der Waals and the Weiss mean-field theories, respectively. The name *mean field* indicates that these theories focus on the behaviour of the order parameter discarding thermal fluctuations: each degree of freedom is described as interacting with the averaged behaviour of the rest of the system. Universality was also explained in a unifying framework by the phenomenological theory of Landau, who showed that different problems could be described by means of very similar functional forms.

While it correctly predicts power-law behaviours and universality, mean-field theory is not quantitatively correct. For instance, the critical exponents calculated exactly in the two-dimensional Ising model do not have the expected mean-field values. Second, the magnetization exponent β was measured in simple liquids finding $\beta \simeq 1/3$ instead of the mean-field value $\beta = 1/2$.

These facts could not be explained in the framework of Landau mean-field theory: they have been finally addressed with the introduction of the Renormalization Group.

1.1.3 Kadanoff's idea

In 1966 L.P. Kadanoff [10] proposed an argument which could qualitatively account for the existence of Widom scaling hypothesis [11] and for the scaling relations among critical exponents. The whole argument rests upon the divergence of the correlation length ξ at the critical point.

Let us consider a system with short-range interactions at some inverse temperature β and let a be a microscopic characteristic length scale, *e.g.*, the lattice spacing: degrees of freedom in regions of size ℓa, with $a \ll \ell a \ll \xi(\beta)$, will essentially behave as a single variable. Hence, we perform a coarse graining of the degrees of freedom in a block of dimension ℓa and write a new Hamiltonian for these new variables, called *block variables*. Now, Kadanoff's hypothesis was that, since in the original theory the interactions were short-ranged, so should they be for the block variables. What are the consequences of the scale transformation on the new system? Let us consider a magnetic system coupled to a thermal bath at reduced temperature $\varepsilon = (\beta - \beta_c)/\beta_c$ and to a reduced external field $h = \beta H$, keeping in mind that the critical values of the control parameters are $\varepsilon = h = 0$. The singular part of the free energy is assumed to scale as

$$N f_s(\varepsilon, h) = N \ell^{-d} f_s(\varepsilon_\ell, h_\ell), \tag{1.11}$$

since the number of the block variables is $N\ell^{-d}$, where ε_ℓ and h_ℓ are the control parameters for the blocked lattice. Hence, the singular part of the free-energy density scales as

$$f_s(\varepsilon, h) = \ell^{-d} f_s(\varepsilon_\ell, h_\ell), \tag{1.12}$$

Under a scale transformation the correlation length decreases by a factor ℓ

$$\xi_\ell = \xi/\ell, \tag{1.13}$$

so that after the transformation the system gets further away from criticality. Hence, the block variables, which are correlated on shorter distances, are coupled to a different value of the reduced temperature. A similar argument holds for the external field.

Since we are interested in studying the scaling behaviour near the critical temperature, we assume that the control parameters transform as

$$\varepsilon_\ell = \varepsilon \ell^{y_\varepsilon}, \qquad h_\ell = h\ell^{y_h}, \qquad y_\varepsilon, y_h > 0. \tag{1.14}$$

The singular part of the free-energy density scales therefore as

$$f_s(\varepsilon, h) = \ell^{-d} f_s(\varepsilon \ell^{y_\varepsilon}, h\ell^{y_h}). \tag{1.15}$$

We are free to choose the value of ℓ, and we do so by fixing $|\varepsilon|\ell^{y_\varepsilon} = 1$, obtaining

$$\ell = |\varepsilon|^{-1/y_\varepsilon}. \tag{1.16}$$

As a result, we get the following scaling form for the free energy

$$f_s(\varepsilon, h) = |\varepsilon|^{d/y_\varepsilon} f_s(\pm 1, h/|\varepsilon|^{y_h/y_\varepsilon}) = |\varepsilon|^{-d/y_\varepsilon} F_\pm(h/|\varepsilon|^{y_h/y_\varepsilon}). \tag{1.17}$$

This expression allows one to derive Widom scaling form [11] for the equation of state

$$h = m^\delta g\left(\frac{\varepsilon}{m^{1/\beta}}\right). \tag{1.18}$$

where g is some scaling function. Let us set $h = 0$ in (1.17): we have that $f_s(\varepsilon, h) \propto |\varepsilon|^{d/y_\varepsilon}$, so that the specific heat reads

$$c \propto |\varepsilon|^{d/y_\varepsilon - 2}, \tag{1.19}$$

allowing us to identify the critical exponent α as

$$2 - \alpha = \frac{d}{y_\varepsilon}. \tag{1.20}$$

For the correlation length ξ we can repeat the scaling argument we just discussed for the free energy, obtaining

$$\xi(\varepsilon) = \ell\, \xi(\varepsilon_\ell) = \ell\, \xi(\varepsilon \ell^{y_\varepsilon}) = |\varepsilon|^{-1/y_\varepsilon} \xi(1), \tag{1.21}$$

where $\xi(1)$ is a constant. Thus, we identify the exponent ν as

$$\nu = \frac{1}{y_\varepsilon}. \tag{1.22}$$

Comparing equations (1.20) and (1.22), we obtain Josephson identity

$$2 - \alpha = d\nu. \tag{1.23}$$

Indeed, Kadanoff's argument gives us a deep insight into the physics of the problem, allowing us to determine some scaling relations linking critical exponents. At the same time it leaves us with no means for calculating the value of critical exponents nor with an explanation for universality.

1.1.4 The Renormalization Group

Let us now introduce the main concepts of the Renormalization Group (RG) as proposed in the seminal papers of K.G. Wilson [12, 13] and F. J. Wegner [14]. The RG stands as a mathematical generalization of Kadanoff's ideas. First of all, it is easy to understand that after a scale transformation the new Hamiltonian cannot have the same functional form as the original one: considering an Ising model with nearest-neighbour interactions only, after a block transformation the new Hamiltonian does include next-to-nearest neighbour interactions too. It is possible to take this behaviour into account by considering the most general Hamiltonian compatible with the symmetries of the problem in the form of eq. (1.1). Such Hamiltonian functions are defined on the space of all possible coupling constants $\mathbf{K}$. However, we stress that this construction should not be considered as 'rigorous', since many technical aspects are still under debate [15, 16, 17, 18]. Nonetheless, it is useful for the implementation of approximation schemes which eventually lead to controlled quantitative results.

Hence, the starting point is that the functional form of the Hamiltonian changes under scale transformations. The mapping between Hamiltonians is called **Renormalization Group transformation** acting on the coupling space. Indicating the RG transformation of length scale $\ell > 1$ as R_ℓ we write

$$\mathbf{K}' = R_\ell(\mathbf{K}). \tag{1.24}$$

We immediately notice that, by definition, R_ℓ is not invertible and that the transformations form a semi-group since they satisfy the composition rule

$$R_{\ell_1\ell_2}(\mathbf{K}) = R_{\ell_1}\, R_{\ell_2}(\mathbf{K}), \tag{1.25}$$

i.e. two successive transformations of length scales ℓ_1 and ℓ_2 must correspond to a single one of scale $\ell_1\ell_2$. Finally, we stress that eq. (1.24) represents a non-linear transformation, hence the new coupling constants $\mathbf{K}'$ are generally non-linear functions of $\mathbf{K}$.

Let us now characterize the action of the RG transformation on the partition sum and on the free-energy density: the partition function of the original system reads

$$Z_N(\mathbf{K}) = \sum_{\{\mathcal{C}\}} \exp\left[\mathcal{H}_N(\{\mathcal{C}\}, \mathbf{K})\right]; \tag{1.26}$$

the block transformation acts on the exponential as

$$\exp\left[\mathcal{H}_{N'}(\{\mathcal{C}'\}, \mathbf{K}')\right] = \sum_{\{\mathcal{C}\}} P(\{\mathcal{C}\}, \{\mathcal{C}'\}) \exp\left[\mathcal{H}_N(\{\mathcal{C}\}, \mathbf{K})\right], \tag{1.27}$$

where $N' = \ell^{-d}N$, and $P(\{\mathcal{C}\}, \{\mathcal{C}'\})$ is a projection operator.

It is generally assumed that the projector possesses the following properties:

1. $P(\{\mathcal{C}\}, \{\mathcal{C}'\}) \geq 0$,
2. $P(\{\mathcal{C}\}, \{\mathcal{C}'\})$ should possess the symmetries of the original system,
3. $\sum_{\{\mathcal{C}'\}} P(\{\mathcal{C}\}, \{\mathcal{C}'\}) = 1$.

The first property guarantees that the right-hand side of (1.27) is positive, so that one can rewrite is as the exponential of a new effective Hamiltonian which has the same symmetries of the original one because of the second property. Finally, the third property implies that the partition sum is invariant under the RG transformation, in fact we could write

$$\begin{aligned} Z_{N'}(\mathbf{K}') &= \sum_{\{\mathcal{C}'\}} \exp\left[\mathcal{H}_{N'}(\{\mathcal{C}'\},\mathbf{K}')\right] = \sum_{\{\mathcal{C}'\}}\sum_{\{\mathcal{C}\}} P(\{\mathcal{C}\},\{\mathcal{C}'\}) \exp\left[\mathcal{H}_N(\{\mathcal{C}\},\mathbf{K})\right] \\ &= \sum_{\{\mathcal{C}\}} \exp\left[\mathcal{H}_N(\{\mathcal{C}\},\mathbf{K})\right] = Z_N(\mathbf{K}). \end{aligned} \tag{1.28}$$

It follows that the singular part of the free-energy density scales as

$$f_s(\mathbf{K}) = \frac{1}{N} F_s(\mathbf{K}) = \frac{\ell^{-d}}{N'} F_s(\mathbf{K}') = \ell^{-d} f_s(\mathbf{K}'). \tag{1.29}$$

One remark is in order: although the transformation (1.24) is highly non-trivial it is, however, analytic and the main advantage given by the RG framework is that it is possible to approximate it.

We can now identify the origin of the singular behaviour: the RG transformation (1.24) defines a recursion relation, the so-called **renormalization flow** for the coupling constants, which, after an infinite number of iterations, *i.e.*, when all degrees of freedom have been integrated out, reaches a fixed point

$$\mathbf{K}^* = R_\ell(\mathbf{K}^*). \tag{1.30}$$

We can understand the behaviour of (1.30) by using an analogy, studying a damped dynamic system described by the differential equation

$$\dot{x} = -\frac{\mathrm{d}\, V(x)}{\mathrm{d}x} = -\frac{\mathrm{d}}{\mathrm{d}x}\left[a(x-x_c)^2 + b(x-x_c)^4 + c\right], \tag{1.31}$$

with $a < 0$ and $b > 0$ so that there are two minima, x_1 and x_2, for the potential $V(x)$, and $x = x_c$ is a maximum, *i.e.* an unstable point, with $x_1 < x_c < x_2$. The infinite-time limit of the solution of the differential equation (1.31), $x = f(t, x_0)$, is a discontinuous function of the initial condition x_0

$$x_2 = \lim_{x_0 \to x_c^+} \lim_{t\to\infty} f(t,x_0) \neq \lim_{x_0 \to x_c^-} \lim_{t\to\infty} f(t,x_0) = x_1. \tag{1.32}$$

If we associate x_1, x_2 and x_c to the RG fixed points and the time variable to the number of applications of the RG transformation, this is exactly what happens in the limit of an infinite number of iterations: performing a continuous change in the initial choice of the coupling constants $\mathbf{K}$, the fixed point can change discontinuously.

It is possible to characterize fixed points by using the recursion relation for the correlation length which reads

$$\xi(\mathbf{K}^*) = \xi(\mathbf{K}^*)/\ell. \tag{1.33}$$

Hence, the fixed-point value for the correlation length can either be $\xi(\mathbf{K}^*) = 0$, defining **trivial fixed points**, or $\xi(\mathbf{K}^*) \to \infty$, defining **critical fixed points**.

It is easy to see that the basin of attraction of a critical fixed point is the set of initial conditions for which the correlation length is divergent, *i.e.* critical fixed points must have unstable directions for the RG flow.

Let us now study the properties of the RG transformation near fixed points. Indeed, the recursion relation (1.24) is non-linear but it is possible to give a linear approximation of it near a fixed point. Hence, we write each coupling constant as its fixed-point value plus a small deviation

$$K_n = K_n^* + \delta K_n, \tag{1.34}$$

which clearly transform as

$$K_n'(\mathbf{K}) = K_n^* + \delta K_n'. \tag{1.35}$$

We can write the Taylor expansion of the last expression as

$$K_n'(\mathbf{K}^* + \delta\mathbf{K}) = K_n^* + \sum_m \frac{\partial K_n'}{\partial K_m}\bigg|_{K_m=K_m^*} \delta K_m + O\left((\delta K)^2\right), \tag{1.36}$$

and finally write the transformed variations as

$$\delta K_n' = \sum_m \frac{\partial K_n'}{\partial K_m}\bigg|_{K_m=K_m^*} \delta K_m = \sum_m M_{nm}\delta K_m. \tag{1.37}$$

Now, we make a rather simplifying assumption: the diagonalizability of the matrix M. The eigenvalue equation for the matrix M reads

$$\sum_m M_{mn}\, e_m^{(\sigma)} = \Lambda^{(\sigma)} e_n^{(\sigma)}, \tag{1.38}$$

where $e_m^{(\sigma)}$ and $\Lambda^{(\sigma)}$ stand for the eigenvector and the eigenvalue respectively, both labeled by σ. Because of the semi-group property of R_ℓ one has that $M^{(\ell_1)}M^{(\ell_2)} = M^{(\ell_1\ell_2)}$, which implies for the eigenvalues

$$\Lambda_{\ell_1}^{(\sigma)}\Lambda_{\ell_2}^{(\sigma)} = \Lambda_{\ell_1\ell_2}^{(\sigma)}. \tag{1.39}$$

Obviously, $\Lambda_{\ell=1}^{(\sigma)} = 1$. We take the derivative of both sides of equation (1.39) as follows

$$\begin{aligned} \frac{\partial}{\partial\ell_2}\left(\Lambda_{\ell_1}^{(\sigma)}\Lambda_{\ell_2}^{(\sigma)}\right)\bigg|_{\ell_2=1} &= \frac{\partial}{\partial\ell_2}\Lambda_{\ell_2}^{(\sigma)}\bigg|_{\ell_2=1}\Lambda_{\ell_1}^{(\sigma)} = y_\sigma\,\Lambda_{\ell_1}^{(\sigma)}, \\ \frac{\partial}{\partial\ell_2}\Lambda_{\ell_1\ell_2}^{(\sigma)}\bigg|_{\ell_2=1} &= \frac{\mathrm{d}}{\mathrm{d}\ell}\Lambda_{\ell}^{(\sigma)}\frac{\partial\ell}{\partial\ell_2}\bigg|_{\ell_2=1} = \ell_1\frac{\mathrm{d}}{\mathrm{d}\ell_1}\Lambda_{\ell_1}^{(\sigma)}, \end{aligned} \tag{1.40}$$

where we have defined the intermediate variable $\ell = \ell_1\ell_2$. Equating these two last lines we get the differential equation

$$\ell_1\frac{\mathrm{d}}{\mathrm{d}\ell_1}\Lambda_{\ell_1}^{(\sigma)} = y_\sigma\,\Lambda_{\ell_1}^{(\sigma)}, \tag{1.41}$$

which can be integrated as

$$\int_{\Lambda_1^{(\sigma)}}^{\Lambda_\ell^{(\sigma)}} \frac{\mathrm{d}\Lambda_{\ell_1}^{(\sigma)}}{\Lambda_{\ell_1}^{(\sigma)}} = y_\sigma\int_1^\ell\frac{\mathrm{d}\ell_1}{\ell_1}, \tag{1.42}$$

thus leading to the following expression for the eigenvalue of the linearized RG transformation

$$\Lambda_\ell^{(\sigma)} = \ell^{y_\sigma}. \tag{1.43}$$

Now, we continue by expanding the coupling constants variation $\delta\mathbf{K}$ on the eigenvector basis as

$$\delta\mathbf{K} = \sum_\sigma \left(\delta\mathbf{K}\cdot\mathbf{e}^{(\sigma)}\right)\,\mathbf{e}^{(\sigma)} = \sum_\sigma u_\sigma\,\mathbf{e}^{(\sigma)}, \tag{1.44}$$

where the projection of the coupling constants on the eigenvector basis $\delta\mathbf{K}\cdot\mathbf{e}^{(\sigma)} = u_\sigma$ is the so-called **scaling field**. The RG flow of $\delta\mathbf{K}$ reads

$$\delta\mathbf{K}' = \mathbf{M}^{(\ell)}\,\delta\mathbf{K} = \mathbf{M}^{(\ell)}\sum_\sigma u_\sigma\,\mathbf{e}^{(\sigma)} = \sum_\sigma u_\sigma\,\Lambda_\ell^{(\sigma)}\mathbf{e}^{(\sigma)} = \sum_\sigma u'_\sigma\,\mathbf{e}^{(\sigma)}. \tag{1.45}$$

Hence, some scaling fields will grow along the RG flow, some will shrink and some will be unchanged according to the corresponding eigenvalues. Scaling fields are usually classified as

- **relevant**: $\Lambda_\ell^{(\sigma)} = \ell^{y_\sigma} > 1, \Rightarrow\ y_\sigma > 0$;
- **marginal**: $\Lambda_\ell^{(\sigma)} = \ell^{y_\sigma} = 1, \Rightarrow\ y_\sigma = 0$;
- **irrelevant**: $\Lambda_\ell^{(\sigma)} = \ell^{y_\sigma} < 1, \Rightarrow\ y_\sigma < 0$;

where, clearly, $\ell > 1$. The relevant scaling fields are those corresponding to the control parameters which need to be tuned experimentally in order to reach the critical point. For ferromagnets we need to be at the critical temperature and at zero external field, hence temperature and magnetic field are relevant variables.

It is possible to classify the nature of fixed points according to their **codimension**, *i.e.* the difference between the dimensionality of the coupling-constant space and that of the basin of attraction. This concept is useful because it holds also in the limit of infinite dimensional spaces. Since we are dealing with continuous phase transitions we only consider two cases:

1. codimension 0 and $\xi(\mathbf{K}^*) = 0$: this fixed point is called **sink** and its basin of attraction is defined by values of the coupling constants belonging to a definite phase of the system;
2. codimension 2 or greater and $\xi(\mathbf{K}^*) \to \infty$: this is a critical fixed point.

We can now examine the scaling behaviour of the singular part of the free-energy density in proximity of the critical fixed point. Let u_{r_k}, $k = 1, \ldots, R$, be the relevant scaling fields, u_{m_k}, $k = 1, \ldots, M$, and u_{i_k}, $k = 1, \ldots, \infty$, the marginal and irrelevant ones respectively. Hence, we get

$$\begin{aligned} &f_s(u_{r_1}, \ldots, u_{r_R}, u_{m_1}, \ldots, u_{m_M}, u_{i_1}, \ldots) \\ =&\ell^{-d} f_s(u_{r_1}\ell^{y_{r_1}}, \ldots, u_{r_R}\ell^{y_{r_R}}, u_{m_1}, \ldots, u_{m_M}, u_{i_1}\ell^{y_{i_1}}, \ldots). \end{aligned} \tag{1.46}$$

We are always left with the freedom of choice for the length scale ℓ, and since we want to study the critical limit $u_{r_k} \ll 1$ as the scale changes we can choose $\ell = |u_{r_1}|^{-1/y_{r_1}}$,

so that we keep fixed the product $|u_{r_1}|\ell^{y_{r_1}} = 1$. Indeed, there is no need to choose the scaling field u_{r_1} over the other relevant ones, we do so to keep the notation simple. We substitute in the previous expression obtaining

$$\begin{aligned} &f(u_{r_1}, \ldots, u_{r_R}, u_{m_1}, \ldots, u_{m_M}, u_{i_1}, \ldots) \\ &= |u_{r_1}|^{d/y_{r_1}} f(\pm 1, \ldots, u_{r_R}|u_{r_1}|^{-y_{r_R}/y_{r_1}}, u_{m_1}, \ldots, u_{m_M}, u_{i_1}|u_{r_1}|^{-y_{i_1}/y_{r_1}}, \ldots). \end{aligned} \tag{1.47}$$

By definition $y_{i_k}/y_{r_1} < 0$ and $y_{r_k}/y_{r_1} > 0$ so that in the critical limit $u_{r_k} \to 0$ the irrelevant scaling fields vanish, while the marginal scaling fields give rise to logarithmic corrections.

As we pointed out for the Kadanoff picture, in a much more naive way, the scaling-field exponents y_σ are related to the usual critical exponents, and the scaling behaviour near the critical point naturally emerges performing the RG transformation. Scaling forms of other quantities, *e.g.* the correlation function, allow us to recover scaling relations between different critical exponents.

The behaviour near the critical point is dictated only by relevant scaling fields and not by the starting point of the RG transformation. Hence, the original form of the Hamiltonian plays no role: this mechanism accounts for the existence of universality. Finally, in the RG framework it is possible to compute the values of critical exponents once a reasonable approximation of the RG transformation is given. Since in this work we are mainly interested in numerical estimations of critical parameters we do not go into further details.

Finally, we summarize the relations linking the various critical exponents of a magnetic system to those of the relevant scaling fields, y_ε and y_h:

$$\begin{aligned} \nu &= \frac{1}{y_\varepsilon}, & \alpha &= 2 - \frac{d}{y_\varepsilon}, \\ \eta &= d + 2(1 - y_h), & \delta &= \frac{y_h}{d - y_h}, \\ \beta &= \frac{1}{y_\varepsilon}(d - y_h), & \gamma &= \frac{1}{y_\varepsilon}(2y_h - d). \end{aligned} \tag{1.48}$$

1.2 Scaling in Monte Carlo Simulations

Monte Carlo (MC) simulations are one of the most valuable techniques when an analytic approach is not available for a given model. Basically, the probability distribution of the degrees of freedom is evolved according to a Markov chain, and eventually converges to the equilibrium distribution defined as

$$P_N(\{\mathcal{C}\}, \mathbf{K}) = \frac{1}{Z_N} \exp\left[\mathcal{H}_N(\{\mathcal{C}\}, \mathbf{K})\right]. \tag{1.49}$$

However, since we know that phase transitions cannot occur for finite systems, which are indeed the only ones we can simulate, we review first some general properties which can be effectively employed in numerical simulations.

1.2.1 Finite-Size Scaling

Let us consider a magnetic system for which in the thermodynamic limit the two relevant scaling fields are those associated with the reduced temperature and the

external field, *i.e.* u_ε and u_h. Near the critical point, where $u_\varepsilon = u_h = 0$, they can be expanded as

$$u_\varepsilon = \varepsilon + O(\varepsilon^2, h^2), \qquad u_h = h + O(\varepsilon h), \tag{1.50}$$

with $\varepsilon = (\beta - \beta_c)/\beta_c$. The RG transformation is local, hence it is still meaningful to apply it to a finite system. Let L be the linear size of the system, the associated scaling field should be $u_L \sim L^{-1}$ since under the RG transformation one has

$$L^{-1} \to \ell L^{-1}, \tag{1.51}$$

so that the associated eigenvalue exponent is $y_L = 1$. We want to stress that this is a good approximation in the case of *periodic* boundary conditions since in the general case the scaling field should contain higher order terms such as $u_L \sim L^{-1} + L^{-2} + L^{-3} + \cdots$, which is the case for fixed boundary conditions [19, 20, 21, 22]. There is no evidence of such behaviour for periodic systems.

Setting $h = 0$ and discarding marginal scaling fields, the scaling form for the singular part of the free-energy density reads

$$f_s(\varepsilon, L^{-1}, u_{i_1}, \ldots, u_{i_I}) = |\varepsilon|^{d/y_\varepsilon} f_s(\pm 1, L^{-1}|\varepsilon|^{-1/y_\varepsilon}, u_{i_1}|\varepsilon|^{-y_{i_1}/y_\varepsilon}, \ldots, u_{i_I}|\varepsilon|^{-y_{i_I}/y_\varepsilon}). \tag{1.52}$$

Now, we substitute the expressions for the critical exponents, $d/y_\varepsilon = 2 - \alpha$ and $1/y_\varepsilon = \nu$ and define a new scaling function such that the scaling form explicitly depends on the system size

$$\begin{aligned} &f_s(\varepsilon, L^{-1}, u_{i_1}, \ldots, u_{i_I}) \\ &= |\varepsilon|^{2-\alpha} f_s(\pm 1, L^{-1}|\varepsilon|^{-\nu}, u_{i_1}|\varepsilon|^{-y_{i_1}\nu}, \ldots) \\ &= |\varepsilon|^{2-\alpha} (L^{-1}|\varepsilon|^{-\nu})^{\frac{2-\alpha}{\nu}} (L^{-1}|\varepsilon|^{-\nu})^{\frac{\alpha-2}{\nu}} f_s(\pm 1, L^{-1}|\varepsilon|^{-\nu}, u_{i_1}|\varepsilon|^{-y_{i_1}\nu}, \ldots), \end{aligned} \tag{1.53}$$

where, up to now, we have just added two prefactors whose product is one. It is possible to write the product $(L^{-1}|\varepsilon|^{-\nu})^{\frac{\alpha-2}{\nu}} f_s(\pm 1, L^{-1}|\varepsilon|^{-\nu}, \ldots) = \tilde{f}_s(\pm 1, L^{-1}|\varepsilon|^{-\nu}, \ldots)$ thus defining a new scaling function $\tilde{f}$. We continue by writing

$$\begin{aligned} &|\varepsilon|^{2-\alpha} (L^{-1}|\varepsilon|^{-\nu})^{\frac{2-\alpha}{\nu}} \tilde{f}_s(\pm 1, L^{-1}|\varepsilon|^{-\nu}, u_{i_1}|\varepsilon|^{-y_{i_1}\nu}, \ldots) \\ &= |\varepsilon|^{2-\alpha} (L^{-1}|\varepsilon|^{-\nu})^{\frac{2-\alpha}{\nu}} \tilde{f}_s(\pm 1, (\varepsilon L^{1/\nu})^{-\nu}, u_{i_1} L^{y_{i_1}} (\varepsilon L^{1/\nu})^{-y_{i_1}\nu}, \ldots), \end{aligned} \tag{1.54}$$

where it appears that the dependence on $\varepsilon L^{1/\nu}$ can be factorized on each variable of $\tilde{f}$. Thus we define a new scaling function F_L such that

$$\begin{aligned} &|\varepsilon|^{2-\alpha} (L^{-1}|\varepsilon|^{-\nu})^{\frac{2-\alpha}{\nu}} \tilde{f}_s(\pm 1, (\varepsilon L^{1/\nu})^{-\nu}, u_{i_1} L^{y_{i_1}} (\varepsilon L^{1/\nu})^{-y_{i_1}\nu}, \ldots) \\ &= |\varepsilon|^{2-\alpha} (L^{-1}|\varepsilon|^{-\nu})^{\frac{2-\alpha}{\nu}} F_L(\varepsilon L^{1/\nu}, u_{i_1} L^{y_{i_1}}, \ldots) \\ &= L^{\frac{\alpha-2}{\nu}} F_L(\varepsilon L^{1/\nu}, u_{i_1} L^{-\omega_1}, \ldots) \end{aligned} \tag{1.55}$$

where we have renamed the irrelevant scaling fields exponents as $\omega_k = -y_{i_k}$, with $\omega_k > 0$. Supposing to sort them in an increasing order, *i.e.* $\omega_1 < \omega_2 < \ldots$, we can expand the previous expression as

$$\begin{aligned} &f_s(\varepsilon, L^{-1}, u_{i_1}, \ldots) \\ &= L^{\frac{\alpha-2}{\nu}} G_L(\varepsilon L^{1/\nu}) \left[1 + \frac{u_{i_1}}{L^{\omega_1}} g_{1L}(\varepsilon L^{1/\nu})\right] + O\left(L^{-2\omega_1}, L^{-\omega_{k>1}}\right). \end{aligned} \tag{1.56}$$

Hence, we expect data belonging to different linear sizes and reduced temperatures for $L^{(2-\alpha)/\nu} f_s$, to lie on the same curve as a function of $\varepsilon L^{1/\nu}$, modulo scaling corrections proportional to $L^{-\omega_1}$.

For RG invariant quantities the scaling relations do not involve any prefactor proportional to powers of L, hence it is possible to use such quantities, such as Binder cumulants or the rescaled correlation length $R_\xi = \xi/L$, to estimate the critical temperature and, at least, the critical exponent ν: non-rescaled data belonging to different sizes and temperatures will intersect at the critical point, modulo scaling corrections; the exponent ν can be determined by the scaling of data onto a single curve in the parameter $\varepsilon L^{1/\nu}$.

This is the technique we will generalize in the out-of-equilibrium relaxation setting.

1.2.2 Finite-Time Scaling

Now, we discuss the scaling in the out-of-equilibrium relaxation regime. We define the relevant scaling field for the time variable t as

$$u_t \simeq t^{-1}. \tag{1.57}$$

We can think of this scaling field as describing an anisotropic direction evolving along the RG flow near the critical point, $u_t = 0$, with an eigenvalue ℓ^{y_t} defined by some function of the coupling constants. Let us write the scaling expression for a generic observable A scaling as $L^{\lambda/\nu}$ and introducing the new scaling field at zero external field and limiting the dependence to the first irrelevant operator

$$\begin{aligned} &A(\varepsilon, L^{-1}, t^{-1}, u_{i_1}) \\ &= L^{\lambda/\nu} G_{A,L}(\varepsilon L^{1/\nu}, tL^{-y_t}) \left[1 + \frac{u_{i_1}}{L^{\omega_1}} g_{1A,L}(\varepsilon L^{1/\nu}, tL^{-y_t})\right] + O\left(L^{-2\omega_1}\right). \end{aligned} \tag{1.58}$$

Usually one defines $y_t = z$ as the **dynamic exponent** which satisfies $z > 0$. It is possible to write new scaling functions in order to obtain a time-scaled relation as

$$\begin{aligned} &A(\varepsilon, L^{-1}, t^{-1}, u_{i_1}) \\ &= L^{\lambda/\nu} \left(t^{-1/z} L\right)^{-\lambda} G_{A,t}(\varepsilon L^{1/\nu}, tL^{-z}) \\ &\times \left[1 + \frac{u_{i_1}}{L^{\omega_1}} \left(t^{-1/z} L\right)^{\omega_1} g_{1A,t}(\varepsilon L^{1/\nu}, tL^{-z})\right] + O\left(L^{-2\omega_1}\right) \\ &= t^{\lambda/z} G_{A,t}(\varepsilon L^{1/\nu}, tL^{-z}) \left[1 + \frac{u_{i_1}}{t^{\omega_1/z}} g_{1A,t}(\varepsilon L^{1/\nu}, tL^{-z})\right] + O\left(L^{-2\omega_1}\right). \end{aligned} \tag{1.59}$$

Once again it clearly appears that the equilibrium limit is obtained when $t^{-1/z} L \ll 1$. This relation gives a qualitative explanation for the divergence of the equilibration time τ as a function of the correlation length near the critical point

$$\tau \sim \xi^z, \tag{1.60}$$

which is usually called **critical slowing down**. This behaviour is what hampers the MC simulation approach to equilibrium at the critical temperature. The dynamic

evolution of the system should behave as in the thermodynamic limit as long as $a \ll \xi(t) \ll L$, where a is the lattice spacing, or equivalently for $tL^{-z} \ll 1$. This is the basis of the so-called Non-Equilibrium Relaxation (NER) method [23].

Systems sharing the same values of the static critical exponents, *i.e.* belonging to the same static universality class, may belong to different **dynamic universality classes** characterized by different values of z. In the famous work by Hohenberg and Halperin [24] the latter are classified according to symmetry criteria.

In this work we will deal with the purely relaxational dynamics, or type-A dynamics, resulting from the Metropolis algorithm which does not conserve the order parameter.

1.3 Spin-Glasses

Spin-glasses (SGs) represent one of the most attractive test grounds for an out-of-equilibrium method for the estimation of critical parameters. The reason is easily explained: for the three-dimensional Edwards-Anderson model with bimodal disorder (EA3D) the type-A dynamic exponent is $z = 6.86(16)$ [25], while for the three-dimensional Ising model $z \sim 2$. This is the source of most difficulties in obtaining equilibrium estimates for large systems. Such a problem has often been faced through the construction of dedicated supercomputers [26, 27, 28, 29] and the introduction of a new kind of dynamics, *i.e.* the Parallel Tempering (PT) dynamics [30], which has slightly reduced the problem. However, still today the largest linear size for which a reasonable number of disorder samples has been equilibrated is $L = 40$ [31].

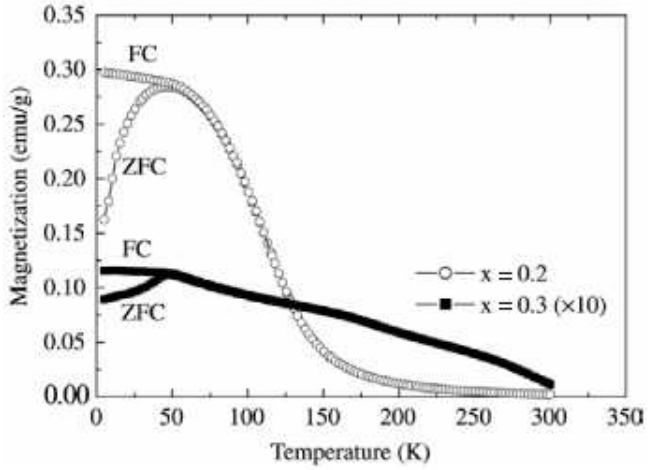
Let us now briefly review the main concepts and characteristics of SGs.

1.3.1 Phenomenology

SGs are disordered metallic alloys where the local magnetic dipoles interact with each other via ferromagnetic and antiferromagnetic exchange interactions. This fact introduces, alongside with disorder, the so-called **frustration**: the ground-state of the system is generally not one where all dipoles minimize their exchange interactions.

The physical behaviour of such alloys is qualitatively the following: there exists a transition temperature T_c above which the system behaves as a paramagnet, *i.e.* the magnetization is proportional to the external magnetic field strength and inversely proportional to the absolute temperature, according to Curie's law; below the critical temperature, if one switches off the external field the magnetization follows a two-step decay, first to a plateau whose value is called **thermoremnant magnetization**, then towards zero. The low-temperature behaviour is due to a complex free energy landscape below T_c allowing the existence of aging effects similar to those measured in structural glasses.

Cooling the system below the critical temperature in the presence of a small external magnetic field, h_{FC}, leads to a slowly varying value of the magnetization as a function of the temperature. This is the so-called **field-cooled magnetization** m_{FC}. The first evidence for an aging behaviour [32] was obtained using the so-called **zero-field-cooled magnetization** m_{ZFC}, which is measured after cooling the system below the transition temperature and then applying an external magnetic



Temperature dependence of the magnetization of the system $La_{0.7}Sr_{0.3}Mn_{1-x}Fe_xO_3$ for x = 0.2 and 0.3 measured for ZFC and FC in a field of 1000 Oe.

Figure 1.1. Plot reprinted from reference [33].

field. One can compare the behaviour of m_{ZFC} to that of m_{FC} by applying the same magnetic field h_{FC} in both cases. Once the sample has been cooled at zero magnetic field, at first, immediately after the magnetic field has been applied, the magnetization takes a value m_{ZFC} that is lower than m_{FC}. Afterwards, as time goes on at a fixed magnetic field h_{FC}, the magnetization evolves towards the expected m_{FC} value. Moreover, the dynamics leading the magnetization from m_{ZFC} to m_{FC} depends on the waiting time t_w, which is time elapsed from the quench below the critical temperature to the application of the magnetic field [32], *i.e.* the derivative $\partial m(t_w + t)/\partial t$, with $t > 0$, depends on t_w. This behaviour signals the presence of an aging regime. This means that memory effects are present and that the system has not reached an equilibrium state on the probed timescales: this is one of the most important characteristics of spin-glasses.

In Fig. 1.1 we show typical results for m_{ZFC} and m_{FC} as reported in [33] where measures are taken at a given temperature and for a constant magnetic field. The different curves are related to different dopings of the considered metallic alloy.

1.3.2 Mean-field models

The Sherrington-Kirkpatrick (SK) model [34] is a mean-field model for SGs and it is defined by the following Hamiltonian

$$H = -\sum_{i<k} J_{ik}\, \sigma_i \sigma_k, \tag{1.61}$$

where the coupling constants $\{J_{ik}\}$ are randomly drawn according to a Gaussian probability distribution of mean J_0 and variance J

$$P(J_{ik}, J_0, J) = \frac{1}{\sqrt{2\pi J^2}} \exp\left[-\frac{(J_{ik} - J_0)^2}{2J^2}\right] \tag{1.62}$$

and the spins variables take on the integer values $\sigma_i \in \{-1, +1\}$.

The equilibrium solution for the minimization of the free-energy density has been found in 1979 by G. Parisi [35, 36] through the so-called **replica trick**. Replicas are different systems sharing the same disorder realization and are involved in the calculation of the quenched average of the free-energy density

$$\begin{aligned} f(N,\beta) = [f(N,\beta,\{J_{ik}\})] &= \sum_{\{J_{ik}\}} P(J_{ik},J_0,J)\, f(N,\beta,\{J_{ik}\}) \\ &= -\frac{1}{N\beta} \sum_{\{J_{ik}\}} P(J_{ik},J_0,J)\, \log Z_N(\beta,\{J_{ik}\}) \\ &= -\frac{1}{N\beta} \lim_{n\to 0} \sum_{\{J_{ik}\}} P(J_{ik},J_0,J)\, \frac{Z_N^n - 1}{n} \end{aligned} \tag{1.63}$$

where n denotes the number of replicas and the squared brackets the disorder average. This strategy for the computation of the free energy is motivated by the fact that the disorder average of a power of the partition sum is easier to evaluate with respect to the average of the logarithm. The order parameter of this system is the so-called **overlap** matrix $Q_{\alpha\beta}$, which is defined as

$$Q^{ab} = \frac{1}{N} \sum_{i=1}^{N} \sigma_i^a \sigma_i^b, \quad \Rightarrow \quad -1 \leq Q^{ab} \leq 1, \tag{1.64}$$

where σ_i^a is the i-th spin belonging to the replica a. In the limit $n \to 0$ one needs to give a different parametrization of the overlap matrix which is substituted by the one parameter overlap function $q = q(x)$. Since we average over the disorder the order parameter has its own probability distribution $P(q)$. A continuous phase transition is signaled by a discontinuous change in the $P(q)$: above the critical temperature $P(q)$ is a delta function centered at $q = 0$, while below the transition the $q = 0$ peak vanishes and two new symmetric peaks develop linked by a continuous function.

In the paramagnetic phase the solution is invariant with respect to replicas permutations, while in the low temperature phase this symmetry is broken signaling the existence of a complex free energy landscape where the thermodynamic states are arranged in a binary-tree or ultrametric fashion. This symmetry breaking is called full Replica Symmetry Breaking (fRSB).

One of the most important physical properties of the fRSB phase is that it is still present even when an external magnetic field is applied, thus defining a line of critical temperatures, $T_c = T_c(H)$ which is known as the deAlmeida-Thouless (dAT) line. This is one of the most striking differences with respect to the mean-field Ising model.

1.3.3 Finite-dimensional models

The picture of finite-dimensional spin-glass models is far less clear than the mean-field one. The three-dimensional Edwards-Anderson model (EA3D) [37] is defined by the Hamiltonian

$$H = -\sum_{\langle ik \rangle} J_{ik}\, \sigma_i \sigma_k, \quad \sigma_i \in \{-1, +1\}, \tag{1.65}$$

where this time the sum $\sum_{\langle ik \rangle}$ is on nearest neighbours only in a cubic lattice discretization. We studied the bimodal disorder distribution which is defined as

$$P(J_{ik}) = \frac{1}{2}\left\{\delta_K(J_{ik}-1) + \delta_K(J_{ik}+1)\right\} \tag{1.66}$$

where $\delta_K(a-b) = \delta_{ab}$ is the Kronecker delta, so that $J_{ik} \in \{-1, +1\}$. Now, the very existence of a continuous phase transition in the two real replicas overlap as order parameter has been the subject of intense studies for almost 30 years. One of the first estimates for the critical temperature was given in 1985 by R.N. Bhatt and A.P. Young in [38] using the Binder cumulants crossing technique, while the latest measure was obtained, with the same method, by the Janus collaboration [31] in 2013, using the dedicated Supercomputer Janus which simulated a number of disorder realization orders of magnitudes higher than that of [38]. In between, many works on the subject have appeared using both equilibrium or out-of-equilibrium techniques. Interestingly, many different estimates of critical parameters have been published and for the critical temperature a systematic shift as a function of the largest equilibrated system size is clearly visible. This drift is surely due to the presence of large finite-size corrections to scaling [39].

As for the nature of the low temperature phase the quest for a clarifying answer is still open with two major players on the scene: the fRSB and the droplet [40, 41, 42] scenarios. The droplet picture is exact for some types of hierarchical lattices [43] where the Kadanoff approach we described above can be exactly applied. According to the droplet approach the low-temperature phase is characterized by the presence of only two states which are related by the global spin-flip symmetry thus leading to an overlap distribution characterized by two delta functions in contrast with the fRSB mean-field solution describing an infinite number of states. The system relaxes through droplet excitations just as a ferromagnet with an applied magnetic field would do. The thermodynamic state is a mixture of the these two pure states. In the fRSB description states would be continuously linked thus allowing to transform the entire system from one state to another. The main assumptions of the droplet description are the following: given a droplet excitation of characteristic length ξ the excitation free energy F_ξ, the activation energy A_ξ and the probability distribution $P(F_\xi)$ scale as

$$F_\xi \sim \xi^\theta, \quad A_\xi \sim \xi^\psi, \quad P(F_\xi) \sim \frac{1}{\Upsilon \xi^\theta}\rho(F_\xi/\Upsilon\xi^\theta), \tag{1.67}$$

where $0 < \psi < \theta$, Υ is a dimensional constant linked to the interfacial tension and the function ρ satisfies the property $\rho(0) > 0$.

So far, no definitive evidence has been gathered validating one of the two pictures. Many technical difficulties for the fRSB mean-field interpretation come from the high value of the estimated lower critical dimension $d_L = 6$. However, the droplet and the fRSB descriptions are much different. As an example, the droplet scenario differs from the fRSB picture for the action of the external magnetic field which would destroy the transition, *i.e.* no dAT line is expected in the droplet picture. Much effort has been spent in simulations in order to check for the presence of a dAT line in the three-dimensional Edwards-Anderson model. From the most recent

results [44] it appears that usual finite-size scaling techniques might not be the best tool to locate the transition temperature.

The new out-of-equilibrium technique we are about to describe in this thesis, which is generally applicable to any second-order phase transition, might turn out to be a valuable tool also with respect to the problem of the low temperature phase of finite-dimensional spin-glasses.

Chapter 2

GPUs and High Performance Computing

In this chapter we give a brief introduction to the main concepts involved in the Graphic Processing Unit (GPU) programming for scientific applications. We begin giving a rapid historical overview on how and why GPU entered the scene of High Performance Computing (HPC). We then continue describing one of the available frameworks for programming GPUs: the Compute Unified Device Architecture (CUDA) developed and released by nVidia Corporation. We summarize the most important new concepts of CUDA which are usually not present in single CPU programming. In order to define the GPU programming limitations we give a summary of the programmer-exposed hardware architecture whose knowledge is necessary in order to optimize algorithm performances. We describe the two most important resources: the computing cores and the memory hierarchy. Finally, we discuss some lessons we learned with the programming practice. Most of the contents can be found in [45].

2.1 Historical Introduction

The name 'Graphic Processing Units' is rather recent but the use of coprocessors for handling graphical tasks has a long history which originates in 1970s video games. In these systems fast calculations for the graphic output are necessary for enabling a real-time interaction between the machine and the user. Later on, graphic coprocessors were produced for accelerating industrial drawing applications. In a first stage only two-dimensional images could be processed and three-dimensional projections capabilities were eventually introduced. Since the basic task is to compute the colour of each pixel of the screen independently, the hardware naturally evolved towards a **many-core** architecture where different cores with limited capabilities could be used in parallel to accelerate the calculations.

Just ten years ago these coprocessors could be programmed through the OpenGL Application Programming Interface (API). However, the programmer was forced to use functions and pipelines which were designed for graphical purposes but nonetheless there have been some successful attempts of using GPUs for Monte Carlo (MC) Ising spin model simulations [46].

A major leap forward has finally been taken in 2007 by nVidia when the new Tesla architecture and the first version of CUDA were released: it was possible to use GPUs for general purpose computing using a limited set of new instructions in the C/C++ language. This step allowed the scientific community to access, in a much easier way, the GPU computational resources. Indeed, many scientific problems can be solved using algorithms that naturally expose an intrinsic parallelism. These have greatly benefitted from the use of GPUs: considerable speed-ups, with respect to usual CPU implementations, have been obtained in several applications with a limited effort in the code porting. Most recent GPUs can remarkably sustain some TFlops for single-precision computations. However, naive code porting still does not allow to fully use the GPU capabilities. As we will describe in the Chapter 3 best GPU performances must be sought with great care.

Finally, it is worth noticing that CUDA is not the only framework available for programming on GPUs, nor nVidia is the only industry producing programmable cards. There exist an open-source programming environment called OpenCL which can execute programs on nVidia GPUs as well as on ATI graphic cards. However, we won't describe OpenCL. Nonetheless, our implementation strategies are completely general and can be used in OpenCL.

2.2 CUDA programming model

Let us now review the basic programming concepts one has to handle in order to successfully program on a GPU.

2.2.1 Threads

Through some programming language extensions it is possible to define special functions, called **kernel**, to be run directly on the GPU. A given instance of the kernel is handled by one **thread**. Threads are organized in a two-level hierarchy.

- **thread blocks**: a block is a set of N_b threads which can be organized in a cubic lattice topology where each thread is identified by the triple of built-in variables `threadId.x, threadId.y` and `threadId.z`. These variables are read-only and can be accessed only within a kernel function. Different block directions might have different limitations: the maximum number of threads along the x or y directions, $N_{b,x}^{max}$ or $N_{b,y}^{max}$, up to the latest architecture, differs from that of the z direction $N_{b,z}^{max} \neq N_{b,x}^{max}, N_{b,y}^{max}$. Moreover, the maximum number of threads in a single block N_b^{max} is different from the product of the maximum number of threads for each direction: *i.e.*

$$N_b^{max} \neq N_{b,x}^{max} \times N_{b,y}^{max} \times N_{b,z}^{max}, \tag{2.1}$$

  so that one has to pay attention when defining the block size. The only case in which different threads can communicate, without imposing a serialization of the execution, is when they belong to the same block. Data can be exchanged via a programmable cache which is called **shared memory** or via specific instructions which are available for the latest architecture (*i.e.*, warp shuffle instructions).

- **block grid**: the grid is a set of thread blocks which can be organized in a cubic lattice topology where each block is identified by the triple of built-in variables `blockIdx.x`, `blockIdx.y` and `blockIdx.z`. As we will discuss below, the execution order of different blocks is not deterministic and it may change at every kernel invocation. Also for the grid dimensions different limits in the number of blocks apply according to the direction, so that the maximum number of blocks on the x or y directions $N_{g,x}^{max}$ or $N_{g,y}^{max}$ differs from that along the z direction $N_{g,z}^{max}$.

The CUDA programming model is called SIMT (single instruction multiple thread) and it is similar to the SIMD (single instruction multiple data) model. The simplest case is that in which each thread executes the same set of instructions on different data. The set of possible instructions which can be executed by a single thread is almost the same as the usual CPU one with some differences: some double-precision floating point operations do not comply with the IEEE-754 standard and the possibility of recursion is available only for the latest cards and CUDA versions.

In order to launch a kernel function one has to specify the grid configuration, *i.e.*, in the one-dimensional case for both blocks and grid, the number of threads per block and the total number of blocks. Another important property of kernel functions is that they are **asynchronous**: when a host thread launches a kernel on the device the control of the flow immediately return to the host, so that it is possible to overlap the execution of a kernel on the device to some different task executed on the CPU. Of course, it is also possible for the host to wait for the kernel execution to terminate.

Another interesting feature we will mention again in Chapter 3 are the so-called CUDA streams using which it is possible to launch concurrently on the same device more than one kernel. Assigning different tasks to different streams it is possible to organize the device workload in such a way that a direct interaction with the host is possible while keeping the device busy. We will leverage this feature in order to implement the multi-GPU version of our code. Data transfer between the host and the device can be performed both synchronously and asynchronously.

Now that we have a global view of the thread structure we can exemplify the typical workflow as follows:

1. data structures are initialized on the host and copied on the device;
2. kernels are executed on the device;
3. the result of the execution is copied from the device to the host.

When writing a kernel the general strategy is to map the global thread index[1] to some array index so that each thread can load its own data onto which perform the programmed calculations. Reading from array entries which are written to by other threads generates a non deterministic behaviour since the thread execution order is determined at runtime.

[1]The global kernel index for one dimensional blocks and grid reads `thIndex = threadIdx.x + blockIdx.x*blockDim.x`.

2.3 GPU architecture

Let us now discuss some essential details of the GPU architecture which are shared by all GPUs. The threads execution we discussed in the previous section has a rather complex mapping onto the actual hardware of the GPU. We present such a mapping describing the two principal elements of a GPU, the streaming multiprocessors and the memory hierarchy.

2.3.1 Streaming Multiprocessors

Each nVidia GPU, among other parts, is composed by one or more Streaming Multiprocessors (SMs) which is the part of the card where the CUDA cores reside. The number of cores N_c per SM varies with the hardware architecture, *e.g.*, for the Fermi architecture $N_c = 32$ while for the Kepler architecture $N_c = 192$. Each SM has also two different caches: the L1/shared memory and the texture cache. The number of available registers per SM, R_{SM}, on which calculations are executed also varies with the hardware architecture.

When a thread grid is instantiated each block is assigned to a given SM. Each SM has a scheduler which divides the threads into groups of 32 which are called **warps**: if the number of threads in a block is not a multiple of the warp the number of instantiated warps is rounded to the next greater multiple directly by the hardware. The warp is the basic execution unit of the SM. Each thread of a warp has its own private registers, hence all the threads of a block have their own private registers. Now, each thread of a warp executes the instructions on a given core which might execute two different instructions at once, for certain operations, if the results are independent. If the instructions present a data-dependent **branching**, *i.e.*, an `if` statement, the scheduler serializes the execution of the warp branches: this means a great loss in performances. In the Chapter 3 we will see how to implement the Metropolis MC dynamics without using `if` statements in order to avoid warp branchings.

The SM has mainly two limitations which has to be satisfied at the same time:

1. there is a maximum number of thread blocks which can be run concurrently B_{max};

2. there is a maximum number of threads that can be instantiated for each clock cycle T_{max}, hence a maximum number of warps W_{max}.

Each kernel requires a certain number of registers R_{th} in order to be executed by a thread, so that the total amount of registers that are needed by a one-dimensional block is $R_{bl} = R_{th}N_{b,x}$. If the block size is such that the required register are more than those physically present on the SM, *i.e.*, $R_{bl} > R_{SM}$, the kernel launch fails since the private state of a block has to be entirely available on the SM registers at any time. Another scenario might be one where just two blocks can be instantiated at once per SM: this situation might give some trouble to the scheduler which could optimize the performances shuffling the execution of more than two blocks on the same SM. An opposite example might be that of a kernel using just a few registers so that choosing small block sizes would lead to reach the maximum SM block occupancy while underusing the available registers.

These examples show that the block dimension is a crucial control parameter for tuning the performances: there exist an optimal grid launch configuration which may vary from GPU to GPU for the same data set.

It is possible to have a one-to-one mapping of the CUDA cores to the number of single-precision floating-point units while this mapping is not possible for integer or bitwise units which are less than CUDA cores. CUDA cores are also in charge of the calculations of the physical memory addresses for memory loads and writes.

Each SM has a separated texture unit which is in charge of loading the read-only memory regions, *i.e.*, execute **texture fetches**, which are bound to a texture. The texture memory physical addresses are not computed by CUDA cores.

At this point it is worth introducing a difference in the production of nVidia GPU cards: a given card is tested during the production phase and according to the results it is sold as a certified scientific-applications card, belonging to the Tesla series, or as a gaming card, belonging to the GTX series[2]. Beside the certification for intense computing usage of the Tesla cards, there are two important difference between the scientific-application and the gaming cards: Tesla cards have the so-called Error-correction code (ECC) which allows to determine whether there has been a data corruption both in storage data and in the computation of physical addresses for memory loads and writes; the GTX cards have a limited double-precision capability (with the exception of the GTX Titan so far) which is completely enabled on Tesla cards.

2.3.2 Memory hierarchy and caches

All data allocated in the GPU are stored in the Random Access Memory (RAM) and are divided in three different categories:

- Global Memory data, which can be read or written by any thread;
- Constant Memory data, which are read-only;
- Texture Memory data, which are read-only.

All memory transactions from the RAM are served through an L2 cache whose size is GPU-dependent. Now, this distinctions mainly relates to a difference in the use of these memory regions:

- Global Memory is read and written through the SM L1/shared memory caches and each transaction requested by a thread is served from the L2 cache in lines of 128 Bytes. This means that, if threads in a warp request non-contiguous Global Memory regions, more than one transaction, in the worst case 32, are needed to serve the warp requests. When the memory request refer to contiguous data, transactions are said to be **coalesced** and the bandwidth is used in the most efficient way. An example is when all threads in a warp request 32 contiguous integer values. Having a non-coalesced access-pattern may turn into a sizeable loss in efficiency.

[2]Of course this is only a rough explanation of the process.

- Constant Memory is served through a dedicate SM cache which streams the requested values to all threads in the instantiated block. Hence, coalescing is not requested.

- Texture Memory is served through another SM cache and transactions are optimized on a two-dimensional data locality criterion. When the coalescing of data cannot be achieved in the case of read-only data in Global Memory, the use of texture fetches, *i.e.* texture memory reads, might give a sizeable gain.

We want to stress that arrays which belong to Texture Memory can still be modified on the device if they are used as normal arrays. Any array can be read through the texture cache when it is bound to a texture. On the contrary constant memory data can only be modified by the host. As we will see in Chapter 3 we will alternatively access to the spins as if they belong to the Global or to the Texture Memories.

Hence, the memory hierarchy is the following. At the lowest level there is the RAM which is alternatively used as Global, Constant or Texture Memory. Each request is served through the common L2 cache on the GPU level, and afterwards through the L1/shared memory, constant or texture caches respectively, all belonging to a single SM. At the highest level data are finally manipulated on the SM registers and eventually stored back in the RAM, through the L2 cache.

Finally, one has always to remember that the memory requests are served with a large latency, which is expected to be overlap with the algebraic computations performed on the SM registers.

2.4 What practice taught us

So far, we gave a brief account of the basic features of GPUs which are, nonetheless, important in order to obtain good performances. Indeed, when optimizing the code, the only way to determine whether a given modification leads to an improvement or not lies in a direct test: this is due to the fact that most of the workload management is done directly by the SM schedulers. Moreover, some implementations may be the best choice for one GPU and not for another one. Hence, it seems reasonable to develop different implementations for the same problem in order to be able to choose the best one according to the available hardware.

We want now to give some guidelines we used in the implementations we will discuss in detail in Chapter 3 which we have learned from practice. Generally speaking, algorithms can be classified as memory- or compute-bound depending on how many algebraic operations per memory transaction are executed. In the first case it is possible to leverage the very high memory bandwidth of the GPU, while in the second case its computational capabilities. As an example the Metropolis algorithm for the three-dimensional Edwards-Anderson model is memory-bound: we need to perform 13 reads and 1 write to update one spin, while using only integer or bitwise operations. We now present a general strategy which has turned to be useful in such a case.

2.4.1 Data layout in Global Memory

For memory-bound algorithms it is of utmost importance to optimize the data layout in memory in order to suite in the best way possible the memory-access requirements we discussed above. We list here some of the possible strategies.

- If the data structure is bipartite, *e.g.* square or cubic lattices of even size, it is often more convenient to allocate two different arrays whereas in the usual CPU case one single array is allocated.
- Whenever an array is only read during a kernel execution it is useful to bind it to a texture in order to delegate to the dedicated texture hardware, rather than to CUDA cores, the computations of the physical memory addresses. This choice might also alleviate problems arising from non-coalesced data loads.
- Another strategy might involve a different ordering of the data structure in order to alleviate the non-coalescence of memory loads because of some boundary condition. A different data order might be useful to obtain a balanced load of threads avoiding repeated thread divergences in different warps.
- Considering an array of three-dimensional coordinates data structures it is better to allocate three different arrays so that threads can access them sequentially.
- We will show in Chapter 3 that for lagged-Fibonacci-like pseudo-random number generators it is better to access the state array in a strided way.

All these solutions share a common feature: to minimize the number of data transactions needed for a warp to be ready to elaborate the loaded data. Hence, one should look at usual data structure differently in order to find a new, more suitable, layout.

2.4.2 Calligraphy

Surprisingly, most of these solutions are implemented at the kernel level just by tiny modifications of the code. Under this point of view it seems that GPU optimizations are a matter of *calligraphy* in writing the code. We will see a practical example in the next chapter. This fact can be justified by the complexity of the GPU behaviour which depends on a large number of different factors.

Chapter 3

Edwards-Anderson model on GPU

In this chapter we present a highly optimized implementation of a Monte Carlo (MC) simulation for the three-dimensional Edwards-Anderson model with bimodal disorder, *i.e.* the 3D Edwards-Anderson model running on CUDA enabled GPUs. We begin with a review of the literature according to different criteria: the memory allocation strategies, the nearest-neighbours access pattern, the chosen pseudo-random number generator (PRNG) and the multi-GPU strategy. We then describe a new memory access pattern for nearest neighbours on a cubic lattice which gives a better memory alignment compared to previous implementations together with new implementations for lagged-Fibonacci-like PRNGs, *e.g.* the Parisi-Rapuano and the well-known Mersenne-Twister MT19937. Performance results will be given concerning various metrics.

3.1 Introduction

Let us begin by writing once more the Hamiltonian of the three-dimensional Edwards-Anderson model (EA3D)

$$H = -\sum_{\langle ik \rangle} J_{ik}\, \sigma_i \sigma_k, \tag{3.1}$$

where the $\sigma_i \in \{-1, +1\}$ are the spin variables, the $J_{ik} \in \{-1, +1\}$ are the coupling constants (representing *quenched* variables) which are randomly drawn according to a given probability distribution $P(J_{ik})$, and the sum $\sum_{\langle ik \rangle}$ is restricted to nearest neighbouring spins. Since we deal with bimodal disorder the probability distribution reads

$$P(J_{ik}) = \frac{1}{2}\left[\delta_K(J_{ik} - 1) + \delta_K(J_{ik} + 1)\right], \tag{3.2}$$

where $\delta_K(a - b) = \delta_{ab}$ stands for the Kroneker delta. Such a model describes, in three dimensions, a disordered and frustrated magnetic system showing a glassy dynamics below a finite critical temperature $T_c = 1.1019(29)$ [31]. Because of a very high critical exponent $z = 6.86(16)$ [25] for the MC Metropolis dynamics, this model has represented a long standing challenge for numerical simulations. For almost thirty years special-purpose machines have been employed [26, 27, 28, 29] in order

to get equilibrium measures for always larger systems. Large systems are needed because of the severe finite-size corrections to scaling [39, 31]. Moreover, new kinds of dynamics have been developed in order to reach equilibrium as fast as possible: Parallel Tempering has proved to be the best choice so far [30].

However, the hegemony of special purpose hardware for this class of problems might be about to end because of the widespread use of (multi) GPU devices, enabling us to reach computational horsepower exceeding by orders of magnitude those of common CPUs.

Since the very early use of CUDA, and even before [46], it has been understood that MC simulations of Ising spin systems would have enjoyed benefits from the use of GPUs. This is not surprising: for even cubic lattice sizes $L = 2n$ the system can be simply partitioned according to a checkerboard scheme into two coloured subsystems, which we will refer to as *reds* and *blues*, which can be updated separately since nearest neighbours of one colour belong to the other colour, *i.e.*, a red spin has blue nearest neighbours only. Hence, the problem has an intrinsic parallelism which perfectly suits the GPU architecture: the update of each spin of a given color does not require any coordination with the update process of other spins of the same color so that the large amount of computing threads needed for the best use of the GPU can be programmed to update concurrently independent spins of the system.

Using the Metropolis dynamics, the update of the entire system is performed via two separate kernels, one for each colour. This is an easy way to enforce the independence of the update of the two subsets, a necessary condition for a correct implementation of the Markov chain. Of course, this kind of update does not ensure *detailed balance* but *stationarity* holds nonetheless, which is enough to guarantee that the system probability distribution converges to the equilibrium Gibbs distribution [47, 48]

$$P(\{\sigma_i\}, \beta) = \frac{e^{-\beta H[\{\sigma_i\}]}}{\sum_{\{\sigma_i\}} e^{-\beta H[\{\sigma_i\}]}}. \tag{3.3}$$

Now, we will review the previous works on spin systems for GPUs, taking into account also different models and dimensionalities. We will analyze them according to:

- memory-allocation strategies and spins-thread mappings;
- the kind and implementation of PRNGs;
- multi-GPU implementation techniques.

We can classify the previous works on spin systems according to the allocation and the memory access strategies starting from the lowest level (Global Memory) up to the highest level (Shared Memory and registers) of the memory hierarchy:

1. **Global Memory allocation**. Two different strategies are mainly used for the memory allocation of spins in the GPU Global Memory: a first one uses a mixed scheme where one array is allocated containing both colours in a cubic lattice topology [49, 50, 51, 52, 53, 54]; a second one allocates two separate buffers for the two colours breaking the cubic lattice topology [46, 55, 56, 57, 58].

Though the first strategy seems to be more natural, one has to take into account that memory transactions are served from the L2 cache in blocks of 128 bytes so that for each transaction one loads both the to-be-changed spins and the nearest neighbours which remain unchanged during the kernel execution. Different threads will update different spins sharing some neighbours, thus rendering the access pattern highly non-trivial. As a matter of fact, many of the works adopting the first strategy use the Shared Memory to improve the locality of such mixed access to the coloured spins [50, 51, 54, 53].

In the second case there are several benefits but also drawbacks. It is possible to achieve good memory loading performances, since many threads look for the same neighbouring spin allowing for a higher second hit probability in L1/L2 caches. However, one has to deal with an algebraically demanding access pattern due to the loss of the cubic topology: it is necessary to take into account the *parity* of the lattice site in order to correctly determine the right and left neighbours. Usually this strategy does not require the use of Shared Memory as reported in [55, 56, 57, 58].

Other considerations are in order. Being the two colours allocated in two different arrays, it is possible to bind each buffer to a texture in order to load the neighbouring spins through the dedicated texture unit of each Streaming Multi-processor (SM) with a separated cache different from the L1/Shared. This choice offers a two-fold advantage: on one side, the slowing down due to occasional non-coalescence of loads for nearest neighbours is softened because texture fetches work on a memory locality principle, on the other side the dedicated texture hardware is in charge of the computation of physical memory addresses rather than CUDA cores. Using texture fetches for nearest neighbours and couplings gives a sizeable gain of the order of 10-20%.

2. **Spins arrangement in Global Memory**. For both allocation strategies it is still possible to choose several spins arrangements. Such a choice aims at maximizing the loading efficiency from the Global Memory, *i.e.*, reducing the number of non-coalesced loads.

   In case of a single buffer for both colours, two different strategies have been proposed in [53] and [54]. In [53] the authors divide the cubic lattice in sublattices linearly organized in memory in a "snake" fashion so that every block taking charge of one sublattice could load more efficiently the spins to the Shared Memory. As for [54], the authors studied a so-called *shuffled* scheme where spins coming from different replicas where mixed saving memory transactions. However they found that such a strategy performs worse than the so-called *unified* one, where each array contains spins belonging to one replica.

   In the case of a separate allocation of colours, the authors of [55] proposed, for the two-dimensional Potts model, a coordinate transformation of the lattice that leads to have three of the four nearest neighbours lying sequentially in

the array. The coordinate transformation reads

$$\begin{cases} x' = & (((x+y) \bmod 2)L + x)/2 \\ y' = & y \end{cases}$$

3. **Spins-threads correspondence**. The mapping between spins and threads can also be done in different ways.

   Some of the works adopting the unified allocation scheme resort to (per-block) Shared Memory [50, 51, 54]. In one of the most commonly used mappings each thread-block evaluates the MC move on a sublattice [50, 51, 52, 53, 54] which usually is a square in two dimensions or a cubic sublattice in three. Clearly, with this strategy one needs to look for neighbours in the boundaries which will also be retrieved by neighbouring blocks thus leading to a duplication of data served by memory transactions. The most frequently used technique, in this case, is loading in Shared Memory the neighbours and, if needed, the couplings. However, this might represent a serious limitation since the number of thread blocks running on a single Streaming Multiprocessor (SM) depends on the amount of Shared Memory needed by each of them. In the case of models with complex degrees of freedom only few blocks can be run concurrently thus leading to under-use the SM.

   A rather different, but apparently less effective, approach has been tried in [49] where thread-blocks were associated to stripes of the cubic lattice of size $L \times 2 \times 2$. A thread-block is associated to each region and each thread updates 4 spins in the three-dimensional case.

   As for the separated scheme there are no particular restrictions on the dimensionality of the thread blocks which can also be taken as one-dimensional. Hence, no specific correspondence between blocks and lattice portions has to be considered resulting in a more tunable and flexible scheme [56, 57, 58]. Indeed, such a choice allows to decrease memory transfers redundancy. Moreover, as shown in [56] and verified in the present work, it turns out that nearest neighbours values are loaded in a more efficient way directly from global memory making use of texture fetches, *i.e.* texture cache and hardware.

We continue our classification considering the choice of the PRNG which is one of the most important aspects of a MC simulation. The reliability of the estimates depends on the quality of the sequence generated by the chosen PRNG. As an example, it is well known that the use of one single Linear Congruential PRNG with a period $p = 2^k$ in equilibrium MC for the 2D Ising model leads to systematic discrepancies on lattice sizes $L = 2^\ell$ due to resonance phenomena between the size of the system and the systematic long range correlations which affect Congruential PRNGs. It is then important to provide fast implementations for reliable PRNGs. Nonetheless, many of the previous works on MC simulations of spin systems on GPU used such PRNGs [46, 49, 51, 52, 56, 57, 58], mainly for benchmarking reasons. The principal motivation is that the state of these PRNGs is limited to one integer value making them the best choice in terms of speed but a questionable choice as for the quality of the produced numbers. However, it is interesting to notice

that in [51] compatible results with theoretical predictions for the two-dimensional Ising model were reported. In the most straightforward implementation of Linear Congruential PRNGs, each thread accesses its own memory location storing the one-valued state of the generator so that one deals with a battery of generators rather than a single one used to update the entire lattice. Little is known about the behaviour of such parallelized implementations of Linear Congruential PRNGs and it would be interesting to study them carefully.

In [55, 53] the authors used the so-called multiply-with-carry PRNG which consists in a modified Linear Congruential PRNG where the result of the modulus operation is used for the successive update, hence needing two integers to store the state.

Otherwise in [54] the cuRand implementation of the XORWOW has been used. This PRNG consists in a XOR-Shift summed to a Weyl generator.

There are a few other works using the so-called lagged-Fibonacci PRNG [59, 60]: these generators use a state of a certain length from which two 'lagged' entries are read and combined, usually summed, giving the random number and updating at the same time the state. Usually, this kind of generators have very long periods, much longer with respect to Linear Congruential PRNGs. As we are mainly interested in the memory access scheme for the GPU implementation we can label as lagged-Fibonacci-like all those PRNGs sharing a scattered read pattern of the state. Since there are lags between the reads of the state, a certain amount of random numbers can be produced in parallel by different threads [61, 62] using Shared Memory to store the state which will be used in a thread block. One of the most popular generators of this kind is the Mersenne Twister MT19937 [63] which has a very long period $p = 2^{19937} - 1$. However, since its state needs at least 624 entries, a Shared Memory implementation would be too memory-consuming, thus strongly limiting the SM occupancy. Hence in [64] the authors chose to use the so-called Warp generator [61] which has been written along the same lines of MT19937.

To the best of our knowledge, there is only one work [60] using the so-called Parisi-Rapuano generator [65] on a GPU. This generator basically consists in a lagged-Fibonacci PRNG. It has been long known that the 32-bit version of the Parisi-Rapuano cannot be safely used for sequential updating [66] and in [62] the authors reported the failure of the Crush and Big Crush tests for that generator. Indeed, in [60] the authors used the Parisi-Rapuano summed to a 64-bits congruential generator as proposed in [66] and they showed that a battery of such combined generators passes the Marsaglia tests [67], paying some care in the initialization process. However, no details about the GPU implementation were reported. Indeed, in [62] it has been observed that such a PRNG is not well suited for a Shared Memory-based GPU implementation because of the lags values.

Finally, as for multi-GPU implementations we are only aware of [50, 58, 57] which show stron scaling. Other works present weak scaling [60, 54] although in [60] communication between different nodes is needed because of the adopted Parallel Tempering implementation. However, the requirement of strong scaling depends on the physical features of the simulated systems one is interested in.

Let us now discuss the solutions we found as for the implementation of optimized access patterns for nearest neighbours in the cubic stencil and for lagged-Fibonacci-like PRNGs.

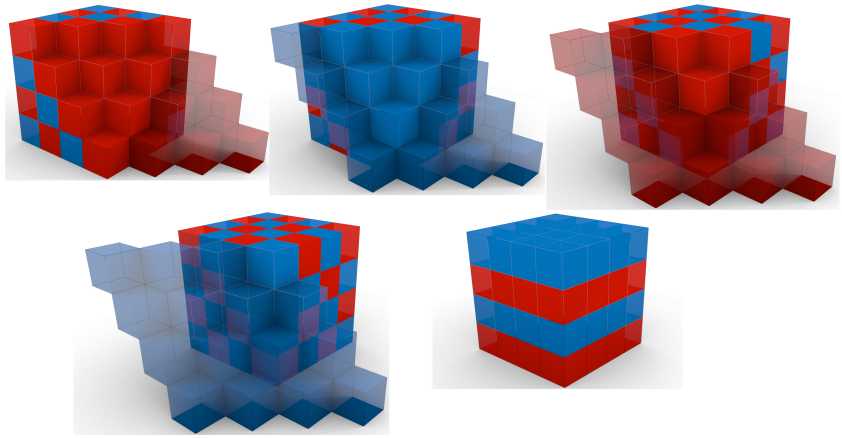


Figure 3.1. A depiction of the slicing procedure. Lighter cells are the periodic ones.

3.2 Cubic Stencil

With "Cubic Stencil" we refer to a set comprising one vertex of a cubic lattice together with its six nearest neighbouring vertices and edges. It represents the set of data needed to perform the update of a spin. It is also the fundamental data element of many other algorithms based on the cubic lattice discretization, *e.g.* for the solution of partial differential equations.

We now discuss the features of a new cubic stencil access pattern which we will refer to as **sliced**. We store red and blue spins in two separate arrays of Global Memory bound to two different textures. The novelty of the approach is in the spin arrangement. Here, we analyze the three-dimensional case, however the approach naturally extends to lower, *i.e.* two-dimensional, and higher dimensional cases. In three dimensions, under the assumption of having periodic boundary conditions, there exist a way of separating the colours while keeping the cubic lattice topology: let us consider the cubic lattice starting from the origin of a Cartesian reference frame where the coordinates take on integer values, hence $\vec{x} \in \mathbb{Z}^3$; vertices belonging to planes orthogonal to the direction $\vec{n} = (1, -1, 1)$ are all one-coloured. That is the reason why we call this scheme *sliced*. With periodic boundary conditions, vertices belonging to one slice will have all nearest neighbours either in the upper or in the lower slice. Such a slicing procedure is depicted in Fig. 3.1, whereas in Fig. 3.2 the transformation from the usual spin arrangement is shown. Hence, one starts from a three-dimensional checkerboard and ends up with a cubic lattice where each horizontal plane is one-coloured.

It is easy to write down the transformation and its inverse given that the vertices coordinates read $\vec{x} = (x, y, z)$ and the transformed coordinates read $\vec{x}' = (x', y', z')$

$$\begin{cases} x' = x \\ y' = y - x \\ z' = x - y + z \end{cases} \qquad \begin{cases} x = x' \\ y = y' + x' \\ z = z' + y' \end{cases} \tag{3.4}$$

recognizing in the equation for z' the expression of a plane orthogonal to the direction $\vec{n} = (1, -1, 1)$. The generalization to any number of dimensions for the vector $\vec{n}$ is

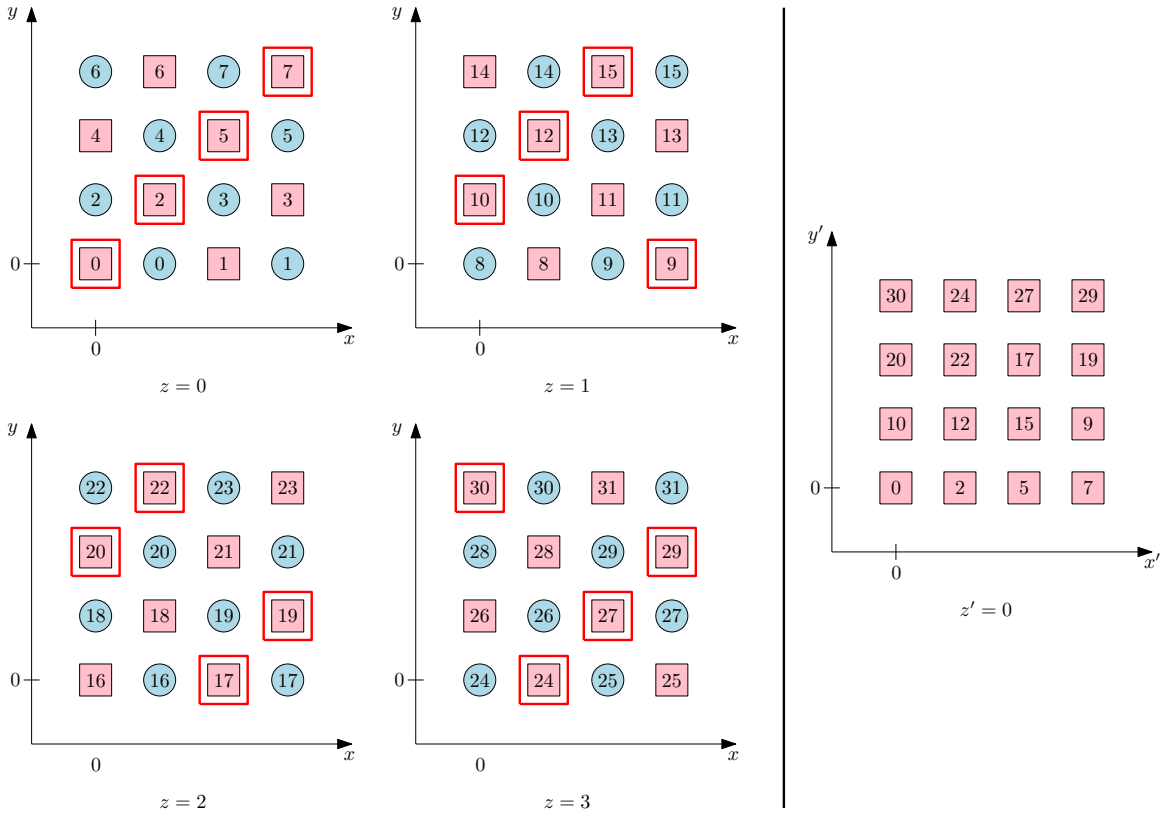


Figure 3.2. Mapping from the separated allocation checkerboard scheme to the sliced one for an $L = 4$ lattice. All four planes of the real lattice are involved in the definition of the first plane. The procedure is iterated starting from the $z = 1$ plane taking the blue diagonal: a blue slice is obtained for $z' = 1$ and so on.

simply given by $n_i = (-1)^{i+1}$ so that in two dimensions one-coloured vertices lie on lines orthogonal to $\vec{n} = (1, -1)$, and in four dimensions they lie on hyperplanes orthogonal to $\vec{n} = (1, -1, 1, -1)$ and so on. Hence, as long as one deals with regular (hyper)cubic lattices the scheme is completely general.

The transformed coordinates of nearest neighbours are:

$$\begin{aligned}
\vec{x}_{spz} &= (x, y, z+1) \quad \rightarrow \quad \vec{x'}_{spz} = (x', y', z'+1) \\
\vec{x}_{smy} &= (x, y-1, z) \quad \rightarrow \quad \vec{x'}_{smy} = (x', y'-1, z'+1) \\
\vec{x}_{spx} &= (x+1, y, z) \quad \rightarrow \quad \vec{x'}_{spx} = (x'+1, y'-1, z'+1) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (3.5) \\
\vec{x}_{smz} &= (x, y, z-1) \quad \rightarrow \quad \vec{x'}_{smz} = (x', y', z'-1) \\
\vec{x}_{spy} &= (x, y+1, z) \quad \rightarrow \quad \vec{x'}_{spy} = (x', y'+1, z'-1) \\
\vec{x}_{smx} &= (x-1, y, z) \quad \rightarrow \quad \vec{x'}_{smx} = (x'-1, y'+1, z'-1)
\end{aligned}$$

where the labels $spz, \ldots$ are self-explaining[1]. The order of the new coordinates

[1]The induced modification in the kernel is tiny: this is an example of the calligraphy concept we talked about at the end of Chapter 2

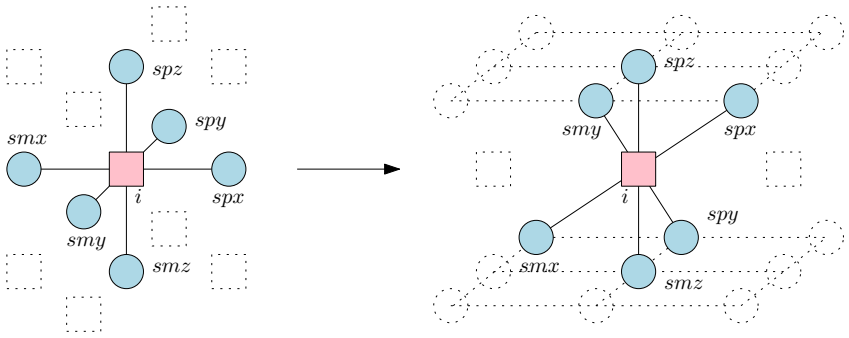


Figure 3.3. Cubic stencil for the real lattice (left) and for the transformed lattice (right).

makes apparent that for the calculations of, say, $\vec{x}'_{spz}$ can be reused for $\vec{x}'_{smy}$ and $\vec{x}'_{spx}$, hence one can write

$$\vec{x}'_{spx} = \hat{i}'(x'+1) + \vec{x}'_{smy} = \hat{i}'(x'+1) + \hat{j}'(y'-1) + \vec{x}'_{spz}, \tag{3.6}$$

where $\hat{i}'$, $\hat{j}'$ and $\hat{k}'$ are the unit basis vector for the x', y' and z' direction respectively. This feature gives some advantage in terms of calculations for the memory accesses, and it does not hold for the usual expressions. Thus, the cubic stencil layout in both set of coordinates looks as in Fig. 3.3, and periodic boundary conditions apply also to the new coordinates.

Memory transactions for bulk spins are completely coalesced with some per-warp redundancy for *smy*/*spx* and *spy*/*smx* which is completely handled by the hardware. The new layout naturally shows a two-dimensional locality of data. Indeed, such a remapping is in the same spirit as the one proposed in [55], with the difference that being this approach geometric it can be extended to other dimensionalities as shown above.

After having explained the memory arrangement we finally explain the spins-threads mappings. Some general remarks are in order:

- we simulate four replicas at once, systems with the same quenched disorder but different initial conditions and evolutions, which are stored in different arrays. Such a choice is a common practice [31]. However, the extension to an arbitrary number of replicas is trivial just requiring to handle a new stride in the kernels.

- colours and couplings are bound to textures in order to delegate addresses calculations to the texture hardware rather than to CUDA cores. Indeed one colour is constant while updating the other.

- for the sliced scheme, since the arrays are separated, one does not really need to consider z' as running from 0 to $L-1$, but rather from 0 to $L/2-1$. Let us assign the same z' label to pairs of differently coloured $x'-y'$ planes starting from the bottom with red vertices. The blue *spz* of a red vertex i has its same index, *i.e.* $spz = i$, whereas the blue vertex j will have its bottom red spin at $smz = j$. This scheme further simplifies calculations.

- Currently, physically interesting behaviour of the EA3D model can be studied only for relatively small sizes, hence we should try to saturate the GPU resources also for small-size lattices. The easiest way to achieve this goal is to simulate different coded disorder realizations numbered by k. Thus, the stride separating in the spins arrays different coded samples is $L^3/2 = V/2$.

  As far as we know only in [54, 60] such a technique has been adopted and indeed it is possible to sustain almost stable performances while varying the linear size L. We reserve the y, z block grid dimension as disorder index, which seems a reasonable choice since `gridDim.y,z` < 65536.

- couplings are indexed as if they were red vertices and they are allocated in six different arrays `Jpx, Jpy, Jpz, Jmx, Jmy, Jmz`. This choice introduces an asymmetry in the kernels which can easily be fixed by allocating a copy of the couplings suitably transformed in order to be indexed as blue vertices. However, we do not show the results since the difference in the performances of the two updating kernels is negligible.

We implemented two different ways of mapping the vertices index to the threads index:

1. a one-dimensional mapping that associates s vertices to a single thread. By tuning s is possible to find the best performance. Though, one needs to compute two divisions and two modulus operations in order to calculate nearest neighbours indices.

2. a multi-dimensional mapping exploiting the grid algebra provided by the GPU which allows to avoid divisions and modulus operations at the price of a more rigid choice for the total number of threads. The corresponding Kernels are tagged as *Grid*.

The kernel launch parameters for the first case are defined as follows

```
dim3 block(blockSize,1,1);
int fitGrid = (V/2/s + blockSize - 1)/blockSize;
dim3 grid(fitGrid, k, 1);
```

where `s` $= s$ is the number of spins per thread and `blockSize` $= 32n$, *i.e.* a multiple of the warp size. For Grid kernels the launch parameters are

```
dim3 blockG(L, l, 1);
dim3 gridG(A/(blockG.x*blockG.y), L/2, k);
```

where `A=L*L` and `l` is the number of lines of a single plane updated by a thread block. We highlight that for the latter case we use `threadIdx.x`, `threadIdx.y` and `blockIdx.y` as x, y and z indices respectively. For stantard-Grid kernels one has

```
dim3 blockG(L/2, l, 1);
dim3 gridG(A/(blockG.x*blockG.y)/2, L/2, k);
```

The nearest neighbours indices are always calculated from the one-dimensional index of the spin that is updated. Here we report the calculation which are the same for the sliced and sliced-Grid kernels in order to be as clear as possible

```
int smz = i + (SM(z - 1, d_hL) - z)*d_A;
int spy = smz + (SP(y + 1, d_L) - y)*d_L;
int smy = i + (SM(y - 1, d_L) - y)*d_L;
int smy = i + (SM(y - 1, d_L) - y)*d_L;
int smx = spy - x + SM(x - 1, d_L);
int spx = smy - x + SP(x + 1, d_L);
```

where `i = kk + off`, begin `kk` $< V/2$ and `off = blockId.y,z*d_hV` a disorder offset (with `d_hV` $= V/2$). We have implicitly set `spz = i`. In order to avoid the modulus operation enforcing the periodic boundary conditions we defined the macros `SM` and `SP` which read

```
#define SP(a, m) (a&(~(-(a >= m))))
#define SM(a, m) (a+((-(a < 0))&m))
```

Let us briefly comment the definition of `SP`: if `a >= m` evaluates to `1` then

```
(~(-(a >= m))) = 0x00000000
```

i.e. all bits set to zero, otherwise one has

```
(~(-(a >= m))) = 0xffffffff
```

i.e. all bits set to one. The macro `SM` is completely analogous. Hence we have reproduced the periodic boundary conditions since `SP(m + 1, m) = 0` and `SM(-1, m) = m - 1`.

We remark that we had to define another macro `SMM` for the Grid version in order to handle the fact that `threadIdx` and `blockIdx` variables are unsigned integers.

We tested the new access scheme comparing it with an implementation of the classic checkerboard spins arrangement, which we will refer to as **standard**, and with another scheme [56] using mainly bitwise operations, which however works only in the case $L = 2^\ell$. We will refer to this last implementation as **bitwise**.

We also wrote the Grid version of the standard scheme so that we end up with five different kinds of kernels: bitwise, standard, standard-Grid, sliced, sliced-Grid.

3.3 Pseudo-Random Numbers Generators

We chose to implement as a baseline the so-called Lehmer-Park-Miller MINSTD Linear Congruential PRNG which is defined as

$$R_{n+1} = (16807\, R_n) \mathrm{mod}(2^{31} - 1). \tag{3.7}$$

Its period is a prime number, more precisely a Mersenne prime $M_{31} = 2^{31} - 1$. This generator can be used for the coupling values J_{ik} and it is also a reasonable choice

for the critical off-equilibrium relaxation dynamics under the hypothesis of a number of simulation steps not larger than the period.

One difficulty comes with the implementation of the module which cannot be carried out by hardware truncation, so that we need to directly handle the overflow due to the multiplication and then take the module. This can be done by means of a swap 64-bit variable or by means of 32-bit variables only as proposed by Carta in [68, 69]. The latter solution does not require the module operation. We followed the implementation proposed in [69] but we substituted the conditional statements with bitwise operations to avoid warp branchings.

However, since we plan to extend this MC implementation to the equilibrium regime we also developed a GPU version of the Parisi-Rapuano PRNG [65] which is mostly used in the spin glass community. keeping in mind that one would also need to sum such a generator to a congruential one in order to obtain reliable random numbers for long parallel sequences [66, 60]. The Parisi-Rapuano is a lagged-Fibonacci-like PRNG with a minimal state of 62 words. One instance of the generator reads

```
ira[i] = ira[i - 24] + ira[i - 55];
R = ira[i]^ira[i - 61];
```

where `ira` denotes the state array and `R` is the new random number. A common approach [70, 62] consists in exploiting the lags and let the threads in a block share one or more states which can be concurrently updated storing them in Shared Memory. However lags as those of the Parisi-Rapuano PRNG are not well-suited for this scheme [62].

Hence, we propose a new simple alternative: allocate an array of $N_{threads} \times N_{state}$ entries and let each thread access it with its own global grid index and load the lagged entries just using a stride, *i.e.* the number of threads. Thus, defining `d_threads` as the number of threads and `globalId` as the thread global grid index, a sketch of the kernel implementation simply reads

```
swap = ira[(i - 24)*d_threads + globalId]
     + ira[(i - 55)*d_threads + globalId];
R = swap^ira[(i - 61)*d_threads + blobalId];
ira[i*d_threads + globalId] = swap;
```

although in a real implementation one has to take into account the periodic conditions for the access to the state. This can be easily obtained through the `SP` macro we defined before, thus avoiding modulus operations.

In order to show the validity of this scheme we chose to implement the much known Mersenne Twister MT19937 and compare its performance to that of the cuRand MTGP32 which is a modified version of the Mersenne Twister. Because of the adopted implementation in which one state is shared in a thread block the device API version requires the block size to be no larger that 256. We can also directly compare it to the cuRand host API version of MT19937 which is available for CUDA 6.0, only for devices of compute capability equal to or higher than 3.5.

We want to stress that our implementation does not share any of these restrictions. For the host API comparison we use the criterion proposed in [71]. Nonetheless, we

	Tesla M2090			Tesla K20X			GTX Titan		
PRNG	t_{INIT}(s)	t_{GEN}(s)	t_{TOT}(s)	t_{INIT}(s)	t_{GEN}(s)	t_{TOT}(s)	t_{INIT}(s)	t_{GEN}(s)	t_{TOT}(s)
cuRand MTGP32	0.09	12.34	12.43	0.12	13.46	13.58	0.21	10.11	10.32
cuRand XORWOW	0.01	2.91	2.92	0.01	2.90	2.91	0.01	2.31	2.32
cuRand MT19937	0	0	0	0	0	0	0.01	3.23	3.24
MT19937	3.86	6.40	10.26	4.63	6.12	10.75	3.92	4.66	8.58
Parisi-Rapuano	0.40	8.17	8.57	0.45	5.87	6.32	0.41	4.18	4.59
MINSTD	0.01	1.72	1.73	0.01	1.34	1.35	0.01	1.13	1.14

Table 3.1. PRAND benchmark [71] results using cuRand host API. The task consists in filling an array of 2^{29} single-precision floating point variables. In the upper half of the table cuRand library results are reported while in the lower half those of our implementations. Two different measures are reported: t_{INIT} is the time needed to initialize the PRNG; t_{GEN} is the generation time. For the M2090 ECC is off, while for K20x ECC is on.

PRNG	M2090	K20X	GTX Titan	GTX 680
cuRand MTGP32	$4.5 \cdot 10^9$	$3.9 \cdot 10^9$	$5.2 \cdot 10^9$	$5.1 \cdot 10^9$
cuRand XORWOW	$2.9 \cdot 10^{10}$	$7.6 \cdot 10^{10}$	$10.7 \cdot 10^{10}$	$6.1 \cdot 10^{10}$
MT19937	$10.1 \cdot 10^9$	$10.7 \cdot 10^9$	$14.1 \cdot 10^9$	$9.6 \cdot 10^9$
Parisi-Rapuano	$9.4 \cdot 10^9$	$12.5 \cdot 10^9$	$16.8 \cdot 10^9$	$8.3 \cdot 10^9$
MINSTD	$4.1 \cdot 10^{10}$	$7.6 \cdot 10^{10}$	$8.9 \cdot 10^{10}$	$6.7 \cdot 10^{10}$

Table 3.2. Device API test. Number of instances per second. The launch configuration is the following: 64 blocks of 256 threads, each thread producing 2^{15} instances, repeated 10 times. For the M2090 ECC is off, while for K20x ECC is on.

propose as standard benchmark for a PRNG its kernel version counting the fraction of odd numbers (just as the example reported in the cuRand manual [72]). Such a benchmark should be more suitable for kernel-use PRNGs.

Results are reported in Table 3.1 for the PRAND test, and in Table 3.2 for the device API test. Tests were run on GTX 680, GTX Titan, Tesla M2090 and Tesla K20x GPUs. Looking at Table 3.1, where we report the execution times for filling an array of 2^{29} single-precision floating point variables, we see that our implementation of MT19937 runs roughly twice as fast as the cuRAND MTGP32 implementation. We could only test the most recent Host API cuRAND implementation of the MT19937 on the GTX Titan and not on the K20x (the M2090 is ruled out being too old), and our implementation performs 44% slower than cuRAND. The large t_{INIT} values for our implementation are due to the fact that the seed are read from the system random pool, slowing down the process. It is clearly possible to reduce those times implementing some initialization algorithms as those proposed in [73]. In Table 3.2 the metric is changed to the number of PRNG instances per second. The trends are qualitatively the same although our implementation of the MT19937 on the GTX Titan runs almost three times faster than the cuRAND MTGP32.

However, such benchmarks only give a qualitative comparison of different PRNGs. As we will see one should always compare different PRNGs in a given real-life implementation.

3.4 Asynchronous Multispin Coding

Multispin coding techniques are rooted in lattice gauge theory simulations [74]. They have been employed later in Ising models simulations [75, 76, 77, 78]. The search for a close packing of data was motivated by the limited memory resources of that time, and by the intrinsic bit-level parallelism which can be obtained through bitwise operations. Indeed, since the quantities involved in the simulations are two-valued, *i.e.* $\sigma_i \in \{-1, +1\}$ and $J_{ik} \in \{-1, +1\}$, the optimal solution is to store couplings and spins in single bits rather than use a single byte, *e.g.* using a `char`.

Multispin coding comes in two different flavours:

- synchronous multispin coding (SMSC) consisting in storing in one word spins belonging to one single system, usually aligned along one specific direction. This allows to get faster simulations in terms of wall-clock time compared to a simple one-variable-one-spin setting. Indeed, such a technique is used in the Janus supercomputer [28] for reaching thermal equilibrium. Clearly, the update of each bit-spin requires one instance of the PRNG;

- asynchronous multispin coding (AMSC) consists in storing spins belonging to different systems, located at the same vertex, in the same word. The total wall-clock time does not decrease, but it is possible to update all spins contained in a word with only one instance of the PRNG at the cost of the introduction of a certain amount of correlation, which can be taken care of easily.

We chose to implement the AMSC because we were interested in the off-equilibrium critical relaxation regime, hence being able to simulate a large number of samples is preferable over obtaining a long simulation time. The AMSC for spin systems was clearly explained in [78] where each system was considered to be at a different temperature. We are aware of some AMSC implementations on GPU: [50] for the 2D Ising model and [54] for the EA3D model with external field. In particular in [54] the proposed AMSC technique stores in one word spins of the same system, *i.e.* with the same couplings, which are evolved at different temperatures. This scheme has been adopted for implementing the PT dynamics. Transition probabilities are stored in a look-up table indexed by the energy difference ΔE of the proposed flip and the spin direction (with respect to an external magnetic field). Hence, the swap of two temperature-replicas simply requires to swap two lines in the look-up table. However, in order to speed up the access to the look-up table, some space in the spins words is reserved so that not all bits of a word codify for a spin. We will see that for non-PT dynamics this represents a bottle-neck for memory use efficiency. Again, each spin update is served by one PRNG instance.

As we anticipated, we associate to each spin a different disorder realization, thus only one PRNG instance is needed for all spins contained in a word. Considering the contribution to the Hamiltonian due to a single cubic stencil, it is clear that the possible energy differences after a proposed spin flip on σ_a are

$$\Delta E = H[\{\sigma_{i\neq a}, -\sigma_a\}] - H[\{\sigma_{i\neq a}, \sigma_a\}] = -12, -8, -4, 0, 4, 8, 12. \tag{3.8}$$

The Metropolis dynamics is defined by the acceptance probability

$$P_{\text{flip}}(\Delta E) = \begin{cases} 1, & \Delta E \leq 0 \\ e^{-\beta \Delta E}, & \Delta E > 0 \end{cases} \tag{3.9}$$

where $\beta = T^{-1}$ is the inverse temperature. The value of $P_{\text{flip}}(\Delta E > 0)$ has to be compared to a flat-distributed random number $r \in [0, 1]$ so that if $r < P_{\text{flip}}(\Delta E > 0)$ the proposed flip is accepted otherwise it is rejected. However, since PRNGs are defined for integers, one does not really need to use a normalized r. The most direct way is to multiply the transition probability for the value of the biggest random number R_{max} and compare it with the PRNG instance R, *i.e.* $R \lesseqgtr R_{max} \exp(-\beta \Delta E)$. We label the non-trivial normalized transition probabilities as $R_{max} \exp(-\beta \Delta E) =$ `EXP12`, `EXP8`, `EXP4`. We employ the following mapping of spins and couplings to bits

$$\begin{aligned} J_{ik} = -1 \ &\rightarrow \ \texttt{J}_{ik}\texttt{=1}, \qquad \sigma_i = -1 \ \rightarrow \ \texttt{s}_i\texttt{=0}, \\ J_{ik} = +1 \ &\rightarrow \ \texttt{J}_{ik}\texttt{=0}, \qquad \sigma_i = +1 \ \rightarrow \ \texttt{s}_i\texttt{=1}. \end{aligned} \tag{3.10}$$

The value of the interaction energy with one of the nearest neighbours is then converted for each bit as

$$\begin{aligned} -J_{ik}\,\sigma_i\,\sigma_k = -1 \ &\rightarrow \ \texttt{e}_{ik} = \texttt{J}_{ik}\texttt{\^{}s}_i\texttt{\^{}s}_k = \texttt{0}, \\ -J_{ik}\,\sigma_i\,\sigma_k = +1 \ &\rightarrow \ \texttt{e}_{ik} = \texttt{J}_{ik}\texttt{\^{}s}_i\texttt{\^{}s}_k = \texttt{1}. \end{aligned} \tag{3.11}$$

If we sum the six energy variables per stencil $\texttt{e}_{ik}$ we obtain a three bits result

$$\sum_k \texttt{e}_{ik} = (\texttt{sum2, sum1, sum0}) = 2^2 \times \texttt{sum2} + 2 \times \texttt{sum1} + \texttt{sum0}, \tag{3.12}$$

which directly maps to the seven possible values of ΔE since flipping the spin leads to flip the partial values $\texttt{e}_{ik}$.

$$\begin{array}{llll} (\texttt{0, 0, 0}) = 0 & \rightarrow & \Delta E = -12 & (\texttt{1, 0, 0}) = 4 \ \rightarrow \ \Delta E = 4 \\ (\texttt{0, 0, 1}) = 1 & \rightarrow & \Delta E = -8 & (\texttt{1, 0, 1}) = 5 \ \rightarrow \ \Delta E = 8 \\ (\texttt{0, 1, 0}) = 2 & \rightarrow & \Delta E = -4 & (\texttt{1, 1, 0}) = 6 \ \rightarrow \ \Delta E = 12 \\ (\texttt{0, 1, 1}) = 3 & \rightarrow & \Delta E = 0 & \end{array} \tag{3.13}$$

Now, the aim is to define a mask in order to flip the right spins with a XOR operation

$$\texttt{spin} = \texttt{spin\^{}mask}; \tag{3.14}$$

As a first step we compare the random number `R` with the non-trivial transition probabilities defining the variables

```
cond12 = -(R < EXP12);
cond8 = -(R < EXP8);
cond4 = -(R < EXP4);
```
(3.15)

i.e. if `R` < `EXP4` then `cond4 = 0xffffffff` (all bits to one), while if `R` > `EXP4` then `cond4 = 0x00000000` (all bits to zero). Clearly, if `cond12 = 0xffffffff`, *i.e.* the

most improbable flip can be accepted, then all spins must be flipped. Also all spins with `sum2 = 0` must be flipped so that we write

```
mask = cond12 | (~sum2);
```
(3.16)

where the `|` stands for the logic OR operator. We still need to handle the two remaining non-trivial cases corresponding to $\Delta E = 4, 8$. A first selection is obtained using `sum2` as a mask, although we still need to discard the case $\Delta E = 12$, hence we write `sum2 & (sum2 ^ sum1)`. The flipping condition for the cases $\Delta E = 4, 8$, when `R < EXP8`, simply reads `(sum2 & (sum2 ^ sum1)) & cond8`. The last step is to consider $\Delta E = 4$ when `R < EXP4` which leads to `(sum2 & (sum2 ^ sum1)) & (cond8 | (cond4 & (~sum0))`. All in all the mask reads

```
mask = cond12 | (~sum2)
      | ((sum2 & (sum2^sum1)) & (cond8 | (cond4 & (~sum0))));
```
(3.17)

This expression has the same number of bitwise operations of the natural extension of [78].

3.4.1 Results

We present now the results concerning the performances of the different GPU implementations which are labeled as **sliced**, **standard** and **bitwise**. The **sliced** one uses the sliced checkerboard scheme we propose in this work, whereas the **standard** and **bitwise** implementations are based on the usual checkerboard scheme with the difference that the last one only works for linear sizes which are powers of two, $L = 2^\ell$ and the calculations are implemented mainly through bitwise operations. We checked that all these schemes give the same bit-to-bit results so that they are completely equivalent[2].

Before discussing the results let us define the principal metric we will use in order to measure performances: the pico-second-spin-flip $\mathrm{psFlip}_{n,\mathrm{X}}$ that is how many pico-seconds are needed in order to reject or accept a proposed spin-flip. Here n stands for the number of GPUs and 'x' for the used PRNG. The mathematical definition is the following

$$\mathrm{psFlip}_{n,\mathrm{X}}(L,k) = t_{\mathrm{SW}} \cdot n \cdot \left(32 \cdot k \cdot 4 \cdot L^3\right)^{-1}, \tag{3.18}$$

where $32 \cdot k$ is the number of different disorder realizations (32 multispin-coded times k different codings), 4 is the number of simulated replicas, and t_{SW} is the wall-clock time needed to perform one sweep, *i.e.* update red and blue spins, for *all* disorder realizations. Indeed, t_{SW} is always measured on a single node. Data were taken for four different GPUs: GTX 680, GTX Titan, Tesla M2090 and Tesla K20x.

Let us begin by analyzing a behaviour which has been seldomly explored in the past literature: how to saturate the GPU resources for small lattices. The solution we adopted, as others did [54, 60], is to allocate at the same time different systems. As it is shown in Fig.3.4, in the case of $L = 8$, t_{SW} is almost constant up to $k = 64$

[2]Precisely, the bitwise check between the sliced and the other implementations requires to remap all random numbers after one of the two colours has been updated.

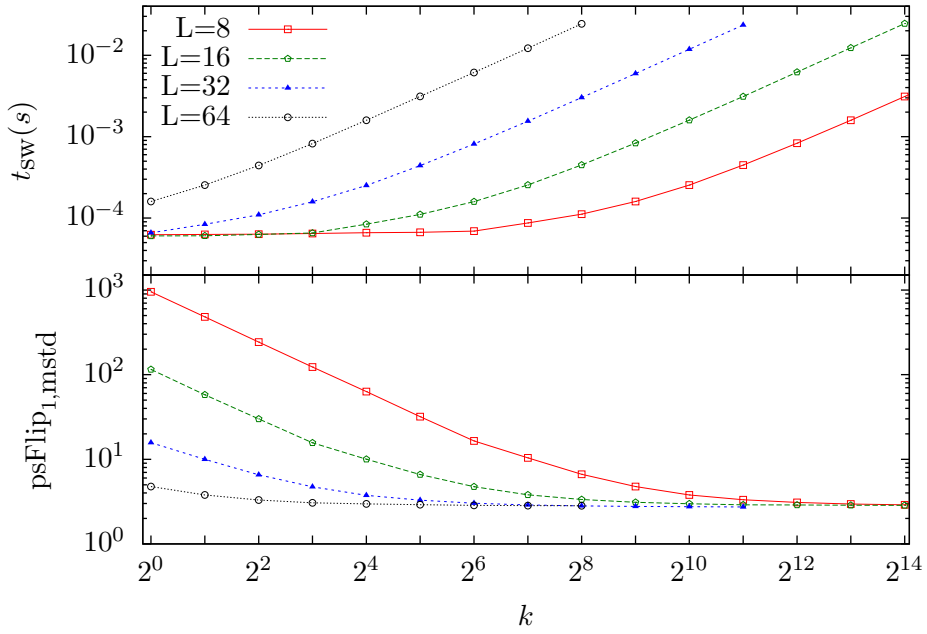


Figure 3.4. Values for t_{sw} and $\mathrm{psFlip}_{1,\mathrm{mstd}}$, *i.e.* the number of pico-seconds needed to accept or refuse a flip proposal with the MINSTD PRNG, as a function of the number of coded systems k for a GTX Titan. Data refer to the best performances at varying grid launch parameters for a given value of k for different sizes L.

which means that simulating one or 64 coded systems has the same cost for the GPU. This means that a factor 64 can be gained for free. Indeed, this observation is important since the accessible physics for the EA3D is still confined to relatively small lattices, hence obtaining the best result also for $L \leq 32$ is crucial. We notice that even though, for $k > 64$, t_{sw} starts to increase, a linear regime is attained only for $k \geq 4096$ in the case $L = 8$. Indeed, the case $L = 32$ saturates the GPU almost at the beginning and the metric $\mathrm{psFlip}_{1,\mathrm{mstd}}$ only evolves from 4 psFlip to 3 psFlip, which however is a $\sim 25\%$ gain.

Now, in order to make a fair comparison with Janus FPGA hardware [79, 29], it is important to stress that those machines sustain comparable performances in terms of pico-seconds-spin-flip (16 psFlip for Janus and 3-5 psFlip for Janus II) for a **single** sample also for small lattice sizes. In our case we need to simulate several samples to saturate the GPU resources. Hence, Janus and Janus II are the fastest solution in terms of wall-clock time to bring a single sample to equilibrium and for small lattice sizes GPUs are still far away. A direct comparison with Janus supercomputers can be only performed when a single system is large enough to saturate the GPU resources. However, the game is subtle since saturation is attained only for large sizes which might be out of the domain of physical interest, at least for equilibrium simulations.

In Figures 3.5 and 3.6 we report benchmarks results for different GPUs and different algorithms on all even lattice size in the range $8 \leq L \leq 256$. They all share the MINSTD as PRNG. Benchmarks were performed measuring the sweep wall-clock

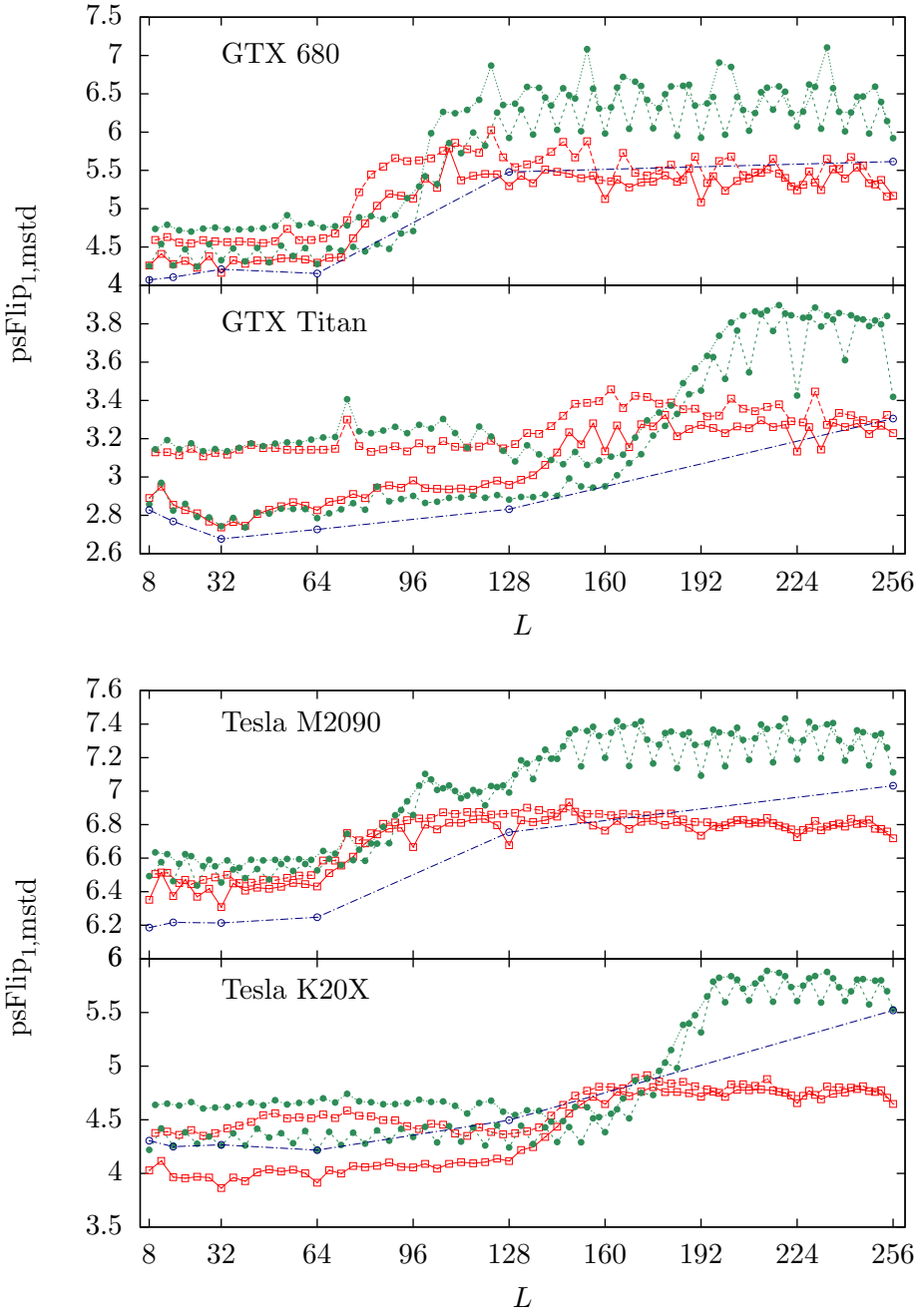


Figure 3.5. Best performances for $\text{psFlip}_{1,\text{mstd}}$. The red empty squares refer to the **sliced** implementation, the light-green filled circles refer to the **standard** implementation while the blue empty circles to the **bitwise** one. The value of L_{thr} is larger for the GTX Titan and Tesla M2090.

time t_{SW} while varying L, k and the grid configuration for the kernel, in order to find the best configuration for each lattice size, *i.e.* only the best configurations times are reported. There are some qualitative features which are shared by the

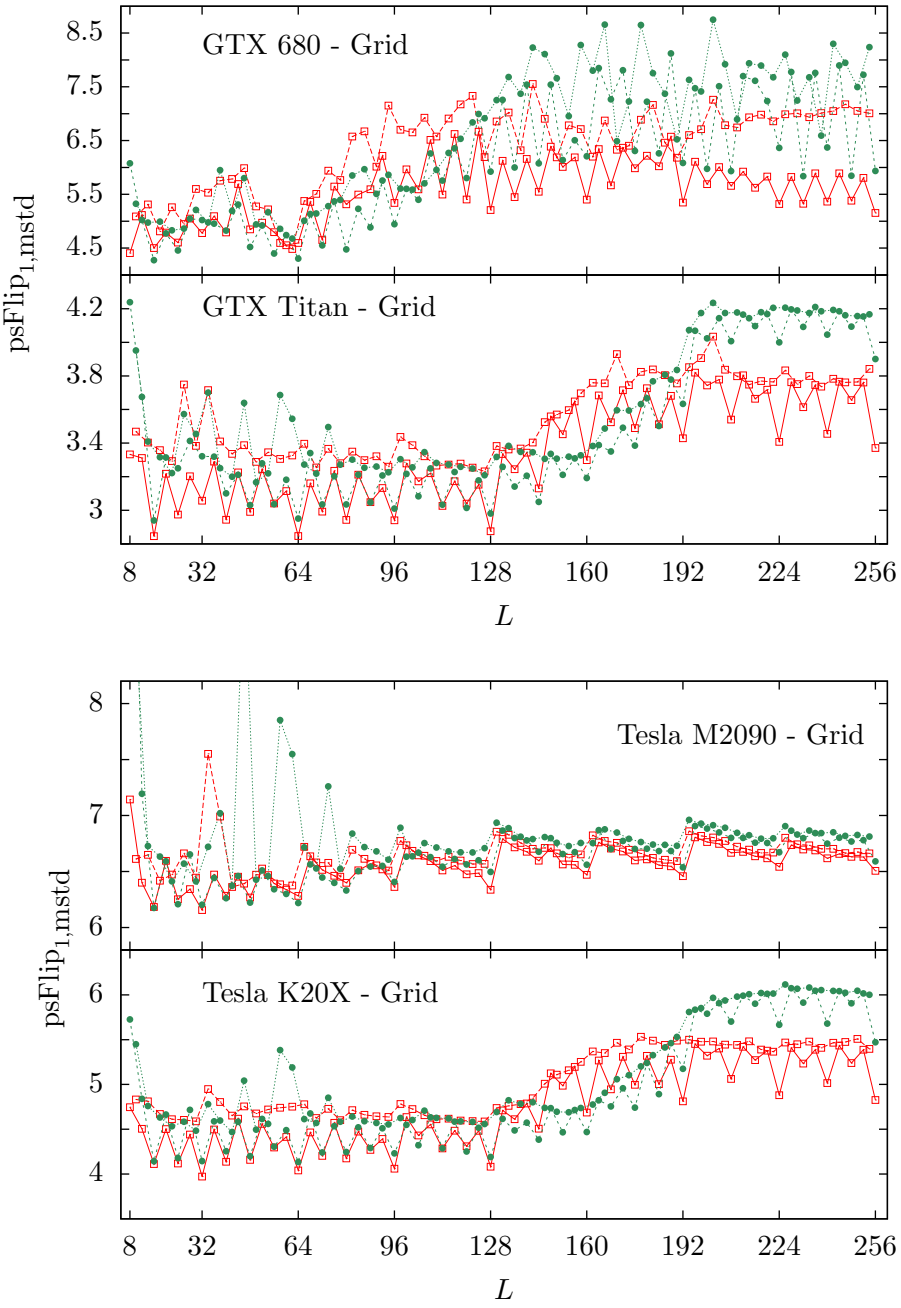


Figure 3.6. Best performances for $\text{psFlip}_{1,\text{mstd}}$ for Grid implementations. The red empty squares refer to the **sliced-Grid** implementation and the light-green filled circles refer to the **standard-Grid** implementation.

different GPUs

- the best performances are obtained in the first range of lattice sizes $L < L_{thr}$, where the threshold L_{thr} varies according to the GPU and the algorithm, assuming larger values for latest GPUs; L_{thr} is defined as the first value of L

for which $\mathrm{psFlip}_{1,\mathrm{mstd}}$ begins to grow significantly;

- the sliced scheme performances get worse always before those of the standard scheme do;
- the sliced scheme gives always the best performance for $L > L_{thr}$;
- we split the data in two different branches defined by two subsequences of the lattice size $L_0 = 4m$ (faster) and $L_1 = 2(2m+1)$ (slower), which converge for $L > L_{thr}$, and this splitting is most evident for small lattice sizes.

We want to stress that it is the first time, in our knowledge, that results are reported in such a wide range of lattice sizes with such a stability in performances.

The sliced scheme worsen before the standard does probably because the latter deals with boundary conditions on the y-axis only after $L^2/2$ elements have been processed whereas for the sliced scheme the boundary conditions on the y'-axis are treated after L^2 elements. Hence, a cache hit for the standard scheme is more likely. It appears that the behaviour of the GPU memory is somehow correlated to the number of memory requests for the periodic boundaries. As a matter of fact, the following scaling relation $L_{thr}^{standard} \sim \sqrt{2}\, L_{thr}^{sliced}$ roughly holds.

Data related to the standard-Grid and sliced-Grid implementations are clearly less stable. We notice that the standard-Grid implementation performs much worse than the sliced-grid. This should be related to the fact that the block size is fixed to the number of one-coloured spins in a z slice, which means for the standard-Grid scheme $L^2/2$, starting from 32 threads, and for the sliced-Grid L^2, starting from 64 threads. Hence, having blocks which coincide with a warp does not seem to be an optimal choice for the GPU. Looking at the data for the Tesla M2090 in figure 3.6 there is a modulation as a function of the lattice size with a period $\Delta L = 32$. We notice that for such values of L the blocks are always multiple of a warp.

The 'Grid' algorithms perform slower than the others in the examined range so that we can safely discard this implementation choice which relies on the inherent algebra of the thread-grid indices. Hence, we will focus hereafter mainly on the non-grid implementations.

In Fig.3.7 we report the optimal values of the number of coded systems k as a function of the lattice size which we can see decreases roughly in a power-law fashion. In particular we also show the ratio between the number of branching warps and k: for lattices belonging to the subsequence $L_1 = 2(2m+1)$ this ratio is always equal to one, whereas for the subsequence $L_0 = 4m$ it is always equal to zero. This result is explained by the fact that half of the volume of a lattice $V_i/2 = L_i^3/2$, *i.e.* all the one-coloured spin, is always a multiple of the warp size for the even subsequence $V_0/2 \propto 32$ whereas it is not so for the odd subsequence $V_1/2$,

$$\frac{V_0}{2} = \frac{(4m)^3}{2} = 32m^3, \qquad \frac{V_1}{2} = \frac{[2(2m+1)]^3}{2} = 4(2m+1)^3. \tag{3.19}$$

In order to prove this let us look if there exist a value of m for which $V_1/2$ is a multiple of the warp size

$$\frac{V_1}{2} = 4(2m+1)^3 = 32n, \qquad 2m+1 = 2n^{1/3}, \qquad m = n^{1/3} + \frac{1}{2}, \tag{3.20}$$

which has integer solutions for m for non-integer n. Since we are looking for integer values of n this proves the previous assertion. Hence, the subsequence of cubic lattices of linear size L_1 is intrinsically uncommensurate to the actual warp size which is characteristic of the CUDA framework. Thus, as long as the warp size is fixed to the actual value, there will always be warp branchings for checkerboard algorithms updating one colour at the time. Indeed, this result is correlated to the fact that the two subsequences L_0 and L_1 have different performances, but does not provide a full explanation.

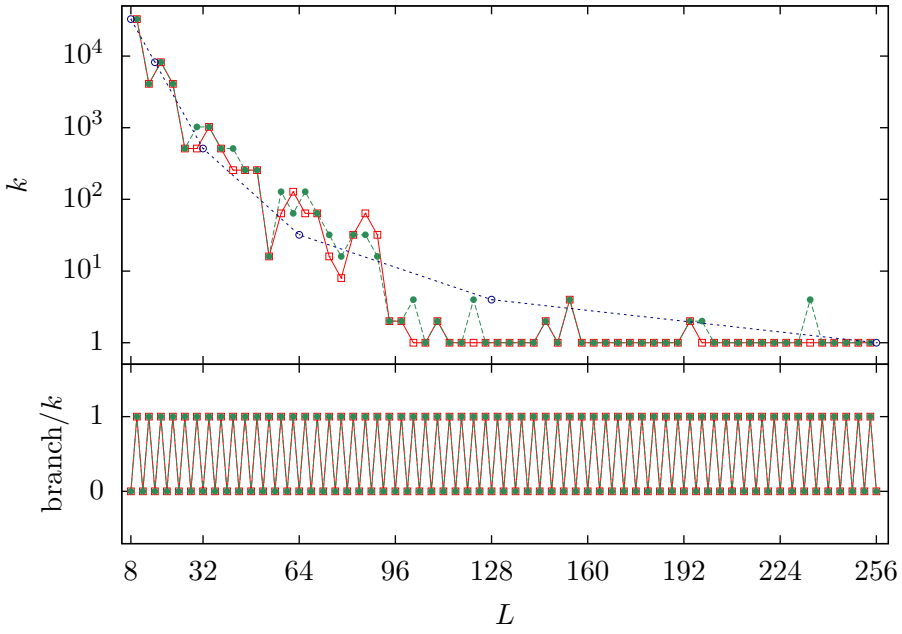


Figure 3.7. Upper panel: best values of k for the GTX 680. The red empty squares refer to the **sliced** implementation, the light-green filled circles refer to the **standard** implementation while the blue empty circles to the **bitwise** one. Lower panel: number of branched warps divided by k: for the sliced and the standard implementations the subsequence $L_1 = 2(2m + 1)$ has a divergent warp for each system.

To complete the analysis for the best performances, we report in Fig. 3.8 the results for the bandwidths measures for the best launch configurations. Except for the Tesla M2090 the sliced and the standard algorithms saturate the available bandwidth in the entire range with some fluctuations.

Let us now examine the results for different PRNGs: our implementations of the Parisi-Rapuano and the usual Mersenne Twister MT19937, together with the cuRand XORWOW (which is the standard cuRand PRNG) and MTGP32 which is a reduced version of the MT19937. While for the first three PRNGs we could perform full benchmarks with the only limitation of the memory usage, for the MTGP32 we could use a maximum number of 200 blocks and a maximum blocks size of 256 threads. As for the number of blocks, this is a limitation of the standard usage which, however, can be by-passed with some effort as reported in the cuRand documentation [80].

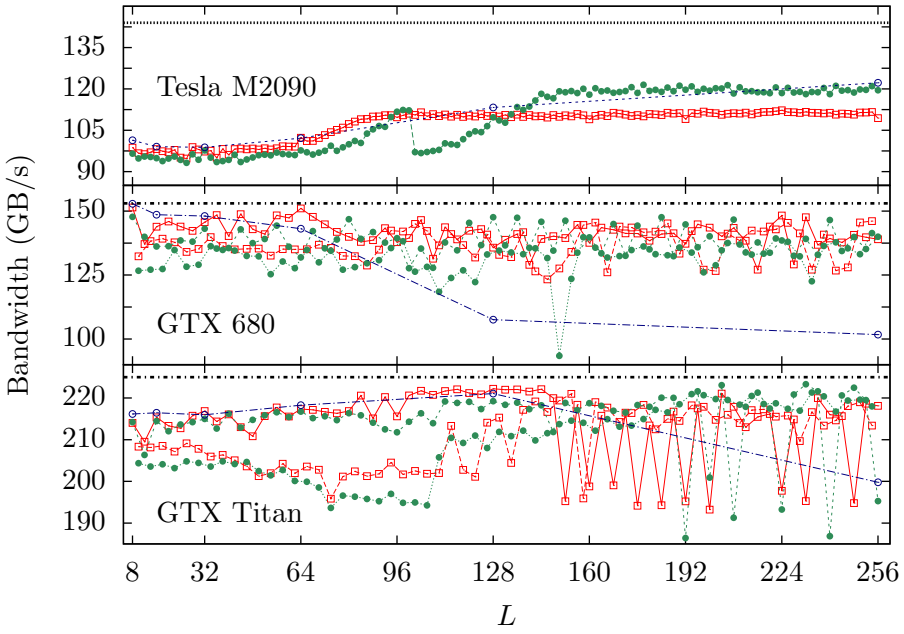


Figure 3.8. Data for the GTX 680 and GTX Titan GPUs. The red empty squares refer to the **sliced** implementation, the light-green filled circles refer to the **standard** implementation while the blue empty circles to the **bitwise** one. The horizontal lines are the peak bandwidths.

Results are reported in Fig. 3.9: in the upper panel we show the values of $\mathrm{psFlip}_{1,\mathrm{x}}$ normalized to the MINSTD performances $\mathrm{psFlip}_{1,\mathrm{mstd}}$, which we use as a baseline, for the three different algorithm implementations. In $\mathrm{psFlip}_{1,\mathrm{x}}$, 'x' labels three different PRNGs: Parisi-Rapuano, MT19937 and XORWOW. All data refer to the GTX Titan GPU. It is clear that the lowest ratio for the XORWOW is obtained for the sliced implementation for which it is ~ 2 whereas for the standard and bitwise versions the ratio is ~ 3. The Parisi-Rapuano and the MT19937 have roughly the same ratio for the three different algorithms.

In the lower panel of Fig. 3.9 we report the absolute values for $\mathrm{psFlip}_{1,\mathrm{x}}$. it is possible to see that the performances for our implementations of the Parisi-Rapuano and MT19937 and of cuRand MTGP32 weakly depend on the chosen algorithm, while there is a considerable difference for the XORWOW for which $\mathrm{psFlip}_{1,\mathrm{xor}} \sim 6$ps for the sliced scheme while $\mathrm{psFlip}_{1,\mathrm{xor}} \sim 9$ps for the sliced and bitwise implementations. There are two main results emerging from the data:

- the standard cuRAND XORWOW performs slower than our best-quality PRNG, the MT19937;
- the sliced scheme is more robust with respect to a change in the memory bandwidth load.

The first point can be easily understood by considering that the data structure of the cuRand XORWOW PRNG has a size of 48 bytes: each 128 byte transaction, which is served from the L2 cache, only loads the data needed by two threads, so

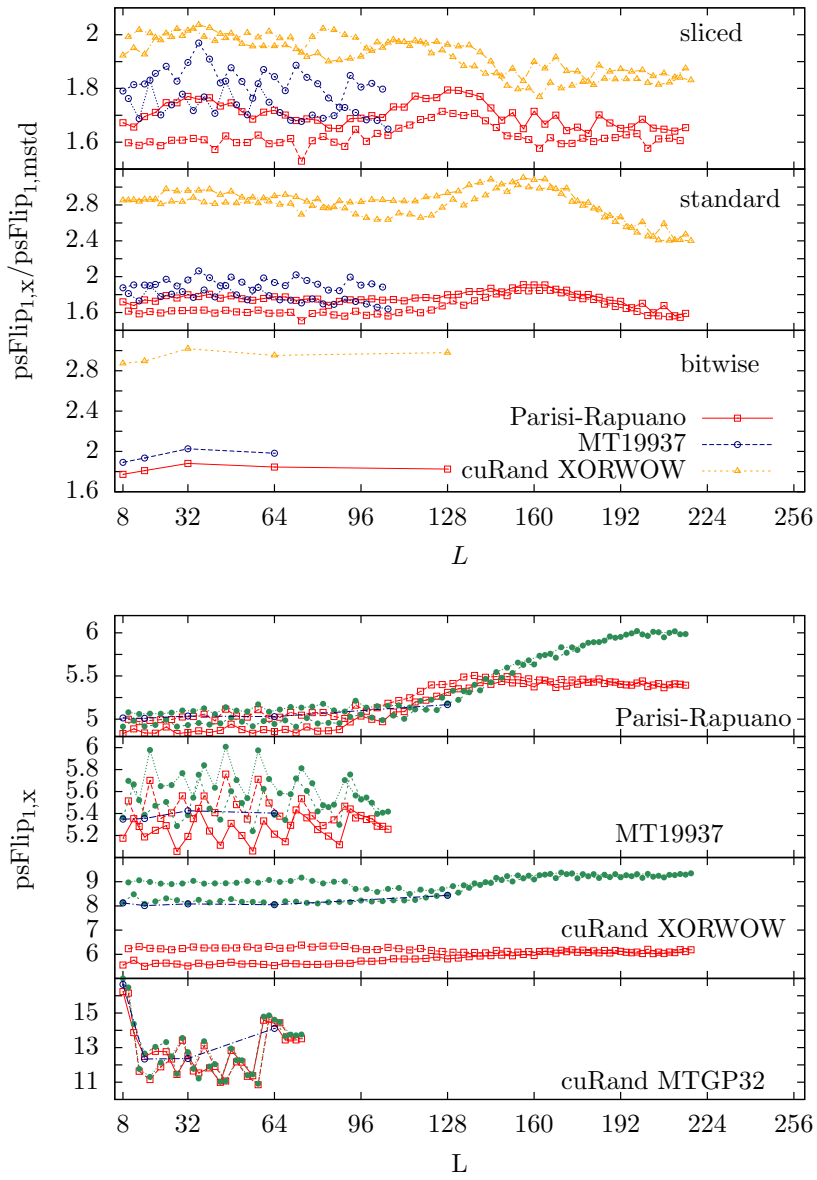


Figure 3.9. Top panel: $\text{psFlip}_{1,\text{x}}$ normalized to the MINSTD are shown. Bottom panel: the absolute performance is reported where the red squares represent the sliced scheme data, the filled green circles and the empty blue circles those of the standard and bitwise schemes respectively. All data refer to the GTX Titan.

that we need roughly 16 memory transactions for a warp to be ready, whereas in our approach we only need $O(1)$ memory transactions, *e.g.* 3 for the Parisi-Rapuano and the MT19937. Indeed, also the MTGP32 follows a similar pattern in that every thread in a warp loads in the shared memory one entry of the state. The strategy used for the XORWOW implementation is not adequate for intense memory usage algorithms.

As for the second point this should be a proof that the memory alignment given

by the sliced scheme is better suited for second hits in the caches: indeed the amount of needed data transfers is the same for the three schemes in the XORWOW case but for the sliced scheme there is a $\sim 33\%$ gain with respect to the standard and bitwise schemes.

As a final remark, the MTGP32 performs from 3 to 5 times worse than the MINSTD implementation. We stress that this result is strongly influenced by some limitations of the cuRand implementation which, however, can be softened with some further work.

3.5 Multi-GPU Implementation

As far as we know, there are just a few works showing strong scaling results for spin systems [50, 57, 58]. We chose to adopt the same technique proposed in [57, 58] where the partitioning is performed along the z'-axis of the system. All communications among nodes are handled by MPI and the overlap between calculations and communications is achieved by using CUDA streams. We maintain the single-GPU version flexibility for a customary number of spins per thread and coded systems k. A priori, it should not be taken for granted that the bulk update, executed on one CUDA stream, can mask the boundary update and data copy/transfer, executed on the other stream, since the algebraic intensity of the algorithm is rather low.

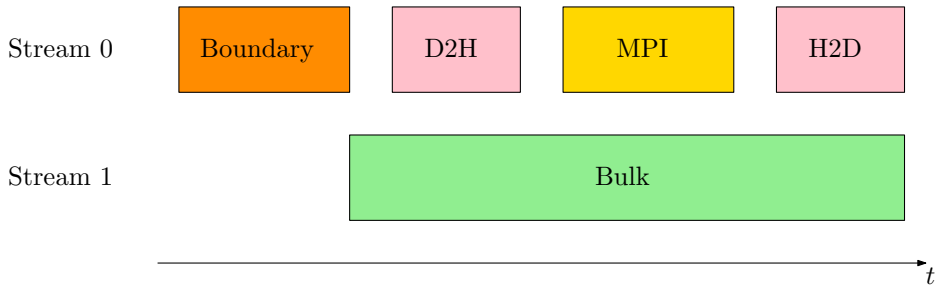


Figure 3.10. Scheme representing the multi-GPU strategy leveraging CUDA streams. Here 'Boundary' and 'Bulk' represent two kernels launched on the same GPU. After the boundary update on the stream 0 an asynchronous 'D2H' device-to-host copy of the only one-coloured boundary is performed, then 'MPI' handles the one-directional boundary exchange between nodes and an asynchronous 'H2D' host-to-device memory copy updates the boundary spins.

The multi-GPU version of the sliced Kernel is rather different from the one using the standard checkerboard scheme [57, 58] since the disposition of colours in the cubic lattice is different. At fixed z' value spins are one-coloured so that for every partition of the system the lowermost plane is always red whereas the highermost one is always blue. This means that when updating red spins the only boundary coincides with the lowermost red plane or with the highermost when updating the blue spins. Hence, the communication between the nodes goes in the downward direction for red spins and in the upward one for the blue spins: there is no need for all nodes to communicate with all nearest neighbours after a colour update. To-be-sent boundary spins are stored in the bulk array and copied to an auxiliary

buffer by the same kernel that performs the update. To-be-received boundary spins are stored in a separate array, bound to a texture, which is just read when updating the spins of the other colour. This scheme automatically handles the z'-axis periodic boundary conditions and reduces the number of intra-node communications. In Fig. 3.11 we report a depiction of the multi-GPU sliced scheme.

This is an interesting property which might be of use in cases where the amount of data to transfer is low and the latency time is comparable to the data-exchange time. Then, one would expect to have a significant speed-up in the communication.

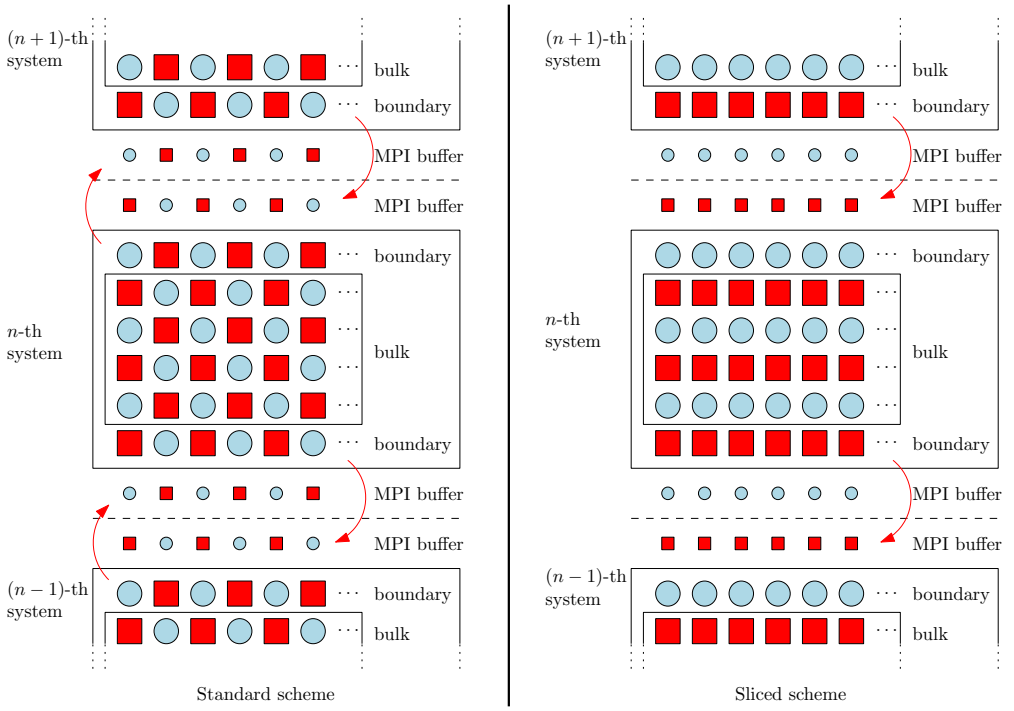


Figure 3.11. A depiction of the standard (on the left) and the sliced (on the right) checkerboard schemes for the multi-GPU version. Updating red spins one needs to update the bulk and the boundaries of each system partition. The standard scheme has two-coloured boundaries while for the sliced scheme these are one-coloured: communication (red arrows) must be two-ways for the standard implementation while is only one-directional for the sliced scheme. Clearly, in both cases the same amount of data is transferred.

3.5.1 Results

Let us now discuss the results we obtained for the multi-GPU implementation of the three-dimensional Edwards-Anderson model[3]. Given the definition (3.18) of $\mathrm{psFlip}_{N,\mathrm{X}}$, it clearly appears that for $N > 1$ we consider the time spent by a single GPU on its own system partition rather than the wall-clock time spent by the N

[3]For these measures we kept the number of spins per thread fixed to 4, one for each replicas, since in the single-GPU results this was the optimal value.

GPUs as a whole. However, the strong-scaling efficiency η_{SC} is directly defined as

$$\eta_{SC} = \frac{\text{psFlip}_{1,\text{x}}}{\text{psFlip}_{N,\text{x}}}, \tag{3.21}$$

and thus the performance referred to the multi-GPU system as a whole is defined as

$$\text{psFlip}_{\text{multi,x}} = \text{psFlip}_{N,\text{x}}/N, \tag{3.22}$$

allowing to recover the usual strong-scaling efficiency definition.

$$\eta_{SC} = \frac{\text{psFlip}_{1,\text{x}}}{\text{psFlip}_{N,\text{x}}} = \frac{\text{psFlip}_{1,\text{x}}}{N \times \text{psFlip}_{\text{multi,x}}}. \tag{3.23}$$

All data have been gathered on the Piz Daint Supercomputer which uses Tesla K20x GPUs [81]. In the top panel of Fig. 3.12 we report the strong scaling efficiency up to 8 GPUs. Indeed, the saturation efficiency is remarkable, $\eta_{SC} \gtrsim 0.9$, although the more the GPUs the further in terms of lattice size L one needs to go to reach a stable regime. Nonetheless up to 8 GPUs the algorithm practically scales linearly with the number of GPUs.

In the bottom panel of Fig. 3.12 where we show the values of $\text{psFlip}_{\text{multi,mstd}}$, hence considering N GPUs as a single system, the linear scaling in N is clearly visible for any number of GPUs. We obtain very good results in absolute terms: for $N = 2$ in the range from $L = 64$ to $L = 128$ we have an almost stable performance of $2\,\text{ps} < \text{psFlip}_{\text{multi,mstd}} < 3\,\text{ps}$.

Lastly, we want to pay some attention to the power-law behaviour visible in 3.12. It is easy to determine that roughly performances scale as

$$\text{psFlip}_{\text{multi,mstd}} \sim L^{-1}. \tag{3.24}$$

Now, looking at the definition (3.18), it is easy to derive that

$$\frac{L}{N}\,\text{psFlip}_{N,\text{x}}(L,k) \propto t_{\text{sw}} \left(\frac{L}{N}\right)^{-2}, \tag{3.25}$$

hence, defining the rescaled variable $x = L/N$ we can plot $x\,\text{psFlip}_{N,\text{x}}$ as a function of x. The result is shown in Fig. 3.13. Indeed, we can see that data for $N \leq 8$ collapse almost everywhere on the same curve whereas there are some deviations for $N \geq 16$, proving that x is a good scaling variable. From the plot two distinct regimes are visible: a first one where data lie on a horizontal line and a second one where they grow linearly in x. In the first regime the sweep wall clock time grows as $t_{\text{sw}} \sim L^2$, *i.e.* the boundary communication, which scales as the system area, dominates. In the second regime $t_{\text{sw}} \sim L^3$, which means that the wall-clock time is dominated by the bulk update task which is then able to mask the communication between the nodes.

Such a good scaling data and collapse show the quality of the communication technology, based on Aries routing, communications ASIC, and Dragonfly network topology, included in the Piz Daint Supercomputer [82, 83]. We propose to use this kind of analysis in order to measure the communication infrastructure quality.

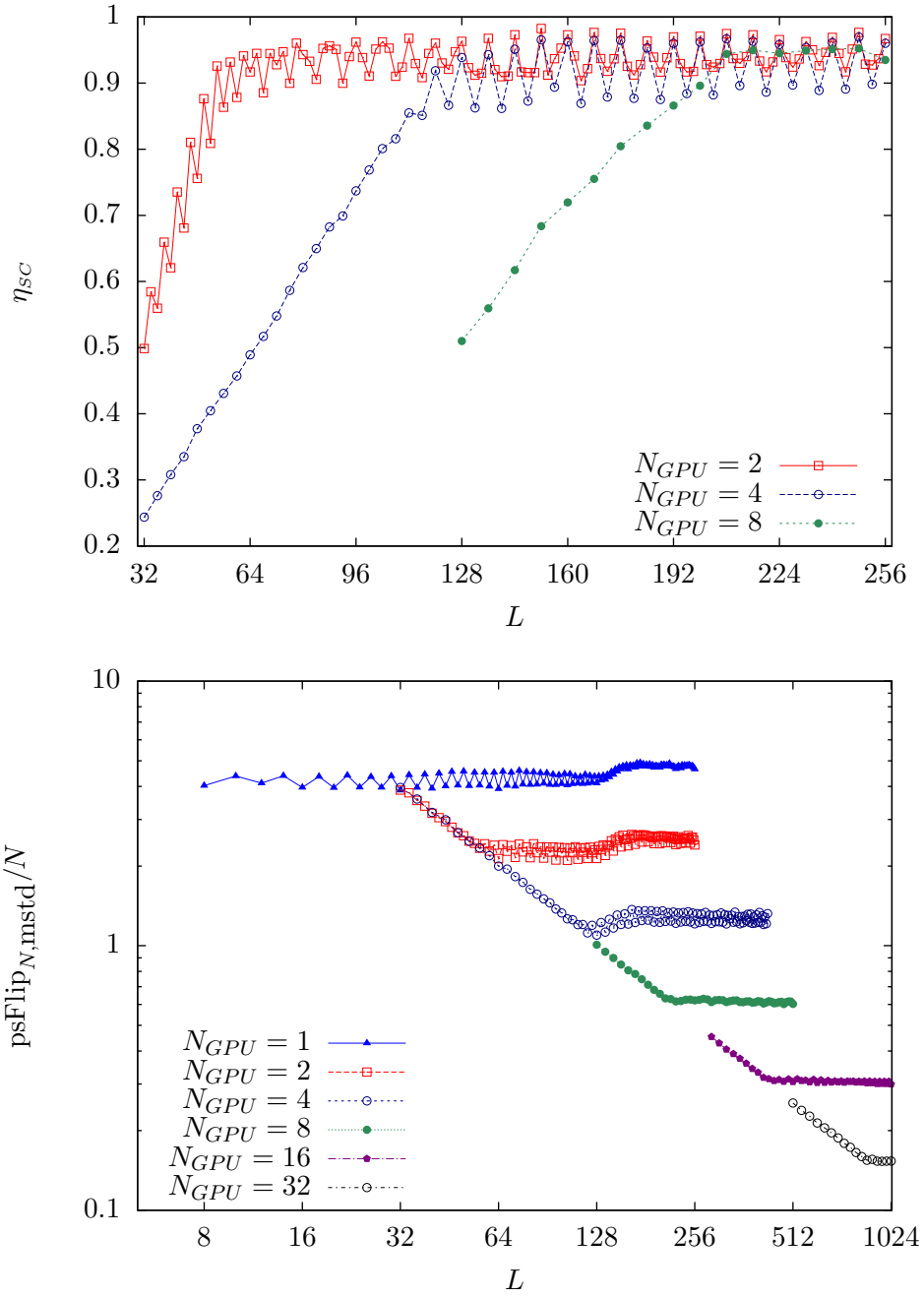


Figure 3.12. Top panel: the strong-scaling efficiency η_{SC} is reported for different numbers of GPUs. Bottom panel: the multi-GPU system performances are shown. A power-law behaviour as $\mathrm{psFlip}_{\mathrm{multi,mstd}} \sim L^{-1}$ is noticeable before saturation is reached for $N \leq 16$.

3.6 Conclusions

We have studied different strategies for the implementation of the Metropolis dissipative dynamics for the three-dimensional Edwards-Anderson bimodal spin glass.

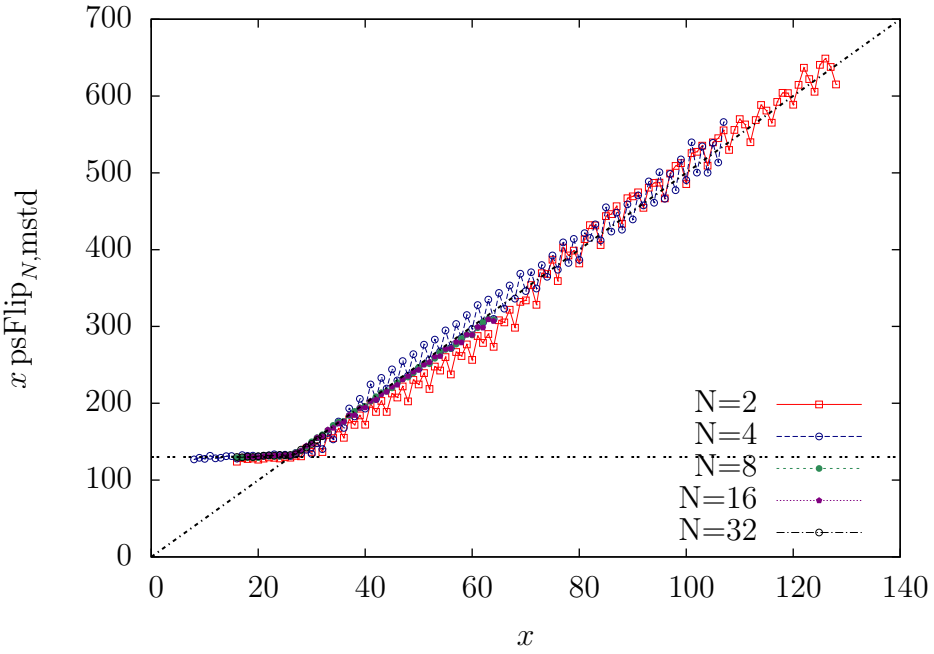


Figure 3.13. Scaling plot for the performances of the multi-GPU system. The initial constant value indicates a scaling for the sweep wall-clock time as L^2, while for the linear growth in x the sweep time scales as L^3 signaling a cross-over from a communication dominated to a bulk calculations dominated regime. The data collapse is possible for the high quality of the inter-node communication.

We proposed new access patterns for both the cubic stencil data structure and lagged-Fibonacci-like PRNGs. We showed, comparing different GPUs and different algorithm implementations that it is possible to obtain stable performances on a wide range of lattice sizes, $8 \leq L \leq 256$. For some GPUs our new **sliced** scheme performs slightly better than the other schemes, for the version which uses the MINSTD PRNG. However, the sliced scheme performs always better for large sizes L and when the data transfer load is increased using more complex PRNGs. In particular the sliced scheme gains roughly the 30% over standard implementation for the cuRand XORWOW. As for the comparison of different PRNGs, we showed that our implementation of the full Mersenne-Twister MT19937 performs better than the standard cuRand XORWOW thus indicating a new implementation strategy for PRNGs which turns out to be very efficient for memory bandwidth demanding algorithms. Indeed, the MT19937 performs only 70% worse than the MINSTD congruential PRNG with our approach.

Of course at the basis of such results there is the possibility of using the asynchronous multispin-coding (AMSC) technique which allows us to store one spin of 32 different systems in a word.

In terms of single GPU we showed that it is possible to obtain performances comparable to those of dedicated FPGA hardware [79] although one should be careful in this respect. However, single GPU performances are enough to obtain competitive results for critical parameters estimations using the out-of-equilibrium

relaxation regime as we shall see in the next chapters.

Furthermore, we explored the multi-GPU version of the sliced scheme which presents the intriguing feature of halving the number of MPI data transactions while, obviously, keeping the total amount of data transfer fixed. We showed that a very high strong-scaling efficiency can be reached leading to scientifically interesting performances in the range $64 \leq L \leq 128$.

Many of these result can be extended and reused outside the statistical mechanics domain since they involve cubic lattice discretization, along with their multi-GPU extension, and high quality random numbers PRNGs implementations.

Chapter 4

Out-of-equilibrium finite-size scaling

The aim of this chapter is the presentation of our new out-of-equilibrium finite-size scaling technique allowing us to measure the critical temperature, the dynamic and all the static critical exponents exploiting the relaxation dynamics of a system starting from a random, *i.e.* infinite temperature, initial condition.

To do so, we first review the past literature for out-of-equilibrium techniques which might be applied to the three-dimensional Edwards-Anderson model (EA3D), or to other systems, in order to have an idea of the frame in which our work is inserted. Then, we will explain our approach in the framework of the finite-size and finite-time scaling hypotheses in Chapter 1.

4.1 Previous techniques

The quest for a reliable off-equilibrium technique for measuring the critical temperature and the critical exponents characterizing a second order phase-transition has been the subject of many efforts in the last thirty years. For homogeneous systems undergoing such phase transitions, standard equilibrium finite-size scaling turns out to be a robust method, as for the determination of the critical parameters and the of the finite-size corrections.

However, for many inhomogeneous systems, such as spin glasses, disorder and frustration render the thermalization of large-size systems ano almost impossible task with the computational resources available today. These difficulties are also enhanced near the critical temperature where the critical slowing down of the dynamics occurs. In many cases the very existence of a phase-transition is an open issue mainly because of the difficulties in applying the equilibrium finite-size scaling framework to such systems.

It is possible to follow a two-fold approach in order to overcome these problems:

- one can use commercial high-end (*e.g.*, GPUs) or purpose-driven (*e.g.*, FPGAs [79, 29]) hardware in order to obtain the fastest MC simulations possible with the present resources;
- alongside, one can look for off-equilibrium techniques in order to obtain data

from system of larger sizes with respect to those accessible in equilibrium simulations.

Scaling laws controlling the critical relaxation of a system undergoing a second-order phase transition are a well-known subject [24, 84] which is under good control. As we have shown in Chapter 1 the critical behaviour is governed by the static critical exponents along with a new dynamic exponent z which is the same for different dynamics in the same dynamic universality class.

Most of the works we analyze below are consider the purely relaxational, or type-A [24], dynamics, which can be implemented using the Metropolis Monte Carlo algorithm. Although there are many different approaches, they all rely on two scaling ansatzes. First the divergence of the relaxation time τ as a function of the correlation length ξ scales as

$$\tau \sim \xi^z. \tag{4.1}$$

This relation is also used in order to probe the very existence of a phase transition in experimental works where the equilibration time of the system exceeds the experimentally accessible time range. Since the correlation length scales near the critical point as $\xi \sim |\varepsilon|^{-\nu}$, we can write

$$\tau \sim |\varepsilon|^{-z\nu} = \frac{1}{\beta_c}|\beta - \beta_c|^{-z\nu}. \tag{4.2}$$

This relation is usually fitted to experimental, or simulation, data in order to get estimations of β_c and of the product $z\nu$. Such a procedure is well-suited for a direct comparison of experimental and simulation data. However, in (4.2) we are discarding corrections to scaling which might give a significant contribution.

The second scaling relation is the finite-time scaling one [85, 86] which we report here for a generic observable $\mathcal{S}$ to first order in the first irrelevant scaling field

$$\mathcal{S} = t^{\lambda/z\nu} S_t(\varepsilon L^{1/\nu}, tL^{-z}) \left[1 + \frac{u_{i_1}}{t^{\omega_1/z}} s_{1t}(\varepsilon L^{1/\nu}, tL^{-z})\right], \tag{4.3}$$

where λ is some critical exponent. The case $\lambda = 0$ applies to a RG invariant observable. One of the possible uses of this scaling ansatz is the following. The thermodynamic limit is defined as $tL^{-z} \to 0$, hence it can be reached either by letting L grow indefinitely or by probing 'small' times such that $tL^{-z} \ll 1$. Indeed, one always need to keep in mind that such regime is only valid if the correlation length ξ is smaller than the lattice size L while being sufficiently larger than the lattice spacing a

$$a \ll \xi(t) \ll L. \tag{4.4}$$

This request is necessary in order to avoid strong corrections to scaling which would be present for $\xi \sim a$, and finite-size effects present for $\xi \lesssim L$.

In the infinite-volume regime one expects the values of $\mathcal{S}\, t^{-\lambda/z\nu}$, as a function of tL^{-z}, to be constant. However, the presence of z as an unknown parameter greatly complicates the scaling analysis introducing another free parameter.

As a consequence, one of the most common approaches consists in fitting simulation data to some effective time power law and then study the behaviour of the measured effective exponents. This is one of the most common approach in the previous works as we illustrate below.

We will now review the past literature on the subject.

4.1.1 Equilibrium dynamics

The paper by Ogielski [87] in 1985, is one of the first works we are aware of in which the dynamic behaviour has been studied in order to measure both the critical temperature and the product of the critical exponents $z\nu$. In this paper the simulated lattice sizes are $L = 8, 16, 32$ and 64. The number of samples is $N_{samp} = 64$ for $L = 8$, $N_{samp} = 32$ for $L = 16$, $N_{samp} = 2$ for $L = 32$ and $N_{samp} = 1$ for $L = 64$. The simulations were performed at equilibrium, studying the dynamic behaviour of the correlation function $q(t)$, defined as

$$q(t) = \frac{1}{V} \sum_x \left[\langle \sigma_x(0) \sigma_x(t) \rangle \right]. \tag{4.5}$$

The angular brackets denote the thermal averages and the square ones the average over different disorder realizations. Starting from a continuous version of the master equation for the evolution of the probability distribution and its Fourier transform

$$\frac{\partial P(\sigma, t)}{\partial t} = \sum_{\{\sigma_i'\}} \Gamma(\sigma|\sigma') P(\sigma', t), \qquad -\omega \tilde{P}(\sigma', \omega) = \sum_{\{\sigma_i'\}} \Gamma(\sigma|\sigma') \tilde{P}(\sigma', \omega), \tag{4.6}$$

it follows that, in the eigenvectors basis, the single-spin correlation function reads

$$\langle \sigma_x(0) \sigma_x(t) \rangle = \sum_\omega |\langle \sigma_x \tilde{P}(\sigma, \omega) \rangle|^2 \exp(-\omega t). \tag{4.7}$$

The different eigenvalues ω are related to different relaxation times as $\tau = 1/\omega$. In order to determine the longest relaxation time τ at a given temperature, equation (4.5) can be equivalently written as

$$q(t) = \int_0^\infty d\tau \rho(\tau) \exp(-t/\tau), \quad \int_0^\infty d\tau \rho(\tau) = 1, \tag{4.8}$$

where $\rho(\tau)$ is interpreted as the correlation coefficient averaged over the disorder distribution. In the case of a finite gap between the lowest eigenvalue ω_0 and zero, integral (4.8) is dominated by the largest-time contribution. On the other hand, if the eigenvalues accumulate at $\omega = 0$ then one must deal with the entire distribution of the relaxation times $\rho(\tau)$. The moments of this distribution are obviously related to the autocorrelation function (4.5) as

$$\int_0^\infty d\tau \, \tau^{k+1} \rho(\tau) = \frac{1}{k!} \int_0^\infty dt \, t^k \, q(t), \tag{4.9}$$

so that one defines the *average* relaxation time τ_{av} as the first moment corresponding to the time integral of $q(t)$. If the dynamic scaling hypotheses holds, then from standard scaling arguments it follows that near the critical point the autocorrelation function should be described in terms of a scaling function (neglecting finite-size corrections) as

$$\begin{aligned} q(t) \simeq & t^{-\beta/z\nu} f(\varepsilon t^{1/z\nu}) = t^{-x} Q(t\varepsilon^{z\nu}) = t^{-x} Q(t/\tau), \\ x = & \frac{\beta}{z\nu} = \frac{1}{2z}(d - 2 + \eta), \\ \tau = & \varepsilon^{-z\nu} = (\beta - \beta_c)^{-z\nu}, \end{aligned} \tag{4.10}$$

where one considers the dependence on $tL^{-z} \ll 1$ as negligible. Given this scaling form it is possible to obtain an estimate of τ observing that

$$\tau_{av} = \int_0^\infty dt\, q(t) = \int_0^\infty dt\, t^{-x} Q(t/\tau) = \tau^{1-x} \int_0^\infty dy\, y^{-x} Q(y), \quad y = t/\tau, \tag{4.11}$$

hence, it is possible to write

$$\tau = \frac{\int_0^\infty dt\, t\, q(t)}{\int_0^\infty dt\, q(t)}. \tag{4.12}$$

Another consequence is that, if the dynamic scaling hypothesis holds, since by Eq. (4.11) one has $\tau_{av} \propto \tau^{1-x}$, then also τ_{av} diverges near the critical temperature but with a different exponent z_{av} given by

$$z_{av} = z(1-x) = \frac{2z - d + 2 - \eta}{2}. \tag{4.13}$$

Combining these different definitions one defines a consistency condition linking z to other static exponents. In [87] the author fitted the estimates of $\tau_{av}(T)$ and $\tau(T)$ at different temperatures to

$$\tau_{av} = a\varepsilon^{-z_{av}\nu}, \qquad \tau = b\varepsilon^{-z\nu}. \tag{4.14}$$

Requiring optimal scaling for both quantities with a consistent set of the critical parameters $\{\beta_c, z, \nu, \eta\}$, and using the equilibrium value of $\eta = -0.22(5)$ and $\nu = 1.3(1)$ from [88] and [38], the outcome of the measures was: $T_c = 1.175(25)$, $\beta_c = 0.851(21)$, $z = 6.1(3)$ and $z_{av} = 5.4(2)$.

4.1.2 Equilibrium and off-equilibrium mixing

Stemming from Ogielski's paper there is a series of works [89, 90, 91, 92] which aim at studying the universality of the spin-glass transition for various three-dimensional models specified by different probability distributions for the coupling constants: typically, the comparison is made using the bimodal, Gaussian and Laplacian distributions. The proposed method uses out-of-equilibrium relaxation measures together with static measures in order to estimate the critical parameters. Indeed, it is closely related to that of [87] and the main difference is that the latter relies on equilibrium dynamics. The out-of-equilibrium relaxation of some observables is defined as its dynamic evolution at a given finite value of β, starting from an infinite-temperature configuration of the system. This choice gives a sizeable gain over the previous method since the dynamic measures are performed completely out of equilibrium.

The proposed method works as follows. The out-of-equilibrium relaxation of $q(t)$, which is the same as defined in (4.5), has the functional form of eq. (4.10) and at the critical point one has

$$q(t) = \lambda t^{-x} \tag{4.15}$$

where λ is a size-independent constant and we have assumed that $L \gg \xi(t)$ so that finite-size effects are negligible. It is also possible to measure the out-of-equilibrium susceptibility of the overlap order parameter χ which at the critical point scales as

$$\chi(t) = \frac{1}{V}\left[\left\langle \sum_x \sigma_x^a(t)\sigma_x^b(t)\right\rangle^2\right] \sim t^{(2-\eta)/z} = t^h, \tag{4.16}$$

where the Latin superscripts denote different replicas, *i.e.*, systems sharing the same disorder realization but with different dynamic histories starting from different initial conditions.

These scaling relations strictly hold only at the critical temperature but one can use them also in an off-critical regime obtaining temperature dependent exponents, $x(\beta)$ and $h(\beta)$. Given this pair of effective exponents, it is possible to obtain another pair of effective exponents, $z(\beta)$ and $\eta_1(\beta)$, which are defined as

$$\eta_1(\beta) = \frac{4x(\beta) - h(\beta)(d-2)}{2x(\beta) + h(\beta)}, \qquad z(\beta) = \frac{d}{2x(\beta) + h(\beta)}. \tag{4.17}$$

At the critical point the equilibrium susceptibility scales as

$$\chi_{SG} = L^{2-\eta}\mu, \tag{4.18}$$

where μ is a size-independent quantity so that from equilibrium simulations it is possible to obtain an independent estimate of η. Again, one extends this scaling form to the off-critical region defining a second effective exponent $\eta_2(\beta)$. Now, the general idea is to match, using two different combinations, the static effective exponents with those determined in the out-of-equilibrium regime. Since the scaling forms are exact only at the critical temperature, a crossing point, signaling the critical temperature, is expected.

In [89, 90] the authors decided to match directly the effective exponents $\eta_1(T)$ and $\eta_2(T)$. In [89] the data of [87] and [93] were used to validate the method obtaining: $T_c = 1.17(1)$, $z = 6.0(2)$ and $\eta = -0.25(2)$. In [90] the authors combined their data from out-of-equilibrium simulations with those of [87] and [94] obtaining new estimations: $T_c = 1.20(1)$ and $\eta = -0.21(2)$. The critical temperature is compatible in three standard deviations with the previous estimate.

In [91] the authors matched to $h(\beta)$ a new exponent $h^*(\beta)$

$$h^*(\beta) = \frac{2x(\beta)(2 - \eta_2(\beta))}{d - 2 + \eta_2(\beta)} \tag{4.19}$$

obtained by using the equilibrium estimates of $\eta_2(\beta)$ and the off-equilibrium estimates of $x(\beta)$, used to eliminate the $z(\beta)$ dependence. In this work the authors allow for the presence of finite-time scaling corrections writing for the overlap susceptibility

$$\chi = At^h(1 - Bt^{-\omega/z}) \tag{4.20}$$

For the out-of-equilibrium measures the authors used data coming from lattices of sizes $L = 20$ up to $L = 28$. At $T = 1.2$ for $L = 28$ the number of the disorder realizations is $N_{samp} = 6766$. For the equilibrium measures data were taken from [94]. The analysis at equilibrium was performed introducing an effective finite-size correction exponent ω. The crossing of $h(\beta)$ and $h^*(\beta)$ led to the results: $T_c = 1.195(15)$, $\omega = 2.9(6)$ and $z = 5.65(15)$.

Finally, in [92] the authors introduce a two-times version of $q(t)$, $q(s,t)$ defined as

$$q(s,t) = \frac{1}{V}\sum_x \left[\langle \sigma_x(s)\sigma_x(t)\rangle\right], \tag{4.21}$$

Defining $\tau = t - s$ with $t > s$ they write the critical scaling relation and its asymptotic form in the limit $s \gg \tau$ for a fixed value of the linear size L

$$C(t,s) = s^{-x} f_c(t/s), \qquad C(t,s) \sim \tau^{-x}. \tag{4.22}$$

In the paper the authors verify the scaling at finite values of the ratio t/s and extrapolate the value of x. However, the data for $x(\beta)$ are taken from the quasi-equilibrium regime $s \gg \tau$. As for the other work the equilibrium and off-equilibrium overlap susceptibilities are used in order to determine the effective exponents $h(\beta)$ and $\eta_2(\beta)$, which, together with $x(\beta)$, are used to extrapolate two different effective dynamic exponents $z(\beta)$ and $z^+(\beta)$. The crossing point again signals the critical temperature. The out-of-equilibrium simulations were performed on a system of linear size $L = 50$ but the number of samples was not specified, nor the sizes and the number of samples of the equilibrium simulations. No corrections to scaling terms were introduced. The estimated critical parameters were: $T_c = 1.19(1)$, $\eta = -0.22(2)$ and $z = 5.7(2)$. The range of waiting times is $0 < t_w \leq 1600$.

4.1.3 Off-equilibrium response function

In [95] the author proposes a technique which only uses out-of-equilibrium measures on a system relaxing at a finite temperature starting from an infinite-temperature configuration. This work can be ssen as an improvement of the scheme proposed in [89, 90, 91, 92], we just reviewed in the previous subsection, where equilibrium and out-of-equilibrium data were needed.

In [95], the author performs a critical review of previous results stressing how important is the choice of the minimum waiting time s in the estimation of the effective exponents. The author uses the two-time correlation function and the integrated response function, obtained using an external field applied up to a waiting time s. The critical scaling form of the response function reads

$$\rho(t,s) = \frac{1}{\beta}\int_0^s du R(t,u) = \frac{1}{N\beta}\int_0^s du \sum_{i=1}^{N}\left[\frac{\delta\langle\sigma_i(t)\rangle}{\delta h_i(u)}\right] = s^{-a} f_\rho(t/s). \tag{4.23}$$

Now, at the critical point, one expects the fluctuation dissipation theorem to hold, implying the equality $x = a$. Again, one measures the effective exponents for different values of β. The value minimizing the (squared) difference between the two measures is taken to be the critical temperature.

The main difference with respect to the previously discussed works is that the exponents are measured from the finite-time behaviour of the correlation and response functions. No collapsing of the exponents is noticed for small values of s. The estimated value of the critical temperature for the bimodal case is $T_c = 1.135(5)$. The simulated lattices have linear size $L = 50$ with a number of samples varying from $N_{samp} = 3000$ to $N_{samp} = 5000$ with waiting times up to $t_w = 1600$, which is the same as in [92]. The same method has recently been applied in [96].

4.1.4 Non-equilibrium relaxation

In the paper [97] the authors propose the so-called non-equilibrium relaxation method (NER) in order to estimate the critical temperature from out-of-equilibrium

simulations. A review of this method can be found in [23]. The authors choose to study the dynamic relaxation of the clone correlation function (CCF) which is defined as follows. Take an all-up state at the beginning of the simulation, then after a waiting time t_w duplicate the configuration and let the two replicas $\{\sigma_i^{(1)}\}$ and $\{\sigma_i^{(2)}\}$ evolve independently using different random number sequences, then the clone correlation function reads

$$Q(t, t_w) = \frac{1}{N} \sum_{i=1}^{N} \left[\langle \sigma_i^{(1)}(t + t_w) \sigma_i^{(2)}(t + t_w) \rangle \right]. \tag{4.24}$$

Since the clones start from the same configuration, we have $Q(0, t_w) = 1$ for any value of t_w. The $Q(t, t_w)$ decays towards the equilibrium overlap value as $t \to \infty$. At the critical temperature, and in the thermodynamic limit, the relaxation of the order parameter is described by a power-law decay, hence it is possible to distinguish the phases measuring the local exponent defined as

$$\lambda(t, t_w) = -\frac{d \log Q(t, t_w)}{d \log t}. \tag{4.25}$$

As a function of the ratio t_w/t the local exponent diverges in the paramagnetic phase as $t_w/t \to 0$, since the decay of $Q(t, t_w)$ is exponential in this limit. In the same limit at the critical temperature it is supposed to go to a constant value. Hence, at least it is possible to determine an upper bound to the critical temperature by observing the divergence of the local exponent in the time range of simulations.

As for the lower bound the authors found that the local exponent still converges to a constant even in the spin-glass phase. Hence, it does not give enough information. So they turned to the dynamic scaling form expected at the critical point for the clone correlation function

$$Q(t, t_w) = t_w^{-\lambda_q} f_Q(t/t_w), \tag{4.26}$$

and found out that in the spin-glass phase there is no scaling for the considered range of t/t_w while at the candidate critical temperature $T_c = 1.2$ scaling holds good. The last temperature at which the scaling is observed is $T = 1.1$.

Using this criterion the authors estimate the critical temperature in the range $1.00 < T_c < 1.25$. As the authors noticed, even though the uncertainty is bigger than those of other works, at least the uncertainty sources are under control.

The simulations were performed on cubic lattices with skew boundary conditions. The considered lattices sizes range from $V = 29^2 \times 30$ up to $V = 127^2 \times 128$ and the temperatures belong to the interval $0.8 \leq T \leq 2.2$. The largest times were $t_w = 10^8$ and $t = 2 \times 10^8$. The number of samples was not specified.

4.1.5 Extrapolation techniques

In [98] the authors analyze the out-of-equilibrium behaviour of the spin-spin correlation function which is defined as

$$C(t, t_w) = \frac{1}{V} \sum_{i=1}^{V} \left[\langle \sigma_i(t) \sigma_i(t_w) \rangle \right]. \tag{4.27}$$

Using the ansatz given in [99, 100]

$$C(t,t_w) = a(t) + b(t)t_w^{-c(t)}, \qquad \lim_{t\to\infty}\lim_{t_w\to\infty} C(t,t_w) = \lim_{t\to\infty} a(t) = q_{EA}. \tag{4.28}$$

the numerical data for $C(t,t_w)$ were fitted to $a(t) + b(t)t_w^{-c(t)}$. Then $a(t)$ was fitted to a constant function for large times obtaining $q_{EA}(\beta)$. The critical behaviour of the order parameter is described by the power-law relation

$$q_{EA}(\beta) = A(\beta - \beta_c)^{\beta_q}, \tag{4.29}$$

where β_q is the critical exponent. The results were: $\beta_c = 0.866(2)$, or equivalently $T_c = 1.155(3)$, and $\beta_q = 0.52(9)$.

Simulations involved two different lattice sizes, and two different series of temperatures and $N_{samp} = 58$ for each of them. For $L = 30$ the simulated temperatures are $\beta = 2.00, 1.67, 1.25, 1.05, 1.00, 0.95, 0.91$; for $L = 60$ the simulated temperatures are $\beta = 2.00, 1.67, 1.25, 1.00$. Hence, the estimated critical temperature lies outside the simulation temperature range.

4.1.6 Dimensionless ratios

Even though the method developed in [101] is not applied to the 3D Edwards-Anderson model it is useful to review it. In the hypothesis of an infinite-size dynamic behaviour, *i.e.* before finite-size effects appear, it is possible to define a dimensionless ratio

$$X(\tau,\beta) = \frac{1}{\tau}\frac{\int_0^\tau dt\, t\, C(t,\beta)}{\int_0^\tau dt\, C(t,\beta)}, \qquad C(t,T) = \langle s_i(t)s_i(0)\rangle, \tag{4.30}$$

where t is the duration of a finite-temperature dynamics starting from an infinite temperature initial state and τ is a fixed time value. The infinite temperature initial configuration justifies the form of the correlation function $C(t,T)$.This ratio is similar to the one used by Ogielski in [87] to define the relaxation time. This ratio should be independent from the choice of τ only at the critical temperature. Hence, for a given lattice size, the crossing of the curves at different β and different τ should signal the phase transition.

4.1.7 Out-of-equilibrium relaxation time

In [102] the authors used the non-equilibrium relaxation method proposed in [97], and studied the dynamic behaviour of the overlap susceptibility which at the critical point and in the thermodynamic limit, scales as $\chi(t) \sim t^{\gamma/z\nu}$.

In order to determine the critical temperature, one defines a relaxation time as a function of the temperature $\tau(\beta)$ which is obtained by collapsing the data of $\chi_{SG}(t)\, t^{-\gamma/z\nu}$ against $t/\tau(\beta)$ for all the simulated temperatures at once. Once $\tau(\beta)$ is measured, the critical temperature and the product of critical exponents $z\nu$ is determined by fitting the data to

$$\tau(\beta) \sim (\beta - \beta_c)^{-z\nu}. \tag{4.31}$$

Then, it is possible to determine the value of γ. The author stresses that it is important to have a large number of replicas for the thermal average. The simulated

lattice size is $L = 49$ and the number of samples is $N_{samp} = 393$ for simulation times up $t = 10^5$, and $N_{samp} = 88$ for times up to $t = 4 \times 10^6$. The number of replicas is not specified. The results are $T_c = 1.17(4)$, $\gamma = 3.6(6)$, $\nu = 1.5(3)$ and $z = 6.2(2)$.

4.1.8 Out-of-equilibrium scaling

In [103] the author introduces a new scaling method which stands on a different footing with respect to the works we reviewed so far. For the first time the finite-size scaling relations are applied to out-of-equilibrium relaxation data. In particular, this method avoids the estimation of the critical exponent z in the determination of the critical temperature.

The author starts by introducing a different definition of correlation length. Since in the relaxation dynamics one has $\xi \ll L$, the three-dimensional correlation function should not decay as

$$G(r) \propto \frac{\exp(-r/\xi)}{r}, \tag{4.32}$$

but rather as $\exp(-r/\xi)$ without the algebraic term r^{-1}, whose presence is expected from mean-field equilibrium arguments. Then, a new definition for a correlation length is given

$$\xi_N = \frac{1}{k_m} \sqrt{\sqrt{\frac{\chi}{\tilde{\chi}(k)}} - 1}, \tag{4.33}$$

which differs from the usual second-moment definition

$$\xi_2 = \frac{1}{2\sin(k_m/2)} \sqrt{\frac{\chi}{\tilde{\chi}(k)} - 1}, \tag{4.34}$$

where $k = (k_m, 0, 0)$, $k_m = 2\pi/L$, and $\tilde{\chi}$ is the Fourier transform of the overlap susceptibility χ. A scaling analysis is presented in order to choose which definition to use. For the three-dimensional Ising model the author claims that the data of ξ/L, with ξ given by (4.34), scale only if an incorrect value for z is used. Hence the author's choice is adopted.

Three different scaling procedures are presented:

- The determination of the exponent z is achieved by the best data collapse of ξ/L as a function of $t^{1/z}/L$. Here it is not discussed whether being at the critical temperature is a necessary condition.

- The anomalous dimension η is obtained considering that since both susceptibility and correlation length grow algebraically near the critical point, one has $\chi \sim \xi^{2-\eta}$, hence $\chi L^{-(2-\eta)} \sim (\xi/L)^{2-\eta}$. Therefore, in a log-log plot one expects a collapsing of data on a straight line of slope $2 - \eta$. The choice of the right critical temperature is said not to be so important, and that any temperature in the critical zone would allow a good estimate of the critical exponent. However, because of the study reported in [39] we know that this is not true because of the severe analytic corrections of the susceptibility.

- The determination of β_c and ν is carried out in an independent way with respect to the previous estimates. Once the data, free from finite-size effects,

are chosen with an unexplained procedure, the ratio $\chi(t,T)/\xi^{2-\eta}$ is plotted versus $\xi(t,T)/|T-T_c|^{-\nu}$ and the critical parameters are estimated by the best collapse onto a single curve. Not many technical details are reported. He used $N_{rep}=256$ replicas for a given disorder realization and considered temperatures from $T=1.60$ to $T=1.00$ with $\Delta T=0.05$. Additionally, he considered $T=0.9$.

The results were: $T_c=1.18(1)$, $\nu=1.40(5)$ and $\eta=-0.20(1)$.

Finally, in [104], the authors applied the same technique to the so-called window measurements: only a bulk portion of the system is considered and the various observables are shown to be rather insensitive to the chosen boundary conditions.

4.2 Similarities and common problems

The methods we listed above, with the exception of the ratio and dynamic out-of-equilibrium scaling methods [101, 103], share the general strategy of fitting values which are the result of previous fits. This scheme leads to not well controlled estimates. This lack of control is also due to the absence of finite-size, or finite-time, scaling corrections in most of the works which rely on the hypothesis of the thermodynamic-limit behaviour. Because of this hypothesis one considers the dependence of the scaling functions on the argument tL^{-z}, as a weak dependence. A careful explanation of the method used to check the validity of such an hypothesis is always neglected. The results of the various works reported in table 4.1.

Method		T_c	ν	η	ω
Equilibrium dynamics	[87]	1.175(25)			
Equilibrium and off-equilibrium mixing	[89]	1.17(1)		−0.25(2)	
	[90]	1.20(1)		−0.21(2)	
	[91]	1.195(15)			2.9(6)
	[92]	1.19(1)		−0.22(2)	
Off-equilibrium response function	[95]	1.135(5)			
Non-equilibrium relaxation	[97]	$1.12(12)_{sys}$			
Extrapolation technique	[98]	1.155(3)	$\beta=0.52(9)$		
Off-equilibrium relaxation time	[102]	1.17(4)	1.5(3)	$\gamma=3.6(6)$	
Off-equilibrium scaling	[103]	1.18(1)	1.40(5)	−0.20(1)	
Equilibrium FSS	[39]	1.109(10)	2.45(15)	−0.375(10)	1.0(1)
	[31]	1.1019(29)	2.562(42)	−0.3900(36)	1.12(10)

Table 4.1. Results for critical parameters for various off-equilibrium techniques and most recent equilibrium finite-size scaling (FSS) estimates [39, 31]. The exponents β and γ are related to ν and η by $\beta=\nu(1+\eta)/2$, $\gamma=\nu(2-\eta)$.

It appears that in most of the cases the estimates of the critical temperature are much different from the most recent equilibrium values [39, 31], in a measure which is not explainable in terms of statistical error. Indeed finite-size corrections seem to be the most important source of error, together with uncontrolled corrections due to the failure of the thermodynamic-limit hypothesis. Indeed, we will ses that such a regime is fully attained only for very large system sizes for a reasonably long time-window. Most of the system sizes used in these works are too small to allow an infinite-size analysis.

Only four of the analyzed works [87, 97, 102, 103] give compatible results for the critical temperature with respect to equilibrium estimates, although with much bigger errors. However, in both [102, 103] the result for ν is compatible within three error-bars, while that for the anomalous dimension η [103] is compatible but not very accurate.

Finally, in table 4.2 we report the various estimates of the dynamic exponent z together with the largest value of the waiting time reached in simulations. The different estimates are compatible.

Method		z	t_w^{MAX}
Equilibrium dynamics	[87]	6.1(3)	
Equilibrium and off-equilibrium mixing	[89]	6.0	
	[90]		
	[91]	5.65(15)	
	[92]	5.7(2)	1600
Off-equilibrium response function	[95]		1600
Non-equilibrium relaxation	[97]		10^8
Extrapolation technique	[98]		$3.7 \cdot 10^8$
Off-equilibrium relaxation time	[102]	6.2(2)	

Table 4.2. Different estimates of the dynamic exponent z and values of the max waiting time t_w^{MAX}.

4.3 Out-of-equilibrium finite-size scaling

Let us now present our novel technique which differs almost completely from the previous works with the exception of [103] with which some similarities can be found: a common feature is the bypassing of the exponent z as for the measure of the critical temperature and critical exponents and the use of the finite-size scaling relation in the out of equilibrium regime. Our approach heavily relies on the dynamic finite-size scaling ansatz [85, 86] of which we gave an explanation in Chapter 1 and the main feature is to substitute the time dependence of the various observables with

a dependence on an RG invariant quantity which exhibits a monotonous growth in time. In this setting it is still possible to use the entire finite-size scaling framework which, in particular, allows for the control of finite-size corrections which we do not claim to be negligible. Indeed, we will show how these corrections are most important in the very beginning of the relaxation dynamics even in the framework of the usual non-equilibrium relaxation scheme [102, 97, 23]: indeed the thermodynamic-limit behaviour is reached only after a certain time and for very large systems.

Another important difference with respect to the previous works lies in the demonstration that a huge number of samples is needed in order to extract the right behaviour while the number of replicas mainly depends on the kind of observables one wants to measure. Indeed to get a competitive estimate we needed to simulate a number of samples compatible with, and sometimes even greater than, that of [31]. Hence, the possibility of simulating large system sizes with this kind of technique seems to be ruled out with the actual GPU computational resources for what concerns the relaxation dynamics of the three-dimensional Edwards-Anderson model.

Let us now define our scaling ansatz. The usual dynamic finite-size scaling hypothesis states that for any given pair, A and B, of RG invariant quantities near the critical point one can write

$$\begin{aligned} A(t, L, \beta) &\simeq \mathcal{A}\left(tL^{-z}, \varepsilon L^{1/\nu}\right) + \frac{u(\beta)}{L^{\omega}} \mathcal{A}_{\omega}\left(tL^{-z}, \varepsilon L^{1/\nu}\right), \\ B(t, L, \beta) &\simeq \mathcal{B}\left(tL^{-z}, \varepsilon L^{1/\nu}\right) + \frac{u(\beta)}{L^{\omega}} \mathcal{B}_{\omega}\left(tL^{-z}, \varepsilon L^{1/\nu}\right), \end{aligned} \tag{4.35}$$

where $\varepsilon = \beta - \beta_c$, ω is the exponent of the first corrections to scaling and $u(\beta)$ is the associated non-linear scaling function. It is then possible, for a monotonic dynamic behaviour and at a given temperature, to write the time dependence tL^{-z} in terms of one of the two observables which we choose to be B

$$tL^{-z} \simeq \mathcal{T}\left(B, \varepsilon L^{1/\nu}\right) + \frac{u(\beta)}{L^{\omega}} \mathcal{T}_{\omega}\left(B, \varepsilon L^{1/\nu}\right), \tag{4.36}$$

so that our scaling ansatz directly follows

$$A(t, L, \beta) \simeq \tilde{\mathcal{A}}\left(B, \varepsilon L^{1/\nu}\right) + \frac{u(\beta)}{L^{\omega}} \tilde{\mathcal{A}}_{\omega}\left(B, \varepsilon L^{1/\nu}\right). \tag{4.37}$$

As we mentioned above there is no need to take into account the presence of z in order to get estimates of the other critical parameters. Indeed using RG invariant observables it is possible with a single fit procedure to obtain three different parameters: β_c, ν and ω. In order to obtain the dependence of A in terms of B one just needs to associate the observables for a given value of the MC time t.

Let us analyze how to implement this procedure in practice. We can choose B as the RG invariant ratio $R_\xi = \xi/L$, and A as one of the possible Binder cumulants, say U_4. The scaling ansatz reads

$$U_4(t, L, \beta) \simeq \tilde{\mathcal{U}}_4\left(R_\xi, \varepsilon L^{1/\nu}\right) + \frac{u(\beta)}{L^{\omega}} \tilde{\mathcal{U}}_{4,\omega}\left(R_\xi, \varepsilon L^{1/\nu}\right), \tag{4.38}$$

which has, for a fixed value of R_ξ, the same functional form of the usual equilibrium finite-size scaling ansatz. Hence, for what concerns the estimate of the critical

temperature, our approach turns out to be an off-equilibrium generalization of the familiar Binder crossing criterion.

In the same way it is possible to obtain a scaling relation for the overlap susceptibility χ allowing us to measure also the anomalous dimension η. Thus we write

$$\chi(t, L, \beta) \simeq L^{2-\eta} v(\beta) \bar{\chi}\left(R_\xi(t), \varepsilon L^{1/\nu}\right) + \frac{v(\beta)u(\beta)}{L^\omega} \bar{\chi}_\omega\left(R_\xi(t), \varepsilon L^{1/\nu}\right), \qquad (4.39)$$

where the function $v(\beta)$ represents the analytic corrections to scaling deriving from the scaling field associated to an external magnetic field [39]. Usually one wants to have as few free parameters as possible so that the strategy we propose is to measure β_c, ω and ν from the scaling of the Binder cumulant in the first place, then it one substitutes these values in the susceptibility scaling in order to obtain an estimate of η. However, from a logical point of view it is possible to estimate all the critical parameters appearing the susceptibility scaling form (4.39) just with the data series of $R_\xi(t)$ and $\chi(t)$, but it would not be an efficient method from a numerical point of view.

We also considered the possibility to use the off-equilibrium generalization[1] of the temperature derivatives of some observables, in order to get different estimates of the critical exponent ν. In particular we computed the temperature derivatives of $U_4(t)$ and $\log \chi(t)$ which are expected to scale as $L^{1/\nu}$.

Finally, it is possible to estimate the value of z using the data collapse of RG invariant quantities once the value of the critical temperature is known. This is just a qualitative explanation of the analysis we performed. A more detailed account will be given in the next chapter.

[1] In the out-of-equilibrium regime it is not possible to give an exact definition of temperature, however it is possible to define observables which in the equilibrium limit would be equal to some temperature derivative.

Chapter 5

Out-of-equilibrium Results

In this chapter we summarize our results for the data analysis using our out-of-equilibrium finite-size scaling ansatz. We start by giving the definitions of the various observables we studied, using some care in describing how to generalize common equilibrium definitions to the out-of-equilibrium regime. We continue with a detailed discussion of the behaviour of RG invariant observables in order to establish a clear link between the static and the dynamic approach. We explain and verify the hypothesis we used in order to fit the data and obtain the estimates for the critical parameters β_c, ν, ω, η and z. We test our results showing different scaling behaviours and we especially analyze the infinite-volume limit approach. Finally, we draw some conclusions.

5.1 Observables Definitions

Here we define the set of observables which have been analyzed in this work. We always considered two-replicas observables which were constructed in order to get the usual equilibrium limit, hence we need to specify that the usual thermal average corresponds to an average over different thermal histories and not merely to an average over different pairs of replicas as it has been argued in [103]: averaging over different short thermal histories at equilibrium leads to the same results one would get by averaging over a single long equilibrium sequence. Practically speaking, one prepares for a given disorder realization a number r of replicas with different infinite-temperature (*i.e.* random) initial configurations which are evolved with a simple Metropolis dynamics at a given inverse temperature β with different random number sequences for each replica. From the evolution of the r replicas one can get $c_2 = r(r-1)/2$ different values of two-replicas observables and average all of them for each time step in a given time sequence. Since the thermal fluctuations due to the different initial states of each replica are negligible, with respect to those due to different disorder realizations, we do not need r to be a large number. In this thesis we are working with $r = 4$. Now, one need to sample the disorder distribution so that for given values of the lattice size L and inverse temperature β the procedure is repeated and the values coming form different disorder realizations are averaged for each time steps. This way we perform both the thermal average and the disorder average at once.

5.1.1 Basic observables

We study the critical behaviour of the overlap order parameter which, in the dynamic regime, is defined as

$$q(t) = \frac{1}{c_2 V} \sum_x \sum_{a<b} \left[\sigma_x^a(t)\sigma_x^b(t)\right] = \frac{1}{c_2 V} \sum_x \sum_{a<b} \left[q_x^{ab}(t)\right], \tag{5.1}$$

where the Latin indices stand for replica indices, the sum $\sum_{a<b}$ denotes that only different pairs of replicas are averaged and the square bracket stand for the average over the coupling distribution. The constant c_2 equals the number of different pairs of replicas: for each sample there are four replicas, then one has that $c_2 = 6$. Since our simulations are performed on lattices with periodic boundary conditions, we write the dynamic susceptibility and its Fourier transform as

$$\chi(t) = \frac{1}{c_2 V} \sum_{a<b} \left[\left(\sum_x q_x^{ab}(t)\right)^2\right], \qquad \tilde{\chi}(t,k) = \frac{1}{c_2 V} \sum_{a<b} \left[\left|\sum_x q_x^{ab}(t) e^{-ikx}\right|^2\right], \tag{5.2}$$

so that the out-of-equilibrium correlation length, or coherence length, is defined as for the static case

$$\xi(t) = \frac{1}{2\sin(k_m/2)} \sqrt{\frac{\chi(t)}{\tilde{\chi}(t,k)} - 1}, \qquad k_m = \frac{2\pi}{L}. \tag{5.3}$$

We calculate the Fourier transform taking alternatively the directions $k_1 = (k_m, 0, 0)$, $k_2 = (0, k_m, 0)$ and $k_3 = (0, 0, k_m)$ so that we need to sum the local overlaps q_x^{ab} only on the orthogonal planes taking the average of the results along the three directions because the system is isotropic. The susceptibility, together with the correlation length, is used to measure the anomalous dimension η.

We have also studied the dynamic behaviour of five Binder cumulants, or renormalized couplings, which are defined as follows

$$U_4(t) = \frac{\left[c_2^{-1} \sum_{a<b} \left(\sum_x q_x^{ab}(t)\right)^4\right]}{\left[c_2^{-1} \sum_{a<b} \left(\sum_x q_x^{ab}(t)\right)^2\right]^2}, \tag{5.4}$$

$$U_{22}(t) = \frac{\left[c_{22}^{-1} \sum_{a<b<c<d} \left(\sum_x q_x^{ab}(t)\right)^2 \left(\sum_y q_y^{cd}(t)\right)^2\right]}{\left[c_2^{-1} \sum_{a<b} \left(\sum_x q_x^{ab}(t)\right)^2\right]^2} - 1, \tag{5.5}$$

$$U_{23}(t) = \frac{\left[c_{23}^{-1} \sum_{a<b<c} \left(\sum_x q_x^{ab}(t)\right)^2 \left(\sum_y q_y^{ac}(t)\right)^2\right]}{\left[c_2^{-1} \sum_{a<b} \left(\sum_x q_x^{ab}(t)\right)^2\right]^2}, \tag{5.6}$$

$$U_{13}(t) = \frac{\left[c_{13}^{-1} \sum_{a<b<c} \left(\sum_x q_x^{ab}(t)\right) \left(\sum_y q_y^{ac}(t)\right) \left(\sum_z q_z^{bc}(t)\right)\right]}{\left[c_2^{-1} \sum_{a<b} \left(\sum_x q_x^{ab}(t)\right)^2\right]^{3/2}}, \tag{5.7}$$

$$U_{14}(t) = \frac{\left[c_{14}^{-1} \sum_{a<b<c<d} \left(\sum_x q_x^{ab}(t)\right) \left(\sum_y q_y^{bc}(t)\right) \left(\sum_z q_z^{cd}(t)\right) \left(\sum_w q_w^{da}(t)\right)\right]}{\left[c_2^{-1} \sum_{a<b} \left(\sum_x q_x^{ab}(t)\right)^2\right]^2}. \quad (5.8)$$

As before the constants c_i accounts for the multiplicity of the n-tuples of different replica pairs: with four replicas the values are, $c_{22} = 3$, $c_{23} = 12$, $c_{13} = 4$ and $c_{14} = 3$. These quantities have been used together with the correlation length in order to determine the critical temperature and the critical exponents ν and ω.

A remark is in order. There is a subtlety one should be aware of when writing the off-equilibrium generalization of some equilibrium observable. Let us consider the Binder cumulant U_{22} whose equilibrium definition reads

$$U_{22} = \frac{\left[\langle (\sum_x q_x^{12})^2 \rangle^2\right]}{\left[\langle (\sum_x q_x^{12})^2 \rangle\right]^2} - 1 \quad (5.9)$$

where the angular brackets denote the thermal equilibrium average. The difference between the definitions (5.5) and (5.9) lies in the fact that the square of an equilibrium average of a two-replicas observable, $\langle x^{12} \rangle^2$, can be obtained using two sufficiently decorrelated part of the same thermal history giving two independent estimates $\langle x^{12} \rangle_1$ and $\langle x^{12} \rangle_2$, so that one has $\langle x^{12} \rangle^2 = \langle x^{12} \rangle_1 \langle x^{12} \rangle_2$. This technique cannot be applied in the off-equilibrium generalization of this kind of observables. Since the thermal average is replaced by the equal time average of different thermal histories one needs more than two replicas in order to define an adequate generalization of the powers of two-replicas thermal averages which we write as

$$\langle x^{12} \rangle^2 \ \rightarrow x^{12}(t) \cdot x^{34}(t). \quad (5.10)$$

Hence, powers of equilibrium estimates of two-replicas observables correspond to products of equal-time values given by different pairs of replicas: for a generic power k, one needs to have k independent pairs of replicas. This argument naturally generalizes to observables involving an arbitrary number of replicas. In figure 5.1 we report a graphical approach for the determination of the terms needed for each Binder cumulant.

Finally, we remark that this kind of Binder cumulant has also been used in the recent work [31] of Janus collaboration. Comparing the equilibrium definition the relation between the observables reads

$$\begin{aligned} U_{4,J} &= U_4, \\ U_{22,J} &= (U_{22} + 1)/U_4, \\ U_{111,J} &= U_{13}^{4/3}/U_4, \\ U_{1111,J} &= U_{14}/U_4. \end{aligned} \quad (5.11)$$

Only the cumulant U_{23} has not been used so far.

5.1.2 Temperature derivatives

We have also studied two different temperature derivatives. Indeed, in an out-of-equilibrium context the concept of temperature derivative is not well-defined.

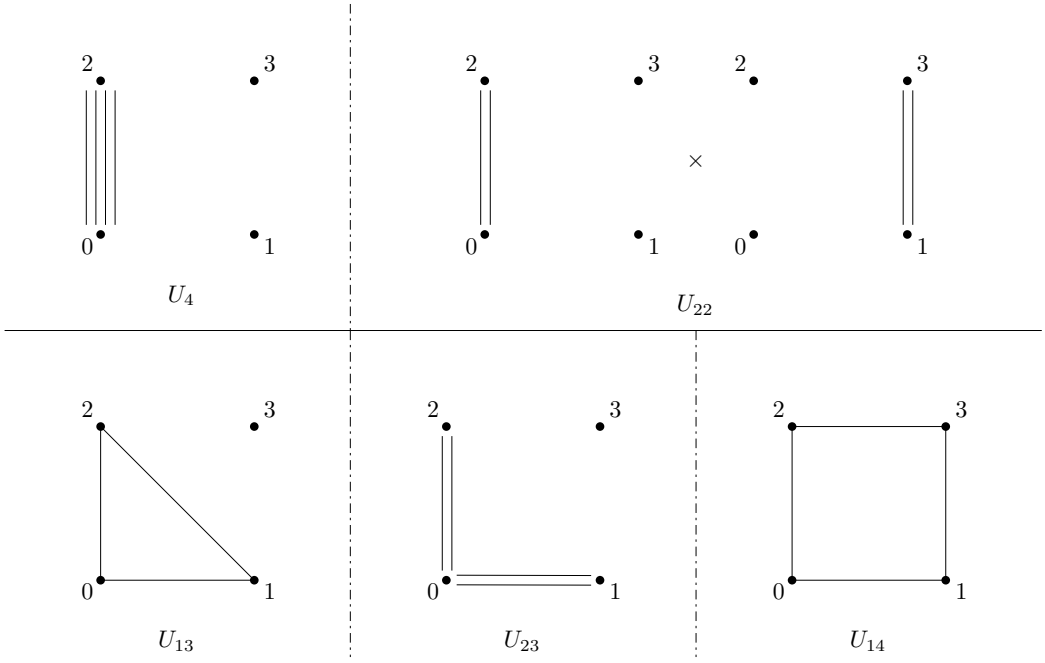


Figure 5.1. A graphical approach for the calculation of the different terms of Binder cumulants for 4 replicas. Each line represents an overlap term. All possible combinations must be taken into account.

However, in the same spirit of the previous definitions it is possible to write quantities that converge to the value of the usual equilibrium temperature derivative. Let us begin from the definition of the thermal average of a two-replicas observable

$$\left[\langle \mathcal{O}^{ab} \rangle\right] = \left[\frac{1}{Z^2} \sum_{\{\sigma_i^a = \pm 1\}, \{\sigma_i^b = \pm 1\}} \mathcal{O}^{ab} \exp\{-\beta \left(H^a + H^b \right)\} \right]. \tag{5.12}$$

One has that its derivative with respect to the inverse temperature β reads

$$\frac{\mathrm{d}\left[\langle \mathcal{O}^{ab} \rangle\right]}{\mathrm{d}\beta} = \left[\langle \mathcal{O}^{ab} \rangle \langle H^a + H^b \rangle - \langle \mathcal{O}^{ab} \left(H^a + H^b \right) \rangle \right]. \tag{5.13}$$

As in the definition of Binder's cumulants any time we need to calculate some power of an equilibrium thermal average we need to multiply different replicas since the powers of equilibrium averages are always computed on uncorrelated ranges of the equilibrium thermal histories. Hence, given that $r = 4$ we can only calculate the temperature derivatives of two-replicas observables. The general off-equilibrium form is

$$\frac{\mathrm{d}\left[\mathcal{O}(t)\right]}{\mathrm{d}\beta} = \left[\frac{1}{c_2} \sum_{a<b<c<d} \mathcal{O}^{ab}(t) \left(H^c(t) + H^d(t) \right) - \frac{1}{c_2} \sum_{a<b} \mathcal{O}^{ab}(t) \left(H^a(t) + H^b(t) \right) \right]. \tag{5.14}$$

In particular we have computed the temperature derivative of the logarithm of the susceptibility $\log \chi(t)$ and of the Binder cumulant $U_4(t)$.

These quantities have been adopted in order to get another estimate of the exponent ν once the the critical temperature has been determined since one expects at the critical point

$$\frac{\mathrm{d}}{\mathrm{d}\beta}\log\chi(t) = \frac{1}{\chi(t)}\frac{\mathrm{d}\chi(t)}{\mathrm{d}\beta} \sim L^{1/\nu}, \qquad \frac{\mathrm{d}}{\mathrm{d}\beta}U_4(t) \sim L^{1/\nu}. \tag{5.15}$$

However, we will report only qualitative results concerning these quantities.

5.1.3 Sublattice observables

We also studied the behaviour of observables computed on limited spatial regions in the cubic lattice we refer to as **sublattices**. The aim was to study the behaviour of finite-size effects while varying the sublattices sizes, and see if for these systems corrections to scaling might be less severe.

We also compared the sublattices observables to their real-lattices counterparts. Indeed, it is possible to observe the superposition of data at the beginning of the dynamics. This superposition however worsens as the correlation length grows marking a distinction between the real lattice and the sublattice.

We chose to examine sublattices of linear size L_s of the form

$$L_s = \frac{L}{2m}, \quad m = 1, 2, \ldots, n \tag{5.16}$$

choosing the smallest sublattice to be $L_{s,min} = 8$ for every system. Since the possible ways to divide a lattice into sublattices are too many we chose to analyze only the exact divisions and their translations along the diagonal direction $\vec{D} = (d, d, d)$ obtaining maximally different systems. The data we show in the Appendix B are those for

$$\vec{D} = (4, 4, 4)\,, \tag{5.17}$$

hence for two translations of $L_{s,min} = 8$. It follows that $L_s = 16$ will have four translations and so on, doubling every time for the next bigger sublattice. At last let us define the number of translation steps for the smallest $L_{s,min} = 8$ as

$$p_m = \frac{L_{s,min}}{d} = \frac{8}{4} = 2 \tag{5.18}$$

The algorithm calculates all fundamental quantities, overlap q_x^{ab}, the susceptibility Fourier Transform $\tilde{\chi}^{ab}(2\pi/L_s)$, and energy E^a, for the smallest sublattices. The it builds the bigger sublattices values and translations up to the real lattice.

Now, let us focus on a fundamental aspect: the sublattices do not have periodic boundary conditions. Hence we need to take it into account in all those quantities which are usually calculated on the hypothesis of translational invariance. Let us take as an example the susceptibility. We have to integrate the two point correlation function not on the whole sublattice but just on some subset which we take at its center for symmetry reasons. We will call this subset **core**, whose linear size is L_c

$$\chi_c(t) = \frac{1}{V_c}\left[\sum_{a<b}\left(\sum_{x\in V} q_x^{ab}(t)\right)\left(\sum_{y\in V_c} q_y^{ab}(t)\right)\right], \tag{5.19}$$

where $V_c = L_c^3$. Indeed for periodic systems where $L = L_c$ we recover the usual definition

$$\chi_c(t) = \chi(t) = \frac{1}{V}\left[\sum_{a<b}\left(\sum_x q_x^{ab}(t)\right)^2\right]. \tag{5.20}$$

It is reasonable to think that the smaller the size of the core the smaller will be the effects due to the loss of translational invariance for sublattices.

Similar considerations hold for the Fourier transform of the correlation function which reads

$$\tilde{\chi}_c\left(\frac{2\pi}{L_s}, t\right) = \frac{1}{V_c}\left[\sum_{a<b}\left(\sum_{y\in V_{core}}\sum_x q_y^{ab}(t)\, q_x^{ab}(t)\, e^{-\frac{2\pi i}{L}(x-y)}\right)\right]. \tag{5.21}$$

Hence, we need to keep track of the different sublattices cores. This task is soften by the use of diagonal translations which allows to share sublattice cores between smaller and bigger sublattices. In fact for $p_m = 2$ the cores of the first translation of $L_s = 8$ are the same as those of the zeroth translation of $L_s = 16$ and so on. We choose as the smallest size of the cores $L_c = 2$. Finally, all sublattices results are averaged over all translations for fixed L_s and L_c. Indeed, because of the cores we gain the condition over the displacements of the smallest sublattice $k_{min} \geq 2$, which guarantees that all cores are shared.

We have thus defined a new set of observables which should scale with the same critical exponents as those of real lattices. We expect to observe scaling for those that have the same values of the following ratios

$$S = \frac{L}{L_s}, \quad C = \frac{L_s}{L_c}. \tag{5.22}$$

Hence, in principle, it is possible to use these class of observables in order to get an estimation of critical parameters. However, it is clear that these estimations would be correlated to those of real lattices.

5.2 Monte Carlo Simulations

All simulations were performed on GPUs using the implementation we described in Chapter 3 with the MINSTD PRNG. This choice is *a priori* justified by the fact that we are using one PRNG for each lattice site which is shared by different replicas. Using random initial seeds eventual long range correlations are to be compared to those due to the critical behaviour which are also long range. Furthermore, the simulation time never reaches the period of this PRNG. *A posteriori*, we will see that the compatibility of the final results with those of [31] justifies such a choice.

Our computing system is inhomogeneous with a total of 9 GPUs: 3 GTX Titan, 4 GTX 680 and 1 Tesla M2090 on one cluster, and a single GTX 680 on a stand-alone computer. The total simulation time is 27397 GPU hours which means 3.1 GPU years on a single GTX Titan which is our fastest GPU. Further details about simulations are reported in the Appendix A.

5.3 Out-of-equilibrium finite-size scaling

Let us now analyze the results concerning the out-of-equilibrium finite-size scaling ansatz we presented in Chapter 4, specifically in (4.38).

The simulated cubic lattices have linear sizes in the range $8 \leq L \leq 64$. We considered five values of β between 0.880 and 0.910. Statistics is a crucial factor in the analysis, hence we considered a very large number N_s of samples for each L and β. Typically, N_s varies between $5 \cdot 10^5$ and a few million. Only for $L = 48$ and 64, is statistics smaller: $N_s = 6.5 \cdot 10^5$, 10^5 in the two cases, respectively. Essentially all runs end when the system is still out of equilibrium. In most of the cases, data extend only up $R_\xi \approx 0.5$, in some cases even less (at equilibrium [31] $R_\xi = 0.652(3)$ at the critical point). Error bars are reported in all figures.

5.3.1 Critical Temperature, ν and ω

Let us begin by writing once again the out-of-equilibrium finite-size scaling ansatz for the Binder cumulant U_4 (4.38)

$$U_4(t, L, \beta) \simeq \tilde{\mathcal{U}}_4 \left(R_\xi, \varepsilon L^{1/\nu}\right) + \frac{u\,(\beta)}{L^\omega} \tilde{\mathcal{U}}_{4,\omega} \left(R_\xi, \varepsilon L^{1/\nu}\right). \tag{5.23}$$

This expression is defined for $R_\xi \leq R_{\xi,\mathrm{eq}}(\varepsilon L^{1/\nu})$, where $R_{\xi,\mathrm{eq}}(\varepsilon L^{1/\nu})$ is the equilibrium value of R_ξ for the given $\varepsilon L^{1/\nu}$.

Indeed, ignoring scaling corrections, close to the critical point Eq.(5.23) can be expanded in powers of ε, obtaining

$$U_4(t, L, \beta) \simeq \tilde{\mathcal{U}}_4 \left(R_\xi, 0\right) + \frac{\partial}{\partial \varepsilon} \tilde{\mathcal{U}}_4 \left(R_\xi, \varepsilon L^{1/\nu}\right) \Big|_{\varepsilon=0} \varepsilon L^{1/\nu} \tag{5.24}$$

At fixed R_ξ, the quantity $U_4(t, L, \beta)$ behaves exactly as in the equilibrium case: β_c is determined as the crossing point and ν is obtained by computing the slope at β_c. However, in this formulation equilibration is not needed. Equation (5.23) is valid for any value of R_ξ, hence one might think of choosing a small value for such a parameter, reducing significantly the length of the runs. However, one must not forget that the method is intrinsically a finite-size method, hence it can only work if finite-size effects are not too tiny, and this, in turn, requires R_ξ not too small.

In figure 5.2 we report the values of $U_4(t)$ as a function of $R_\xi(t)$ for one lattice size $L = 8$ for the lowest and highest values of β we considered. The curves share the origin point which is at $R_\xi = 0$ and $U_4 \sim 3$, *i.e.* the infinite-temperature values. Along the time evolution data belonging to different values of β eventually separate. Data precision at early times is mainly controlled by the number of disorder realizations while towards equilibrium the time dependence becomes stronger. In figure 5.3 we report the data for two different lattice size $L = 12$ and $L = 32$ in the entire range of β: lower/higher curve represent data at smaller/higher values of β. We can see that the lowest $L = 32$ curve is lower than the lowest curve of $L = 12$, both corresponding to $\beta = 0.880$, while the highest of $L = 32$ is higher than the highest of $L = 8$ in correspondence of $\beta = 0.910$. Hence, the relative order of data belonging to $L = 12$ and $L = 32$ has an inversion in the temperature range

$0.880 \leq \beta \leq 0.910$. This result is the out-of-equilibrium generalization of the Binder crossing from a **global** point of view.

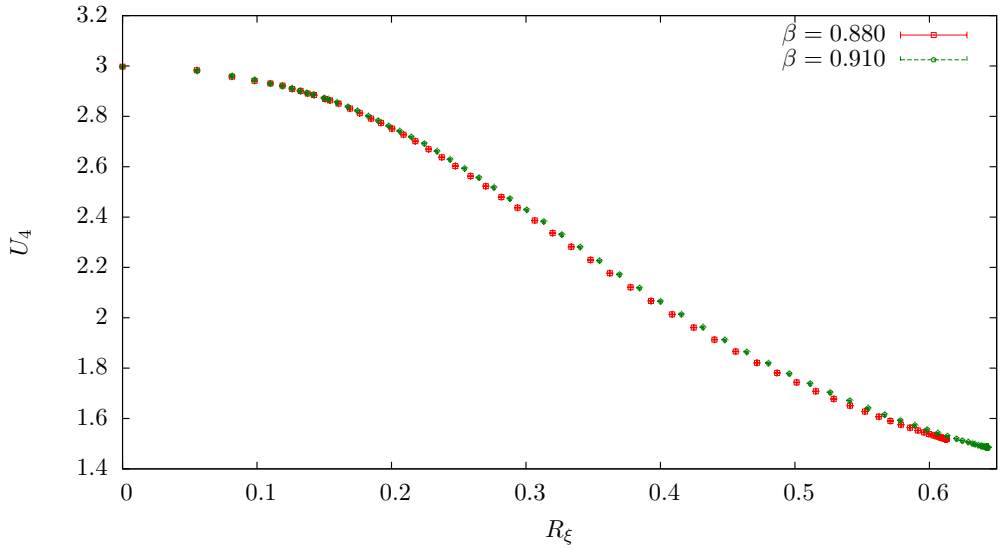


Figure 5.2. Plot of U_4 as a function of R_ξ for $L = 8$ at two boundaries of the considered temperature interval.

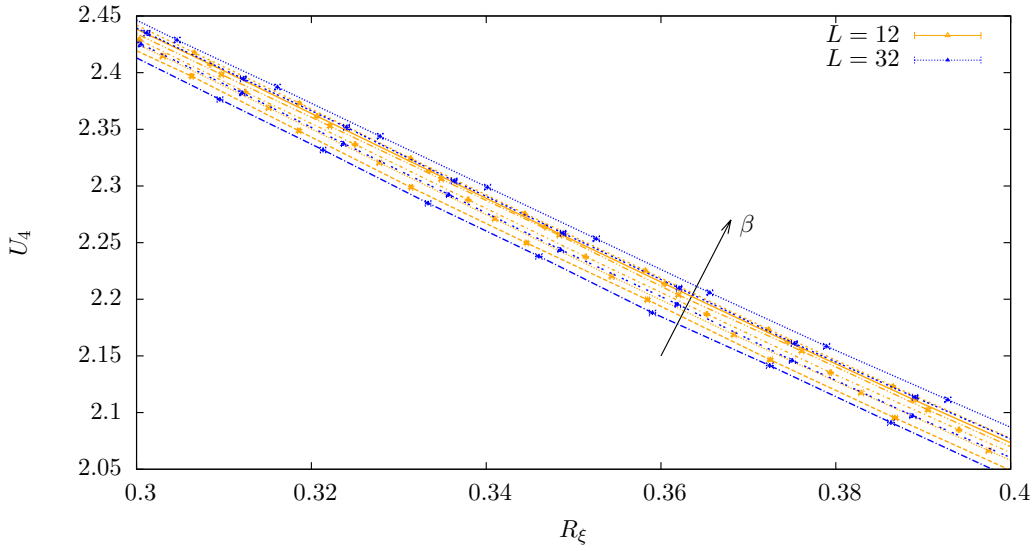


Figure 5.3. Detail of the combined plot of U_4 and R_ξ for two different lattice sizes $L = 12$ and $L = 32$. The direction of increasing inverse temperature β is shown. The temperature interval is the same for both lattices. At the lowest value of β $L = 32$ data stay below those of $L = 12$ while the contrary happens at the highest values. This is how the Binder crossing appears from the global point of view.

Hence, we expect that fixing the value of either U_4 or R_ξ, *i.e.* making a **cut** of the data set, different curves would intersect, thus signaling the existence of a phase

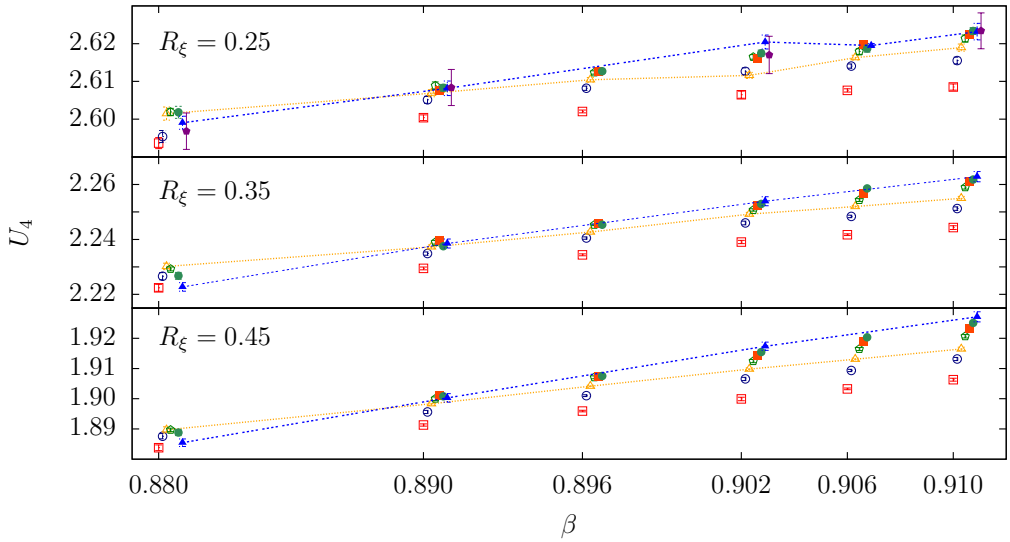


Figure 5.4. Temperature dependence of U_4 for fixed values of R_ξ. A shift in temperature has been added in order to highlight finite-size dependence [1].

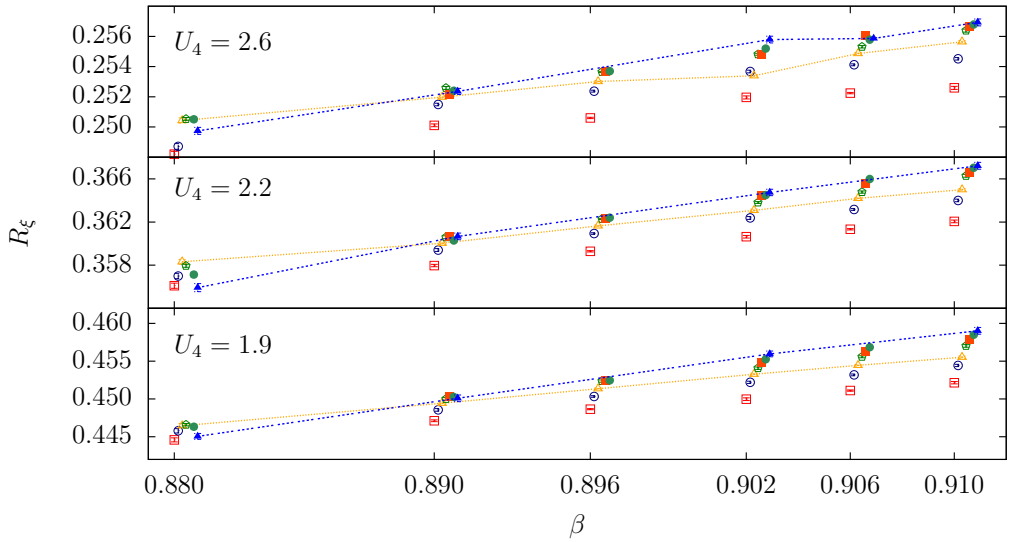


Figure 5.5. Temperature dependence of R_ξ for fixed values of U_4. A shift in temperature has been added in order to highlight finite-size dependence. The top panel is the farthest away from equilibrium since for a random configuration $U_4 \sim 3$.

transition. In figures 5.4 and 5.5 we report such a behaviour: this is exactly what one would expect for a standard equilibrium analysis for the Binder cumulant U_4 or for R_ξ. Although we carried the final analysis substituting the time dependence with a dependence on R_ξ it is clear that any time-monotonous RG invariant quantity, such as U_4, can be used as a time-like parameter. This is a **local** point of view. The

[1] Symbols: empty red square ($L = 8$), blue empty circles ($L = 10$), orange empty triangles ($L = 12$), green empty pentagons ($L = 16$), red filled squares ($L = 20$), green filled circles ($L = 24$),

main difference is that the values are taken from an out-of-equilibrium regime, so instead of having only one set of intersecting curves, one has a correlated succession of intersection points. For such a succession finite-size effects are expected to change. The estimate of the critical temperature, however, has to be stable as the R_ξ or U_4 cut changes. There is yet another way to collect evidence for a phase transition from out-of-equilibrium data. Let us consider now two RG invariant quantities, say U_4 and U_{13}, at a fixed value of R_ξ. We expect data for different sizes and temperature to collapse onto a single curve, when plotting U_{13} as a function of U_4. As shown in

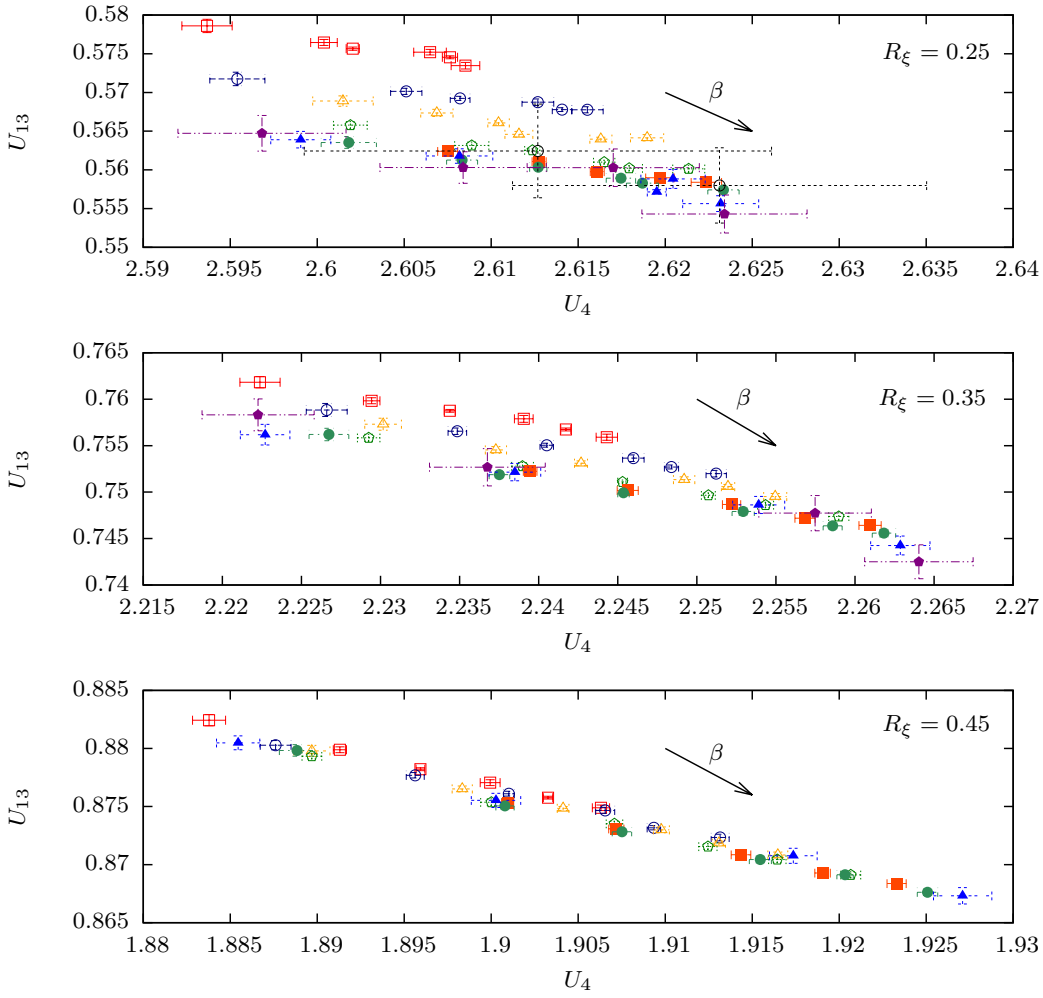


Figure 5.6. Data for U_{13} as a function U_4 for three different values of R_ξ. For small times at $R_\xi = 0.25$ both statistical errors and finite-size corrections are large, while later on at $R_\xi = 0.45$ the expected scaling is clearly visible[2].

figure 5.6, the expected scaling is clearly visible only in the late dynamic evolution at

blue filled triangle ($L = 32$), purple filled pentagon ($L = 48$).

$R_\xi = 0.45$, where statistical errors and scaling corrections are smaller. The scaling ansatz is then verified, since at a fixed value of an RG invariant quantity we have that the two-dimensional space of control parameters L and β is mapped onto a single curve which is a one-dimensional space, which is exactly what we described in Chapter 1 describing Widom scaling.

We just presented also the local point of since it gives us a useful bridge to the standard equilibrium techniques. However, our final estimates are based on the global point of view, so let us continue with the analysis of the complete scaling form. Looking at (5.24) it clearly appears that the derivative term $\partial\tilde{\mathcal{U}}_4/\partial\varepsilon$, when evaluated in $\varepsilon = 0$, must be sensibly different from zero for finite-size effects to allow a determination of the critical temperature. We expect this factor to be small at small values of R_ξ and to increase as equilibrium is approached. Hence, in principle, it would be possible to set a very small maximum value of R_ξ but then one would need a very large statistics in order for this derivatives to significantly differ from zero in the statistical errors. Hence R_ξ should be set to a small value, but still large enough to have a reasonable sensitivity of the results on system sizes.

Since we already knew with good precision the value of the critical temperature [39] we chose a small temperature interval, so that we can expand the scaling relation to first order in ε. We have also performed some analyses using a second-order approximation, without observing significant differences. As for the correction-to-scaling function, we have verified that we can assume it to be independent of temperature. Finally, we should make approximations for the nonlinear scaling fields. Relying on the analysis of Ref. [39], we set $u_\omega(\beta) = 1$, neglecting the additional corrections. Hence, each Binder cumulant was fitted to

$$U_\#(t, L, \beta) = P_1(R_\xi) + P_2(R_\xi)(\beta - \beta_c)L^{1/\nu} + P_3(R_\xi)L^{-\omega}, \tag{5.25}$$

with $P_1(R_\xi)$, $P_1(R_\xi)$, and $P_3(R_\xi)$ polynomials of degree 6, 3, and 3, respectively.

Using the linearized scaling ansatz (5.25) we can give a clear interpretation of the finite-size data dependence which is visible in figures 5.4 and 5.5. The terms $P_2(R_\xi)(\beta - \beta_c)L^{1/\nu}$ and $P_3(R_\xi)L^{-\omega}$ represent corrections to the critical fixed point value P_1 for every value of R_ξ: the first one codifies corrections due to off-critical data while the second one clearly codifies finite-size corrections. As L grows the two contributions behave differently

$$\lim_{L\to\infty} P_2(R_\xi)L^{1/\nu} = \text{const}, \qquad \lim_{L\to\infty} P_3(R_\xi)L^{-\omega} = 0. \tag{5.26}$$

This means that the sign of the sum $P_2(R_\xi)(\beta - \beta_c)L^{1/\nu} + P_3(R_\xi)L^{-\omega}$ changes according to the relative magnitudes of the two terms and weather if the system is in the spin glass, *i.e.* $\beta - \beta_c > 0$, or in the paramagnetic phase, *i.e.* $\beta - \beta_c < 0$. Hence, looking at 5.4 and 5.5, for low values of β in the paramagnetic phase, one can see that the corrections to the critical behaviour change sign as the linear size L increases because the off-critical corrections are larger than finite-size ones, while for higher values of β, in the spin glass phase, the sign of the correction does not depend on the linear size.

[2]Symbols: empty red square ($L = 8$), blue empty circles ($L = 10$), orange empty triangles ($L = 12$), green empty pentagons ($L = 16$), red filled squares ($L = 20$), green filled circles ($L = 24$), blue filled triangle($L = 32$), purple filled pentagon ($L = 48$).

The fit of the five Binder cumulants is quite complex, as we take ω, β_c, ν, and the coefficients of the polynomials as free parameters. As a whole, there are 78 free parameters that must be optimized. Additionally, one should choose the goodness-of-fit function that is optimized in the fitting procedure. Any choice provides correct estimates (i.e., estimates that converge to the exact result when the error on the input data goes to zero), but they may provide results of different precision. Moreover, the results obtained by using different goodness-of-fit functions are affected by the neglected next-to-leading corrections in a different way. We have thus made two different, inequivalent choices, obtaining similar results. More details can be found in the appendix.

As usual in this type of analyses, the most difficult issue is the estimation of the systematic errors due to the neglected correction terms. This is very important here, since the attainable values of L are quite small. We have thus performed fits with several types of cuts. We perform fits including each time only data satisfying $L \geq L_{\min}$, $\xi \geq \xi_{\min}$, and $R_\xi \geq R_{\xi,\min}$, considering several values for $L_{\min}$, $\xi_{\min}$, and $R_{\xi,\min}$. Results obtained by taking $3 \leq \xi_{\min} \leq 5$, $8 \leq L_{\min} \leq 12$, and $0 \leq R_{\xi,\min} \leq 0.4$ show some scatter, which is somewhat larger than statistical errors, indicating that the neglected systematic effects may be as important as the statistical ones. The most crucial parameter is $\xi_{\min}$. When such a parameter is increased from 3 to 4, the exponent ω decreases sharply, by more than one error bar, while β_c increases. Such a systematic drift occurs also when $\xi_{\min}$ is further increased to 5, but now the change is much less than one error bar. Therefore, the results we quote correspond to fits with $\xi_{\min} = 4$. For such a value of $\xi_{\min}$, the analysis, which a *a priori* should be least affected by the additional scaling corrections, gives $\beta_c = 0.911(2)$, 0.916(4), 0.909(4) for $L_{\min} = 8, 10, 12$ and $R_{\xi,\min} = 0$, and $\beta_c = 0.911(2)$, 0.909(2), 0.909(3) for $R_{\xi,\min} = 0$, 0.2, 0.4 and $L_{\min} = 8$. No systematic trends can be observed, all estimates being consistent within errors. Except for one estimate, all results (with their errors) we are quoting here lie in the interval $0.906 \leq \beta_c \leq 0.913$. Hence, we take

$$\beta_c = 0.910(4) \tag{5.27}$$

as our final estimate. The error, which is twice the error affecting the results with $L_{\min} = 8$, is somewhat subjective and should take into account the effect of the neglected next-to-leading scaling corrections. The results of all analyses are consistent with estimate (5.27) within one combined error bar. Analogously, we can estimate ω and ν obtaining

$$\omega = 1.3(2), \qquad \nu = 2.47(10). \tag{5.28}$$

The estimates of ω are strongly correlated with those of β_c: the larger β_c, the smaller ω is. If $\beta_c = 0.906$, fits keeping β_c fixed give $\omega \approx 1.5$, while $\omega \approx 1.1$ is obtained by fixing $\beta_c = 0.914$. The exponent ν is instead much less correlated with β_c, changing at most by ± 0.03 when β_c varies by ± 0.004.

To show the quality of the results, in Fig. 5.7, we report ΔU_4, defined by

$$\Delta U_4(\beta, L, R_\xi) = U_4(\beta, L, R_\xi) - P_3(R_\xi) L^{-\omega}, \tag{5.29}$$

versus $\varepsilon L^{1/\nu}$. We consider $R_\xi = 0.25$, 0.35, and 0.45. Very good scaling is observed, confirming the correctness of the scaling Ansatz and the accuracy of the estimates

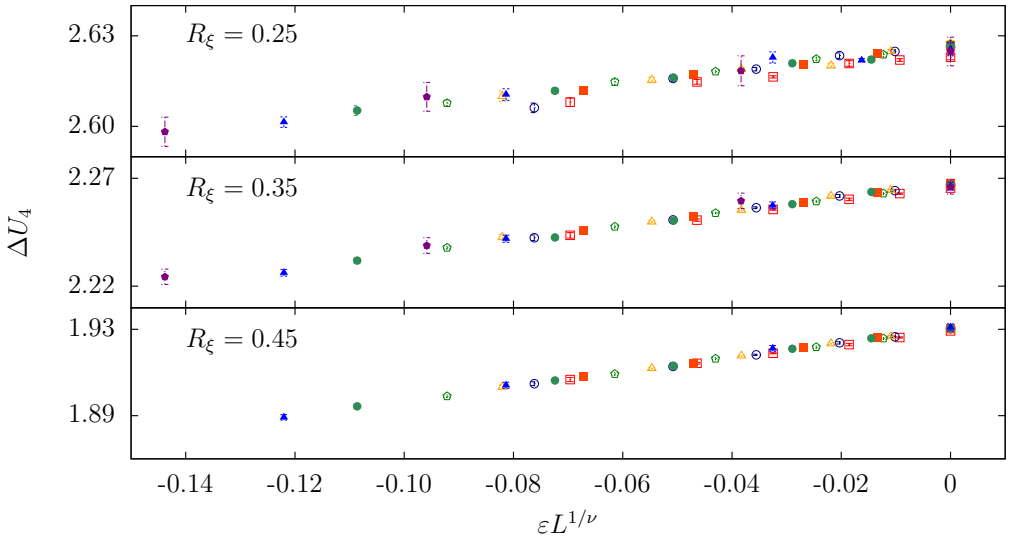


Figure 5.7. Plot of ΔU_4 versus $\varepsilon L^{1/\nu} = (\beta - \beta_c)L^{1/\nu}$. Data collapse onto a single curve also far away from equilibrium.[3]

of the critical exponents. Note also that the data lie on an essentially straight line, validating our choice of expanding $f_U(R_\xi, \epsilon)$ to first order in ϵ. From the figure, we can also clarify why a large number of samples, of order 10^6, is needed to estimate the critical parameters. For instance, U_4 at $R_\xi = 0.35$ varies by 0.04 within our temperature interval. Hence, the temperature dependence of the data can be observed only if the errors on U_4 are significantly less than 10^{-2}, for instance, if they are equal to 10^{-3}. Since errors scale as $a/\sqrt{N_s}$ with $a \approx 1$ for all values of L, a 10^{-3} error is obtained by taking $N_s \approx 10^6$. Note that this requirement is not specific of the off-equilibrium method we use. Also equilibrium analyses require N_s to be large [105, 39, 31].

5.4 Anomalous dimension and dynamic exponent

The analysis we have performed for the Binder cumulants can be extended to the susceptibility. The finite-time scaling behavior can now be written as

$$\ln \chi = (2 - \eta) \ln L + P_1(R_\xi) + (\beta - \beta_c)L^{1/\nu} P_2(R_\xi) + L^{-\omega} P_3(R_\xi) + P_4(\beta), \quad (5.30)$$

where the last term $P_4(\beta)$ is the contribution of the nonlinear scaling field associated with the magnetic field, see Ref. [105, 39] for a discussion. A good parametrization is obtained by taking $P_1(R_\xi)$, $P_2(R_\xi)$, $P_3(R_\xi)$ as polynomials of degree 6, 3, 3, respectively, as before. For the $P_4(\beta)$, we set $P_4 = a_4\beta$. We obtain at the end the final estimate

$$\eta = -0.39(1), \quad (5.31)$$

[3]Symbols: empty red square ($L = 8$), blue empty circles ($L = 10$), orange empty triangles ($L = 12$), green empty pentagons ($L = 16$), red filled squares ($L = 20$), green filled circles ($L = 24$), blue filled triangle($L = 32$), purple filled pentagon ($L = 48$).

which is fully consistent with those of Refs. [39, 31]. Finally, we estimate z by requiring data to satisfy the general scaling form with the dependence on tL^{-z}, rather than on R_ξ. We obtain

$$z = 6.80(15), \tag{5.32}$$

where the error should be quite conservative. In figure 5.8 we report the scaling at the estimated critical inverse temperature $\beta_c \sim 0.910$ for U_4 and χ, in order to show the quality of the estimation of η and z. Indeed, by comparing the top

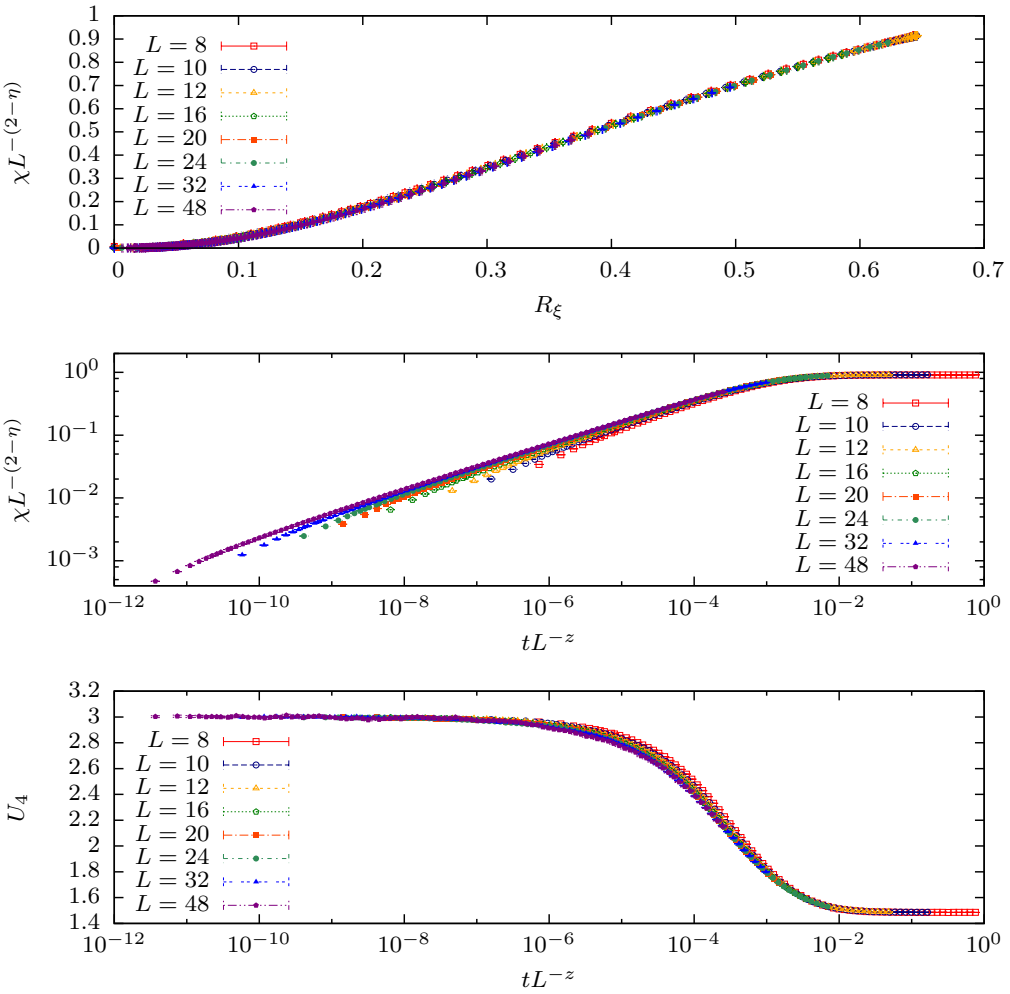


Figure 5.8. Top panel: critical scaling of $\chi L^{-(2-\eta)}$ as a function of R_ξ. Data collapse is very good. Middle panel: critical scaling of $\chi L^{-(2-\eta)}$ as a function of tL^{-z} in log-log scale. Finite-size corrections are much larger with respect to the top panel. Bottom panel: critical scaling of U_4 as a function of tL^{-z}. Also in this case finite-size corrections are considerably large.

and the middle panels, it clearly appears that $\chi L^{-(2-\eta)}$ data have a better scaling as a function of R_ξ than as a function of tL^{-z}. This happens because finite-size

corrections are larger when using the time parametrization. This further justifies our choice for the out-of-equilibrium scaling ansatz in terms of RG invariant quantities.

5.5 Consistency checks

Now that we have estimated all needed critical parameters it is possible to perform two different checks

- we can check the scaling of temperature derivatives with the exponent ν;
- we can check the range of validity of the infinite-volume limit behaviour.

5.5.1 Temperature derivatives

In figures 5.9 and 5.10 we report the critical temperature scaling of $L^{-1/\nu}\mathrm{d}_\beta U_4$ and $L^{-1/\nu}\mathrm{d}_\beta \log\chi$ as functions of R_ξ in order to evaluate the quality of the estimate of ν. The scaling in terms of tL^{-z} would just introduce ulterior finite-size corrections hence we do not report it. Indeed, these two quantities show a very different behaviour.

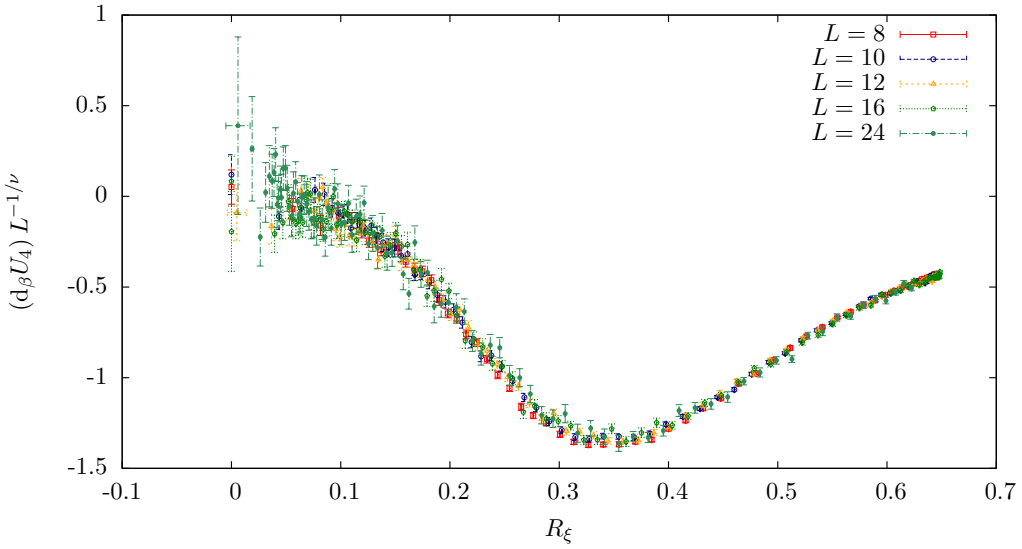


Figure 5.9. Critical scaling of $\mathrm{d}_\beta U_4$ as a function of R_ξ. Finite-size corrections are small.

In figure 5.9 no finite-size corrections are visible. We reported the data at $\beta = 0.910$. However, have we plotted those at $\beta = 0.880$ we would have still obtained a good data collapse. It is clear that this observable requires even more statistics than that we acquired.

In figure 5.10 data for $L^{-1/\nu}\mathrm{d}_\beta \log\chi$ have less severe statistical errors, this is why we could plot data up to $L = 32$ without introducing fuzzy points. Data show, differently from those of $L^{-1/\nu}\mathrm{d}_\beta U_4$, very strong finite-size corrections. Had we plotted the data for $\beta = 0.880$ the order of the curves would have not changed. This means that finite-size corrections are always larger than off-critical corrections at least in this temperature range: a cut at a fixed value of R_ξ would not have shown any crossing point.

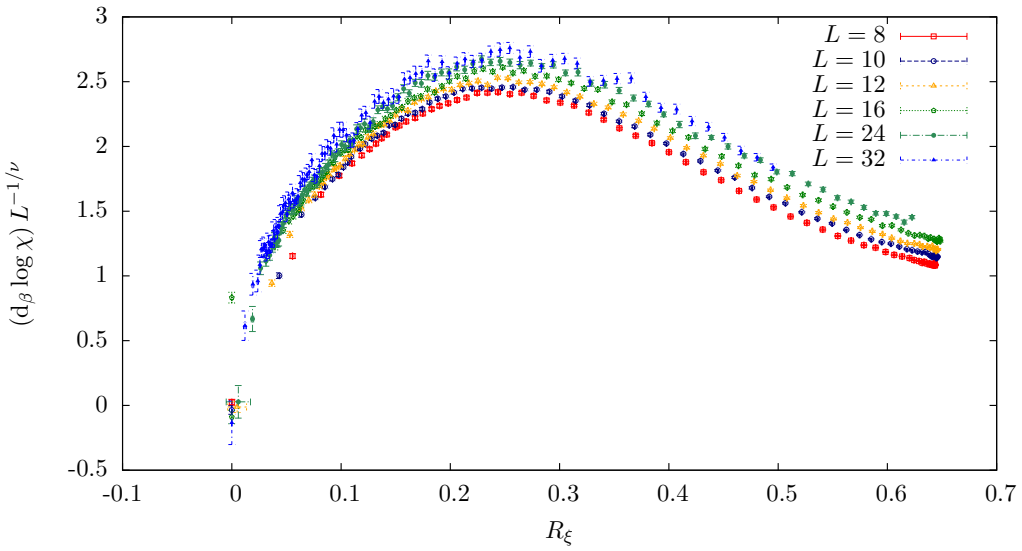


Figure 5.10. Critical scaling of $\mathrm{d}_\beta \log \chi$ as a function of R_ξ. Strong finite-size corrections are visible.

5.5.2 Infinite-volume regime

Let us now check whether it is possible to observe the infinite-volume behaviour we described in Chapter 4 and which is at the basis of most of the out-of-equilibrium techniques that were developed before. Let us write the finite-time scaling ansatz for the susceptibility and the correlation length

$$\begin{aligned} \chi &\sim t^{(2-\eta)/z} F_t(\varepsilon L^{1/\nu}, tL^{-z}) \left[1 + \frac{u_{i_1}}{t^{\omega_1/z}} f_{1t}(\varepsilon L^{1/\nu}, tL^{-z})\right], \\ \xi &\sim t^{1/z} G_t(\varepsilon L^{1/\nu}, tL^{-z}) \left[1 + \frac{u_{i_1}}{t^{\omega_1/z}} g_{1t}(\varepsilon L^{1/\nu}, tL^{-z})\right]. \end{aligned} \tag{5.33}$$

At a fixed value of $\varepsilon L^{1/\nu}$ the infinite-volume regime is located by a weak dependence of the scaling functions on tL^{-z}, since in the thermodynamic limit $tL^{-z} \to 0$. One, then expects that, for sufficiently large lattices and for sufficiently small times, data for $\xi t^{-1/z}$ and $\chi t^{-(2-\eta)/z}$ to assume a nearly constant value as a function of tL^{-z}. We report in figure 5.11 such a scaling plot for both observables at the estimated critical point.

Looking at the figure it is clear that the thermodynamic-limit behaviour begins to be noticeable only for $L = 48$. Moreover, such a regime is reached only after a rather long initial transient. Large finite-size corrections are also present signaling that for the three-dimensional Edwards-Anderson model one needs to simulate much larger systems in order to obtain reliable estimates with this method. Hence, the waiting times of many of the previous works [92, 95] ($t_w \leq 1600$) seem to be insufficient for a good estimation of the infinite-volume behaviour, thus explaining the $\omega \sim 2.9(6)$ result of [91].

In order to appreciate finite-size correction in the tL^{-z} dependence we can cut the U_4 data at fixed values of R_ξ and plot them as in figure 5.12. We should compare

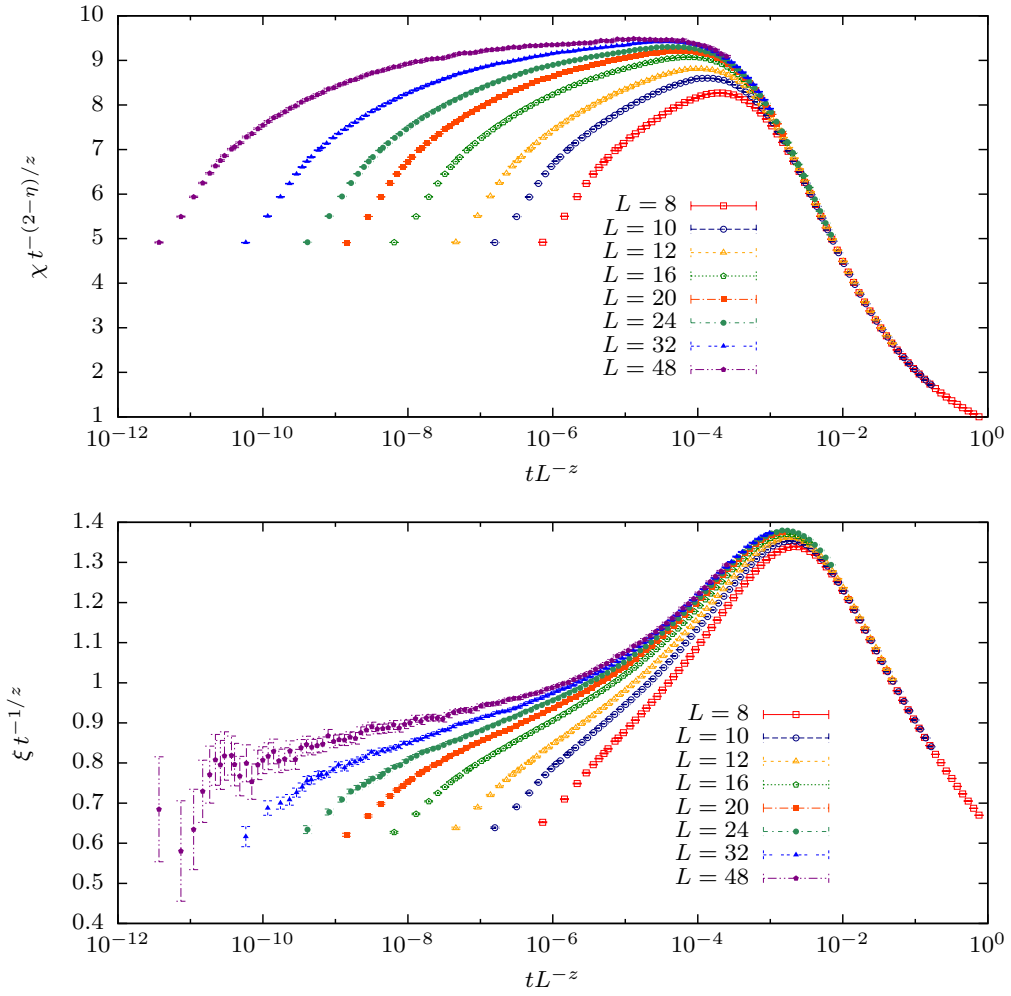


Figure 5.11. Top panel: critical scaling of $\chi t^{-(2-\eta)/z}$ as a function of tL^{-z}. The thermodynamic regime begins to be visible only for $L = 48$. It is clear that a rather long initial regime has to be discarded before the infinite-volume regime is attained, thus explaining the deviations of the previous estimates of the critical temperature. Bottom panel: critical scaling of $\xi t^{-1/z}$ as a function of tL^{-z}. Again, the infinite-volume regime begins to be visible only for $L = 48$. Early-time corrections are masked by the large statistical errors.

this graph with 5.6. Indeed, very strong finite-size corrections are visible for small lattices while some collapsing is present for the larger ones.

These results still supports our approach highlighting the difficulties one has to face in treating spin glasses in the finite-time scaling scheme, thus explaining why earlier works reported wrong estimates for the critical parameters.

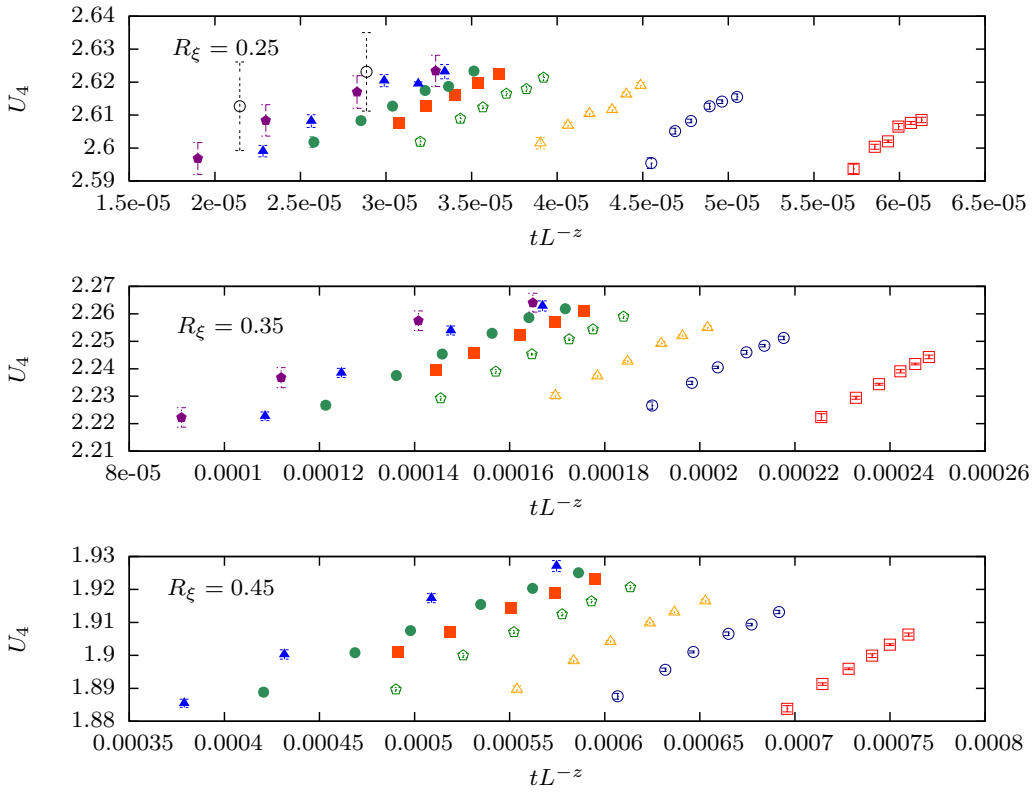


Figure 5.12. Temperature dependence of U_4 for fixed values of R_ξ. A shift in temperature has been added in order to highlight finite-size dependence[4].

5.6 Final remarks

It is interesting to compare these results with previous ones, see Table 5.1. For the critical-point position, our estimate $T_c = 1/\beta_c = 1.099(5)$ agrees within errors with the estimates $T_c = 1.102(3)$ and $T_c = 1.109(10)$ of Refs. [31, 39], obtained from the analysis of equilibrium results. Our error is larger than that reported in Ref. [31], but note that our final error includes a subjective estimate of the systematic error. Had we reported only the statistical error for $L_{\min} = 8$, we would have obtained the same accuracy. The estimates of ν are also consistent, while our final estimate of ω is slightly larger, though still consistent within error bars, than previous ones. In particular, the error quoted in Ref. [39] appears to be underestimated. Previous dynamic estimates of T_c were not consistent with the equilibrium estimates of Refs. [31, 39]. It is now clear that the reported errors are probably underestimated,

[4]Symbols: empty red square ($L = 8$), blue empty circles ($L = 10$), orange empty triangles ($L = 12$), green empty pentagons ($L = 16$), red filled squares ($L = 20$), green filled circles ($L = 24$), blue filled triangle($L = 32$), purple filled pentagon ($L = 48$), black empty circles ($L = 64$).

as a consequence of the neglect of the subleading scaling corrections in the analyses.

Method		T_c	ν	η	ω	z
Equilibrium dynamics	[87]	1.175(25)				6.1(3)
Equilibrium and off-equilibrium mixing	[89]	1.17(1)		−0.25(2)		6.0
	[90]	1.20(1)		−0.21(2)		
	[91]	1.195(15)			2.9(6)	5.65(15)
	[92]	1.19(1)		−0.22(2)		5.7(2)
Off-equilibrium response function	[95]	1.135(5)				
Non-equilibrium relaxation	[97]	$1.12(12)_{sys}$				
Extrapolation technique	[98]	1.155(3)	$\beta = 0.52(9)$			
Off-equilibrium relaxation time	[102]	1.17(4)	1.5(3)	$\gamma = 3.6(6)$		6.2(2)
Off-equilibrium scaling	[103]	1.18(1)	1.40(5)	−0.20(1)		
Equilibrium FSS	[39]	1.109(10)	2.45(15)	−0.375(10)	1.0(1)	
	[31]	1.1019(29)	2.562(42)	−0.3900(36)	1.12(10)	
Dynamics at the critical temperature	[25]					6.86(16)
Our results		1.099(5)	2.47(10)	−0.39(1)	1.3(2)	6.80(15)

Table 5.1. Results for critical parameters for various off-equilibrium techniques and most recent equilibrium finite-size scaling (FSS) estimations [39, 31]. The exponents β and γ are related to ν and η by $\beta = \nu(1+\eta)/2$, $\gamma = \nu(2-\eta)$.

The method we have discussed represents a significant improvement with respect to equilibrium analyses. Indeed, since the scaling variable is tL^{-z}, the time needed to extend Metropolis runs from any value of R_ξ to equilibrium scales as L^z, i.e., as L^7 given that $z \approx 7$ for the Ising spin glass. Hence, the advantage is very large and increases rapidly with L. To make a fair comparison with equilibrium studies, we should, however, take into account that in those studies one combines the parallel-tempering method with the Metropolis or heat-bath algorithm [106]. It is not clear how equilibration times scale for this combined algorithm, and in particular, how long it takes to thermalize the hard samples. However, the results reported in Ref. [31] are consistent with a sample-dependent time that scales as L^2 for the samples that equilibrate fast and as L^7 for those that are slower. The off-equilibrium method is still significantly faster. A more direct comparison can be obtained by using the results published in Ref. [31]. In our simulations at the critical point, runs extending up to $R_\xi \approx 0.5$ require $2.5 \cdot 10^6$, $16 \cdot 10^6$ Metropolis sweeps for $L = 24$ and 32, respectively. In the parallel-tempering simulations for $L = 32$ of Ref. [31], the number of iterations discarded for thermalization varies between $8 \cdot 10^6$ and $500 \cdot 10^6$ (on average $13 \cdot 10^6$) sweeps. Taking into account that 22 different systems are simulated together, our simulations are shorter by a factor of 10 at least. If one were stopping the off-equilibrium runs at $R_\xi = 0.40$, one would gain an additional factor of 3 for this value of L.

In spite of the significant improvement with respect to equilibrium studies, the

computing time needed for a simulation scales as L^z even off-equilibrium, since we need to collect data at fixed R_ξ for all values of L. This requirement makes our method not suitable to investigate large system sizes. As errors are independent of system size as a consequence of the absence of self-averaging, the Monte Carlo time needed to obtain the same statistical errors scales also as L^z. This explains why the bulk of our statistics corresponds to $L \leq 32$. If we increase L, we should increase t at the same time, making simulations far too long. If β_c were known, one could change strategy and consider an infinite-volume method, i.e., perform fixed-length runs in very large volumes. However, one immediately realizes that there is no such thing as a free lunch with spin glasses. Indeed, with this strategy one would collect data up to $\xi \approx \xi_{\rm max}$, for a small value of $\xi_{\rm max}$ (the larger L, the smaller $\xi_{\rm max}$ for a given Monte Carlo time). But now, if $\xi_{\rm max}$ is small, there would be (large) scaling corrections proportional to $\xi^{-\omega}$. To reduce them, one should increase $\xi_{\rm max}$, which in turn requires an increase of the volume to keep size effects small. At the end of the story, it is easy to realize that such an infinite-volume method would be more problematic (size effects would introduce systematic out-of-control corrections) than the one we propose. The reasons are the same that convinced researchers to abandon infinite-volume methods to compute critical exponents in equilibrium, relying instead on finite-size scaling methods.

Let us now summarize our results. We have presented a new dynamic off-equilibrium method suitable for the determination of the critical temperature and for the critical exponents. Such a method represents a significant improvement with respect to previous ones. In particular, there is no need for L to be large enough to avoid finite-size effects—thus, a source of systematic errors is absent—nor does it require an *a priori* knowledge of the critical temperature. We have used the method to determine critical exponents and temperature for the $\pm J$ Ising model. With a relatively modest investment of computing time, thanks also to a very efficient GPU implementation of the MC dynamics, we obtain results that have a comparable precision with that of the estimates of Ref. [31], which are the most precise equilibrium estimates available today.

We want to stress that the method is completely general and can be applied to any pure or disordered system.

Chapter 6

Summary of results

In this thesis we presented a new out-of-equilibrium finite-size scaling analysis based on the Renormalization Group, consisting in an off-equilibrium generalization of the commonly used Binder crossing criterion. This method allowed us to obtain accurate estimates of the critical parameters of the tree-dimensional Edwards-Anderson model, the second most precise so far, which are compatible with previous equilibrium results reported in the following table

Method		T_c	ν	η	ω	z
Equilibrium	[39]	1.109(10)	2.45(15)	−0.375(10)	1.0(1)	
FSS	[31]	1.1019(29)	2.562(42)	−0.3900(36)	1.12(10)	
Dynamics at the critical temperature	[25]					6.86(16)
Our results		1.099(5)	2.47(10)	−0.39(1)	1.3(2)	6.80(15)

Indeed, these results represent a major improvement with respect to those obtained using previously developed out-of-equilibrium methods that we reviewed in this thesis. The most common problem was an underestimation of the errors leading to incompatible results, among different off-equilibrium techniques, and with respect to equilibrium estimates. We carefully described the data analysis procedure, verifying the validity of the approximations we used and showing how to control systematic deviations.

The precise determination of these parameters has been the subject of many efforts in the last thirty years, requiring the use of dedicated machines. The fact that we obtained precise estimates with only 3.1 computational years of a single GTX Titan GPU shows that the method we propose is robust and accurate. Indeed, it can be generalized to any continuous phase transition in order to determine the entire set of the critical parameters.

We also presented some general results we obtained for the GPU implementation of the Monte Carlo simulations we used. We developed a new memory access pattern for the cubic-lattice nearest neighbours, and for lagged-Fibonacci-like PRNGs. In particular we have shown that for the PRNGs we obtain better results with respect to the cuRand library provided by nVidia. Indeed, these results are of very general applicability since the cubic-stencil data structure and PRNGs are of widespread

use. We also implemented a multi-GPU version of the simulation which exhibits a very good strong scaling leading to scientifically interesting performances for large lattice sizes.

Appendix A

Simulations and fitting procedure details

In this appendix, we report some details of the Monte Carlo calculations and explain how we have fitted the large set of numerical data to obtain the estimates of β_c and of the critical exponents reported in the text.

The runs we performed are listed in Tables A.1 and A.2. In particular, we report the number of Monte Carlo Metropolis sweeps and the value $R_{\xi,\max}$ obtained at the end of each run. Such a value varies significantly, but most of the statistics corresponds to $R_{\xi,\max} \approx 0.5$. We have also performed some shorter runs with a significantly larger number of samples and, for $L \leq 16$, we have also runs that reach (barely) equilibrium. At the critical point and at equilibrium [31] $R_\xi = 0.652(3)$. We tried to have comparable statistical errors for all value of L, thereby avoiding fits to be dominated by the results corresponding to small L values. Up to $L = 24$ this is indeed the case. The number of samples for $L = 32$ is instead smaller. We have also data for $L = 48$ and $L = 64$ but the number of samples is small, so that they little contribute to the final results.

A.1 Fitting procedure

Let us discuss how data are fitted. To keep notation as compact as possible we indicate β and L with a single symbol A, so that a *Monte Carlo run* is specified by $A = (\beta, L)$. Each run provides estimates of the different observables averaged over the number of samples considered in the run. Indices a, b, ..., refer to the different independent runs, so that A_a corresponds to the values of β and L of run a. Observables estimates are labeled $\hat{U}_\alpha(A, t)$, and their corresponding expectation values $U_\alpha(A, t)$[1], where t is the Monte Carlo time, and also as $\hat{U}_\alpha(A, \hat{R}_\xi)$, if we parametrize the Monte Carlo evolution in terms of $\hat{R}_\xi = \hat{\xi}/L$. The Greek index α runs over the different observables: in our case it runs from 1 to 5. In the simulation we compute the observables not at all time steps but by using an exponential schedule, i.e., for

$$t = t_i = \lfloor 2^{i/4} \rfloor, \tag{A.1}$$

[1] We generally denote estimators with a hat and their expectation value without it.

where $\lfloor x \rfloor$ indicates the largest integer smaller than x. Latin indices i, j, k, run over the different times at which sampling has been performed. Finally, $\hat{R}_{\xi,ai} = \hat{R}_\xi(A_a, t_i)$ is the estimate of $\hat{R}_\xi$ at time t_i during run a, while the fitting functions are indicated as $F_\alpha(A, \hat{R}_\xi, \{\hat{p}_r\})$, where $\{\hat{p}_r\}$ corresponds to the set of parameters to be determined.

Using these notations, we start by defining a goodness-of-fit function

$$\begin{aligned} G &= \sum_{ai\alpha} \sum_{bj\beta} \left[\hat{U}_\alpha(A_a, \hat{R}_{\xi,ai}) - F_\alpha(A_a, \hat{R}_{\xi,ai}, \{\hat{p}_r\})\right] K_{ai\alpha;bj\beta} \\ &\quad \left[\hat{U}_\beta(A_b, \hat{R}_{\xi,bj}) - F_\beta(A_b, \hat{R}_{\xi,bj}, \{\hat{p}_s\})\right], \end{aligned} \tag{A.2}$$

where the kernel K has yet to be specified but symmetric under the exchange of the two groups of indices $\{ai\alpha\} \leftrightarrow \{bj\beta\}$. The parameters $\{\hat{p}_r\}$ are obtained by requiring G to be stationary with respect to their variations. If K is positive definite, the solution can equivalently be obtained by minimizing G with respect to $\{\hat{p}_r\}$. It is easy to convince oneself that this procedure is correct, whatever the kernel K is. The argument goes as follows. Let us indicate with $U_\alpha(A_a, R_\xi)$ the exact value of the quantity U_α for the given values of the parameters. Then, assume that the functions F_α provide an exact parametrization of the data, so that there are values $\{p_r\}$ of the parameters so that

$$U_\alpha(A_a, R_\xi) = F_\alpha(A_a, R_\xi, \{p_r\}). \tag{A.3}$$

The quantities $\{p_r\}$ are of course the parameters we wish to estimate. If we use the exact values for U_α and $\{p_r\}$, we have $G = 0$, which implies that the parameters $\{p_r\}$ are solutions of the stationary equations for any K if $\hat{U}_\alpha = U_\alpha$. We can thus use a simple continuity argument. Suppose that we estimate $\{p_r\}$ using a given set of estimates of U_α. Call this estimate $\{\hat{p}_r\}_1$. If we double the statistics, we obtain more precise estimates U_α and a new estimate $\{\hat{p}_r\}_2$. If we increase again the precision on U_α, we obtain an estimate $\{\hat{p}_r\}_3$, and so on. It is obvious that, by increasing the statistics, $\hat{U}_\alpha$ converges to U_α. Correspondingly, the estimates $\{\hat{p}_r\}_1$, $\{\hat{p}_r\}_2$, $\{\hat{p}_r\}_3$ converge to $\{p_r\}$. Again, K plays no role here.

One can also analyze the issue from a mathematical point of view. One can prove that the solution of the stationary equations provides a correct estimator of $\{p_r\}$. In order to show this let us define primed greek indices as $\alpha' = (a, i, \alpha)$ and the following shorter notation

$$\hat{U}_\alpha(A_a, \hat{R}_{\xi,ai}) = \hat{U}_{\alpha'}, \quad F_\alpha(A_a, \hat{R}_{\xi,ai}, \{\hat{p}_r\}) = F_{\alpha'}(\{\hat{p}_r\}), \quad K_{ai\alpha;bj\beta} = K_{\alpha'\beta'}.$$

The stationarity equation for G reads

$$-\frac{\partial G}{\partial \hat{p}_u} = 2 \sum_{\alpha'\beta'} \left\{ \frac{\partial F_{\alpha'}(\{\hat{p}_r\})}{\partial \hat{p}_u} K_{\alpha'\beta'} \left[\hat{U}_{\beta'} - F_{\beta'}(\{\hat{p}_s\})\right] \right\} = 0, \tag{A.4}$$

since $K_{\alpha'\beta'} = K_{\beta'\alpha'}$. Now, in the limit of $N_{\beta'} \to \infty$ we obtain that $\hat{U}_{\beta'} \to U_{\beta'}$ so that the stationarity condition (A.4) is satisfied if $\{\hat{p}_r\} \to \{p_r\}$, indipendently on the choice of $K_{\alpha'\beta'}$. I we now assume a linear dependence of the functions $F_{\alpha'}$ on the parameters $\{\hat{p}_r\}$, *i.e.*

$$F_{\alpha'}(\{\hat{p}_r\}) = \sum_r \hat{p}_r \, F_{\alpha',r},$$

the stationarity condition (A.4) becomes

$$-\frac{\partial G}{\partial \hat{p}_u} = 2\sum_s \sum_{\alpha'\beta'} \left\{ F_{\alpha',u}\, K_{\alpha'\beta'} \left[\hat{U}_{\beta'} - \hat{p}_s\, F_{\beta',s} \right] \right\} = 0. \tag{A.5}$$

The last expression is equivalent to

$$\sum_{\alpha'\beta'} F_{\alpha',u}\, K_{\alpha'\beta'}\, \hat{U}_{\beta'} = \sum_s \sum_{\alpha'\beta'} (F_{\alpha',u}\, K_{\alpha'\beta'}\, F_{\beta',s})\, \hat{p}_s = \sum_s B_{us}\, \hat{p}_s, \tag{A.6}$$

where we have defined the matrix $B_{us} = \sum_{\alpha'\beta'} F_{\alpha',u}\, K_{\alpha'\beta'}\, F_{\beta',s}$, whose inverse is denoted by $(B^{-1})_{us}$. The parameters $\{\hat{p}_r\}$ are determined by

$$\hat{p}_r = \sum_u \sum_{\alpha'\beta'} (B^{-1})_{ru}\, F_{\alpha',u}\, K_{\alpha'\beta'}\, \hat{U}_{\beta'}. \tag{A.7}$$

Now, whether the estimates $\{\hat{p}_r\}$ are biased or not depends on the estimatrs $\hat{U}_{\alpha'}$. Since these quantities are Binder cumulants which have been computed through the jackknife method we can write

$$\langle \hat{U}_{\alpha'} \rangle = U_{\alpha'} + \frac{1}{N_{\alpha'}^2} B_{\alpha'}, \tag{A.8}$$

where the bias is at the order $N_{\alpha'}^2$. Hence, the estiates $\hat{p}_r$ will also be biased since

$$\begin{aligned}
\langle \hat{p}_r \rangle &= \sum_u \sum_{\alpha'\beta'} (B^{-1})_{ru}\, F_{\alpha',u}\, K_{\alpha'\beta'}\, \langle \hat{U}_{\beta'} \rangle \\
&= \sum_u \sum_{\alpha'\beta'} (B^{-1})_{ru}\, F_{\alpha',u}\, K_{\alpha'\beta'} \left(U_{\beta'} + \frac{1}{N_{\beta'}^2} B_{\beta'} \right) \\
&= \sum_u \sum_{\alpha'\beta'} (B^{-1})_{ru}\, F_{\alpha',u}\, K_{\alpha'\beta'} \left(\sum_s p_s\, F_{\beta',s} + \frac{1}{N_{\beta'}^2} B_{\beta'} \right) \\
&= p_r + \sum_u \sum_{\alpha'\beta'} (B^{-1})_{ru}\, F_{\alpha',u}\, K_{\alpha'\beta'}\, \frac{1}{N_{\beta'}^2} B_{\beta'}.
\end{aligned} \tag{A.9}$$

Clearly, if $\hat{U}$ would have been a linear estimator then no bias would have occurred. Hence, the estimators $\{\hat{p}_r\}$ are always correct in the limit $N_{\beta'} \to \infty$.

Although the choice of K is irrelevant for the correctness of the estimator, different K provide estimators of different precision: the error on the estimated parameters $\{\hat{p}_r\}$ depends on the choice of the kernel.

It is also important to stress that care should be taken in determining the error on the estimated parameters—we use the jackknife method—and that G is not in general a χ^2 variable—hence χ^2/DOF, where DOF is the number of degrees of freedom of the fit, cannot be taken as an indicator of the goodness of the fit. It is easy to verify that the optimal choice for K, i.e., the one that provides results with the smallest error, corresponds to taking $K = S^{-1}$, where S is the exact covariance of the data.

Let us show this result. The optimal choice of K is the one tha minimizes the covariance of the estimates $\{\hat{p}_r\}$ which can be written as

$$\begin{aligned}
&\operatorname{cov}(\hat{p}_r, \hat{p}_s) \\
=&\sum_{u\,v} \sum_{\alpha'\beta'\gamma'\delta'} (B^{-1})_{ru}\, F_{\alpha',u}\, K_{\alpha'\beta'} \operatorname{cov}(\hat{U}_{\beta'}, \hat{U}_{\delta'})\, K_{\delta'\gamma'}\, F_{\gamma',v}\, (B^{-1})_{vs} \\
=&\sum_{u\,v} \sum_{\alpha'\beta'\gamma'\delta'} (B^{-1})_{ru}\, F_{\alpha',u}\, K_{\alpha'\beta'}\, S_{\beta'\delta'}\, K_{\delta'\gamma'}\, F_{\gamma',v}\, (B^{-1})_{vs},
\end{aligned} \tag{A.10}$$

and the minimum condition reads

$$\frac{\partial \operatorname{cov}(\hat{p}_r, \hat{p}_s)}{\partial K_{\mu'\nu'}} = 0. \tag{A.11}$$

It is possible to write the variation $\delta(B^{-1})_{rs}$ in terms of that of the direct matrix δB_{uv}. Since $\delta(\sum_s (B^{-1})_{us}\, B_{sv}) = \delta(\delta_{uv}) = 0$ we can write $\delta(B^{-1})_{ru} = -\sum_{wz} (B^{-1})_{rw} (B^{-1})_{uz}\, \delta B_{wz}$ leading to

$$\frac{\partial (B^{-1})_{ru}}{\partial K_{\mu'\nu'}} = -\sum_{wz} (B^{-1})_{rw}\, F_{\mu',w}\, F_{\nu',z}\, (B^{-1})_{zu}. \tag{A.12}$$

Finally, we can write the minimum condition (A.11) as

$$\begin{aligned}
&-\frac{\partial \operatorname{cov}(\hat{p}_r, \hat{p}_s)}{\partial K_{\mu'\nu'}} \\
=&2\sum_{u\,v} \sum_{\alpha'\beta'\gamma'\delta'} \sum_{wz} (B^{-1})_{rw}\, F_{\mu',w}\, F_{\nu',z}\, (B^{-1})_{zu}\, F_{\alpha',u}\, K_{\alpha'\beta'}\, S_{\beta'\delta'}\, K_{\delta'\gamma'}\, F_{\gamma',v}\, (B^{-1})_{vs} \\
-&2\sum_{u\,v} \sum_{\gamma'\delta'} (B^{-1})_{ru}\, F_{\mu',u}\, S_{\nu'\delta'}\, K_{\delta'\gamma'}\, F_{\gamma',v}\, (B^{-1})_{vs} = 0,
\end{aligned} \tag{A.13}$$

which is satisfied if one choses $K_{\alpha'\beta'} = (S^{-1})_{\alpha'\beta'}$.

In our case, it is not feasible to compute S (we have both strong time correlations and correlations among the different observables). We have therefore chosen a diagonal kernel. To check how results depend on the correlation of the data, two different fits have been performed:

i) We perform a fit that would be optimal for uncorrelated data. We consider

$$G = \sum_{ai\alpha} \left[\hat{U}_\alpha(A_a, \hat{R}_{\xi,ai}) - F_\alpha(A_a, \hat{R}_{\xi,ai}, p)\right]^2 / \sigma^2_{ai\alpha}, \tag{A.14}$$

where $\sigma_{ai\alpha}$ is the error on $\hat{U}_\alpha(A_a, R_{\xi,ai})$.

ii) If the data in each run were perfectly correlated, it would be natural to require the contribution of each run to G to be independent of the number of measurements performed in the run. If this were the case, an optimal fit would be obtained (we are still ignoring correlations between the different observables, that should play a minor role) by considering

$$G = \sum_{ai\alpha} \left[\hat{U}_\alpha(A_a, \hat{R}_{\xi,ai}) - F_\alpha(A_a, \hat{R}_{\xi,ai}, p)\right]^2 / (\sigma^2_{ai\alpha} N_a), \tag{A.15}$$

where N_a is the number of points of run a that are considered in the fit (note that the index i runs from 1 to N_a).

Both fits provide statistically correct estimates of the parameters, but *a priori* the errors on the parameters are expected to be different. Note also that, in the presence of subleading corrections, the two fits give different weight to the runs with small and large L. Indeed, as L increases, the total time of the simulation increases and so does the number of measurements. Therefore, fit (i) gives more weight to the runs with larger L values with respect to fit (ii). Hence, from the point of view of the systematic errors, results of fit (i) should be more reliable than those of fit (ii).

A.1.1 Results of the fits of the renormalized couplings

Each renormalized coupling $U(t, L, \beta)$ was fitted to

$$U(t, L, \beta) = P_1(R_\xi) + P_2(R_\xi)(\beta - \beta_c)L^{1/\nu} + P_3(R_\xi)L^{-\omega}, \tag{A.16}$$

with $P_1(R_\xi)$, $P_1(R_\xi)$, and $P_3(R_\xi)$ polynomials of degree 6, 3, and 3, respectively. The five RG couplings were fitted together. We perform both fits of type (i) and of type (ii) The results of the fits for β_c, ω, and ν are reported in Tables A.3, A.4, and A.5. They correspond to $\xi_{\rm min} = 4$.

First, we should note the errors on the estimates obtained by using fit (i) and (ii) are essentially identical. Only in a few cases is fit (ii) slightly less precise than fit (i). Therefore, from a statistical point of view, the two fits are essentially equivalent. If we compare the estimates obtained in the two fits, we observe in all cases some systematic deviations, of size comparable with the statistical errors: for instance, for a given $L_{\rm min}$ and $R_{\xi,\rm min}$, fit (ii) always provides an estimate of β_c that is slightly larger than that of fit (i). Correspondingly, the estimates of ω are smaller. This is probably the effect of the next-to-leading scaling corrections, which are treated differently in the two fits. In practice, this implies the presence of neglected scaling corrections that may give errors of the same order of the statistical errors. A subjective (hopefully conservative) estimate of these effects is presented below.

Second, note that the fits with $R_{\xi,\rm min} = 0$ have errors essentially identical to those obtained with $R_{\xi,\rm min} = 0.20$. This is clearly related to the fact that the bulk of the statistics corresponds to $R_\xi > 0.20$. Indeed, since we are reporting results corresponding to $\xi_{\rm min} = 4$, only results with $L \geq 24$ contribute in the region $R_\xi < 0.20$. Since this is a rather small set of data, they do not play much role in the analysis, hence results are the same for $R_{\xi,\rm min} = 0$ and $R_{\xi,\rm min} = 0.20$.

In Figs. A.1 and A.2 we report the scaling functions $P_2(R_\xi)$ and $P_3(R_\xi)$. The errors are determined with a jackknife procedure at *fixed parametrization*. They do not include the systematic error due to the parametrization, hence errors could be underestimated, especially for $R_\xi \lesssim 0.2$, where we have a limited number of data points. From the figures, it is evident that U_{22} and U_{23} play no role in the determination of β_c and ν: the β dependence of these two quantities is too small. For U_{14} and U_{13}, $P_2(R_\xi)$ is determined with increasing accuracy as R_ξ increases. These two renormalized couplings play an increasingly important role in the determination of β_c and ν as R_ξ increases. The usual Binder parameter U_4 appears to be the most reliable quantity. Note, however, that it approximately vanishes for $R_\xi \approx 0.20$, hence

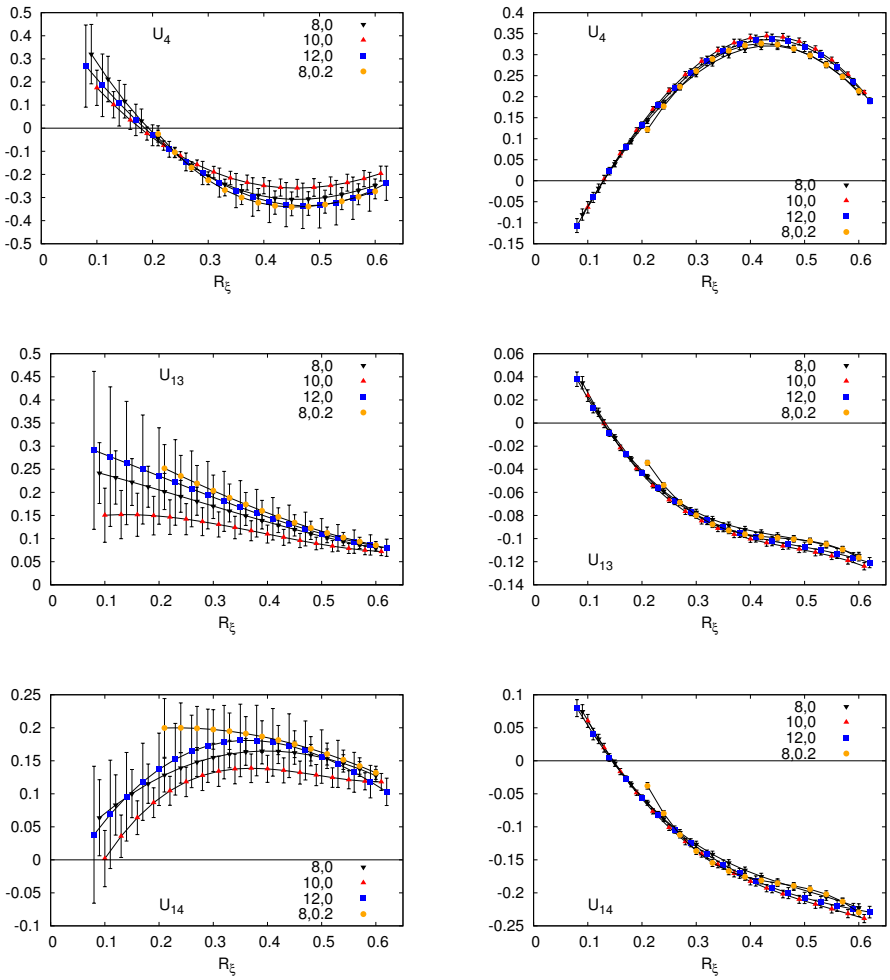


Figure A.1. We report the scaling functions $P_2(R_\xi)$ (left) and $P_3(R_\xi)$ (right) for U_4 (top), U_{13} (middle), and U_{14} (bottom). We report the results obtained for $L_{\rm min} = 8$, $L_{\rm min} = 10$, and $L_{\rm min} = 10$, taking always $\xi_{\rm min} = 4$ and $R_{\xi,\rm min} = 0$. Data labeled "8,0.2" are obtained taking $L_{\rm min} = 8$ and $R_{\xi,\rm min} = 0.2$.

requiring runs to go much beyond this value to allow us to determine accurately the critical parameters.

The plot of the correction-to-scaling function $P_3(R_\xi)$ shows that corrections to scaling are usually larger at equilibrium than during the off-equilibrium transient, the only exception being U_4. This implies that equilibrium estimates are those that are most affected by systematic deviations due to corrections to scaling. It is also interesting to observe that $P_3(R_\xi) \approx 0$ for $R_\xi \approx 0.13, 0.15$ for all observables. The apparent stability of the results is due to the fact that we are reporting data at fixed parametrization, i.e., for a fixed order of the polynomials. If we increase the order of the polynomial parametrizing $P_3(R_\xi)$, we obtain a curve that deviates from that reported in the figure for $R_\xi \lesssim 0.20$. Hence, the graphs we report cannot be trusted below this value of R_ξ. This is not surprising, since in this range of R_ξ we only have

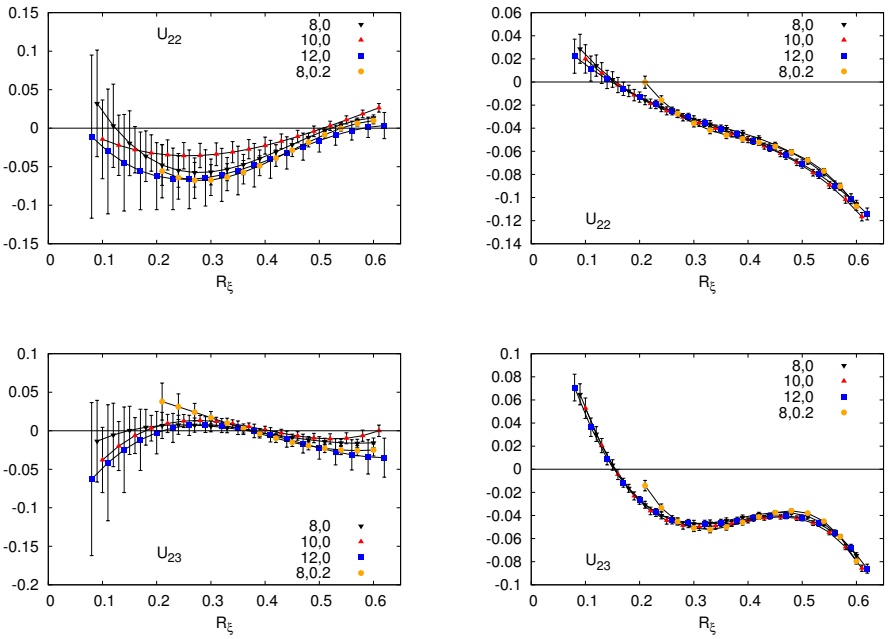


Figure A.2. We report the scaling functions $P_2(R_\xi)$ (left) and $P_3(R_\xi)$ (right) for U_{22} (top), U_{23} (bottom). We report the results obtained for $L_{\min} = 8$, $L_{\min} = 10$, and $L_{\min} = 10$, taking always $\xi_{\min} = 4$ and $R_{\xi,\min} = 0$. Data labeled "8,0.2" are obtained taking $L_{\min} = 8$ and $R_{\xi,\min} = 0.2$.

data with $L = 32, 48$, and 64, which are sufficient to estimate the correction term.

To estimate β_c let us consider the results of fit (i). Results corresponding to $L_{\min} = 8, 10, 12$ lie in the intervals $0.906 \le \beta_c \le 0.913$, $0.910 \le \beta_c \le 0.926$, and $0.905 \le \beta_c \le 0.913$, respectively (we consider the central value plus/minus one error bar). The results for $L_{\min} = 8$ and 12 are consistent, allowing us to exclude the presence of systematic effects at this level of precision. Therefore, we set

$$\beta_c = 0.910(4). \tag{A.17}$$

The results for $L_{\min} = 10$ predict a slightly larger value, but are still consistent with the estimate above. The estimates of ω and β_c are strongly correlated. To take this correlations into account in fixing the final estimate of ω and its error, we have repeated the fits, keeping β_c fixed. The results are reported in Table A.6. The estimates of ω are stable with $L_{\min}$, the error on β_c being the main source of uncertainty. The exponent ν is instead much less correlated with β_c and shows a slight systematic drift as $L_{\min}$ is increased from 8 to 10, so that we only take into account the results with $L_{\min} = 10$ and 12. We obtain finally

$$\omega = 1.3(2), \qquad \nu = 2.47(10). \tag{A.18}$$

A.1.2 Estimates of the exponents η and z

Let us now discuss the determination of the exponent η. Eq. (A.16) can be generalized to the susceptibility, obtaining

$$\chi L^{-(2-\eta)} \overline{u}_h(\beta)^{-2} = f_\chi(R_\xi, \epsilon) + L^{-\omega} g_\chi(R_\xi, \epsilon) + \dots, \tag{A.19}$$

where $\overline{u}_h(\beta)$ is a nonlinear scaling field (see the discussion in Refs. [105, 39]). Taking the logarithm of the previous expression, making the same approximations as for U_α, we obtain the expansion

$$\begin{aligned} \ln \chi &= (2-\eta) \ln L + P_1(R_\xi) + (\beta - \beta_c) L^{1/\nu} P_2(R_\xi) + \\ &\quad L^{-\omega} P_3(R_\xi) + P_4(\beta). \end{aligned} \tag{A.20}$$

The last term $P_4(\beta)$ is the contribution of the nonlinear scaling field $\overline{u}_h(\beta)$. A good parametrization is obtained by taking $P_1(R_\xi)$, $P_2(R_\xi)$, $P_3(R_\xi)$ as polynomials of degree 6, 3, 3, respectively, as before. For the $P_4(\beta)$, we set $P_4 = a_4\beta$.

Fits have been made keeping β_c, ω, and ν fixed to their estimates. Results are reported in Table A.7. Statistical errors are tiny and can be neglected. The main source of error on the results is due to the uncertainty on the estimates of β_c and ω. If we vary β_c and ω, setting $\beta_c = 0.914$ and $\omega = 1.1$, the exponent η varies by -0.01, changing from -0.39 to -0.40. The final estimate is therefore

$$\eta = -0.39(1). \tag{A.21}$$

Finally, we determine the dynamic exponent z. We shall use the scaling form in terms of tL^{-z}, i.e.,

$$R(t, L, \beta) = f_R(tL^{-z}, \epsilon) + u_\omega(\beta) L^{-\omega} g_R(tL^{-z}, \epsilon) + \dots \tag{A.22}$$

At variance with the previous analyses, at fixed tL^{-z}, data are not linear in ϵ, nor are corrections temperature-independent. Hence, we expand $f_R(tL^{-z}, \epsilon)$ to second order in ϵ and $g_R(tL^{-z}, \epsilon)$ to first order. Hence, we write the fitting function as

$$\begin{aligned} R(t, L, \beta) &= P_1(tL^{-z}) + (\beta - \beta_c) L^{1/\nu} P_2(tL^{-z}) + \\ &\quad (\beta - \beta_c)^2 L^{2/\nu} P_3(tL^{-z}) + L^{-\omega} P_4(tL^{-z}) + \\ &\quad (\beta - \beta_c) L^{1/\nu - \omega} P_5(tL^{-z}). \end{aligned} \tag{A.23}$$

The scaling functions $P_1(tL^{-z})$, ..., $P_5(tL^{-z})$ are approximated as polynomials in $x = \exp(-Lt^{-1/z}) t^{1/z}/L$ (with this choice of variable, data are mapped in the interval $[0, x_{\rm max}]$, with $x_{\rm max} \lesssim 0.3$). To have a reasonable fit (in fit (i) we require G to be of the order of the number of data points), we need to consider high-order polynomials. The leading term P_1 is approximated by a 12th-order polynomial in x, while for the other scaling functions we use 5th-order polynomials. We have performed fits for both U_4 and R_ξ, fixing β_c, ω, and ν as before. The results are reported in Table A.8.

The estimates of z obtained by fitting U_4 show a tiny dependence on $R_{\xi,\rm min}$—they slightly increase as $R_{\xi,\rm min}$ increases— and on $L_{\rm min}$—they decrease as $L_{\rm min}$ increases. These changes are, however, significantly smaller than the systematic error due to

the uncertainty on β_c and ω, which changes z by ± 0.10. The estimates of z obtained by fitting R_ξ show a strong dependence on $R_{\xi,\mathrm{min}}$. Only for $R_{\mathrm{xi,min}} = 0.2$ and 0.4 are the estimates of z consistent with those obtained from the analysis of U_4. As final result we take

$$z = 6.80(15), \tag{A.24}$$

which is consistent with all estimates.

Table A.1. Details of the Monte Carlo simulations. For each run we report the values of β and L, the number N_s of samples, the Monte Carlo Metropolis sweeps (MCS), the maximum value of R_ξ obtained, and the GPU time needed on a Titan GPU. Here $8 \leq L \leq 16$.

L	β	N_s	$R_{\xi,\max}$	MCS	t (hours)
8	0.880	524288	0.613	2^{20}	1.4
	0.890	524288	0.623	2^{20}	1.4
		8388608	0.564	2^{12}	1
		8388608	0.564	2^{12}	1
	0.896	8388608	0.565	2^{12}	1
	0.902	524288	0.635	2^{20}	1.4
		8388608	0.566	2^{12}	1
	0.906	8388608	0.566	2^{12}	1
	0.910	524288	0.643	2^{20}	1.4
		8388608	0.567	2^{12}	1
10	0.880	524288	0.613	2^{20}	2.7
	0.890	524288	0.624	2^{20}	2.7
		4194304	0.512	2^{13}	0.7
	0.896	4194304	0.511	2^{13}	0.7
	0.902	524288	0.636	2^{20}	2.7
		4194304	0.509	2^{13}	0.7
	0.906	4194304	0.508	2^{13}	0.7
	0.910	524288	0.646	2^{20}	2.7
		4194304	0.507	2^{13}	0.7
12	0.880	524288	0.611	2^{20}	4.3
	0.890	524288	0.623	2^{20}	4.3
		4194304	0.601	2^{17}	5
	0.896	4194304	0.605	2^{17}	5
	0.902	524288	0.637	2^{20}	4.3
		16777216	0.608	2^{17}	0.3
	0.906	4194304	0.610	2^{17}	5
	0.910	524288	0.646	2^{20}	4.3
		16777216	0.612	2^{17}	5
16	0.880	552960	0.609	2^{23}	123
		2097152	0.578	2^{19}	29.2
	0.890	532480	0.624	2^{23}	118
		2097152	0.582	2^{19}	29.2
	0.896	2097152	0.584	2^{19}	29.2
		16777216	0.370	2^{17}	58.3
	0.902	521728	0.638	2^{23}	116
		2097152	0.585	2^{19}	29.2
		16777216	0.366	2^{15}	13.7
	0.906	2097152	0.586	2^{19}	29.2
	0.910	524288	0.648	2^{23}	117
		2097152	0.585	2^{19}	29.2

Table A.2. Details of the Monte Carlo simulations. For each run we report the values of β and L, the number N_s of samples, the Monte Carlo Metropolis sweeps (MCS), the maximum value of R_ξ obtained, and the GPU time needed on a Titan GPU. Here $20 \leq L \leq 64$.

L	β	N_s	$R_{\xi,\max}$	MCS	t (hours)
20	0.890	4194304	0.485	2^{19}	136
	0.896	2097152	0.536	2^{20}	136
		16777216	0.270	2^{15}	34
	0.902	2097152	0.533	2^{20}	136
		19398656	0.267	2^{15}	39
	0.906	2097152	0.530	2^{20}	136
	0.910	2097152	0.528	2^{20}	136
24	0.880	512000	0.596	2^{24}	657
	0.890	532480	0.606	2^{24}	684
		1048576	0.500	2^{21}	168
		2097152	0.500	2^{21}	337
	0.896	2097152	0.496	2^{21}	337
		8650752	0.381	2^{19}	347
	0.902	520192	0.617	2^{24}	668
		2097152	0.491	2^{21}	337
	0.906	2097152	0.487	2^{21}	337
	0.910	516096	0.622	2^{24}	663
		2097152	0.483	2^{21}	337
32	0.880	331904	0.521	2^{24}	1010
	0.890	295328	0.515	2^{24}	899
	0.902	298272	0.505	2^{24}	908
	0.906	5783552	0.286	2^{20}	1100
	0.910	307264	0.495	2^{24}	936
48	0.880	65536	0.428	2^{26}	2964
	0.890	65536	0.412	2^{26}	2964
	0.902	65536	0.394	2^{26}	2964
	0.910	65536	0.382	2^{26}	2964
64	0.890	10304	0.321	2^{27}	2008
	0.902	11552	0.302	2^{27}	2251

Table A.3. Estimates of β_c for $\xi_{\min} = 4$ for the two types of fits discussed in the paper for different $L_{\min}$ and $R_{\xi,\min}$.

	Fit (i)			Fit (ii)		
$L_{\min}/R_{\xi,\min}$	0	0.20	0.40	0	0.20	0.40
8	0.911(2)	0.909(2)	0.909(3)	0.912(3)	0.910(2)	0.911(3)
10	0.916(4)	0.913(3)	0.920(6)	0.918(5)	0.915(5)	0.923(8)
12	0.909(4)	0.909(3)		0.913(4)	0.910(3)	

Table A.4. Estimates of ω for $\xi_{\min} = 4$ for the two types of fits discussed in the paper for different $L_{\min}$ and $R_{\xi,\min}$.

	Fit (i)			Fit (ii)		
$L_{\min}/R_{\xi,\min}$	0	0.20	0.40	0	0.20	0.40
8	1.22(10)	1.30(9)	1.26(12)	1.20(10)	1.28(9)	1.21(11)
10	1.05(14)	1.19(13)	0.88(18)	0.99(17)	1.15(15)	0.81(18)
12	1.28(20)	1.39(18)		1.12(18)	1.37(17)	

Table A.5. Estimates of ν for $\xi_{\min} = 4$ for the two types of fits discussed in the paper for different $L_{\min}$ and $R_{\xi,\min}$.

	Fit (i)			Fit (ii)		
$L_{\min}/R_{\xi,\min}$	0	0.20	0.40	0	0.20	0.40
8	2.57(6)	2.39(6)	2.39(5)	2.31(5)	2.35(6)	2.34(6)
10	2.52(6)	2.49(6)	2.43(6)	2.53(6)	2.50(6)	2.42(6)
12	2.46(8)	2.41(8)		2.47(8)	2.40(8)	

Table A.6. Estimates of ν and ω for $\xi_{\min} = 4$ and $R_{\xi,\min}$ at fixed $\beta_c = 0.910$ (in brackets the change of the central value as β_c is increased by 0.004).

	Fit (i)			Fit (ii)		
$L_{\min}$	8	10	12	8	10	12
ν	2.37(5)[+0.03]	2.49(6)[+0.02]	2.46(7)[+0.02]	2.29(5)[+0.02]	2.49(6)[+0.02]	2.45(7)[+0.02]
ω	1.25(3)[-0.16]	1.30(5)[-0.18]	1.24(8)[-0.20]	1.29(3)[-0.16]	1.30(5)[-0.17]	1.27(9)[-0.20]

Table A.7. Summary of the estimates of η for $\xi_{\min} = 4$ at fixed $\beta_c = 0.910$, $\omega = 1.3$, $\nu = 2.47$. Estimates are insensitive to changes of ν by ± 0.010. In brackets the change of the last three digits of the estimate as ω is changed to 1.1 and β_c to 0.914. The minus sign in brackets that η becomes approximately equal to -0.40 as β_c increases.

$L_{\min}$	8	10	12	16
Fit (i)	−0.3925(3)[−103]	−0.3904(4)[−106]	−0.3909(6)[−113]	−0.3907(10)[−124]
Fit (ii)	−0.3923(3)[−103]	−0.3901(4)[−105]	−0.3910(6)[−114]	−0.3912(10)[−125]

Table A.8. Results of the fits [fit of type (i)] of U_4 and R_ξ at fixed $\beta_c = 0.910$, $\omega = 1.3$, $\nu = 2.47$. In brackets the change of the last two digits if β_c is 0.914 and ω is 1.1. We only use data $U(t, L, \beta)$ satisfying $\xi(t, L, \beta) > \xi_{\min}$, $L \geq L_{\min}$, $R_\xi(t, L, \beta) > R_{\xi,\min}$.

	$R_{\xi,\min}/L_{\min}$	8	10	12	16
U_4	0	6.854(4)[+94]	6.842(5)[+95]	6.837(6)[+91]	6.817(11)[+81]
	0.20	6.858(4)[+98]	6.846(5)[+98]	6.840(7)[+94]	6.826(11)[+86]
	0.40	6.870(5)[+108]	6.853(6)[+104]	6.845(11)[+96]	6.846(19)[+88]
R_ξ	0	6.736(6)[+37]	6.719(7)[+34]	6.703(8)[+31]	6.604(13)[+12]
	0.20	6.834(3)[+85]	6.826(4)[+84]	6.822(5)[+82]	6.817(7)[+81]
	0.40	6.841(4)[+87]	6.826(5)[+84]	6.813(8)[+76]	6.812(15)[+70]

Appendix B

Sublattice results

In this appendix we want to report some of the results we collected for the sublattice observables we described in Chapter 5. We show these results separately since they belong to a different data set with respect to those presented in Chapter 5 with much less disorder realizations and a few lattice sizes $L = 8, 16, 32, 64$ and $L = 128$. Moreover, the scaling analysis was performed assuming that $\beta_c = 0.902$ as estimated in [39].

We will show finite-size effects of the centered susceptibility χ_c and correlation length ξ_c for varying values of the sublattice and core sizes. We will see that the centered sublattice observables behave more similarly to those of real lattices of the same size the smaller one chooses the size of the core. We also report the behaviour of Binder cumulants.

Finally, we report the critical dynamic scaling behaviour. Some care is necessary in understanding which of the many observables are expected to show the same scaling behaviour.

B.1 Comparison with real lattices

As a first step let us consider the time behaviour of the centered susceptibility χ_c and correlation length ξ_c for different values of L_s and L_c. From figures B.1 and B.2 it appears that data associated to small values of the core volume L_c are closer to those of the corresponding real lattice. This can be understood since the smaller the centered volume with respect to which correlations are calculated the larger the distance from the boundary of the system. Hence the dependence on different boundary condition is small for small times when the correlation length is not sufficiently large to discriminate between the two systems. In figure B.3 we keep fixed the values of the sublattice and core sizes $L_s = 8$ and $L_c = 2$ while varying the embedding lattice size L. We can see that all curves share the early time behaviour and then separate. The first to separate is the real lattice. For sublattice curves, the bigger the embedding lattice the longer the curves superpose.

B.2 Scaling at the critical temperature

We now show the critical dynamic scaling for two different sets of observables: the non-centered version of Binder cumulants and the centered version of the susceptibility χ_c and correlation length ξ_c.

B.2.1 Binder cumulants

In figure B.4 we report the scaling behaviour of the five Binder cumulants we defined in Chapter 5 computed for both real lattices and sublattices.

B.2.2 Centered susceptibility and correlation length

In figure B.5 we report the thermodynamic-limit behaviour for the centered susceptibility and correlation length for both real lattices and sublattices. Again it is possible to see data superposition at early times. Finally, in figure B.6 we report the values of $\chi_c\, L^{-(2-\eta)}$ as a function of R^c_ξ. We consider two different sets of scaling observables belonging to $S = L/L_s = 2$, $C = L_s/L_c = 2$ and $S = 2$, $C = 4$. Our out-of-equilibrium finite-size scaling ansatz holds good also for sublattice observables.

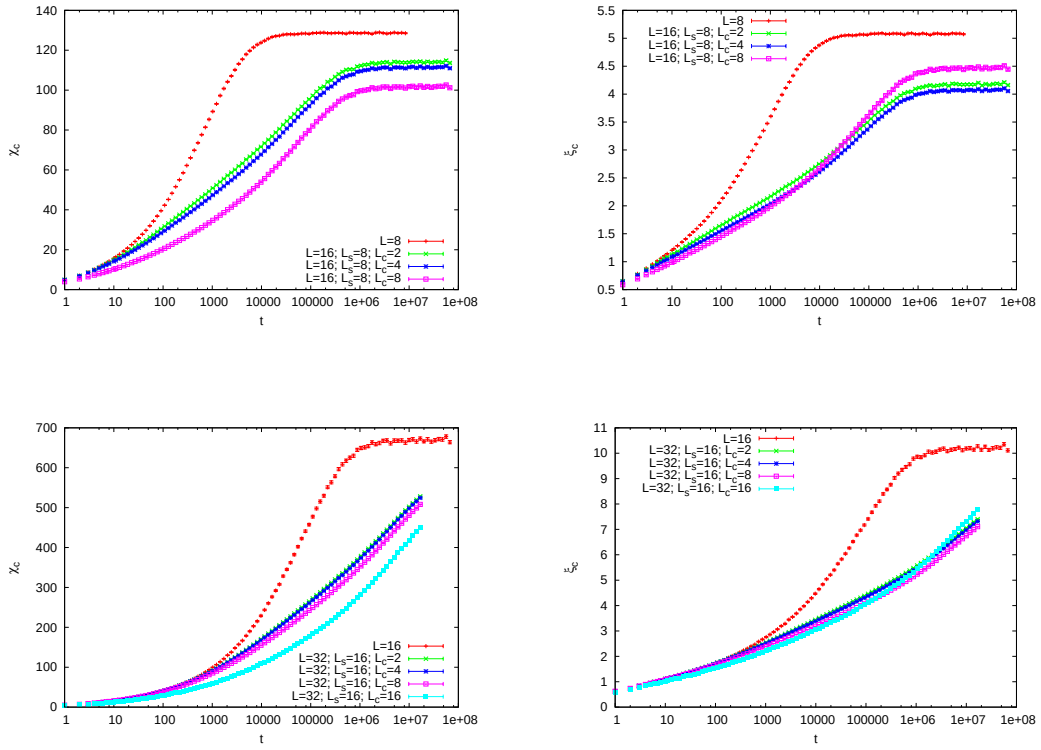


Figure B.1. Left column: χ_c as a function of t for $L_s = 8$ and $L_s = 16$ and different values of the core size from the smallest $L_c = 2$ to the largest $L_c = L_s$. Right column: same as left one for ξ_c. The smaller the value of L_c the better the agreement with the real lattice data.

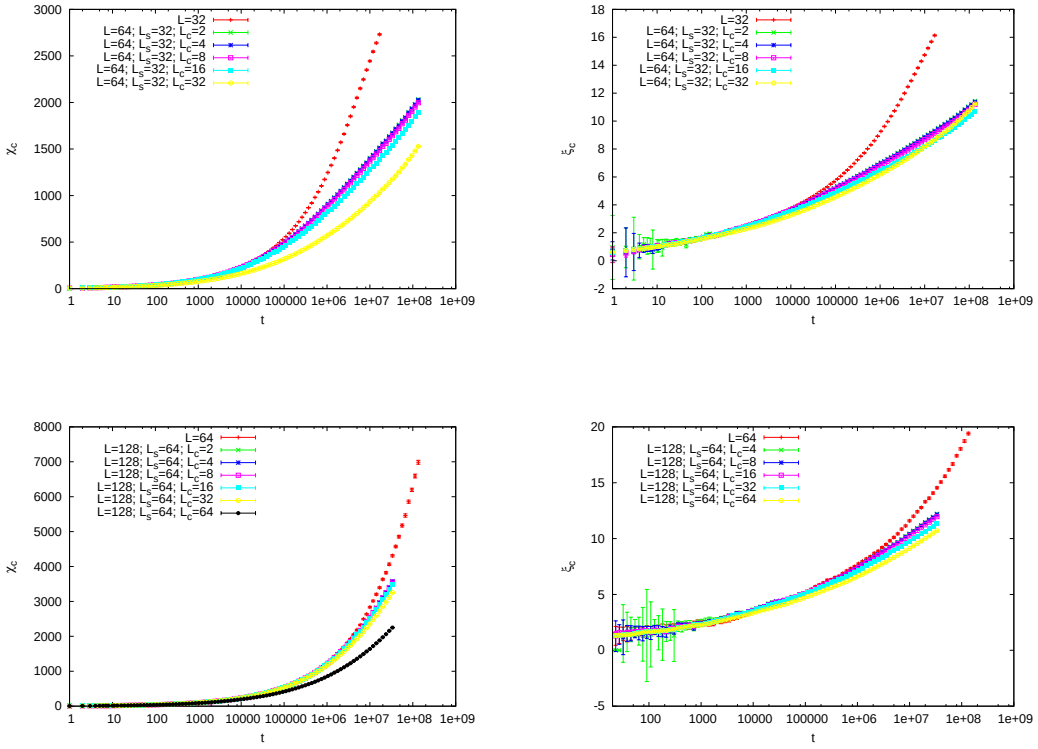


Figure B.2. Left column: χ_c as a function of t for $L_s = 32$ and $L_s = 64$ and different values of the core size from the smallest $L_c = 2$ to the largest $L_c = L_s$. Right column: same as left one for ξ_c. The smaller the value of L_c the better the agreement with the real lattice data.

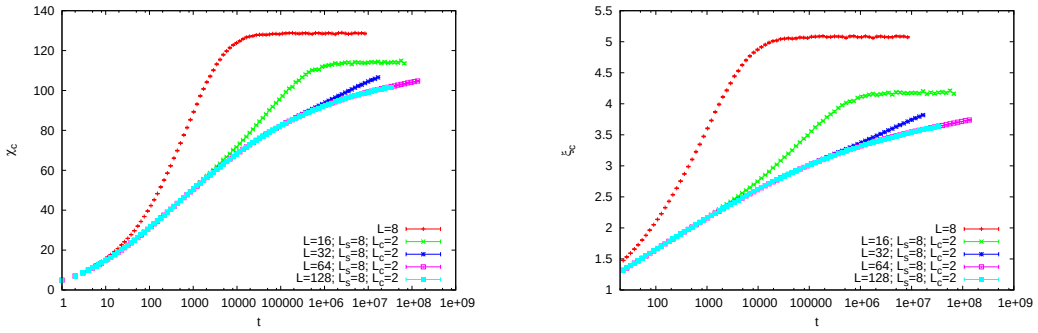


Figure B.3. Data for χ_c (left panel) and ξ (right panel) on the sublattice $L_s = 8$ with $L_c = 2$ for different real lattice sizes. The bigger the embedding lattice the longer data stay on a common curve.

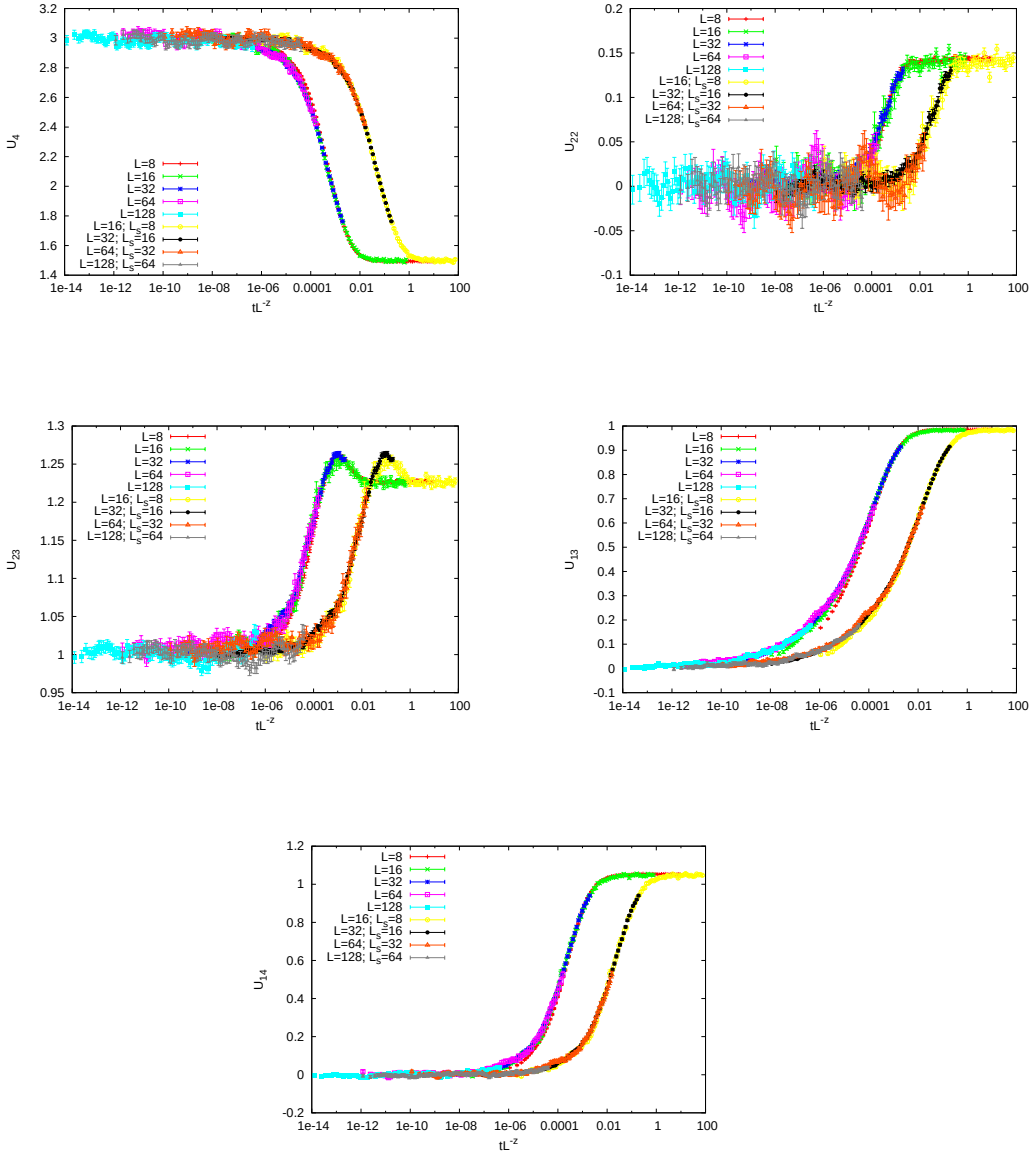


Figure B.4. Critical scaling for real lattice and non-centered sublattice Binder cumulants.

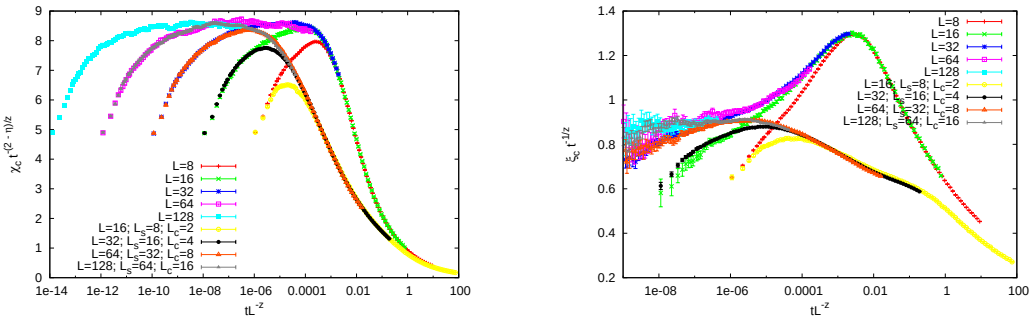


Figure B.5. Critical infinite-volume regime for χ and ξ evaluated for both real lattices and sublattices. Sublattices observables superpose at the beginning of the dynamics. The scaling is obtained for observables sharing the values of the ratios $S = L/L_s = 2$ and $C = L_s/L_c = 2$

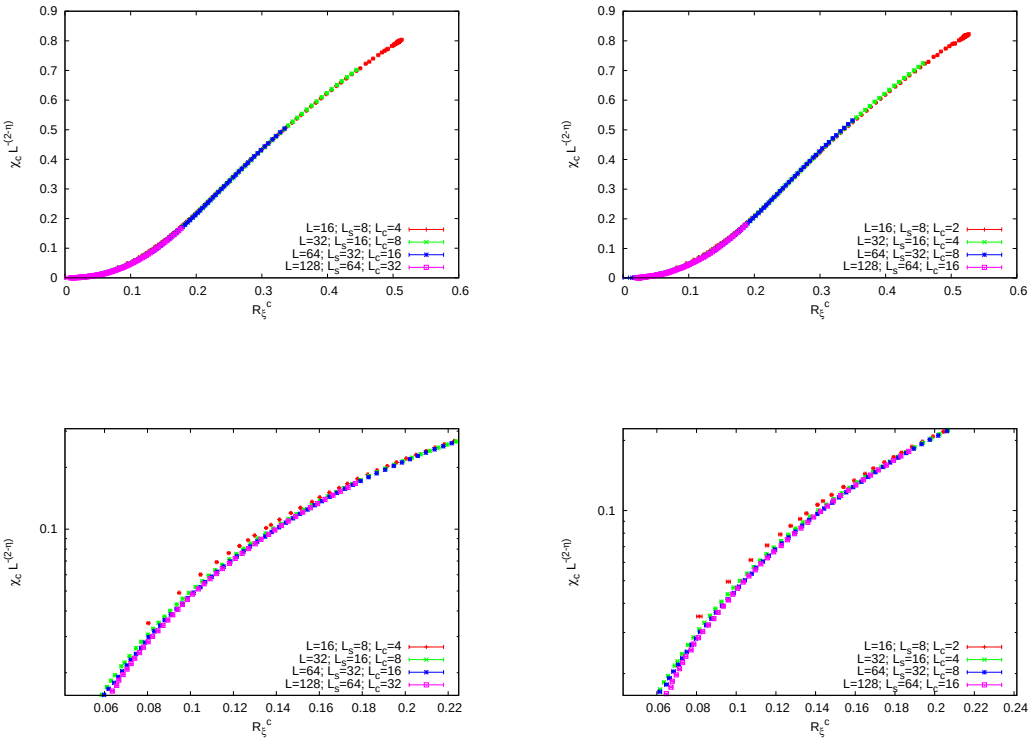


Figure B.6. Critical scaling for the rescaled centered susceptibility $\chi_c L_s^{-(2-\eta)}$ as a function of the centered RG invariant ratio $R_\xi^c = \xi_c/L$, for the two class of observables $C = 2$, $S = 2$ (left) and $C = 2$, $S = 4$ (right). Top panels are in linear scales while bottom panels are zoomed in log scale in order to see finite-size corrections.

Bibliography

[1] N. Goldenfeld, *Lectures on phase transitions and the renormalization group*. 1992.

[2] J. L. Cardy, *Scaling and renormalization in statistical physics*. 1996.

[3] L. Onsager, "Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition," *Phys. Rev.*, vol. 65, pp. 117–149, Feb 1944.

[4] G. S. Rushbrooke, "On the Thermodynamics of the Critical Region for the Ising Problem," *The Journal of Chemical Physics*, vol. 39, no. 3, pp. 842–843, 1963.

[5] J. W. Essam and M. E. Fisher, "Padé Approximant Studies of the Lattice Gas and Ising Ferromagnet below the Critical Point," *The Journal of Chemical Physics*, vol. 38, no. 4, pp. 802–812, 1963.

[6] B. Widom, "Surface Tension and Molecular Correlations near the Critical Point," *The Journal of Chemical Physics*, vol. 43, no. 11, pp. 3892–3897, 1965.

[7] B. Widom, "Degree of the Critical Isotherm," *The Journal of Chemical Physics*, vol. 41, no. 6, pp. 1633–1634, 1964.

[8] B. D. Josephson, "Inequality for the specific heat: I. Derivation," *Proceedings of the Physical Society*, vol. 92, no. 2, p. 269, 1967.

[9] A. Sokal, "More inequalities for critical exponents," *Journal of Statistical Physics*, vol. 25, no. 1, pp. 25–50, 1981.

[10] L. P. Kadanoff, "Scaling laws for Ising models near Tc," *Physics*, vol. 2, no. 6, pp. 263–272, 1966.

[11] J. Yang, Y. Lee, and Y. Li, "Critical behavior of the electron-doped manganite $La_{0.9}Te_{0.1}MnO_3$," *Phys. Rev. B*, vol. 76, p. 054442, Aug 2007.

[12] K. G. Wilson, "Renormalization Group and Critical Phenomena. I. Renormalization Group and the Kadanoff Scaling Picture," *Phys. Rev. B*, vol. 4, pp. 3174–3183, Nov 1971.

[13] K. G. Wilson, "Renormalization Group and Critical Phenomena. II. Phase-Space Cell Analysis of Critical Behavior," *Phys. Rev. B*, vol. 4, pp. 3184–3205, Nov 1971.

[14] F. J. Wegner and A. Houghton, "Renormalization Group Equation for Critical Phenomena," *Phys. Rev. A*, vol. 8, pp. 401–412, Jul 1973.

[15] R. H. Schonmann, "Projections of Gibbs measures may be non-Gibbsian," *Communications in mathematical physics*, vol. 124, no. 1, pp. 1–7, 1989.

[16] A. C. Van Enter, R. Fernández, and A. D. Sokal, "Regularity properties and pathologies of position-space renormalization-group transformations: Scope and limitations of Gibbsian theory," *Journal of Statistical Physics*, vol. 72, no. 5-6, pp. 879–1167, 1993.

[17] L. Bertini, E. Cirillo, and E. Olivieri, "Renormalization-Group Transformations Under Strong Mixing Conditions: Gibbsianness and Convergence of Renormalized Interactions," *Journal of Statistical Physics*, vol. 97, no. 5-6, pp. 831–915, 1999.

[18] L. Bertini, E. N. Cirillo, and E. Olivieri, "Renormalization Group in the Uniqueness Region: Weak Gibbsianity and Convergence," *Communications in Mathematical Physics*, vol. 261, no. 2, pp. 323–378, 2006.

[19] M. Campostrini, A. Pelissetto, and E. Vicari, "Finite-size scaling at quantum transitions," *Phys. Rev. B*, vol. 89, p. 094516, Mar 2014.

[20] M. Hasenbusch, "The thermodynamic Casimir effect in the neighbourhood of the lambda-transition: a Monte Carlo study of an improved three-dimensional lattice model," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, no. 07, p. P07031, 2009.

[21] M. Hasenbusch, "Thermodynamic Casimir effect: Universality and corrections to scaling," *Phys. Rev. B*, vol. 85, p. 174421, May 2012.

[22] H. W. Diehl, D. Grüneberg, M. Hasenbusch, A. Hucht, S. B. Rutkevich, and F. M. Schmidt, "Exact thermodynamic Casimir forces for an interacting three-dimensional model system in film geometry with free surfaces," *EPL (Europhysics Letters)*, vol. 100, no. 1, p. 10004, 2012.

[23] Y. Ozeki and N. Ito, "Nonequilibrium relaxation method," *Journal of Physics A: Mathematical and Theoretical*, vol. 40, no. 31, p. R149, 2007.

[24] P. C. Hohenberg and B. I. Halperin, "Theory of dynamic critical phenomena," *Rev. Mod. Phys.*, vol. 49, pp. 435–479, Jul 1977.

[25] F. Belletti, A. Cruz, L. Fernandez, A. Gordillo-Guerrero, M. Guidetti, A. Maiorano, F. Mantovani, E. Marinari, V. Martin-Mayor, J. Monforte, A. Muñoz Sudupe, D. Navarro, G. Parisi, S. Perez-Gaviro, J. Ruiz-Lorenzo, S. Schifano, D. Sciretti, A. Tarancon, R. Tripiccione, and D. Yllanes, "An In-Depth View of the Microscopic Dynamics of Ising Spin Glasses at Fixed Temperature," *Journal of Statistical Physics*, vol. 135, no. 5-6, pp. 1121–1158, 2009.

[26] J. H. Condon and A. T. Ogielski, "Fast special purpose computer for Monte Carlo simulations in statistical physics," *Review of Scientific Instruments*, vol. 56, no. 9, pp. 1691–1696, 1985.

[27] A. Cruz, J. Pech, A. Tarancón, P. Téllez, C. Ullod, and C. Ungil, "SUE: A special purpose computer for spin glass models," *Computer Physics Communications*, vol. 133, no. 2–3, pp. 165 – 176, 2001.

[28] M. Baity-Jesi, R. Baños, A. Cruz, L. Fernandez, J. Gil-Narvion, A. Gordillo-Guerrero, M. Guidetti, D. Iñiguez, A. Maiorano, F. Mantovani, E. Marinari, V. Martin-Mayor, J. Monforte-Garcia, A. Muñoz Sudupe, D. Navarro, G. Parisi, M. Pivanti, S. Perez-Gaviro, F. Ricci-Tersenghi, J. Ruiz-Lorenzo, S. Schifano, B. Seoane, A. Tarancon, P. Tellez, R. Tripiccione, and D. Yllanes, "Reconfigurable computing for Monte Carlo simulations: Results and prospects of the Janus project," *The European Physical Journal Special Topics*, vol. 210, no. 1, pp. 33–51, 2012.

[29] M. Baity-Jesi, R. Baños, A. Cruz, L. Fernandez, J. Gil-Narvion, A. Gordillo-Guerrero, D. Iñiguez, A. Maiorano, F. Mantovani, E. Marinari, V. Martin-Mayor, J. Monforte-Garcia, A. M. Sudupe, D. Navarro, G. Parisi, S. Perez-Gaviro, M. Pivanti, F. Ricci-Tersenghi, J. Ruiz-Lorenzo, S. Schifano, B. Seoane, A. Tarancon, R. Tripiccione, and D. Yllanes, "Janus II: A new generation application-driven computer for spin-system simulations," *Computer Physics Communications*, vol. 185, no. 2, pp. 550 – 559, 2014.

[30] E. Marinari and G. Parisi, "Simulated Tempering: A New Monte Carlo Scheme," *EPL (Europhysics Letters)*, vol. 19, no. 6, p. 451, 1992.

[31] M. Baity-Jesi, R. A. Baños, A. Cruz, L. A. Fernandez, J. M. Gil-Narvion, A. Gordillo-Guerrero, D. Iñiguez, A. Maiorano, F. Mantovani, E. Marinari, V. Martin-Mayor, J. Monforte-Garcia, A. M. n. Sudupe, D. Navarro, G. Parisi, S. Perez-Gaviro, M. Pivanti, F. Ricci-Tersenghi, J. J. Ruiz-Lorenzo, S. F. Schifano, B. Seoane, A. Tarancon, R. Tripiccione, and D. Yllanes, "Critical parameters of the three-dimensional Ising spin glass," *Phys. Rev. B*, vol. 88, p. 224416, Dec 2013.

[32] L. Lundgren, P. Svedlindh, P. Nordblad, and O. Beckman, "Dynamics of the Relaxation-Time Spectrum in a CuMn Spin-Glass," *Phys. Rev. Lett.*, vol. 51, pp. 911–914, Sep 1983.

[33] M.M. Xavier, F.A.O. Cabral, J.H. de Araujo, T. Dumelow, and A.A. Coelho, "Reentrant spin glass behavior in polycrystalline $La_{0.7}Sr_{0.3}Mn_{1-X}Fe_XO_3$," *Materials Research*, vol. 7, pp. 355 – 357, 06 2004.

[34] D. Sherrington and S. Kirkpatrick, "Solvable Model of a Spin-Glass," *Phys. Rev. Lett.*, vol. 35, pp. 1792–1796, Dec 1975.

[35] G. Parisi, "Infinite Number of Order Parameters for Spin-Glasses," *Phys. Rev. Lett.*, vol. 43, pp. 1754–1756, Dec 1979.

[36] G. Parisi, "The order parameter for spin glasses: a function on the interval 0-1," *Journal of Physics A: Mathematical and General*, vol. 13, no. 3, p. 1101, 1980.

[37] S. F. Edwards and P. W. Anderson, "Theory of spin glasses," *Journal of Physics F: Metal Physics*, vol. 5, no. 5, p. 965, 1975.

[38] R. N. Bhatt and A. P. Young, "Search for a transition in the three-dimensional ± J Ising spin-glass," *Phys. Rev. Lett.*, vol. 54, pp. 924–927, Mar 1985.

[39] M. Hasenbusch, A. Pelissetto, and E. Vicari, "Critical behavior of three-dimensional Ising spin glass models," *Phys. Rev. B*, vol. 78, p. 214205, Dec 2008.

[40] D. S. Fisher and D. A. Huse, "Ordered Phase of Short-Range Ising Spin-Glasses," *Phys. Rev. Lett.*, vol. 56, pp. 1601–1604, Apr 1986.

[41] D. S. Fisher and D. A. Huse, "Absence of many states in realistic spin glasses," *Journal of Physics A: Mathematical and General*, vol. 20, no. 15, p. L1005, 1987.

[42] D. S. Fisher and D. A. Huse, "Nonequilibrium dynamics of spin glasses," *Phys. Rev. B*, vol. 38, pp. 373–385, Jul 1988.

[43] Gardner, E., "A spin glass model on a hierarchical lattice," *J. Phys. France*, vol. 45, no. 11, pp. 1755–1763, 1984.

[44] M. Baity-Jesi, R. A. Baños, A. Cruz, L. A. Fernandez, J. M. Gil-Narvion, A. Gordillo-Guerrero, D. Iñiguez, A. Maiorano, F. Mantovani, E. Marinari, V. Martin-Mayor, J. Monforte-Garcia, A. M. Sudupe, D. Navarro, G. Parisi, S. Perez-Gaviro, M. Pivanti, F. Ricci-Tersenghi, J. J. Ruiz-Lorenzo, S. F. Schifano, B. Seoane, A. Tarancon, R. Tripiccione, and D. Yllanes, "The three-dimensional Ising spin glass in an external magnetic field: the role of the silent majority," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2014, no. 5, p. P05014, 2014.

[45] nVidia Corporation, "CUDA Toolkit Documentation." `http://docs.nvidia.com/cuda/#axzz3FN1CEj00`.

[46] S. Tomov, M. McGuigan, R. Bennett, G. Smith, and J. Spiletic, "Benchmarking and implementation of probability-based simulations on programmable graphics cards," *Computers & Graphics*, vol. 29, no. 1, pp. 71 – 80, 2005.

[47] V. I. Manousiouthakis and M. W. Deem, "Strict detailed balance is unnecessary in Monte Carlo simulation," *The Journal of Chemical Physics*, vol. 110, no. 6, pp. 2753–2756, 1999.

[48] R. Ren and G. Orkoulas, "Acceleration of Markov chain Monte Carlo simulations through sequential updating," *The Journal of Chemical Physics*, vol. 124, no. 6, pp. –, 2006.

[49] T. Preis, P. Virnau, W. Paul, and J. J. Schneider, "GPU accelerated Monte Carlo simulation of the 2D and 3D Ising model," *Journal of Computational Physics*, vol. 228, no. 12, pp. 4468 – 4477, 2009.

[50] B. Block, P. Virnau, and T. Preis, "Multi-GPU accelerated multi-spin Monte Carlo simulations of the 2D Ising model," *Computer Physics Communications*, vol. 181, no. 9, pp. 1549 – 1556, 2010.

[51] M. Weigel, "Simulating spin models on GPU," *Computer Physics Communications*, vol. 182, no. 9, pp. 1833 – 1836, 2011. Computer Physics Communications Special Edition for Conference on Computational Physics Trondheim, Norway, June 23-26, 2010.

[52] T. Levy, G. Cohen, and E. Rabani, "Simulating Lattice Spin Models on Graphics Processing Units," *Journal of Chemical Theory and Computation*, vol. 6, no. 11, pp. 3293–3301, 2010.

[53] T. Yavors'kii and M. Weigel, "Optimized GPU simulation of continuous-spin glass models," *The European Physical Journal Special Topics*, vol. 210, no. 1, pp. 159–173, 2012.

[54] Y. Fang, S. Feng, K.-M. Tam, Z. Yun, J. Moreno, J. Ramanujam, and M. Jarrell, "Parallel tempering simulation of the three-dimensional edwards–anderson model with compact asynchronous multispin coding on {GPU}," *Computer Physics Communications*, vol. 185, no. 10, pp. 2467 – 2478, 2014.

[55] E. E. Ferrero, J. P. D. Francesco, and N. W. S. A. Cannas, "q-state Potts model metastability study using optimized GPU-based Monte Carlo algorithms," *Computer Physics Communications*, vol. 183, no. 8, pp. 1578 – 1587, 2012.

[56] M. Bernaschi, G. Parisi, and L. Parisi, "Benchmarking GPU and CPU codes for Heisenberg spin glass over-relaxation," *Computer Physics Communications*, vol. 182, no. 6, pp. 1265 – 1271, 2011.

[57] M. Bernaschi, M. Fatica, G. Parisi, and L. Parisi, "Multi-GPU codes for spin systems simulations," *Computer Physics Communications*, vol. 183, no. 7, pp. 1416 – 1421, 2012.

[58] M. Bernaschi, M. Bisson, and D. Rossetti, "Benchmarking of communication techniques for GPUs," *Journal of Parallel and Distributed Computing*, vol. 73, no. 2, pp. 250 – 255, 2013.

[59] M. Berganza and L. Leuzzi, "Critical behavior of the XY model in complex topologies," *Phys. Rev. B*, vol. 88, p. 144104, Oct 2013.

[60] M. Baity-Jesi, L. A. Fernandez, V. Martin-Mayor, and J. M. Sanz, "Phase transition in three-dimensional Heisenberg spin glasses with strong random anisotropies through a multi-GPU parallelization," *Phys. Rev. B*, vol. 89, p. 014202, Jan 2014.

[61] D. B. Thomas, "Warp Generator." `http://cas.ee.ic.ac.uk/people/dt10/research/rngs-gpu-warp_generator.html`.

[62] M. Manssen, M. Weigel, and A. Hartmann, "Random number generators for massively parallel simulations on GPU," *The European Physical Journal Special Topics*, vol. 210, no. 1, pp. 53–71, 2012.

[63] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, pp. 3–30, Jan. 1998.

[64] Y. Hsieh and Y. Kao and Anders W Sandvik, "Finite-size scaling method for the Berezinskii–Kosterlitz–Thouless transition," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2013, no. 09, p. P09001, 2013.

[65] G. Parisi and F. Rapuano, "Effects of the random number generator on computer simulations," *Physics Letters B*, vol. 157, no. 4, pp. 301 – 302, 1985.

[66] H. G. Ballesteros and V. Martín-Mayor, "Test for random number generators: Schwinger-Dyson equations for the Ising model," *Phys. Rev. E*, vol. 58, pp. 6787–6791, Nov 1998.

[67] G. Marsaglia, "Diehard Battery of Tests of Randomness." `http://www.stat.fsu.edu/pub/diehard/`.

[68] D. F. Carta, "Two Fast Implementations of the Minimal Standard Random Number Generator," *Commun. ACM*, vol. 33, pp. 87–88, Jan. 1990.

[69] "Park-Miller-Carta Pseudo-Random Number Generator." `http://www.firstpr.com.au/dsp/rand31/`.

[70] M. Weigel, "Performance potential for simulating spin models on GPU," *Journal of Computational Physics*, vol. 231, no. 8, pp. 3064 – 3082, 2012.

[71] L. Barash and L. Shchur, "PRAND: GPU accelerated parallel random number generation library: Using most reliable algorithms and applying parallelism of modern GPUs and CPUs," *Computer Physics Communications*, vol. 185, no. 4, pp. 1343 – 1353, 2014.

[72] "cuRand Device API example." `http://docs.nvidia.com/cuda/curand/device-api-overview.html#device-api-example`.

[73] "MT19937 official site." `http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html`.

[74] M. Creutz, L. Jacobs, and C. Rebbi, "Experiments with a Gauge-Invariant Ising System," *Phys. Rev. Lett.*, vol. 42, pp. 1390–1393, May 1979.

[75] S. Wansleben, J. Zabolitzky, and C. Kalle, "Monte Carlo simulation of Ising models by multispin coding on a vector computer," *Journal of Statistical Physics*, vol. 37, no. 3-4, pp. 271–282, 1984.

[76] G. Bhanot, D. Duke, and R. Salvador, "Finite-size scaling and the three-dimensional Ising model," *Phys. Rev. B*, vol. 33, pp. 7841–7844, Jun 1986.

[77] M. Kikuchi and Y. Okabe, "Renormalization, self-similarity, and relaxation of order-parameter structure in critical phenomena," *Phys. Rev. B*, vol. 35, pp. 5382–5384, Apr 1987.

[78] N. Ito and Y. Kanada, "Monte Carlo Simulation of the Ising Model and Random Number Generation on the Vector Processor," in *Proceedings of the 1990 ACM/IEEE Conference on Supercomputing*, Supercomputing '90, (Los Alamitos, CA, USA), pp. 753–763, IEEE Computer Society Press, 1990.

[79] M. Baity-Jesi, R. Baños, A. Cruz, L. Fernandez, J. Gil-Narvion, A. Gordillo-Guerrero, M. Guidetti, D. Iñiguez, A. Maiorano, F. Mantovani, E. Marinari, V. Martin-Mayor, J. Monforte-Garcia, A. Muñoz Sudupe, D. Navarro, G. Parisi, M. Pivanti, S. Perez-Gaviro, F. Ricci-Tersenghi, J. Ruiz-Lorenzo, S. Schifano, B. Seoane, A. Tarancon, P. Tellez, R. Tripiccione, and D. Yllanes, "Reconfigurable computing for Monte Carlo simulations: Results and prospects of the Janus project," *The European Physical Journal Special Topics*, vol. 210, no. 1, pp. 33–51, 2012.

[80] "cuRand MTGP32 Device API." `http://docs.nvidia.com/cuda/curand/device-api-overview.html#bit-generation-2`.

[81] "Cscs piz daint." `http://www.cscs.ch/computers/piz_daint/index.html`.

[82] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-driven, highly-scalable dragonfly topology," in *ACM SIGARCH Computer Architecture News*, vol. 36, pp. 77–88, IEEE Computer Society, 2008.

[83] B. Alverson, E. Froese, L. Kaplan, and D. Roweth, "Cray XC series network," *Cray Inc., White Paper WP-Aries01-1112*, 2012.

[84] P. Calabrese and A. Gambassi, "Ageing properties of critical systems," *Journal of Physics A: Mathematical and General*, vol. 38, no. 18, p. R133, 2005.

[85] M. Suzuki, "Linear and nonlinear dynamic scaling relations in the renormalization group theory," *Physics Letters A*, vol. 58, no. 7, pp. 435 – 436, 1976.

[86] M. Suzuki, "Static and Dynamic Finite-Size Scaling Theory Based on the Renormalization Group Approach," *Progress of Theoretical Physics*, vol. 58, no. 4, pp. 1142–1150, 1977.

[87] A. T. Ogielski, "Dynamics of the three-dimensional Ising spin glasses in thermal equilibrium," *Phys.Rev.B*, vol. 32, pp. 7384–7398, 1985.

[88] A. T. Ogielski and I. Morgenstern, "Critical behavior of three-dimensional Ising spin-glass model," *Phys.Rev.Lett.*, vol. 54, pp. 928–931, 1985.

[89] L. W. Bernardi, S. Prakash, and I. A. Campbell, "Ordering Temperatures and Critical Exponents in Ising Spin Glasses," *Phys. Rev. Lett.*, vol. 77, pp. 2798–2801, Sep 1996.

[90] P. O. Mari and I. A. Campbell, "Ising spin glasses: Corrections to finite size scaling, freezing temperatures, and critical exponents," *Phys. Rev. E*, vol. 59, pp. 2653–2658, Mar 1999.

[91] P. O. Mari and I. A. Campbell, "Ordering temperature and critical exponents of the binomial Ising spin glass in dimension 3," *Phys. Rev. B*, vol. 65, p. 184409, Apr 2002.

[92] M. Pleimling and I. A. Campbell, "Dynamic critical behavior in Ising spin glasses," *Phys. Rev. B*, vol. 72, p. 184429, Nov 2005.

[93] D. A. Huse, "Remanent magnetization decay at the spin-glass critical point: A new dynamic critical exponent for nonequilibrium autocorrelations," *Phys. Rev. B*, vol. 40, pp. 304–308, Jul 1989.

[94] N. Kawashima and A. P. Young, "Phase transition in the three-dimensional ±J Ising spin glass," *Phys. Rev. B*, vol. 53, pp. R484–R487, Jan 1996.

[95] F. Romá, "Universality in three-dimensional Ising spin glasses: Nonequilibrium dynamics from Monte Carlo simulations," *Phys. Rev. B*, vol. 82, p. 212402, Dec 2010.

[96] F. Romá and D. Domínguez, "Critical behavior of spin and chiral degrees of freedom in three-dimensional disordered XY models studied by the nonequilibrium aging method," *Phys. Rev. B*, vol. 89, p. 024408, Jan 2014.

[97] Y. Ozeki and N. Ito, "Nonequilibrium relaxation study of Ising spin glass models," *Phys. Rev. B*, vol. 64, p. 024416, Jun 2001.

[98] S. P. Gaviro, J. J. Ruiz-Lorenzo, and A. Tarancón, "Study of the phase transition in the 3D Ising spin glass from out-of-equilibrium numerical simulations," *Journal of Physics A: Mathematical and General*, vol. 39, no. 27, p. 8567, 2006.

[99] G. Parisi, F. Ricci-Tersenghi, and J. J. Ruiz-Lorenzo, "Equilibrium and off-equilibrium simulations of the Gaussian spin glass," *Journal of Physics A: Mathematical and General*, vol. 29, no. 24, p. 7943, 1996.

[100] D. Stariolo, M. Montemurro, and F. Tamarit, "Aging dynamics of Edwards-Anderson spin glasses," *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 32, no. 3, pp. 361–367, 2003.

[101] Lippiello, E. and Sarracino, A., "Identification of the critical temperature from non-equilibrium time-dependent quantities," *EPL*, vol. 90, no. 6, p. 60001, 2010.

[102] T. Nakamura, S. Endoh, and T. Yamamoto, "Weak universality of spin-glass transitions in three-dimensional ± J models," *Journal of Physics A: Mathematical and General*, vol. 36, no. 43, p. 10895, 2003.

[103] T. Nakamura, "Nonequilibrium dynamic correlation-length scaling method," *Phys. Rev. B*, vol. 82, p. 014427, Jul 2010.

[104] T.Nakamura and T. Shirakura, "Window measurements of simulations in random systems," *arXiv:1404.7590*, 2014.

[105] M. Hasenbusch, A. Pelissetto, and E. Vicari, "The critical behavior of 3D Ising spin glass models: universality and scaling corrections," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 02, p. L02001, 2008.

[106] K. Hukushima and K. Nemoto, "Exchange Monte Carlo Method and Application to Spin Glass Simulations," *Journal of the Physical Society of Japan*, vol. 65, no. 6, pp. 1604–1608, 1996.