2016

Study of Condensed Matter Systems with Monte Carlo Simulation on Heterogeneous Computing Systems

Sheng Feng
Louisiana State University and Agricultural and Mechanical College

STUDY OF CONDENSED MATTER SYSTEMS WITH MONTE CARLO SIMULATION ON HETEROGENEOUS COMPUTING SYSTEMS

A Dissertation

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in

The Department of Physics and Astronomy

by
Sheng Feng
B.S., University of Science and Technology of China, 2010
May 2016

Acknowledgments

First, I'd like to thank Prof Mark Jarrell for being my advisor. Mark has always been a great example of what a physicist should be. From his example, I learned how a physicist should think, work, communicate and collaborate when tackling physical problems. I also want to thank my co-advisor, Prof Juana Moreno. Together with Mark, she always gave valuable advises in our regular meetings, and was always nice and patient. I would remain grateful for all their guidance.

Throughout my study at LSU, Dr. Ka-Ming Tam has always been a tutor and a helpful friend to me. He introduced me to the Spin Glass field, and almost spoon-fed me most of the physics and numerical techniques in this field. I thank him for all his knowledge and kindness.

I also appreciate the help of my friends and colleagues at LSU, including Ye Fang, Jianpin Lai, Yun Ding, Qinqin Lv, Gaomin Wang, Yan Wu, Mengxi Wu, Enzhi Li, Jian Tao, Sammeer Abu Asal, Shuxiang Yang, Kuangshing Chen, Peng Zhang, etc.

Finally, Wanshu Zhang has been my love and has been supporting me through my highs and lows. She gave me all the dreams that I could ever imagine. Together we would see those dreams come true.

My research was supported by the National Science Foundation through the Louisiana Alliance for Simulation Guided Materials Applications (LASiGMA) program.

Table of Contents

Abstract

We study the Edwards-Anderson model on a simple cubic lattice with a finite constant external field. We employ an indicator composed of a ratio of susceptibilities at finite momenta, which was recently proposed to avoid the difficulties of a zero momentum quantity, for capturing the spin glass phase transition. Unfortunately, this new indicator is fairly noisy, so a large pool of samples at low temperature and small external field are needed to generate results with a sufficiently small statistical error for analysis. We thus implement the Monte Carlo method using graphics processing units to drastically speed up the simulation. We confirm previous findings that conventional indicators for the spin glass transition, including the Binder ratio and the correlation length do not show any indication of a transition for rather low temperatures. However, the ratio of spin glass susceptibilities does show crossing behavior, albeit a systematic analysis is beyond the reach of the present data. This reveals the difficulty with current numerical methods and computing capability in studying this problem.

One of the fundamental challenges of theoretical condensed matter physics is the accurate solution of quantum impurity models. By taking expansion in the hybridization about an exactly solved local limit, one can formulate a quantum impurity solver. We implement the hybridization expansion quantum impurity solver on Intel Xeon Phi accelerators, and aim to apply this approach on the Dynamic Hubbard Models.

Chapter 1
General Introduction

1.1 Computer Simulations

Computer simulation is the discipline of designing a model of an actual or theoretical system, executing the model on a digital computer, and analyzing the execution output. Computer simulations have become a useful part of mathematical modeling of many natural systems in various disciplines of science.

In science, typically two types of computer simulations are used. First is a numerical simulation of differential equations that cannot be solved analytically. In the recent discovery of gravitational waves from a binary black hole merger, LIGO scientists [1] used numerical simulations to provide estimates of the mass and spin of the final black hole, the total energy radiated in gravitational waves, and the peak gravitational-wave luminosity. They were also able to verify that their observation is consistent with general relativity equations.

Another type is the stochastic simulation. A stochastic simulation is a simulation that traces the evolution of variables that can change stochastically (randomly) with certain probabilities. With a stochastic model, we create a projection which is based on a set of random values. Outputs are recorded and the projection is repeated with a new set of random values of the variables. These steps are repeated until a sufficient amount of data is gathered. In the end, the distribution of the outputs shows the most probable estimates as well as a frame of expectations regarding what ranges of values the variables are more or less likely to fall in.

One of the most widely used stochastic simulation methods is the Monte Carlo methods. Named after the famous casino in Monaco, Monte Carlo methods use re-

peated random sampling to obtain numerical results. By the law of large numbers, the average of the results obtained from a large number of trials should be close to the expected value, and gets closer as more test trials are performed. Therefore, an integral can be evaluated by randomly sampling the phase space and sum over the samples. In physics, Monte Carlo methods are very useful for complex systems such as disordered systems, strongly coupled systems, etc.

1.2 Disordered Systems

In physics, the terms order and disorder designate the presence or absence of translational symmetry of a system. The disorder can be put into two main categories according to their dynamics: annealed disorder and quenched disorder.

A system is said to present quenched disorder when some parameters defining its behavior are random variables which do not evolve with time. In these systems, the disorder is explicitly present in the Hamiltonian, typically under the form of random coupling J among the degrees of freedom σ,

$$H = H(\sigma, J). \tag{1.1}$$

Spin glasses [2] are a classical example of quenched disorder. The term spin glass was given to materials that display the lack of long-range magnetic ordering down to zero temperature in the 1970s. Several families of spin glass materials have been identified. The classic examples include dilute metallic alloys such as CuMn with 0.9% Mn, and concentrated insulators such as $Eu_xSr_{1-x}S$. In these systems, the quenched disorder originates in the random dilution. The magnetic moments in Mn and Eu can be described in term of the Heisenberg spins. An example for real material featuring Ising-like spin is the $LiHo_xY_{1-x}F_4$ insulator[3–8]. Since the single-ion crystal field anisotropy is strong compared to the magnetic interaction between Ho^{3+} ions, the magnetic moments can be mapped on to Ising spins that

only point parallel or antiparallel to the c-axis of the tetragonal crystalline structure of the lattice.

The discovery of the spin glass materials is due to its unique behaviors of the susceptibility. The now defining signature of spin glass materials is the cusp in the low-field AC susceptibility, first found by Cannella and Mydosh [9]. In contrast to the usual ferromagnetic systems, this signature strongly suggests the lack of long range order. The physical picture of the lack of divergence in the susceptibility indicates a transition to a state of randomly frozen spins.

In addition to the cusp, some interesting slow dynamic behaviors are also observed. For example, the remanent magnetization is found if one cools the spin glass in a field to below the transition temperature and turns off the field. The magnetization then decays very slowly as it approaches zero, signaling a very long relaxation time. Indeed, experimental spin glass systems can never truly attain equilibrium. We further discuss the experimental properties in section 2.1.

The simplest model that captures the quenched random magnetic interaction is the Edwards-Anderson model[10],

$$H = -\sum_{<i,j>} J_{ij}\sigma_i\sigma_j, \tag{1.2}$$

where the spins $\sigma_i = \pm 1$ are the degree of freedom, and the random coupling J_{ij} can be either Gaussian random variables, or binary random variables.

The mean-field variant of this model, the Sherrington-Kirkpatrick model[11, 12], was solved by Parisi[13–15] using the replica symmetry breaking approach. In this picture, there is a hierarchy of the replica overlap, which can be described by an ultrametric tree. Although this theory has been accepted as the exact solution to an Edwards-Anderson like model with infinite range interactions, its applicability in lower dimensions has been debated. Below the upper critical dimension ($d < 6$)[16–

18], especially at the most physically relevant three dimensions, the nature of the spin glass phase is still not clear. The main competing picture is the Droplet picture, proposed by Fisher and Huse[19, 20], in which there is only a pair of spin-flip related pure states in the thermal dynamic limit.

One of the most significant differences between the two pictures is the effect of an external field[21]. The droplet picture predicts that there is no phase transition in a field, while the replica symmetry breaking picture predicts that there is a transition, and there is an AT line that separates the two phases.

There have been a lot of intensive numerical studies invested in this problem over the last four decades. As we will explain in more detail in this thesis, numerical simulations of spin glass system present a tremendous challenge. Obtaining the ground state by minimization method such as branch-and-cut method is mostly useful for two dimensions.

The difficulty of using minimization at three dimensions can be considered as consequence of the proof that the ground state energy of a three-dimensional Edwards-Anderson model is NP-complete[22]. Therefore, the study of the three-dimensional cases is only practically feasible at finite temperatures. The best available method is the Monte Carlo. By the very nature of spin glass systems, the long relaxation time, Monte Carlo simulation is bound to be very slow. This problem can be attacked from two directions, the advancements in algorithms and computer implementations. Over the last few decades, different methods to accelerate the Monte Carlo have been proposed and tested, we will explain more detail on these methods in this thesis. In addition, computer implementations have been improved to shorten the simulation time. This is not solely on the software programming. Due to the relatively simple Monte Carlo method, various dedicated machines were built exclusively for the simulation of spin glass systems.

In the following, we will review a few important milestones in the numerical study of Edwards-Anderson model. Although the model was proposed at 1975, reliable simulations only appear from around mid-80s.

Bhatt and Young [23] studies used Monte Carlo simulations to study the three-dimensional Edwards-Anderson model in zero field for samples with $3 < L < 20$. Results for $T \geq 1.2$ are consistent with a conventional phase transition at $T_c = 1.2$. However, at lower temperatures, the results indicate marginal behavior. This existence of a spin glass transition has long been considered as an open question due to the results from this paper.

Ballesteros et al. [24] used the parallel tempering technique to study the three-dimensional Edwards-Anderson model in helicoidal geometry. By measuring the correlation length in the critical region, evidence for a second order finite-temperature phase transition was obtained and critical exponents such as ν and η were calculated. This is the beginning of a new chapter in the numerical simulation of spin glass. The wisdom from this paper is that the conventional indicators, in particular, the ratio of cumulants of the order parameter, are not sufficient for detecting a transition. The results from this paper bring to a universal consensus that the spin glass transition does exist in the three-dimensional Edwards-Anderson model.

Katzgraber and Young studied the model in a magnetic field–known and found the absence of the de Almeida-Thouless line. Later, a one-dimensional power-law diluted Ising spin-glass model has been proposed to produce an effective model at higher dimensions[21]. Their results for the model corresponding to a three-dimensional system are consistent with there being no de Almeida-Thouless line.

Recently, it has been suggested that the correlation length may not be a good indicator for the model in an external magnetic field. The Janus collaboration [25, 26] used their special purpose computer with FPGA to simulate four-dimensional

spin glass in a field. They studied the ratio of susceptibilities at the two smallest momenta, and found a crossing at finite temperature, which indicates that the spin glass phase can exist without time-reversal symmetry below the upper critical dimension.

1.3 Strongly Correlated Systems

Strongly correlated materials are a wide class of compounds containing ions with $d-$ or $f-$orbitals, that show unusual electronic and magnetic properties. They include insulators, magnets, paramagnets and superconductors. In transition metals, such as vanadium, iron, and their oxides, for example, electrons experience strong Coulombic repulsion because of their spatial configuration, and their interaction cannot be described by a static mean field generated by other electrons. [27, 28] The interplay of the d and f electrons' internal degree of freedom, such as spin, charge and orbital moment, can exhibit many interesting ordering phenomena at low temperatures, and makes strongly correlated systems sensitive to small changes in external parameters such as temperature, pressure, or doping.

The most important feature that defines these materials is that the behavior of the electrons cannot be described in terms of non-interacting entities. Methods that work well in weakly correlated electron materials, such as Fermi liquid theory, or density functional theory (DFT) [29], are not accurate enough when applied to strongly correlated materials.

Traditionally strongly correlated materials have been described using the model Hamiltonian approach, in which the full many-body Hamiltonian is reduced to a simpler, effective model that retains the essence of the physical phenomena we want to understand. One of the simplest models is the Hubbard Hamiltonian,

$$H = \sum_{i,j,\sigma} t_{ij} c^{\dagger}_{i\sigma} c_{j\sigma} + U \sum_{i} n_{i\uparrow} n_{i\downarrow}. \tag{1.3}$$

This Hamiltonian describes electrons with spin directions σ moving between lattices i and j, and they only interact when they meet on the same lattice site i. The kinetic term favors the delocalization of electrons, while the potential term favors localization, and, therefore, they compete against each other. The system property is then determined by parameters such as the ratio of the Coulomb interaction U and the bandwidth W, the temperature T and the hopping or number of electrons.

One of the most popular approaches used to study the Hubbard model and other related models is the dynamical mean field theory (DMFT) [30–34]. In DMFT, a many body lattice problem is mapped onto a single site impurity problem with effective parameters. DMFT has been deployed to understand the Mott transition [35–37], which has been confirmed by experiments[38–40]. DMFT has also been applied to a range of other strongly correlated materials. With the computing power growing, and new algorithms and ideas emerging, one can expect much a better understanding in complicated strongly correlated materials.

1.4 Heterogeneous Computing

Heterogeneous computing refers to systems that use more than one kind of processor or cores. These systems gain performance and efficiency by adding different processors/accelerators, and divide the task among them to utilize their specialized capabilities. Popular examples of such accelerators, according to this definition, include GPUs, FPGAs, Intel Xeon Phi coprocessors, etc.

Graphic processing units, or GPUs, are typically used in computers for image processing. Due to the performance, it is becoming increasingly common to use a general purpose graphics processing unit (GPGPU). A CPU consists of a few cores optimized for sequential serial processing, while a GPU has a massively parallel architecture consisting of thousands of small cores designed for handling multiple tasks simultaneously. For example, a Nvidia K80 GPU [41] has 2496 cores, and can

achieve up to 2.91 TFlops double precision performance and up to 8.74 TFlops single precision performance, and has a bandwidth of 480GB/s. The dominant framework for GPU programming is Nvidia CUDA [42], which allows programmers to write codes with C, C++ and Fortran and run the program on CUDA-enabled GPUs.

Intel Xeon Phi[43] is a coprocessor developed by Intel that features a X86-compatible architecture. With up to 61 cores, 244 threads, and 1.2 teraFLOPS of performance, a Xeon Phi delivers up to 2.3 times higher peak FLOPS than Intel Xeon processor E5 family-based servers. Since languages, tools, and applications are compatible with both Intel X86 processor and Intel Xeon Phi coprocessors, it is easier for programmers to design, write, compile and optimize their code.

Heterogeneous systems are capable of delivering better performance than traditional homogeneous systems, by exploits the diversity offered by different processors/instruction set architectures(ISAs). The huge performance increase over CPUs makes accelerators popular choices for supercomputers. Tianhe-2[44], a supercomputer developed by Chinas National University of Defense Technology, leads the list of top 500 supercomputers[45]. It utilizes 48,000 Intel Xeon Phi coprocessors and 32,000 Ivy Bridge-EP Xeon processors to achieve 33.86 petaFLOPS.

Heterogeneous systems also have much better energy efficiency. The power usage, as well as the cooling cost required to support the computers, are now the primary cost of ownership for HPC data centers. Therefore, the HPC community now understands that supercomputers should not be evaluated solely on the basis of speed, but should also consider metrics related to energy efficiency. The most popular metric is the FLOPS/watt. By using energy-efficient accelerators, heterogeneous systems can significantly reduce the energy footprint. In fact, heterogeneous accelerator-based systems have been dominating the top places of the

Green500 [46], a ranking of the most energy-efficient supercomputers in the world. In the November 2015 edition of the list[47], the top 40 supercomputers listed all used accelerators of one form or another.

Albeit the advantages in the hardware, the heterogeneous nature present new and unique challenges for programmers and scientists. In parallel programming, programmers need to explore the problem for the possibility of parallelization, map the tasks onto threads, to schedule for communication and/or synchronization, and to solve problems such as race conditions, etc. Communication and barriers could result in parallel slow-down, a situation in which the overhead from communication outweighs the performance gain from parallelism, and further parallelization increases the time to finish the workload. Accelerators use a huge number of cores to achieve great performance. To utilize all the cores, a lot of parallelisms is required, and efficient parallelism is critical.

Different processors/accelerators often feature different memory architecture. To achieve the best possible performance, architecture aware memory access is very important to utilize the high memory bandwidth and reduce the latency. This is even more important in programming for accelerators such as GPUs than in CPU programming, since the latency is not hidden by a large cache. For example, in CUDA there are different types of memory such as registers, local memory, shared memory, global memory, and constant memory, etc. Each of these types is different in terms of size, latency, bandwidth, and performance profile for various access modes, and one has to make conscious decisions when using them in order to avoid performance penalties and make the best of the hardware. In CUDA, this means programmers need to put the correct qualifiers in front when declaring their variables and move their variables around the memory hierarchy when needed.

The different architecture of processors makes it hard to write portable codes. First, some programming languages, such as CUDA, and associated compilers and tools, are exclusive for their specific platforms. In order to program, debug, profile and optimize a code for different platforms, programmers often need to understand more than one set of tools. Second, the performance profiles for different hardware are very different from each other, thus, an efficient code on one platform may not be optimal for another. Fortunately, tools that aim at portable accelerated codes with great performance, such as OpenMP[48], OpenCL[49] and OpenACC[50], are evolving along with the hardware, and they allow programmers to worry more about the problem rather than the language and hardware they are using, and make it much easier to develop and maintain codes that run on heterogeneous systems.

1.5 Scope and Structure

In this dissertation, I cover my work in two projects.

The first is the work on the Three-Dimensional Edwards-Anderson Model in an External field. We first discuss the model and the theoretical understanding in chapter 2. Then, an efficient GPU implementation is described in chapter 3. We show the results obtained from this study in chapter 4. Some additional research using the covariance matrix is covered in the appendix.

Second, we show our work on a Continuous-Time Quantum Monte Carlo solver implemented on the Intel Xeon Phi platform in chapter 5. We first discuss the motivation and formalism behind this implementation. Then, we show the detail of the implementation and the optimization. We also included some preliminary benchmarking results.

Chapter 2
Spin Glass

2.1 Introduction and Experimental Features

Spin glasses [2] are magnetic systems where the frozen-in quenched disorder leads to conflicting couplings among magnetic moments, which prevents the formation of long-range magnetic ordering, e.g.,ferromagnetic ordering.

The prototype material is a dilute magnetic alloy, with a small amount of magnetic impurity (such as Fe, Mn) randomly substituted into the lattice of a noble metallic host (i.e. Cu, Au). On the other hand, insulators such as $Eu_xSr_{1-x}S$, and $LiHo_xY_{1-x}F_4$ also show spin glass behavior.

The physics underlying the spin glass behavior comes from the quenched randomness: a pair of spins has a roughly equal a priori probability of having a ferromagnetic or an anti-ferromagnetic interaction. For the dilute magnetic alloy, the conduction electron-mediated Ruderman-Kittel-Kasuya-Yosida (RKKY) interactions between the localized moments oscillates strongly with distance (as shown in Figure 2.1),

$$J(R) = J_0 \frac{\cos(2k_F R + \phi_0)}{(k_F R)^3}, R \to 0. \tag{2.1}$$

Here J_0 and ϕ_0 are constants, and k_F is the Fermi wave number of the host metal. Since the distance between any pair of spins is random, some of the R will be positive, some will be negative, thus forming ferromagnetic/antiferromagnetic bonds randomly, and no spin alignment would satisfactory all exchange bonds. In the other words, the ground state energy cannot be obtained by minimizing the local energy of every pair of spins. Experiments demonstrated many unusual features of spin glass materials.

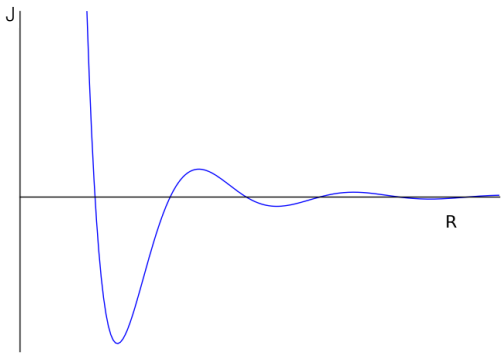


FIGURE 2.1. Sketch of RKKY interaction.

The experimental observation of a sharp cusp in the AC susceptibility, first carried out on a metallic AuFe by Cannella and Mydosh [9], as shown in as shown in Figure 2.2, sparked the research on spin glass materials. Tholence and Wassermann [51], Wassermann and Tholence [52], Kline et al. [53] also found similar cusp occurring in PtMn alloys. Upon applying a magnetic field, Cannella and Mydosh also found that even a weak magnetic field strongly rounds the cusp of $\chi(T)$. In Figure 2.2, the susceptibility data is presented for samples with 1 and 2 at.% Fe. For each concentration, as the applied field increases, the maximum becomes smaller and broader, similar to those observed by Lutes and Schmit [54].

The AC susceptibility of spin glass also features pronounced frequency dependence. In Figure 2.3, we show the AC susceptibility of CuMn for various AC frequencies ν ranging from 10 Hz to 10,000 Hz. As shown in the figure, the peak of $\chi(T)$ gradually shift to lower temperature with decreased ν. This is natural considering that as one probes the spin motion at longer time scales, the slowing down and freezing of the spins tend to occur at lower temperatures.

These behaviors are in sharp contrast to conventional magnetic systems say a ferromagnet. The susceptibility diverges at the critical point, the susceptibility, in this case, is the linear response function to the field. This can be understood

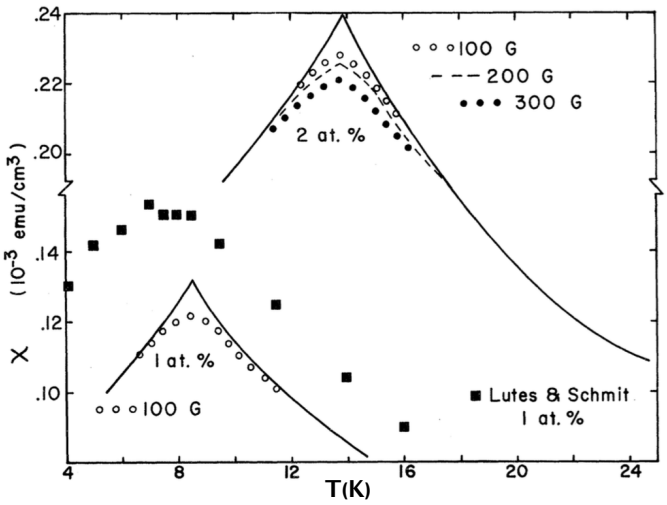


FIGURE 2.2. Susceptibility of Au-Fe alloys ploted vs. temperature, showing the curve for zero field, and for various applied fields. from Cannella and Mydosh [9]. Full curves refer to zero field. A field of ∼100G destroys the peak. The data of Lutes and Schmit [54] for 1 at.% is also included. See text for details.

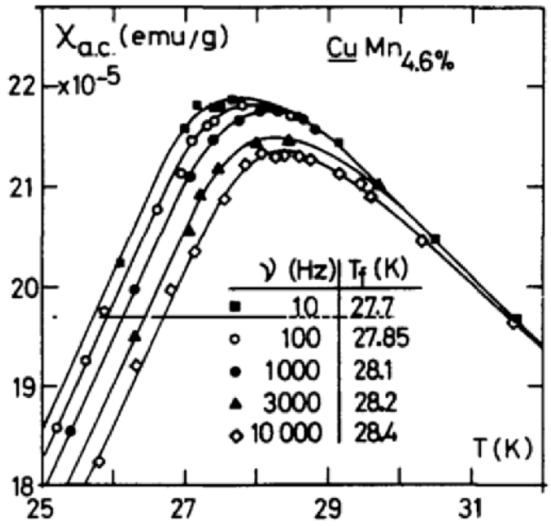


FIGURE 2.3. The AC susceptibility of CuMn with 4.6% Mn plotted vs temperature, for various ac frequencies, from Tholence [55]. The slowing down and freezing of spins happen at lower temperatures as one probes the spin motion at longer time scales. See text for details.

as the conjugate variable of the magnetization, defined as $M = \lim_{N\to\infty}\langle s_i\rangle/N$, is the linear magnetic field. The spins point to a fixed preferential direction in the ordered phase. The cusp in the susceptibility has long been associated as the defining nature of a spin glass system. However, if the spin glass transition is a truly thermodynamic transition, we expect divergence in the response function corresponding to the order parameter.

It was soon found by Edwards and Anderson that the order parameter should be characterizing an order in which each individual spin can point to a fixed direction, but the direction of each spin is random. The original form of the Edwards-Anderson order parameter can be defined as $Q = \lim_{N\to\infty}\langle s_i\rangle^2/N$. It is clear the absence of the divergence in the usual linear response is due to that the conjugate variable of an external field does not diverge. If the magnetization is expanded in term of the magnetic field for two lowest orders, we can define the nonlinear susceptibility, $M = \chi h - \chi_{nl}h^3$, where χ_{nl} is the nonlinear susceptibility. One can easily show that the nonlinear susceptibility is proportional to the spin glass susceptibility as the fluctuations of the Edwards-Anderson order parameter Q.

The direct evidence of a thermodynamic spin glass transition can be deduced from the study of the non-linear susceptibility. The measurement is usually done by superconducting-quantum-interference-device (SQUID). The non-linear susceptibility can be extracted from the curve of the magnetization as a function of magnetic field. This provides a direct access to the critical temperature and the exponent for the spin glass susceptibility.

The magnetization process of spin glass is characterized by strong remanence effects. Figure 2.4 shows remanent magnetization of AuMn measured in two ways: the thermoremanent magnetization is measured by cooling the sample in a field H from above T_g to $T < T_g$, and removing the field afterward; the isothermal

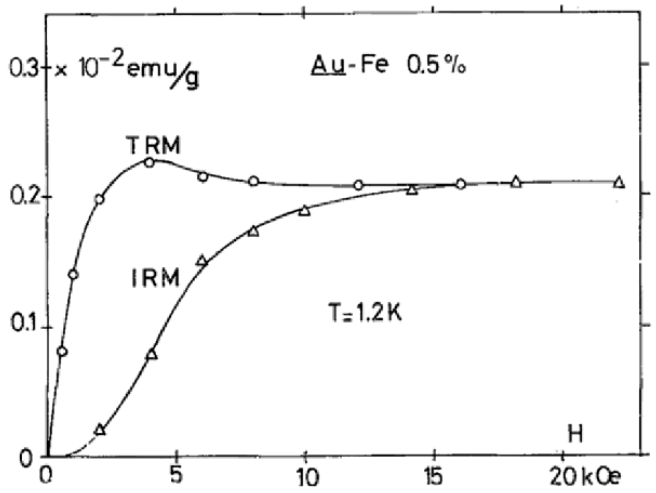


FIGURE 2.4. The isothermal remanent magnetization (IRM) and thermoremanent magnetization (TRM) of AuMn with 0.5% Mn vs magnetic field, from Tholence and Tournier [56]. The temperature is 1.2K. See text for details.

remanent magnetization is measured by first cooling down the sample in zero field to below T_g, then applying a field and removing it. At lower fields, there is a clear difference between the two cases. Upon increasing H, the TRM increases to a maximum and the decrease, while the IRM shows a monotonic increase. With a large magnetic field, both TRM and IRM saturate and agree.

A particularly interesting feature of this irreversible behavior in spin glasses is the slow decay of the various remanent magnetizations with time. Relaxation phenomena occur below Tc on a typical timescale of 1 sec – 1hr [56–58]. This relaxation is distinctly non-exponential. It can be described in terms of power-laws[58] or even, at not too late stages, by a logarithmic behavior[57] (see Figure 2.5).

The specific heat of various spin glass materials has been measured and analyzed. In Figure 2.6, we show the temperature dependence of magnetic specific heat of CuMn with 1.2% Mn. The arrow on the x-axis indicates the spin glass transition temperature T_g. The data shows a broad maximum above T_g, with no anomaly at T_g. Below T_g, the specific heat exhibits T-linear behavior.

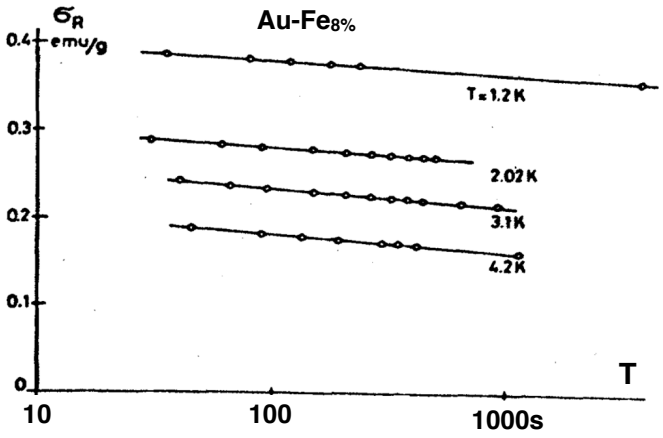


FIGURE 2.5. Isothermal remanent magnetization of Au-Fe with 8 at.% Fe, plotted vs. time (logarithmic scale) for several temperatures, showing the logarithm decay of remanent magnetization. Taken from Holtzberg et al. [57]. See text for details.

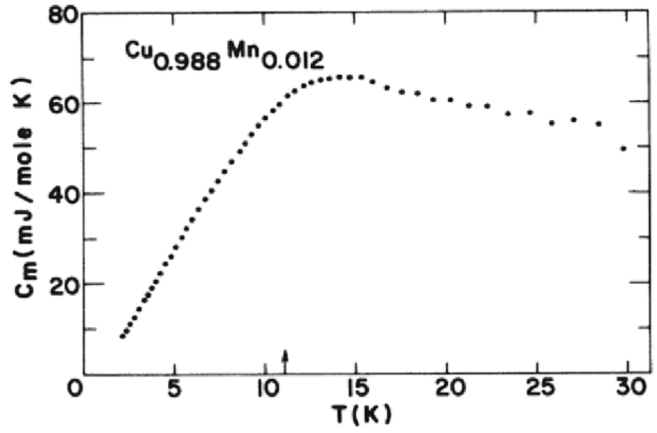


FIGURE 2.6. Magnetic part of specific heat of a Cu-Mn alloy plotted vs. temperature, showing a broad maximum above T_g, with no cusp at T_g. Arrow indicates where susceptibility has its cusp. Taken from Wenger and Keesom [59]. See text for details.

Other properties, such as DC susceptibility [60, 61], imaginary part of AC susceptibility [62, 63], etc., were also extensively studied. These experimental facts suggest that spin glass system has no conventional long range magnetic ordering and exhibits very slow dynamics. For experimental systems, equilibrium can never be achieved.

2.2 Theoretical Understanding on Spin Glass

A simple model that captures the consequences of disorder is an Ising model with quenched randomly disordered couplings, first proposed by Edwards and Anderson[10]:

$$H = -\sum_{\langle i,j \rangle} J_{ij} S_i S_j - h \sum_i S_i. \tag{2.2}$$

Here S_i is the spin in a d-dimensional lattice that can take values ± 1, $\langle i, j \rangle$ indicates nearest neighbors with the coupling J_{ij} between them, and h is the external field.

Numerical evidence suggests that the criticality of the three-dimensional spin glass systems are largely independent of the distribution of the randomness, that is they are in the same universality class for different distributions. But, the two main paradigmatic cases for the J_{ij} in Edwards-Anderson model are:

- Gaussian distribution of random coupling:

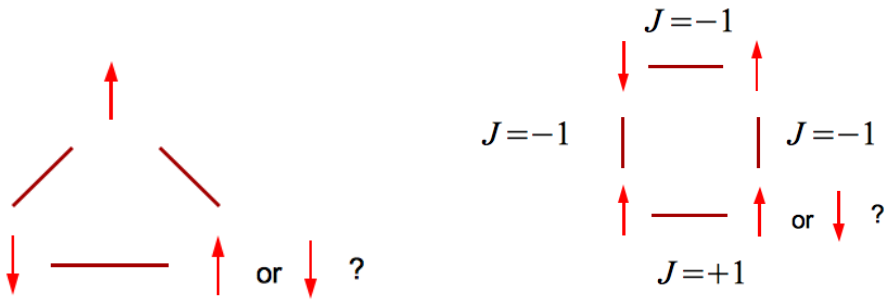
$$P(J_{ij}) = \frac{1}{\sqrt{2\pi}} \exp^{-J_{ij}^2/2}; \tag{2.3}$$

- Bimodal ($\pm J$) distribution of random coupling:

$$P(J_{ij}) = \frac{1}{2}[\delta(J_{ij} - 1) + \delta(J_{ij} + 1)]. \tag{2.4}$$

2.2.1 Frustration

Frustration naturally presents in the Hamiltonian in Equation 2.2, when no spin configurations can satisfy all couplings at the same time. Figure 2.7 demonstrate



(a) Geometrical Frustration. Here all couplings are anti-ferromagnetic.

(b) Frustration due to randomness. Here $J = -1$ indicates an anti-ferromagnetic coupling, while $J = +1$ means ferromagnetic coupling.

FIGURE 2.7. Frustration in Edwards-Anderson model.

two situations where frustrations happens. In Figure 2.7(a), the two spins on the top and the left are anti-parallelly aligned due to the antiferromagnetic coupling between them, but there is not a preferred spin direction for the third spin that can satisfy both the antiferromagnetic bonds. In Figure 2.7(b), the frustration comes from the random distribution of J_{ij}.

The frustration is a key factor that leads to many features which make spin glass a complex system. These features include: the existence of many metastable states; the rugged energy landscape; and dynamical behaviors such as slow relaxation, irreversibility, memory effects, hysteresis, etc.

2.2.2 Mean field Solution

An infinite-ranged version of spin glass models was proposed by Sherrington and Kirkpatrick (SK) [11, 12].

$$H = -\frac{1}{\sqrt{N}} \sum_{1 \le i \le j \le N} J_{ij} S_i S_j. \tag{2.5}$$

Here J_{ij} is chosen from a Gaussian distribution in equation 2.3. This model has an equilibrium phase transition at $T_g = 1$.

For the spin glass phase below T_g, Parisi[13–15] employed a novel ansatz and developed a possible physical interpretation of the nature of spin glass, which is now

known as the "Replica Symmetry Breaking" (RSB) picture. The main idea behind the picture is that the spin glass phase consists of an infinite number of "pure states" that form a hierarchy rather than follow simple symmetry transformation.

Here we discuss the Replica Symmetry Breaking picture by looking at the replica trick. The free energy F satisfies

$$\exp(-\beta F(\{J_{ij}\})) = \mathrm{Tr}_{\{S\}} \exp[-\beta H(\{S_i\};\{J_{ij}\})]. \tag{2.6}$$

In an experimental situation, it is usually assumed that the sample is sufficiently large that it may be considered to be composed of a large number of sub-systems, each of which can be described as a disordered realization of $\{J_{ij}\}$. Thus, a measurement of any observable in such a systems corresponds to an average over all sub-systems, i.e., an average of the ensemble of all realizations of J_{ij}.

Thermodynamic quantities may be calculated from the free energy.

$$\begin{aligned} F_e = [F(\{J_{ij}\})] &= \int P(\{J_{ij}\}) F(\{J_{ij}\}) d\{J_{ij}\} \\ &= -k_B T \int P(\{J_{ij}\}) \log Z(\{J_{ij}\}) d\{J_{ij}\} \end{aligned} \tag{2.7}$$

The logarithm is usually calculated by the replica trick:

$$\log Z = \lim_{n\to 0} \frac{Z^n - 1}{n}. \tag{2.8}$$

Then we need to calculate

$$\begin{aligned} [Z^n] &= \left[\prod_{\alpha=1}^{n} Z_\alpha\right] \\ &= \left[\mathrm{Tr}_{S^1} \ldots \mathrm{Tr}_{S^n} e^{-\beta \sum_{\alpha=1}^{n} H(\{S^\alpha\},\{J_{ij}\})}\right] \\ &= \mathrm{Tr}_{S^1} \ldots \mathrm{Tr}_{S^n} \left[e^{-\beta \sum_{\alpha_1}^{n} H(\{S^\alpha\},\{J_{ij}\})}\right]. \end{aligned} \tag{2.9}$$

Here $S^1, \ldots S^n m$ refers to n replicas of the same disorder realizations. And thus the effective Hamiltonian H_n of the replica system is given by

$$e^{-\beta H_n(S^\alpha)} = \left[e^{-\beta \sum_{alpha=1}^{n} H(\{S^\alpha\},\{J_{ij}\})}\right]. \tag{2.10}$$

The Hamiltonian for the n replicas of the system is symmetric under interchange or permutation of the replicas, but is not expressible as the sum of n Hamiltonians, one for each replica, since the disorder average has coupled the replicas. As long as n is an integer, the permutation symmetry of the Hamiltonian H_n is manifest. However, in the limit $n \to 0$, this symmetry can be spontaneously broken.

Consider the quantity q, which measures the overlap of the samples after a long time relaxation process:

$$q^{\alpha\beta} = \frac{1}{N} \sum_i S_i^\alpha S_i^\beta, \tag{2.11}$$

where α and β are two copies of lattice with the same disorder configuration, but simulated with different random seeds, so they are statistically independent of each other.

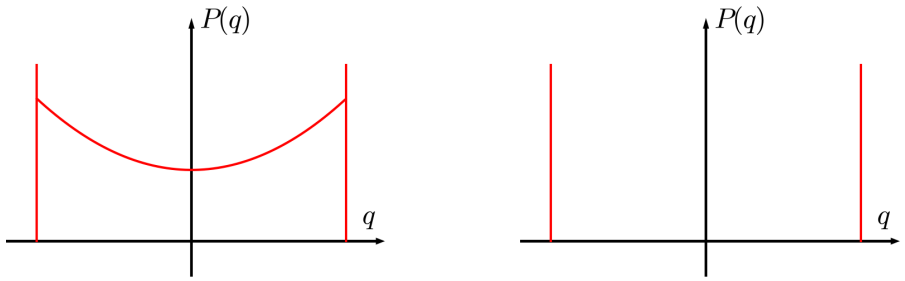
The Edwards-Anderson order parameter, q, is the order parameter of measuring the breaking of ergodicity. In the thermodynamic limit, if this is zero, the system is ergodic, if this is finite, the system breaks the ergodicity. Some parts of the phase space can never be sampled.

Since H_n is symmetric under permutation of the replicas, it might be considered that $q^{\alpha\beta}$ is independent of which replicas α and β are chosen, i.e. that $q^{\alpha\beta}$ is equal to a number q. However, it turns out that this is not correct. Below the transition temperature, $q^{\alpha\beta}$ is not independent of α and β, and replica permutation symmetry is spontaneously broken. The order parameter q is a $n \times n$ matrix, and therefore one cannot simply take the limit of $n \to 0$.

Replica symmetry breaking is successful in the description of infinite range models, but whether it holds in finite dimensions remains the most prominent open question in the study of spin glass systems. A competing picture, known as droplet/scaling[19, 20], is based on domain-wall renormalization group ideas. In this picture, there is only a single of pure states that are spin-flip-symmetrical at

low temperature in any finite dimension. The difference between the consequence from these two pictures will show up in the order parameters for spin glass as we will define in the following.

According to the Parisi solution, for fixed J and (large) N, the structure of the overlap is nontrivial, as displayed in Figure 2.8(a); while in droplet picture, in the thermodynamic limit, the distribution is just a pair of delta functions at ± 1, as displayed in Figure 2.8(b).



(a) Replica symmetry breaking picture (b) Droplet picture

FIGURE 2.8. Sketch of the overlap distribution $P(q)$ for the replica symmetry breaking picture, and the droplet picture.

2.3 Finite Size Scaling

Most phase transitions can be described by an order parameter, which measures the degree of order in a system. Usually, an order parameter is zero in one phase, and non-zero in the other. At the critical point, the susceptibility of the order parameter should diverge at the thermodynamic limit. An example of an order parameter is the magnetization in a ferromagnetic system.

Second-order phase transitions, such as the magnetic transition in the Ising model, and spin glass transition in the Edwards-Anderson model, can be characterized by their power law behaviors for various quantities, such as heat capacity and susceptibility, close to the critical point. The systems can be categorized into

different universality class according to the values of the exponents of these power law behaviors. For example,

$$\begin{array}{lrcl} \text{Magnetization} & M & \sim & |T-T_g|^{\beta}, \\ \text{Magnetic susceptibility} & \chi_M & \sim & |T-T_g|^{-\gamma}, \\ \text{Heat capacity} & C_V & \sim & |T-T_g|^{-\alpha}, \\ \text{Correlation length} & \xi & \sim & |T-T_g|^{-\nu}. \end{array} \tag{2.12}$$

Close to the critical temperature, one can use the following ansatzes:

$$M = L^{-\beta/\nu} g_M(tL^{1/\nu}), \tag{2.13}$$

$$\chi = L^{\gamma/\nu} g_\chi(tL^{1/\nu}), \tag{2.14}$$

$$C_V = L^{\alpha/\nu} g_C(tL^{1/\nu}), \tag{2.15}$$

where $t = (T - T_g)/T_g$.

A frequently used method to determine the critical point is to use the intersection points of the Binder cumulants:

$$U_L = \frac{1}{2}\left(3 - \frac{\langle M^4\rangle_L}{\langle M^2\rangle_L^2}\right). \tag{2.16}$$

For $T > T_g$, $\langle M^4\rangle_L = 3\langle M^2\rangle_L^2$. For $T < T_g$, $\langle M^4\rangle_L = \langle M^2\rangle_L^2$. As a result,

$$U_L = \frac{1}{2}\left(3 - \frac{\langle M^4\rangle_L}{\langle M^2\rangle_L^2}\right) = \begin{cases} 0 & \text{for} \quad T > T_g \\ 1 & \text{for} \quad T < T_g \end{cases}. \tag{2.17}$$

At T_g, the Binder ratio does not depend on L since

$$\frac{\langle M^4\rangle_L}{\langle M^2\rangle_L^2} = \frac{L^{-4\beta/\nu} g_{M^4}(tL^{1/\nu})}{\left(L^{-2\beta/\nu} g_{M^2}(tL^{1/\nu})\right)^2} = g_c(tL^{1/\nu}). \tag{2.18}$$

Therefore, U_L tends towards an universal value independent of the system size.

So one can use various system sizes L, calculate U_Ls as functions of T, and find the point where the $U_L(T)$ curve cross to identify T_g. The catch of the finite size

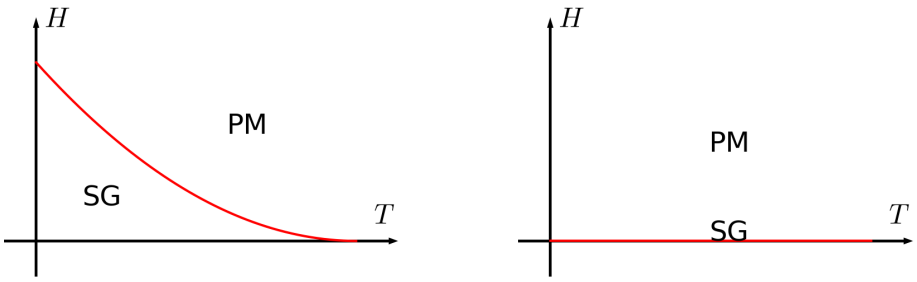
scaling is that the power law scaling behavior is valid only for the system is large, usually one would like to have the system sizes to be larger than the correlation length. Even though, the finite size scaling can in principle extract the exponent of the thermodynamic system, it is still desirable to simulate large system sizes.

2.4 Difficulties and Outstanding Problems

The Edwards-Anderson model is a deceptively simple problem. Since it is a classical spin model, one may think that its numerical study can be simply carried out by Monte Carlo methods on conventional hardware. One of the defining signatures of spin glass systems is their long relaxation time. For sufficiently low temperatures, the system becomes very sluggish and equilibration is prohibitively difficult even for modest systems sizes. Moreover, it has been shown that finding the ground state of the three-dimensional Edwards-Anderson model is an NP-hard problem. [22] Until recently, there has been no consensus on whether there is a finite spin glass critical temperature in the three-dimensional Edwards-Anderson model.

Due to the difficulty in the simulation, there is still no general consensus on which of the two competing pictures is correct. An import discriminator between the theories is the predicted behavior of the system when the temperature is decreased in the presence of an applied magnetic field. In the mean-field approximation, the de Almeida-Thouless line separates the high-temperature paramagnetic phase from the spin glass phase (Figure 2.9(a)). With the droplet/scaling theory, an applied magnetic field is predicted to remove the phase transition completely (Figure 2.9(b)).

Recent work supports that there is only a pair of two ground states in two dimensions. In four dimensions, the JANUS show evidence for the presence of a spin glass phase transition in a field. In three dimensions, numerical simulations give conflicting results.



(a) Replica symmetry breaking picture (b) Droplet picture

FIGURE 2.9. Sketch of the de Almeida-Thouless line in the replica symmetry breaking picture, and the droplet picture.

2.5 Algorithms for Spin Glass Simulation

The breakthrough in the numerical study of spin glass systems came with the introduction of the parallel tempering method[64–66]. Parallel tempering (PT), also known as replica exchange, is a simulation method aimed at improving the dynamic properties of Monte Carlo sampling. Instead of simulating one Markov chain at a time, one runs N copies of the system with random seeds at different temperatures and exchange configurations based on the detailed balance condition. By making the configurations at high temperatures available to simulations at low temperatures, this method allows better sampling over the entire energy landscape. We discuss this method further in 3.2.4.

Simulated annealing[67] is another commonly used algorithm for heuristic optimization, due to its simplicity and effectiveness. In physics, one usually select the energy as the cost function, start the Monte Carlo simulation at a high temperature and slowly tune down the temperature during the simulation, and in the end, the configuration of the system would stay in a local minimum. With slow-enough annealing and multiple repetitions, one would expect to find the global minimum.

Population annealing[68] combines simulated annealing and Boltzmann weighted differential reproduction within a population of replicas to sample equilibrium

states. Similar to simulated annealing, population annealing involves lowering the temperature of the system at a sequence of temperatures. However, population annealing uses a population of replicas and this population is resampled at each time step. By doing this, population annealing aims to ensure that the population always stay close to the Gibbs distribution. Population annealing is naturally a massively parallel algorithm, as realistic spin glass simulation using population annealing require population sizes of the order 10^6 or more.

Another possibility to overcome the diverging autocorrelation time problem is the multicanonical reweighting method[69]. Instead of sampling the Boltzmann distribution $P = \exp(-\beta Ep)$, multicanonical ensemble uses the MetropolisHastings algorithm with a sampling distribution given by the inverse of the density of states of the system. The density of states has to be known a priori or be computed using techniques such as the Wang and Landau algorithm.

Chapter 3
Implementation on Graphic Processing Units

This following chapter is a published work titled **Parallel Tempering Simulation of the three-dimensional Edwards-Anderson Model with Compact Asynchronous Multispin Coding on GPU** that appears in Computer Physics Communications. [70]

In this paper, we discussed an efficient GPU implementation of Monte Carlo Simulation of 3D Edwards-Anderson model, with parallel tempering and multispin coding technique.

This paper is written in collaboration with Ye Fang, Ka-Ming Tam, Zhifeng Yun, Juana Moreno, J. Ramanujam and Mark Jarrell. This work will also appear in Ye Fang's doctoral dissertation.

The idea of the project was first proposed by Ka-Ming Tam. Ye Fang and I developed the implementation together. During the collaboration, I focused on the validity of the code, including the Monte Carlo sampling procedures, the parallel tempering movements, and the measurements, and benchmarked the code against published results. Ye Fang dedicated his efforts on the efficiency of the code, including introducing and testing various multispin coding paradigms, optimizing the procedure calls, the memory accesses, and core computations, and profiling the code to evaluate the performance.

When writing this paper, Ye started the first draft and wrote the majority part of implementation, optimization and benchmark. Ka-Ming and I contributed to the physics part of this paper, including the introduction, the background, and the algorithm, and improved the overall writing. I also plotted many of the diagrams

and figures. During the writing of this paper, Zhifeng Yun, Juana Moreno, J. Ramanujam and Mark Jarrell review the paper draft in several round and give valuable feedbacks.

3.1 Introduction

Stochastic or Monte Carlo simulation is one of the most important methods in the study of complex interacting systems. However, even with the huge success of Monte Carlo methods, many systems remain very difficult to simulate. The main obstacle very often is the long required simulation time, while the memory demands are quite modest. A prominent example is the Edwards-Anderson (EA) model, where the inherent randomness and frustration lead to very long relaxation times. Although the Edwards-Anderson model has been intensively simulated over the past few decades, including implementations using gate-level reconfigurable processors [71] and some dedicated computers designed specifically for solving this model, [72–76] many aspects are still far from completely understood. Some prominent topics, such as the nature of the spin glass phase below the upper critical mdimension, remain highly debated issues. [21, 25, 77–86]

The Graphics Processing Unit (GPU) provides an opportunity to significantly improve the computational performance of Monte Carlo simulations of classical systems. Massive parallelism and acceleration can be achieved by implementing these algorithms on GPUs. In the past few years some GPU accelerated simple spin models have been proposed, including the two-dimensional Ising model by Hawick et al. [87] and Block et al. [88], and the Ising model in the cubic and network lattices by Preis et al. [89]. Weigel [90, 91] studied the Ising and the Heisenberg models in both two- and three-dimensional lattices. These implementations focus predominately on unfrustrated systems with large lattice sizes. In this study, we mainly focus in the simulation of a random frustrated Ising system in equilibrium.

Due to its slow relaxation rate, a large number of Monte Carlo steps are required, at the same time the system sizes that can be simulated are relatively small, in most cases limited to only a few thousands sites. Precisely because of these characteristics, Monte Carlo simulations of random frustrated systems are a good match for the GPU computing architecture.

Our implementation targets cluster computers with NVIDIA Fermi GPUs. Using C/CUDA we control and tune details of the program. We expose the inherent parallelism of the algorithm to the GPU accelerator, including parallel computation on multiple sites, multiple temperature replicas and multiple disorder realizations. The memory requirements are efficiently handled through memory tiling. In addition, the computation is simplified and vectorized using table look-ups and the Compact Asynchronous Multispin Coding (CAMSC). We also substitute all floating point arithmetic with integer or bit string computations while preserve the same precision. Combining various tuning techniques, we achieve an average spin flip time of 33.5 picoseconds. This is the fastest GPU implementation for the random frustrated Ising system on a 16^3 cubic lattice, and is comparable to that obtained with a field programmable gate array (FPGA) hardware [92] for small to intermediate system sizes. We note that a very recent preprint reported a faster speed in a new FPGA system[26].

The paper is organized as follows. In Section 2, we discuss the algorithm. In section 3, we present an outline of the code framework. The implementation and optimization methods are described in Section 4. Section 5 shows the experimental results. Conclusions and future directions are described in Section 6.

3.2 Theoretical Background

3.2.1 Spin Glass

The discovery of a plethora of unusual magnetic behaviors in disordered materials initiated the field of glassy systems.[2] Spin glasses are beyond the conventional description of long range magnetic ordering, e.g., ferromagnetic ordering. Some of their features, including their frequency-dependent susceptibilities and the discrepancy between zero-field and field cooling measurements, suggest that spin glasses have very slow dynamics. Notwithstanding most experimental spin glass systems, which exhibit glassy behavior, randomness and frustration seem to share some common properties. In real materials, dilution introduces randomness and directional or distance-dependent couplings, such as dipolar interactions in insulating systems and the Ruderman-Kittel-Kasuya-Yoshida coupling in metallic systems, introduce frustration.

The simplest model that captures the consequences of disorder is an Ising model with quenched randomly disordered couplings. This model was first proposed by Edwards and Anderson. [10] The mean field solution of the Edwards-Anderson model for infinite dimensions was first attempted by Sherrington and Kirkpatrick. [11] However, the replica symmetric mean field solution was found to be unstable below the Almeida-Thouless line, [93, 94] a line in the temperature-field plane below which replica symmetry is broken. The difficulty of obtaining a stable solution was solved by Parisi with his replica symmetry breaking ansatz. [13–15, 95–97] Although the mean field solution has been proven to provide the exact free energy for the spin glass phase in infinite dimensions, [98, 99] the spin glass physics in finite dimensions, which presumably is more relevant to experiments, is still not fully understood. Indeed, it had long been debated whether a spin glass phase at finite temperatures exists in three dimensions.

The Edwards-Anderson model may be deceptively simple. Since it is a classical spin model, one may think that its numerical study can be simply carried out by Monte Carlo methods on conventional hardware. One of the defining signatures of spin glass systems is their long relaxation time. For sufficiently low temperatures, the system becomes very sluggish and equilibration is prohibitively difficult even for modest systems sizes. Moreover, it has been shown that finding the ground state of the three dimensional Edwards-Anderson model is an NP-hard problem. [22] Until recently, there has been no consensus on whether there is a finite spin glass critical temperature in the three dimensional Edwards-Anderson model.

The breakthrough in the numerical study of spin glass systems came with the introduction of the parallel tempering method. It allowed the study of larger systems at lower temperatures than the simple single spin flip method. [64–66] Combined with improved schemes for finite size scaling, it is now widely believed that the thermodynamic finite-temperature spin glass phase does exist in the three dimensional Edwards-Anderson model [24]. As the upper critical dimension of the model is six [16–18], a prominent remaining question is the nature of the spin glass phase below the upper critical dimension [21]. In particular, if the spin glass can still be described by the replica symmetry breaking scenario, there should be an Almeida-Thouless line below the upper critical dimension. A possible test of whether the Almeida-Thouless line exists is to determine whether a spin glass phase exists under an external magnetic field. Correlation length scaling analysis seems to suggest the absence of the spin glass phase in cubic lattices when a finite external field is applied.[21] On the other hand, a recent study in four-dimensional lattices suggests that by using a different quantity for the finite size scaling analysis, a spin glass phase can be revealed. [25] Given the relevance of spin glasses and the on-going controversy on the nature of the spin glass phase below the upper critical dimension,

it is desirable to implement an efficient parallel tempering Monte Carlo algorithm using graphics processing units to accelerate the simulations. In this work we show that using the multispin coding method, [100] an efficient Monte Carlo algorithm can be implemented on the GPU.

3.2.2 Edwards-Anderson Model

We consider the Edwards-Anderson Model [10] on a simple cubic lattice. Spins on each lattice site have two states $S_i = +1$ or -1. The couplings J_{ij} are between nearest neighbors. In this study, we focus on a distribution of the couplings which is bimodal with a mean value of zero. That is, there are equal numbers of anti-ferromagnetic and ferromagnetic couplings. The effect of the distribution is certainly a non-trivial problem. We choose to focus on the bimodal distribution because it is best suited for multispin coding. In addition, a constant external field, h, is included in our implementation. The Hamiltonian is given by

$$H = -\sum_{i,j} S_i J_{ij} S_j + h \sum_i S_i. \tag{3.1}$$

3.2.3 Single Spin Flip Metropolis Algorithm

We implement the Metropolis algorithm as our sampling method. The spins are visited and tested for flipping according to the probability $P = \exp(-\beta \Delta E)$, where β is the temperature and ΔE is the energy change associated with the proposed spin flip. As the algorithm satisfies detailed balance, the sampling will generate a distribution according to the partition function provided that the simulation is performed long enough. This type of Monte Carlo simulation is called a Markov process, because the evolution of the state only depends on the state at the current step, and not on its history.

3.2.4 Parallel Tempering

For the simulation of glasses, the local single spin update algorithm is very slow in thermalizing the system. This problem is particularly severe when the temperature is close to the critical temperature for the second order transition. For certain spin glass models where random dilution is sufficiently large, some form of cluster algorithm can improve the rate of thermalization. Unfortunately, there is no efficient cluster methods for general spin glass systems. The possible exceptions are some random diluted systems or systems in low dimension [101–103]. Various other methods have been proposed in the past to improve the rate of thermalization including the umbrella sampling, the multi-canonical method, and rejection-free methods. It is now widely accepted that the parallel tempering method is one of the most efficient algorithms for improving the thermalization rate of general spin glass systems.

Parallel tempering uses several samples of the system within a range of temperatures (Figure 3.1). The low temperature sample is more difficult to thermalize due to the larger barriers between low energy configurations [65, 66]. However, as the probability to swap the configuration between the high and the low temperature samples increases, the chance of the system to escape from a local minima in the low temperature sample also increases. The efficiency of such a parallel tempering move can be measured by the time it takes for a sample to perform a round trip along the temperature axis, that is from the lowest to the highest temperature and back to the lowest temperature. This largely depends on the system being simulated. Fine tuning the range of temperatures and the spacing between them is crucial to optimize the performance. Some recent proposals have been tested on the non-disorder Ising model [104–106]. Models with explicit disorder such as the Edwards-Anderson lack an efficient general method. For a practical GPU imple-

mentation, one also needs to consider the effect of the number of replicas on the performance.

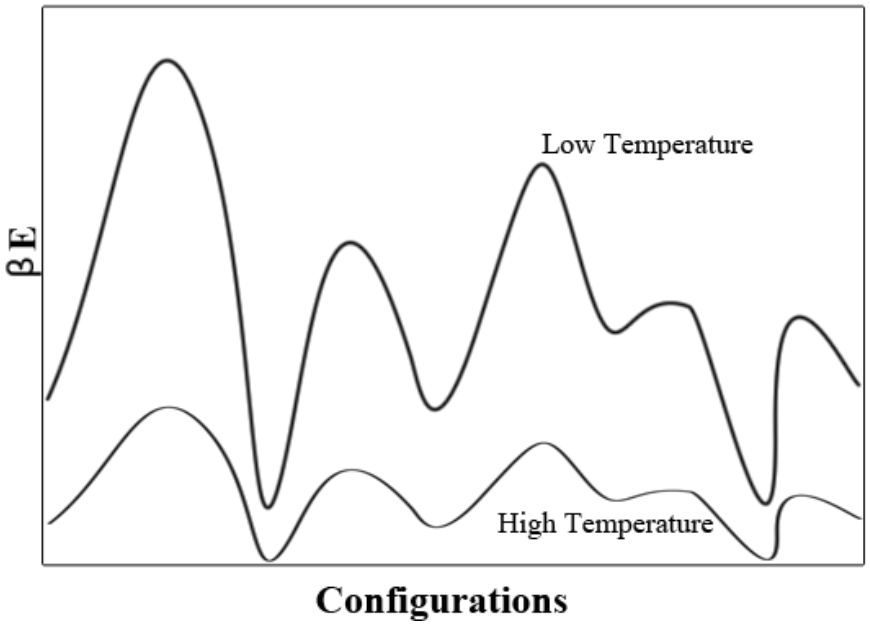


FIGURE 3.1. Schematic diagram of the free energy landscape. At high temperatures (small β) the barriers between configurations are reduced allowing the system to search through configurations more efficiently.

3.3 The Framework

The GPU implementation is discussed in this and the following sections. In our replica exchange spin glass simulation we exploit three levels of parallelism:

1. Several tens of thousands, or more, of independent disorder realizations are required to obtain good statistics.

2. For each disorder realization, usually a few tens of systems at different temperatures are needed to study the physics, such as the possibility of a critical point. We denote these systems as temperature replicas. In the parallel tempering simulation, different temperature replicas communicate with each other only during the parallel tempering swap; these swaps are performed after every few Metropolis single spin sweeps of the lattice.

3. We are mainly interested in systems on bipartite lattices. These are lattices that can be divided in two sub-lattices (A and B) with same sub-lattice spins do not directly coupling with each other. As a result, the update of the A sublattice is independent of the B sublattice.

These three levels of inherent parallelism allows an efficient GPU implementation. In this section we focus on the main structure of the code, which consists of three parts: (i) distributing the spin updates into different GPU threads; (ii) distributing different disorder realizations into different GPU blocks; and (iii) integrating and vectorizing the bit computations of many temperature replicas

3.3.1 Map Lattice Sites to GPU Threads

The spin lattice is represented by a three dimensional primitive cubic system. To update the sites in the lattice, we follow the common practice of employing a checkerboard decomposition that splits the sites into two sub-lattices shown in blue and red in Figure 3.2 . Since a blue site is surrounded by red sites and never directly interacts with other blue sites and vice-versa, it is permissible to update each sub-lattice in parallel. We construct two consecutive stages concentrating independently on each of the sub-lattices for parallel computation. The combination of the two stages delivers a lattice sweep of Monte Carlo updates. The lattice is assigned to a GPU thread block, and sites are split across the threads. Details about the lattice site to thread mapping will be discussed in Section 3.4.2 where we discuss memory optimizations. The total available thread-level parallelism is half of the total lattice sites, and specifically, falls into the range between $8^3/2 = 256$ to $16^3/2 = 2048$ since our simulation targets lattices between 8^3 to 16^3 sites.

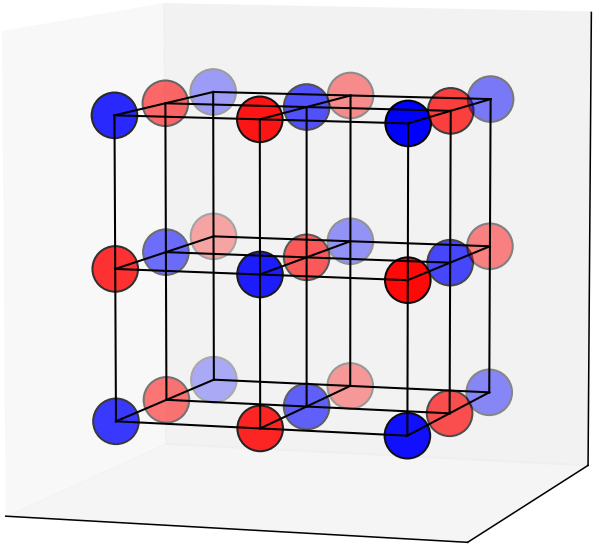


FIGURE 3.2. A demonstration of the 3D checkerboard decomposition. The blue and red sites are on different sub-lattices. Since the sites in a sub-lattice never directly interact with each other, it is permissible to update different sites in parallel.

3.3.2 Map Temperatures Replicas to Bits

The parallel tempering technique facilitates the systems to achieve equilibrium. We choose the temperature as the tempering parameter and generate systems with the same couplings but different temperatures, called temperature replicas. The temperature replicas are uncorrelated during the spin-flip process and can therefore be updated in parallel. However, they communicate and swap temperatures (Figure 3.3) after a few Monte Carlo sweeps. To better utilize the parallelism of multiple temperature replicas and minimize the communication overhead we have developed the Compact Asynchronous Multispin Coding (CAMSC), where spins from different temperature replicas at the same position are encoded into an integer. This leads to sub-word vectorization and a significant reduction of memory transactions. Details of our multispin coding procedure can be found in Section 3.4.3. The number of temperature replicas depends on the system size and the temper-

ature range. In our simulation we used 24 replicas for smaller systems, and 56 temperatures for bigger systems (for example, 10^3 and 12^3).

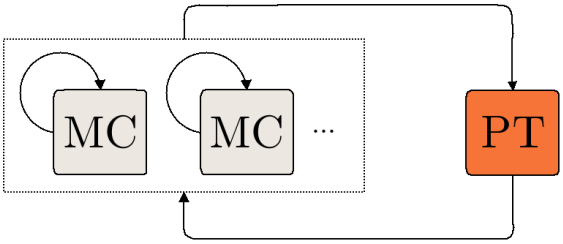


FIGURE 3.3. Many temperature replicas can be simulated simultaneously, each using an independent Monte Carlo process. These replicas may be exchanged after a configurable steps of updates. A single GPU thread block is responsible for updating all the Monte Carlo processes and manipulating the parallel tempering exchange.

3.3.3 Map Realizations to GPU Blocks

Spin glass simulations usually require a larger number of disorder realizations (10^4 or more) for reliable disorder averaging. A realization including all temperature replicas has been designated to a thread block. We launch numerous thread blocks across multiple GPUs of multiple hosts until we get the sufficient number of realizations for disorder averaging (Figure 3.4). To distribute these jobs across multiple nodes, we employ a Pthreads/MPI wrapper for the job distribution.

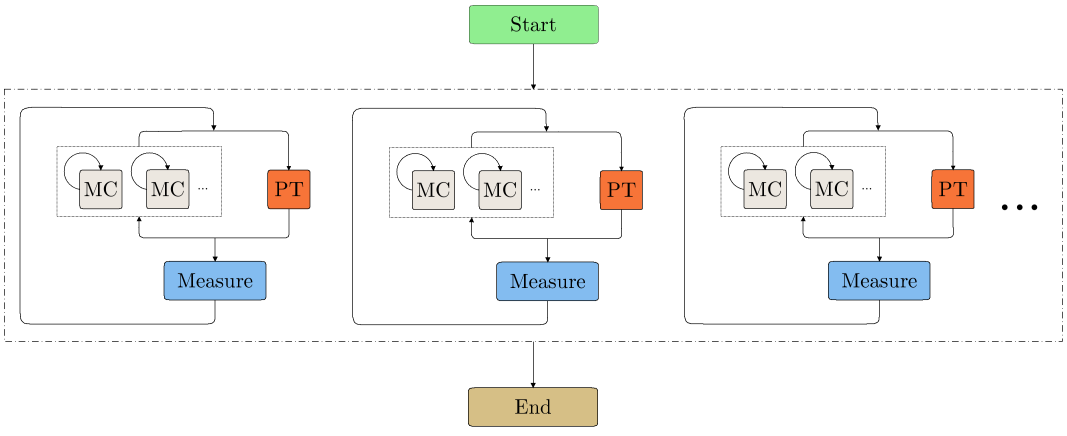


FIGURE 3.4. The outline of the simulation application. Disorder realizations are completely independent and can run simultaneously. Each realization contains a unique Monte Carlo parallel tempering process as depicted in Figure 3.3, and is assigned to a GPU thread block. This task level parallelism yields sufficient number of thread blocks and can fully occupy a parallel computer system.

3.3.4 Discussion

Some parallel processes are sequentialized for better memory locality. For example, although the temperature replicas could be fully parallelized as individual tasks or a lattice may be partitioned across multiple thread blocks, we avoid these forms of parallelism. The remaining parallelism is rich enough (with 10^4 or more thread blocks) to fully occupy the cluster.

To evaluate the performance, we employ a performance metric of average time (in picoseconds) per proposed spin flip for a single GPU card:

$$t = T_{\text{total}}/N_{\text{MCS}}/\left(N_{\text{spins}} \times N_T \times N_{\text{samples}}\right), \tag{3.2}$$

where T_{total} is the total wall time of a simulation; N_{MCS} is the number of Monte Carlo sweeps; N_{spins} is the number of spins within a lattice; N_T is the number of temperature replicas; N_{samples} is the total number of disorder realizations on one GPU card. We develop and benchmark the code on a NVIDIA GeForce GTX 580 GPU. Detailed platform configurations can be found in Section 3.5.

3.4 Implementation

We discuss implementation details in this section, including the construction of the GPU kernel, memory optimization, and various techniques used to simplify the computation.

3.4.1 Kernel Organization Optimization

Our simulation starts with the Pthreads/MPI job dispatcher that forks many CPU processes across the cluster computer system. Each CPU process is responsible for initiating a lattice realization, which is offloaded to its attached GPU for simulation until the spin variables or thermal averaged results are retrieved from the GPU back to the CPU for analysis. The GPU workload has three major components (Figure 3.5):

1. **Metropolis moves**: The Metropolis steps for the single spin update for each temperature replica. This is done by calculating the local energy change and then comparing the acceptance ratio to a uniformly distributed random number.

2. **Parallel tempering moves**: Parallel tempering swaps are performed after a few complete Monte Carlo sweeps of the lattice. This step requires the calculation of total energy for all temperature replicas; we use this to evaluate the acceptance ratio of parallel tempering swaps.

3. **Measurements**: The spin configurations are dumped to the GPU global memory periodically to provide data for the measurements. In practice, we perform one measurement for every few thousands Metropolis sweeps.

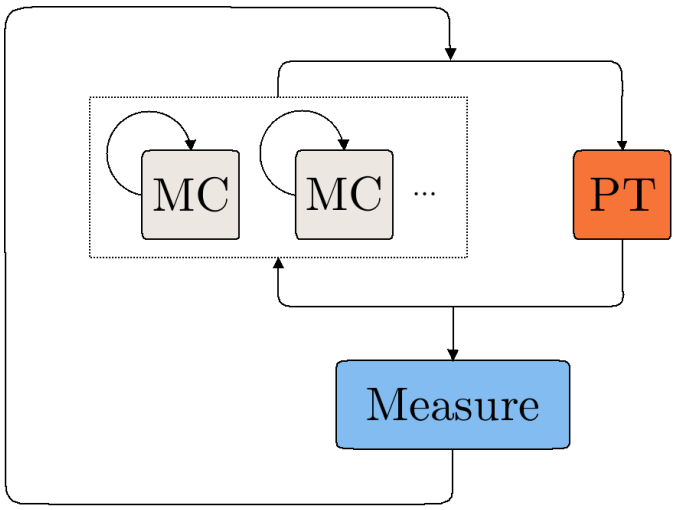


FIGURE 3.5. Three major components of the GPU program. One kernel calls Monte Carlo and parallel tempering, implemented as two device functions. Measurement is implemented as a separate GPU kernel.

The measurement code has little overlap with the Monte Carlo and parallel tempering codes, and it is called much less frequently. We implement this part of the code as an separate GPU kernel.

Both Monte Carlo and parallel tempering functions compute spin local energies. Parallel tempering requires additional steps to sum the local energies. Since an efficient implementation of sum (a form of reduction) consumes a considerable amount of shared memory, it may be efficient to separate the parallel tempering as a dedicated GPU function apart from the Monte Carlo. We denote this scheme **MC-PT separated**. Alternatively, the **MC-PT integrated** scheme combines both the Monte Carlo and parallel tempering in a single GPU kernel. Benchmarks (Figure 3.6) show that the **MC-PT separated** scheme always performs better regardless of the frequency of parallel tempering. However, we find that roughly 10 full Monte Carlo sweeps of the lattice between parallel tempering attempts is a speed/effectiveness sweet point.

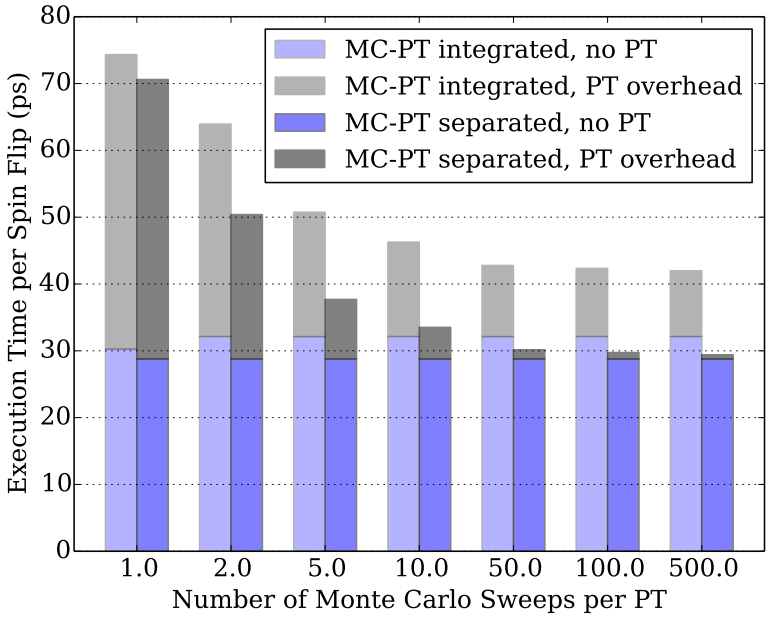


FIGURE 3.6. A comparison of the performance of the **MC-PT integrated** and the **MC-PT separated** schemes with different numbers of Monte Carlo sweeps between an attempted parallel tempering swap. The test is conducted with a 16^3 cubic lattice, shared memory probability table of integers, CURAND, and CAMSC.

3.4.2 Memory Optimization

Each spin interacts with its six nearest neighbors (Figure 3.7) as a seven-point 3D stencil [107, 108] with periodic boundary conditions. Unlike some stencil problems, e.g., the Jacobi finite difference solver for partial differential equations, in which the data for the new time step is completely based on the previous time step, the checkerboard decomposition allows the spin glass simulation to proceed with two consecutive update phases. Only half of the spins are updated in each of the phases. This unconventional stencil, associated with the checkerboard decomposition, leads to a stride-2 memory reference pattern and presents a more challenging memory optimization problem compared to the stride-1 pattern of typical stencils problems.

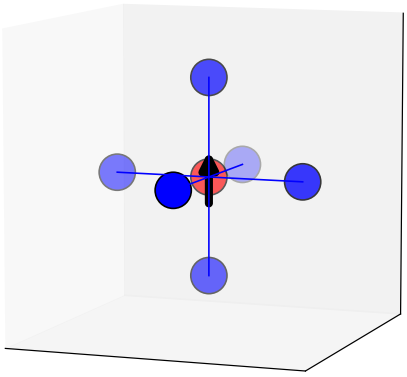


FIGURE 3.7. The memory access pattern for a single spin update where in addition to the state of the local spin, we also need the states of its 6 neighbors. Periodic boundary conditions are used.

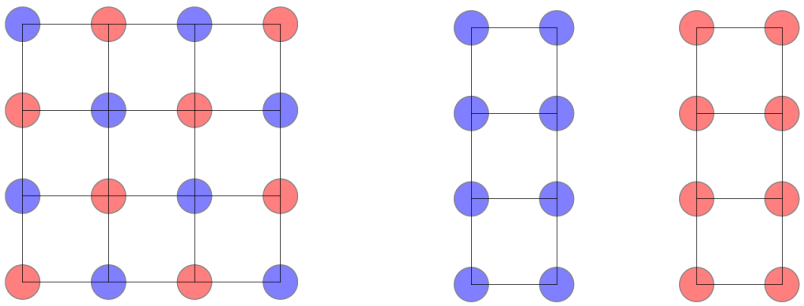
We propose three different schemes to address this problem.

1. The **Unified** allocation (Figure 3.8(a)) stores the checkerboard lattice in its native way as a single piece.

2. The **Separated** allocation (Figure 3.8(b)) breaks the sub-lattices into two chunks stored separately.

3. The **Shuffled** allocation [109] (Figure 3.9) mixes and integrates two temperature replicas, so that the memory access pattern is now identical to the conventional stencil. This is done by mixing the two temperature lattices in such a way that all the A sublattice spins from temperature 1 and the B sublattice spins from temperature 2 are packed together in the memory associated with one lattice. When the spins are being updated on this lattice, they are all independent of each other. They can be considered sequentially and continuously written into memory. Since there is no gap between each memory write, this should theoretically enhance the memory access speed.

The performance comparison on Table 3.1 suggests that the separated allocation is inferior due to its significantly lower memory performance. This is because of the more complicated control flows in the code. Overall, the unified allocation provides the best memory performance in terms of time spent for each spin and is used in our implementation.



(a) Unified (b) Separated

FIGURE 3.8. Unified and separated memory allocation schemes. The unified scheme stores the entire checkerboard lattice together. The separated scheme breaks the memory associated with each sublattice into separate continuous blocks of memory.

[1]These numbers are calculated by measuring the difference of execution time of a memory load and the execution time of assigning a constant to the variable.

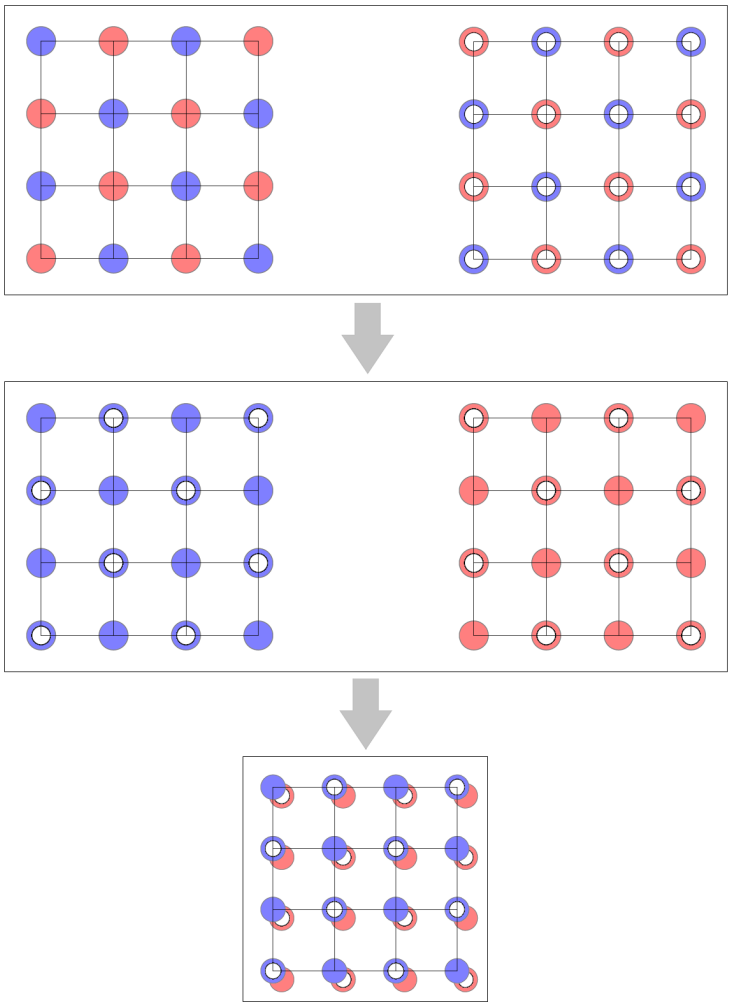


FIGURE 3.9. The shuffled allocation mixes and integrates two lattices, shown on the top of the figure. The first transformation is taking the A sublattice on the left (blue dots) and the B sublattice on the right (blue circles) to construct an intermediate lattice (middle left figure of blue color). Another intermediate lattice of red color is constructed similarly. We then integrate these two intermediate lattices together, which occupy different bit positions under the compact multispin coding scheme (Section 3.4.3). By using one integer lattice instead of two, we avoid doubling the memory consumption. Also, the memory access pattern is identical to that of the 7-point 3D Jacobi stencil.

The basic idea of multispin coding (MSC) is to present many binaries or short vectors in a longer packed word. For example, Ising spins may be stored with a single bit per spin, with 0 being spin down and 1 being up. In our particular implementation, we also encode the 4 bit string of one site's spin-flip probability table's row index (section 3.4.3) into an integer word. MSC [100, 110] yields a more

TABLE 3.1. Performance comparison of the unified/separated/shuffled storage allocation schemes for a 16^3 lattice. The definition of a transaction is a sequence of 7 loads and a store that serve the spin update.[1]

	Unified	Separated	Shuffled
Bandwidth(GB/s)	645.1	279.0	832.6
Time per transaction (ps)	49.608	107.756	38.432
Spins per transaction	24	24	16
Time per spin (ps)	2.067	4.345	2.402

efficient way of calculating local energies (E) and reduces the memory required for the spin configurations. This packing prevents the Arithmetic Logic Unit, which performs integer arithmetic and logical operations, and the memory bandwidth from being under utilized. Also, a memory transaction (7 loads and 1 store) can serve the calculation of multiple spins, which helps improve the relative memory performance.

The usual practice for a single lattice MSC is integrating a line of spins into an integer. We denote this conventional method as Synchronous Multispin Coding (**SMSC**). For the simulation of spin glass models, the temperature replicas provide an alternative approach with a different memory layout. One can pack the spins at a specific site but at different temperature replicas into an integer; we call this the Asynchronous Multispin Coding (**AMSC**). The main idea of these two multispin coding schemes are:

- SMSC: A packed word stores the spins from a single replica, but different sites.
- AMSC: A packed word stores the spins belonging to different temperature replicas of the same site.

We find the ASMC scheme to be more efficient. Its storage consumption is small enough to fit in the GPU shared memory. Furthermore, AMSC's index system is more straightforward, thereby simplifying optimization. The performance of these

different MSC schemes is described below. Here, we briefly discuss how the words associated with either scheme are organized into memory.

Three levels (Figure 3.10) of the memory hierarchy are employed that reflect the GPU memory architecture of global memory, shared memory and registers:

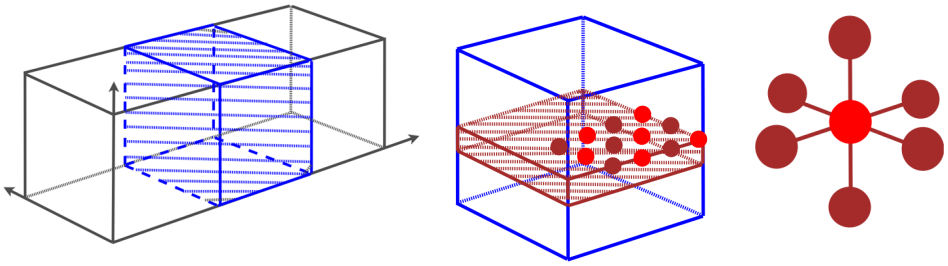
- **Level 1:** The main data resides in the GPU global memory. Due to the limitation that a 32 bit integer represents at most 32 spins, we may need multiple integer cubes (with an integer cube including one integer per site on the cubic lattice) if there are more than 32 temperature replicas.

- **Level 2:** The shared memory scratchpad holds the working set of an entire integer cube (no larger than $4 \times 16^3 = 16KB$). The data transfer between global and shared memory is quite modest because we do not need to switch to another integer cube until the Monte Carlo and parallel tempering swaps within the temperature replicas contained within the current cube are exhausted.

- **Level 3:** The GPU threads scan the shared memory scratchpad for two consecutive sublattices and load the data into registers. The threads are organized as multiple layers of 2D plates. We observe the optimal thread configurations are two or four layers ($16^2/2 \times 2 = 256$ or $16^2/2 \times 4 = 512$).

3.4.3 Optimizing the Computation

We may take advantage of the MSC mapping of the spins onto bits to dramatically reduce the number of floating point operations needed by the Monte Carlo parallel tempering calculations. For example, we may use a bitwise XOR ($\oplus$) as opposed to multiplication to calculate the energy. In the equations below, we denote the variables in the original notation with a superscript ^{o}, and variables without su-



(a) Global Memory (b) Shared Memory (c) Registers

FIGURE 3.10. Memory tiling. The global memory may hold several integer cubes (including one integer per lattice site) if there are more than 32 temperature replicas. The shared memory scratchpad holds the working set of an entire integer cube (no larger than $4 \times 16^3 = 16KB$). The registers hold the data needed for local spin updates.

perscripts are used in the transformed notation. The variables S, J, e and E stand for spin, spin coupling, bound energy and local energy respectively.

$$S^o \in \{-1, 1\}, J^o \in \{-1, 1\}$$

$$E_i^o = \sum_j S_i^o \times J_{ij}^o \times S_j^o, E_i^o \in \{-6, -4, -2, 0, 2, 4, 6\}.$$

$$S \in \{0, 1\}, J \in \{0, 1\}$$

$$E_i = \sum_j S_i \oplus J_{ij} \oplus S_j, E_i \in \{0, 1, 2, 3, 4, 5, 6\}.$$

Note that local energy E_i^o, the energy of a spin i in the field of its nearest neighbors, can only take one of seven values as indicated.

The computation is composed of four steps:

1. Energy: Compute the bound energy (e) and the spin's local energy (E).

$$\begin{aligned} e_{ij} &= S_i \oplus J_{ij} \oplus S_j \\ E_i &= \sum_j e_{ij} \end{aligned} \tag{3.3}$$

2. Probability: Compute the flip probability (P) for the Metropolis Monte Carlo, where the temperature (T) is an input parameters.

$$\begin{aligned} E^o &= 2 \times E - 6 \\ S^o &= 2 \times S - 1 \\ P &= \exp(2 \times (\frac{1}{T} \times E^o + h \times S^o)) \end{aligned} \tag{3.4}$$

3. Rand: Generate a random number (R).

4. Compare: Compare and update spins.

$$S = (P < R) \oplus S. \tag{3.5}$$

Equation 3.4 expresses the straightforward yet expensive method to generate the spin flip probabilities. However, since the number of input/output values is finite (i.e., combinations of 7 possible local energies E, 2 spins S, and no more than 32 temperatures T), a better solution is to deploy a pre-calculated look-up table. The table is a two-dimensional matrix (Figure 3.11), with T as the row index and $(E \times 2 + S)$ as the column index. The column index, as the combination of E and S, requires 4 bits for the address space. The maximum storage consumption of the table is 16 KB, assume that we have 32 rows times 14 columns times 4 bytes per entry (again, assume 32 temperature replicas). When a parallel tempering swap between two replicas at temperatures T_i and T_j is accepted, the two corresponding rows in the table are swapped.

We evaluate four different ways to calculate the probability in Equation 3.4 (Figure 3.12): (a) using the floating point exponential function from the math library, (b) using a less accurate GPU specialized exponential intrinsic function, (c) using the texture memory to store a table, and (d) a shared memory table. The result shows that an optimal table look-up saves close to half of the total

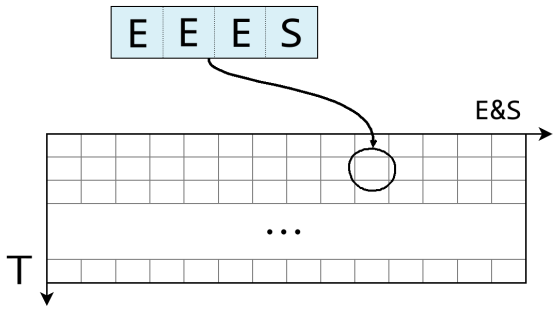


FIGURE 3.11. The organization of the probability look-up table.

computation time compared to direct computation of the probabilities. In addition, the shared memory table outperforms the texture memory table. This is because GPU threads are simultaneously computing on the same temperature replica, and are therefore accessing the same row of the table. This avoids bank conflicts, so that the high bandwidth and low latency performance potential of the shared memory is fully exploited.

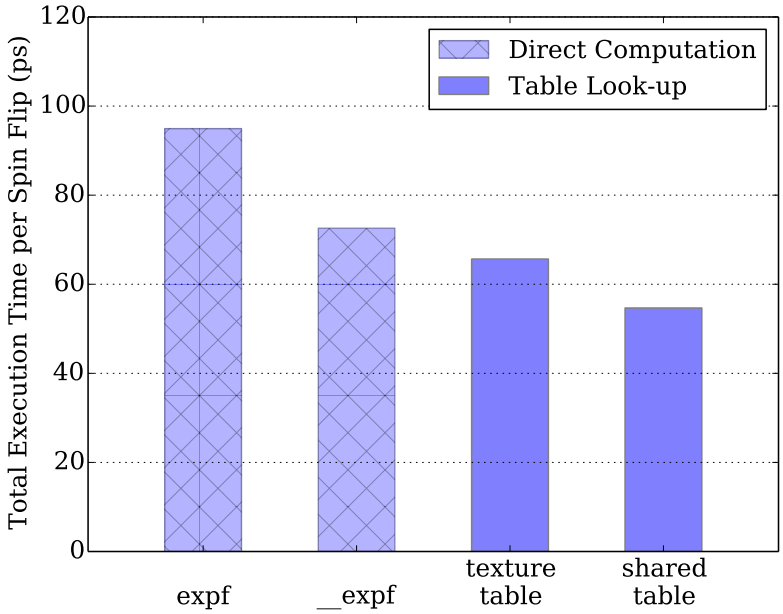


FIGURE 3.12. A comparison of the overall time consumed per spin flip using four different methods to compute the exponential probability in Equation 3.4 as described in the main text. The experiment is done for a 16^3 lattice, fp32 CURAND and AMSC1. No parallel-tempering is performed.

The simulation requires uniformly distributed random numbers between zero and one. However, due to the fact that pseudo random number generators (RNGs) manipulate integer values internally, directly using integer return values from the RNG provides higher performance and preserves identical precision. As a consequence, we convert the pre-generated probabilities from single precision floating point numbers to 32 bit unsigned integers.

We evaluated three random number generators: (i) NVIDIA CURAND library of XORWOW algorithm [111], (ii) rand123 [112] philox4x32_7 (version 1.06), and (iii) our implementation of a multi-threaded 32 bit linear congruential generator (LCG). We decide to adopt CURAND due to its higher performance (Figure 3.13) and quality [113].

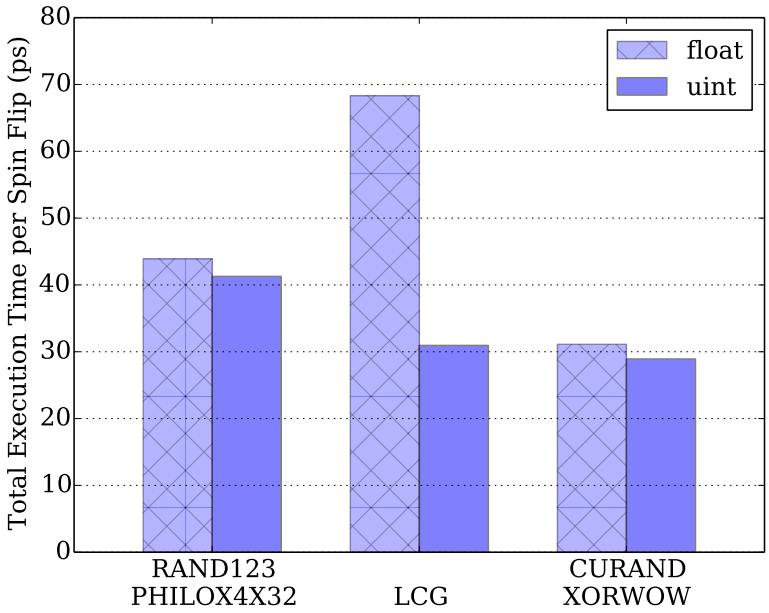
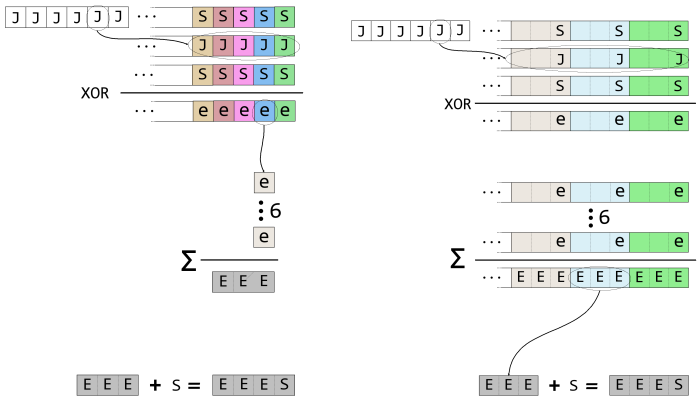


FIGURE 3.13. A comparison of the overall time required per spin flip using different random number generators. The experiment used a 16^3 lattice, a shared memory probability table and CAMSC. No parallel-tempering is performed. The loop that consumes random numbers has been unrolled four times to match the four return values of rand123 philox4x32_7.

We have briefly described the Multispin Coding (MSC). We have developed the Asynchronous Multispin coding (AMSC) as a more efficient alternative to the conventional Synchronous Multispin Coding (SMSC) for calculating the local energies (E), generating the 4 bit string for the column index of the spin-flip probability table (section 3.4.3), and optimizing the memory bandwidth utilization. In our particular GPU implementation, we use four byte unsigned integers, which hold up to 32 bits, as a packed word. Each spin, denoted as 0 or 1, takes only one bit of this packed word. Thus, the calculations in Equation 3.3 can be vectorized via bit-wise operations. We integrate the J bits with the S bits in the same integer, so that we can fetch both the coupling and the spins in only one memory transaction. We then multiply the coupling with a bit-mask to match the pattern of S, and calculate the bond energy with bit-wise XOR operation. The next step is to add the six bond energies around a spin to obtain the local energy. To vectorize this process we need to reserve empty bits to avoid overflow, since the local energy takes 3 bits of storage. In this way, each spin, together with the empty bits reserved for calculation, constitute a virtual segment. We derived three variations of AMSC with different segment width of 1, 3 and 4, denoted as AMSC1, AMSC3 and AMSC4 respectively. In AMSC1 and AMSC3, some calculations are sequentialized to avoid overflow. Figures 3.14 and 3.15 demonstrate how the the different variations of MSC parallelize the computations in Equations 3.3, 3.4 and 3.5.

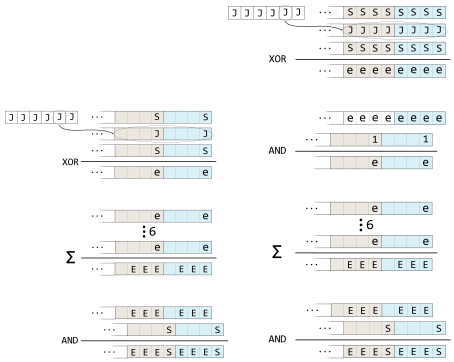
Figure 3.16 illustrates that AMSC3 and AMSC4 are favored over AMSC1 due to improved overall performance. However, we also observe proportionally longer times for the memory transactions. This demonstrates the limitation of the AMSC scheme: there does not exist an optimal segment width that simultaneously provides the highest memory density, and the richest vectorization opportunities in computation.



(a) AMSC1 (b) AMSC3

FIGURE 3.14. This figure demonstrates the computation of $(E \times 2 + S)$ for the purpose of accessing the probability look up table with the deployment of two variations of Asynchronous Multispin Coding (AMSC). Each line in the figure represents an integer, each box of a line represents a bit, and boxes of the same color represent a segment that hold a variable from one of the temperature replicas. We give the name AMSC1 and AMSC3 for these two AMSC schemes according to their segment width. Unlike the AMSC1, the AMSC3 scheme reserves three bits for each segment, and is a less dense storage format. For the calculation of the local energy, we need two spins (S) and the coupling (J) between them. The J bits and S bits are integrated in the same integer, so that we can fetch both the coupling and the spins using only one memory transaction. The local energy (e) of each bond can be calculated by performing two XOR operations. The total local change of energy (EEE) is the sum of the contributions from all six nearest neighbors. Since EEE requires three bits for storage, the AMSC1 scheme compute each segment sequentially to avoid overflow, while the AMSC3 scheme can compute multiple segments in parallel. After we obtain EEE in three-bit format, we combine it with the spin state (S) by doing string concatenation.

To overcome the intrinsic limitation of AMSC, we propose a new scheme named Compact Asynchronous Multispin Coding (CAMSC). We dynamically change the segment width to match the data range. Longer width is adopted for larger data to qualify the vectorization of computing multiple segments. For small range data, we use shorter width to avoid blank bits reservations. For example, we allocate 1 bit per segment for S and e, and then expand to 4 bits when calculating E. The segment width expansion is implemented with shift and mask operations. Figure 3.15(b) demonstrates the procedures of CAMSC and how it differs from traditional AMSCs. Our experiment (Figure 3.16) shows 28.4% performance improvement when we switch from AMSC3 (46.8 ps/spin) to CAMSC (33.5 ps/spin).



(a) AMSC4 (b) CAMSC

FIGURE 3.15. This figure demonstrates how the AMSC4 and CAMSC schemes help exploit bit-level parallelism in computing $(E \times 2 + S)$. Similar to that of the AMSC1 and AMSC3 (see the text and the caption of Figure 3.14), the XOR operations and summation over six nearest neighbors produces the total local energy (EEE). However, since we reserve four bits for each segment, and is capable of holding one more bit over EEE, the string concatenation of EEE with S can now be vectorized. The difference between CAMSC and AMSC4 is that S and J are stored in a more compact format. With such a design, CAMSC avoids waste of space and provides much better parallelism in computing e.

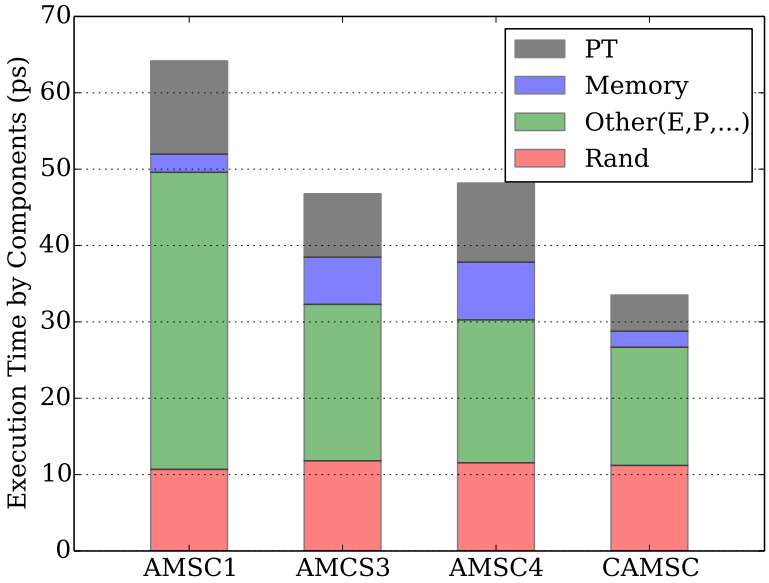


FIGURE 3.16. Comparing the performance using different multispin coding schemes. The experiment is done for a 16^3 lattice, a shared memory probability table with integers and CURAND. A parallel-tempering move is performed every 10 Metropolis single spin sweeps.

3.5 Experimental Results

3.5.1 The Platform Settings

Our development and performance evaluations are carried out on a workstation with an Intel Core i7 x990 CPU and an NVIDIA GeForce GTX 580 GPU card. The GeForce GTX 580 is equipped with a Fermi architecture GPU of 512 stream processors. We use Linux 2.6.32 x86-64, CUDA toolkit version 4.1 and gcc 4.4.6, and optimization flag -O2. We always configure the GPU on-chip memory as 48KB shared memory plus 16KB L1 cache.

3.5.2 Performance Evaluation

To evaluate the performance we use the time spent (in picoseconds) per spin flip proposal, abbreviated as ps/spin (See Equation 3.2).

When we study the equilibrium properties of a spin glass, the system sizes that can be equilibrated within a reasonable time are not very large. Therefore, we used $L = 16$, or $N_{\text{spins}} = 4096$ as the maximum system size. Meanwhile, to achieve efficient parallel tempering moves, we set the number of temperature replicas to $N_T = 24$ or 56, and perform frequent parallel tempering moves (one parallel tempering move after every 5 to 10 Monte Carlo sweeps). The typical number of Monte Carlo steps required to equilibrate such a system is approximately 10^7. Due to the huge sample-to-sample variation, a large number of disorder realizations (10^4 or more) are usually required. However, since there is no correlation among different realizations, we can scatter the jobs to different GPU cards or nodes on a cluster. On each of the cards we only need 16 to 64 realizations to fully utilize all the multiprocessors.

For benchmarking, we simulate 64 disorder realizations of the Edwards-Anderson model on a 16^3 lattice with 24 temperature replicas, and propose to swap adjacent temperatures every 10 Monte Carlo sweeps. We are able to complete 10^7 Monte

Carlo sweeps in 40 minutes. This wall time consists of the single spin flip Monte Carlo time, the parallel tempering swap time, and the measurement time. Discarding the measurements, the average computational speed is 33.5 ps/spin, for a single GPU device. If we simulate without parallel tempering and serve all temperature replicas with the same random number, we could obtain 17.6 ps/spin. Generating random numbers consumes about one third of the total simulations time, as shown in Figure 3.16. We believe we are approaching the limit of performance optimization. For reference, our single thread CPU code (using AMSC4 without parallel tempering on a 16^3 cubic lattice) runs at the speed of 14737 ps/spin; this represents a speed up of almost 440 for the GPU code over the CPU code.

Figure 3.17 compares our implementation with similar existing codes, where not all reference programs target at the random frustrated Ising systems, present the external magnetic field, and feature parallel tempering. Our program is substantially faster than any other GPU implementation [87, 88, 90, 91] for small to intermediate system sizes. We are comparable to the performance achieved by special-purpose FPGA implementations[92].

3.5.3 Simulation Results

We test the code by simulating both the simple ferromagnetic Ising and the Edwards-Anderson spin glass models. In Figure 3.18, our results from the GPU code are found to be consistent with the results from our CPU code for the ferromagnetic Ising model, at various external magnetic fields. We also compare the results with and without parallel tempering as a check to determine whether the parallel tempering swap is performed correctly. We find that the results with and without the parallel tempering swap are consistent with each other. In Figure 3.19 we plot the correlation length for the ferromagnetic Ising model in three dimensions; here, the crossing point for the correlation length coincides with the

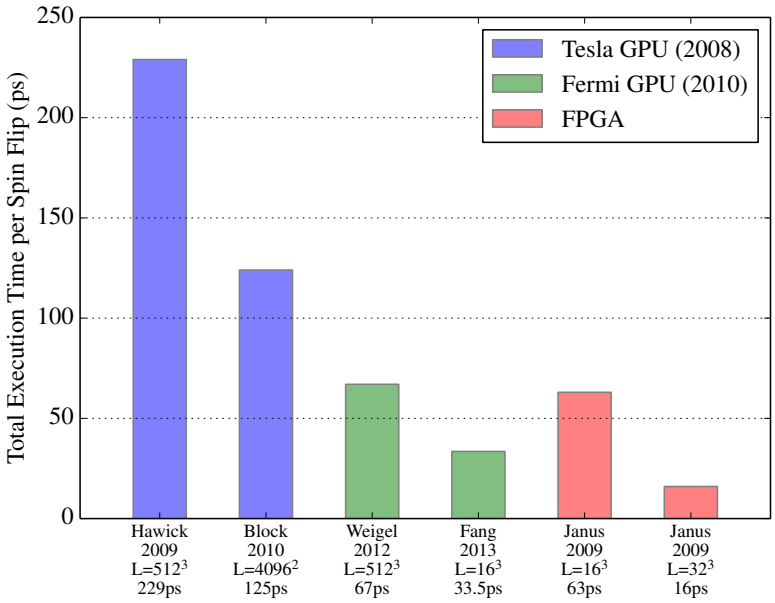


FIGURE 3.17. Performance comparison with other heterogeneous Ising model simulation programs. Hawick et al. [87] reports 4360.1 million Monte Carlo hits per second, which equals to 229 ps/spin. Block et al. [88] reports 7977.4 spin flips per microsecond, which equals to 125 ps/spin.

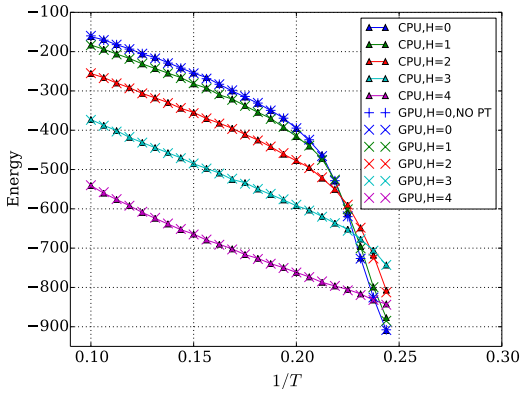


FIGURE 3.18. Comparing the total energy of the 16^3 sites Ising model with nearest neighbors coupling $J = -1$, to CPU generated results. At each value of the external field, the GPU results are nearly identical to the CPU results.

known critical temperature for the ferromagnetic ordering. [114] For the Edwards-Anderson model we calculate the Binder ratio of the system at zero external magnetic field as shown in Figure 3.20. The results match reasonably well with the published data. [115] Figure 3.21 demonstrates the effectiveness of parallel tem-

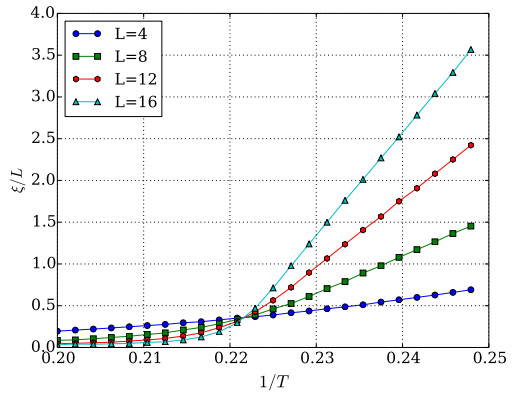


FIGURE 3.19. Correlation length vs. inverse temperature for the Ising model. The lines from different system sizes cross close to $1/T = 0.2217$, which is in agreement with the published result for the critical temperature. [114]

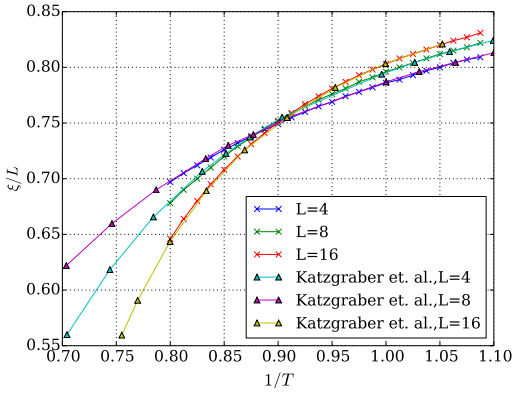


FIGURE 3.20. Binder Ratio for the 3D Edwards-Anderson model. The data generated by our GPU code is compared with the data extracted from the paper by Katzgraber *et al.* [115]

pering for the Edwards-Anderson spin glass. The parallel-tempering simulation reaches equilibrium after 10^5 Monte Carlo sweeps, while without parallel tempering, the system did not reach equilibrium even with 100 times more iterations. This further supports that we have implemented the parallel tempering swapping correctly.

3.6 Conclusion and Future Works

We design and implement a CUDA code for simulating the random frustrated three-dimensional Edwards-Anderson Ising model on GPUs. For small to interme-

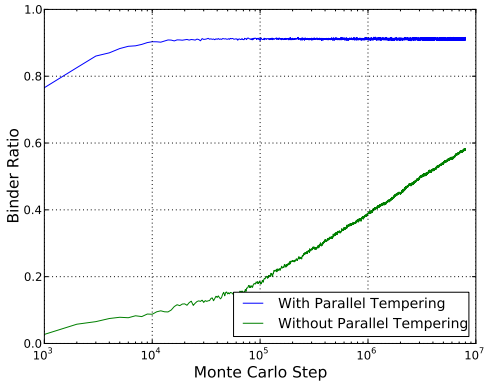


FIGURE 3.21. The convergence of the Binder ratio vs. number of Monte Carlo steps for the Edwards-Anderson model in a system with 8^3 sites, with and without parallel tempering for $1/T = 2.0$. Parallel tempering dramatically improves the convergence to equilibrium.

diate system sizes, our code runs faster than other GPU implementations, and its speed is close to that of the specially built FPGA computer. We note a very recent preprint has reported an improvement in FPGA system. [26] Our performance tuning strategies include constructing three levels (tasks, threads, bits) of parallel workloads for GPU; optimizing the memory access via a proper data layout and tiling; speeding up the computation by translating time consuming floating point operations to integer point operations and table look-ups; and finally, vectorizing bit computations with our binary format, the Compact Asynchronous Multispin coding.

Our program can be extended for other models such as the Potts models and models with different random coupling distributions. The structure of our code may adapt well to upcoming GPUs and future massive parallel accelerators.

Acknowledgments

This work is sponsored by the NSF EPSCoR LA-SiGMA project under award number EPS-1003897. Portions of this research were conducted with high performance computational resources provided by Louisiana State University. Part of

this work was done on the Oakley system at the Ohio Supercomputer Center. We thank Helmut Katzgraber and Karen Tomko for useful discussions. We thank the following collaborators: Bhupender Thakur, Ariane Papke, Sean Hall and Cade Thomasson. We thank Samuel Kellar for his careful reading of the manuscript.

Chapter 4
Results for Three-Dimensional Edwards Anderson Model in an Externel Field

This following chapter is a work titled **Three Dimensional Edwards-Anderson Spin Glass Model in an External Field**, submitted for review to appear in Physics Review E. In this paper, we discussed the results for Monte Carlo Simulation of 3D Edwards-Anderson model.

This paper is written in collaboration with Ye Fang, Ka-Ming Tam, Zhifeng Yun, Juana Moreno, J. Ramanujam and Mark Jarrell. The idea of the project was first proposed by Ka-Ming Tam. Ye Fang and I developed the implementation together. With the code, I ran simulations on the GPU clusters, accumulated and analyzed all the data. This paper is written in collaboration with Ye Fang, Ka-Ming Tam, Juana Moreno, and Mark Jarrell. I started a first draft, including the introduction, the simulation methods, the measured quantities and the conclusions. I also plotted all the figures from the data obtained in the simulation. Ka-Ming contributed to the introduction and conclusions. Juana Moreno and Mark Jarrell also reviewed the paper and improved the conclusions.

4.1 Introduction

Most spin systems order when the temperature is sufficiently low. Conventional magnetic orderings break the spin symmetry, and the moments align in a pattern with long range order. However, magnetic systems with random frustrated couplings can avoid conventional ordering by breaking ergodicity. Typical spin glass systems with such competing magnetic couplings include localized spins in metals coupled via the oscillating Rudermann-Kittel-Kasuya-Yosida exchange as CuFe and CuMn, and in insulators with competing interactions as in LiHoYF and EuSrS

[2, 116, 117]. These systems do not display long range order for a wide range of diluted spin concentrations.

A widely studied model to describe spin glass physics is the Edwards-Anderson (EA) model[10]. It is composed of spins interacting with their nearest neighbors via random couplings. The mean-field variant of the Edwards-Anderson model, the Sherrington-Kirkpatrick (SK) model[11, 12], was solved by the replica technique in 1975 with the striking observation that the entropy can be negative at low temperature[11, 12]. A cavity mean field method was proposed by Thouless, Anderson and Palmer (TAP) in which the local magnetization of each site is considered as an independent order parameter[118]. The hope was to obtain a more physical mean field solution without involving the replica technique. However, multiple solutions were found[119].

Motivated by the deficits of previous approaches, de Almeida and Thouless further studied the replica symmetric mean field solution and found a line in the temperature–magnetic field plane where the replica symmetry solution is unstable towards replica symmetry breaking (RSB) [93, 94] The replica overlap has more structure than simply a constant. The way to characterize this structure for a stable mean field solution was developed by Parisi [13–15]. There is a hierarchy of the replica overlap, and this can be described in terms of an ultra-metric tree. The replica symmetry breaking scheme resolved the negative entropy crisis and naturally explained the many solutions found in the TAP approach.

The replica symmetry breaking theory is accepted to be the correct description of the Sherrington-Kirkpatrick model; indeed it provides the exact free energy [98, 99]. However, its applicability to low dimensional spin glass systems has been intensively debated over the last three decades, especially in the three dimensions case. For systems below the upper critical dimension [16–18] the most prominent

competing theory is the droplet model elaborated by Huse and Fisher [19, 20] and based on the idea of domain wall scaling by Moore, Bray and McMillan [120, 121]. In this theory, there exists a finite characteristic length scale where droplets of excitations can lose energy by aligning with the field. The spin glass phase is thus destroyed by any finite external field. Moreover, those excitations are assumed to be compact and with fractal dimension smaller than the spacial dimension, in contrast with the space-filling excitations in the mean field theory.

Thus a possible scheme to discern between the replica symmetry breaking and the droplet theories is to determine whether a spin glass phase exists at a finite external field [21]. There are other schemes based on the differences in the overlap and the excitations in these two theories. For example, the distribution of the overlap and the parameters that characterize it [122–127], the existence of the ultrametric structure in the overlap [128, 129], and the nature of the ground state and its excitations [124, 130–135]. Unfortunately, the conclusions drawn from different studies are often controversial. This is mostly due to two factors, the limitation in the system sizes that can be simulated and the interpretation of the data.

Using the same techniques on the three dimensional Edwards-Anderson model under an external field, no signal of a crossing of the scaled correlation length for different system sizes can be detected[21]. We will show this is also the case for the Binder ratio. The absence of crossing is powerful evidence that a spin glass phase is absent in the presence of an external field. However, it has been argued that the system sizes studied may be too small and far from the scaling regime. To remedy this problem, one dimensional models with long range power-law decaying interactions [136] which mimic the short range models at higher dimensions have been intensively studied over the last few years [137–139]. In these models much larger systems can be studied [83, 85, 140, 141]. It is worthwhile to mention that

the studies using Migdal-Kadanoff approximations for hierarchical lattice tend to support the droplet picture [142, 143].

On top of these controversies, it has been recently argued that the scaled correlation length is not a good parameter for the spin glass transition in a field since its calculation involves the susceptibility at zero momentum [141]. The latest proposal is to study the ratio of susceptibilities at the two smallest non-zero momenta, denoted it as R_{12} [25]. It has been shown that in four dimensions this quantity displays a crossing at finite temperature which is an important clue that the spin glass can still exist without time reversal symmetry below the upper critical dimension [25]. Giving the success of using R_{12} to capture the spin glass phase at four dimensions, we reexamine the three dimensional Edwards-Anderson model on a simple cubic lattice using a new development in computer architecture, and the recently proposed R_{12}. We will demonstrate that graphic card computing is particularly well suited for equilibrium simulations of spin glass systems, in particular for cases where a huge number of realizations is required such as the model we study in this work.

The paper is organized as follows: The simulation methods and the quantities we measured are introduced in the Section II. In the section III, we show the data from our simulations. We conclude our results and discuss the possible directions for the future study in the section IV.

4.2 Method and Measured Quantities

The Hamiltonian for the Edwards-Anderson model is given as

$$H = -\sum_{<i,j>} J_{ij} S_i S_j - h \sum_i S_i, \tag{4.1}$$

where S_i indicates Ising spins on a simple cubic lattice with $N = L^3$ sites and periodic boundary conditions. The coupling J_{ij} is bimodal distributed with probability $P(J_{ij}) = \frac{1}{2}(\delta(J_{ij} - 1) + \delta(J_{ij} + 1))$, and h is an external field.

The spin glass overlap is defined as

$$q(\mathbf{k}) = \frac{1}{N} \sum_j S_j^{(\alpha)} S_j^{(\beta)} \exp^{i\mathbf{k}\cdot\mathbf{r}_j}, \tag{4.2}$$

where α and β are two independent realizations of the same disorder model. We calculate the overlap kurtosis or the Binder ratio from the overlap as [123, 144]

$$g = \frac{1}{2}\left(3 - \frac{\overline{\left\langle \left(q(0) - \overline{\langle q(0)\rangle}\right)^4 \right\rangle}}{\overline{\left\langle \left(q(0) - \overline{\langle q(0)\rangle}\right)^2 \right\rangle}^2}\right). \tag{4.3}$$

Note that $\overline{(\cdots)}$ indicates averaging over different disorder realizations, and $\langle \cdots \rangle$ denotes thermal averaging.

The wave vector dependent spin glass susceptibility is defined as [123]

$$\chi(\mathbf{k}) = N(\overline{\langle q^2(\mathbf{k})\rangle} - \overline{\langle q(\mathbf{k})\rangle}^2), \tag{4.4}$$

and the correlation length as

$$\xi_L = \frac{1}{2\sin(\mathbf{k}_{\min}/2)} \left[\frac{\chi(0)}{\chi(\mathbf{k}_{\min})} - 1\right]^{1/2}, \tag{4.5}$$

where $\mathbf{k}_{\min} = (2\pi/L, 0, 0)$.

We define R_{12} as the ratio between the susceptibilities with the two smallest non-zero wave vectors [25]

$$R_{12} = \frac{\chi(\mathbf{k}_1)}{\chi(\mathbf{k}_2)}, \tag{4.6}$$

where $\mathbf{k}_1 = (2\pi/L, 0, 0)$, $\mathbf{k}_2 = (2\pi/L, 2\pi/L, 0)$.

Parallel tempering[65, 66] is used to accelerate the thermalization, in which N_T samples of the same disorder coupling are simulated in parallel within a range

of temperatures. In order to compute the spin glass overlap (Equation 4.2) we simulate two replicas of the system with the same bonds $J_{ij} = \pm 1$ and field h at each temperature.

We implement the Monte Carlo simulation with parallel tempering on graphics processing units using the CUDA programming language [42]. Multispin coding[100, 110] is used to pack the N_T replicas into the small but extremely fast shared memory. We achieve a performance of 33ps per spin flip attempt on a GTX 580 card. We use the CURAND implemented XORWOW generator to generate random numbers [111]. Since the GPU is a commodity hardware and widely available in large computer clusters, it is now easy to greatly accelerate these simulations. The details of the implementation can be found in Ref [145].

We list the parameters of our simulation in Table 4.1. We benchmarked the code

TABLE 4.1. Parameters of the simulations. L is the linear system size. N_{samp} is the number of samples, N_{sweep} is the total number of Monte Carlo sweeps for each of the $2N_T$ replicas for a single sample, $\beta_{\max}$ and $\beta_{\min}$ show the temperature region simulated, and N_T is the number of temperatures used in the parallel tempering method. The temperature set in each simulation follows a geometric distribution, i.e. $\beta_n = \beta_{\min}\alpha^{n-1}$, where $\alpha = (\beta_{\max}/\beta_{\min})^{1/(N_T-1)}$, $n \in [1, N_T]$. The first half of the Monte Carlo sweeps are used for thermalization and the second half are used for measurement.

L	N_{samp}	N_{sweep}	N_T	$\beta_{\max}$	$\beta_{\min}$
6	500,000	2,000,000	56	1.8	0.1
8	350,000	2,000,000	56	1.8	0.1
10	240,000	2,000,000	56	1.8	0.1

against existing results at $h = 0$. The smallest β used in the parallel tempering is well below the critical temperature ($1/\beta_c = T_c \approx 1.1019 \pm 0.0029$) [146] of the spin glass transition at zero field[24, 146], while the largest β is about two times larger. The estimated critical field at zero temperature is around $h \approx 0.65$ for the model with zero mean and unit variance Gaussian distributed couplings[147]. We choose to work in a relatively small field, $h = 0.1$. The jackknife method is used to estimate the statistical errors from disorder averaging.

4.3 Results

We plot the spin glass susceptibility in Figure 4.1. As in the zero field case, the susceptibility increases as the temperature is lowered, however there is no obvious asymptotic scaling behavior. In particular, for temperatures below the zero-field critical temperature, the slope of the curves decreases and they begin to bend downward. This result is similar to the one obtained for the one dimensional model[83], but in contrast with the results of the four dimensional lattice which displays asymptotic divergent susceptibilities [123].

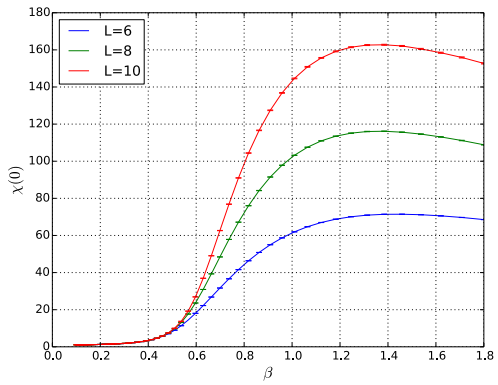


FIGURE 4.1. Spin glass susceptibility at zero momentum, $\chi(0)$, as a function of inverse temperature for system sizes $L = 6, 8, 10$.

As the susceptibility does not show a behavior in accordance with the conventional finite size scaling theory for a second order transition, we move to study various cumulants and ratios of susceptibilities of the overlap parameter. We show the Binder ratio in Figure 4.2. It does not display any signal of crossing. Indeed, the curves for different system sizes do not even tend to merge as the temperature is lowered. Note that the Binder ratio corresponds to the fourth-order cumulant of the distribution, and the possible issues related with the soft mode contributing to the zero momentum susceptibility should likely be canceled in the Binder ratio.

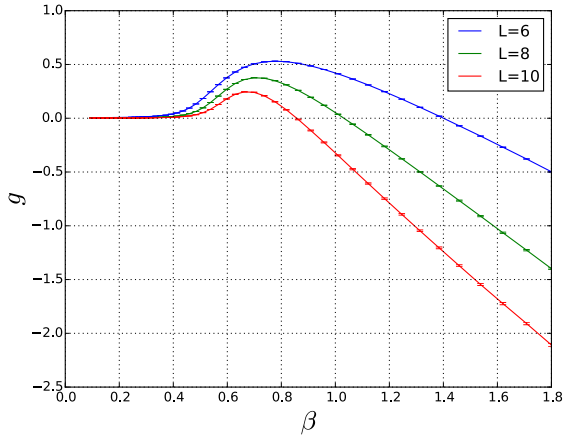


FIGURE 4.2. Binder ratio as a function of inverse temperature in the range $\beta = 0.1 \sim 1.8$ for system sizes $L = 6, 8, 10$.

Figure 4.3 displays the scaled correlation length. This is now a standard diagnosis for the detection of a spin glass transition. The correlation length is extracted from the Ornstein-Zernike form (Equation 4.5), and thus essentially given by the ratio between the zero and the smallest finite momentum susceptibilities. Similar to the Binder ratio, and consistent with other results in the literature, there is no crossing or even merging down to a rather low temperature [21].

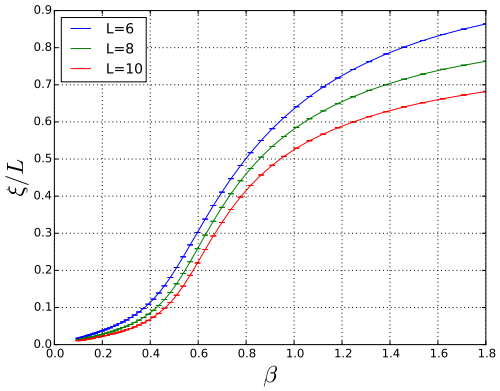


FIGURE 4.3. Scaled correlation length ξ/L as a function of inverse temperature for system sizes $L = 6, 8, 10$.

From now on we focus on R_{12}. We performed simulations in zero field where R_{12} shows a crossing close to the expected critical temperature found from the Binder ratio and the correlation. Therefore, the crossing in R_{12} should be a viable

indicator for the phase transition. Unfortunately, we find that R_{12} is in general much noisier than other quantities. This is due to the fact that the sampling of higher momentum quantities is almost always characterized by larger statistical fluctuations. Taking the ratio between two susceptibilities at finite momenta clearly further harms the quality of the data. To reduce the error bars we generate long runs and larger pools of disorder realizations (see Table 4.1). This is the main reason we have generated a rather large number (2.4×10^5) of realizations for the largest systems size we present here, and even more for smaller sizes. To further reduce the fluctuations, we impose all point group symmetries. For example, when we calculate $\chi(2\pi/L, 0, 0)$ we average the susceptibility at three different directions ($\chi(2\pi/L, 0, 0)$, $\chi(0, 2\pi/L, 0)$, and $\chi(0, 0, 2\pi/L)$). This averaging implicitly assumes that the point group symmetry is restored which is justified only when the number of realizations is rather large.

Figure 4.4 displays R_{12}. In contrast to other quantities, R_{12} shows an intersection

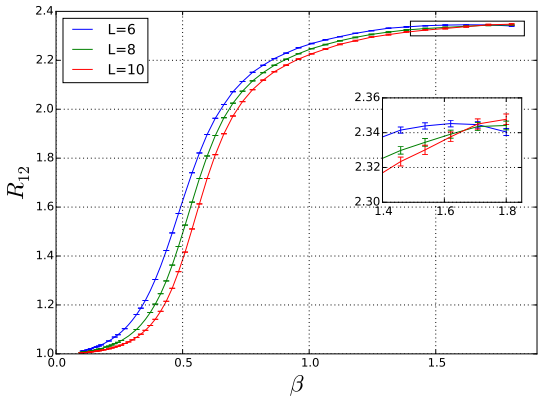


FIGURE 4.4. R_{12} as a function of inverse temperature for different system sizes. An intersection can be seen at around $T \approx 0.6$. We use the jackknife method to estimate the error bar from sample-to-sample variation.

at about $T \approx 0.6$. We do not think we have sufficient data to perform a reasonably accurate finite size scaling analysis to report the exponent or even to quantify the correction [148]. Moreover, the data for $L = 6$ does not seem to fit into a

finite size scaling form with the curve bending downward. Unfortunately, parallel tempering Monte Carlo is not robust enough for simulating larger lattices in a reasonable amount of time; this can be related to the temperature chaos [149–151]. The number of replicas needed to equilibrate the system also increases substantially as the system size increases, we already used 56 temperature replicas for $L = 10$ simulations. We plot R_{12} versus the number of Monte Carlo sweeps in Figure 4.5. We believe the data is sufficiently equilibrated for averaged quantities. The major contribution to the error is from the limited number of disorder realizations. Figure 4.6 shows R_{12} for $L = 10$ for different numbers of realizations. We clearly see that the data converges only when the number of realizations is fairly large. This is one of the prominent hurdles of using higher momentum susceptibility as a diagnosis. We note that the effective one dimensional model also shows crossing behavior, albeit the crossing points do not show a systematic trend[83].

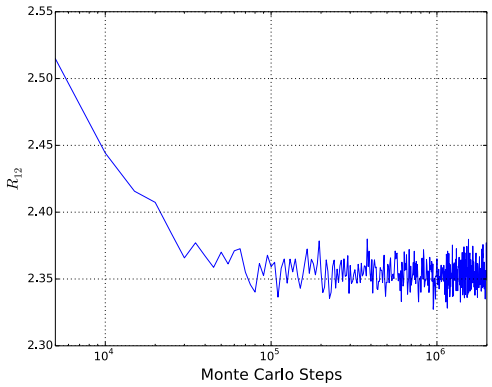


FIGURE 4.5. R_{12} for $L = 10$ at $\beta = 1.8$, as a function of the number of Monte Carlo sweeps. We believe the averaged data is equilibrated for 10^6 sweeps, and it passes the logarithm binning test [152]. The main contribution to the error is from the realization averaging.

Given the difficulty in using the R_{12}, we investigate the source of the noise by studying the distribution of the susceptibility. We calculate the susceptibility for each disorder realization, and plotted the histogram at the lowest temperature, $\beta =$

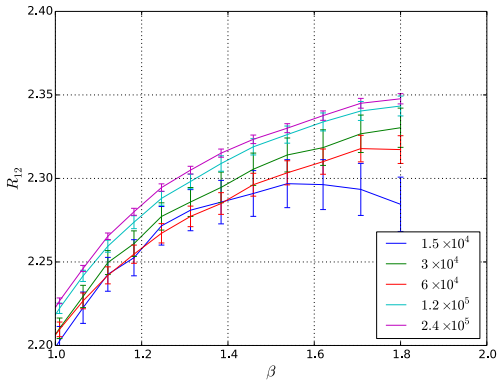


FIGURE 4.6. R_{12} for $L = 10$ and low temperatures ($\beta \geq 1.0$). We show five different numbers of realizations from fifteen thousand to two hundred forty thousand.

1.8. The distribution is highly skewed. The mean of the distribution is dominated by rare events, as suggested in Figure 4.7. The non-Gaussian nature of the distribution suggests that the mean value might not be a good indicator.

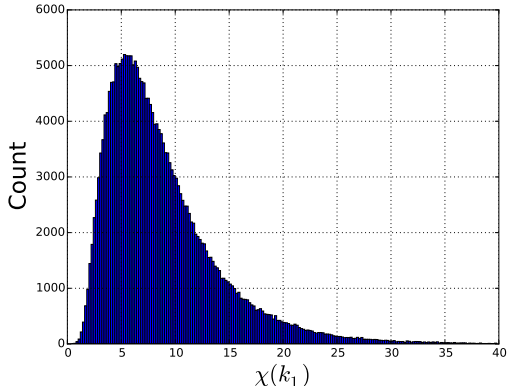


FIGURE 4.7. Histogram for χ_1 at low temperature.

As a first attempt to handle the distribution which is dominated by rare samples, we used the geometrical average [153] over the susceptibilities to find R_{12}. In the Figure 4.8, we show the plot of R_{12} calculated from geometric averaging. We see that the lines are quite different from those obtained with arithmetic mean, and the crossings appear at different temperatures. Rather this approach can provide a better signal for phase transition is unclear at present, but is is worthwhile for further investigation.

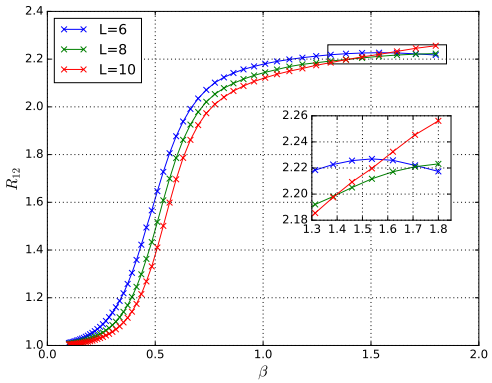


FIGURE 4.8. R_{12} calculated from the geometrical average of susceptibilities, as a function of inverse temperature, for different system sizes.

4.4 Discussions and Conclusions

In summary, we perform Monte Carlo simulations of the three-dimensional Edwards-Anderson model in a finite external field. The goal is to reexamine the long-standing problem of whether mean field behavior, specifically a spin glass phase, can exist in such a model without time-reversal symmetry. We focus on the equilibrium quantities of this notoriously difficult system. By taking advantage of the new commodity multi-threaded graphic computing units architecture we drastically reduce the computation time. The results for the Binder ratio and correlation length for different cluster sizes show no signal of an intersection, thus, they point to the absence of a spin glass transition as found in previous studies. On the other hand, the ratio of susceptibilities R_{12} does show intersections for relatively small system sizes ($L = 6, 8, 10$). We did perform simulations for larger system sizes, but the data for R_{12} is too noisy to draw a conclusion. With the present system sizes and the statistical error bar, a rigorous data analysis does not seem to deliver unbiased information. This situation is rather discouraging, as simulations at this low temperature for much larger system sizes using the present method are already rather challenging. Although we cannot reach a definitive conclusion on whether

a spin glass phase transition exists under a finite magnetic field, we attempt to understand the source of the noise in the R_{12} We showed that the susceptibilities are far from normally distributed and their mean is dominated by rare events. This motivated us to study the geometric average of the distribution. We showed that for different system sizes, there is similar crossing behavior in the R_{12} as those calculated from the linear average.

The results are obtained with a rather small external field, $h = 0.1$. It is possible that a larger field, such as $h = 0.25$, would give us a stronger signal of crossing predominately due to the distance away from the known critical point at $h = 0$. While this may give us a slightly clearer signal on the phase transition, should the transition exist at $h = 0.25$, it would likely occur at a significantly lower temperature which further jeopardizes the quality of the simulation data. As the result stands now, we cannot find a clear phase transition in $h = 0.1$, and if there were no phase transition, it would also rule out the possibility of the transition occurring in a larger field. On the other hand, if we found evidence for the absence of a transition at $h = 0.25$, we cannot determine if the spin glass phase would survive a smaller field like $h = 0.1$. There is no simple rule of thumb which can be used to determine what value of h is the best for the purpose of answering the question on the existence of the transition under an external magnetic field. But we believe that the possible advantage of using a larger field does not outweigh the disadvantages, both in term of the difficulty of the simulation, and its predictive power.

Thanks to our efficient GPU implementation, we are able to leverage the computing power of supercomputing clusters with GPU accelerators, and study a large number of disorder realizations. Our results show that with current numerical methods and computing capability results obtainable in a reasonable amount of

time and resources are still not robust enough to provide dispositive insight into the nature of a spin glass at three dimensions. This makes a call to the method of interpreting the data. We propose that a possible direction for the future study should go beyond simply calculating the average value of critical quantities, such as susceptibilities. The disordered nature in both the spatial and temporal directions should benefit from recent advances in big data analysis. Various clustering and pattern recognition methods develop should provide new opportunities for the analysis of data from spin glass simulations.

We notice a preprint before we finished the present paper where the conditioning variate method is used to expose the silent features from the data [154].

4.5 System with L=16

We further tested with a bigger system size, $L = 16$. We used 11,200 realizations, and divide them into four batches, each containing 2800 samples. We then calculated the R_{12} for the four batches, and compared them, as shown in Figure 4.9. The results shows that a few thousand realizations is not enough, even for a system size of L=16.

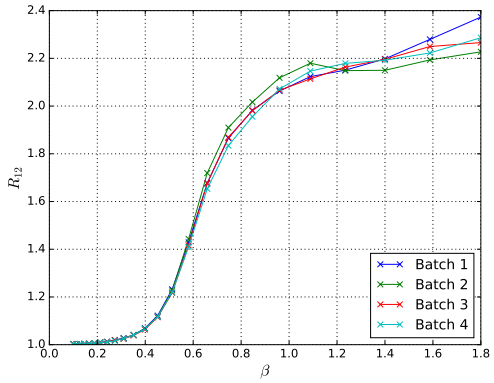


FIGURE 4.9. R_{12} for a larger system with $L = 16$. We used four batches of samples, each batch contains 2800 disorder realizations. The results shows that for different batches R_{12} is quite different, especially at low temperatures.

This work is sponsored by the NSF EPSCoR Cooperative Agreement No. EPS-1003897 with additional support from the Louisiana Board of Regents. This research were conducted with high performance computational resources provided by Louisiana State University (http://www.hpc.lsu.edu). We thank Helmut Katzgraber and Karen Tomko for useful conversations. We thank the following collaborators: Bhupender Thakur, Ariane Papke, Sean Hall, and Cade Thomasson.

Chapter 5

Continuous Time Quantum Monte Carlo Solver for Strongly Correlated Materials

5.1 Introduction

Many interesting phenomena in materials can be simply described in term of a picture in which particles are independent of each others. For a periodic system, Bloch's theorem provides the basis for the description of materials in term of band structure. The energy eigenstates can be expanded in term of the periodic function with definite momentum number. Even such a seemingly simple independent particle picture harbors a vast amount of interesting phenomena. Nine decades have passed after the Bloch's theorem, the physics of non-interacting electrons are still being investigated intensively today. The exotic physics, such as the topological structure of the band theory still have not been exhausted, typical examples include quantum hall effect and topological insulators [155].

Effects from the interaction among particles cannot be completely ignored in many interesting systems. The physics can quickly become very complicated when the interaction becomes a dominating factor in the system. Unlike the simple single particle picture, there is no generic efficient method for the study of quantum interacting systems. From a computational point of view, the size of the quantum system, Hilbert space, grows exponentially as the number of particles. It is in general very difficult to analyze such systems by simple numerical mean.

However if one can reduce the systems into a somewhat single particle like systems, the analysis can be greatly simplified. The Landau Fermi liquid theorem, a landmark achievement in the study of correlated systems, tells us that in many circumstances the interacting system can be reduced into a single particle system

with some modifications. [156, 157] The non-interacting particle can be adiabatically deformed into quasi-particle with a finite lifetime. This important prediction by Landau can only be explained few decades later when the concepts of renormalization group of fermion systems are applied in the condensed matter physics.

Although the Fermi liquid theorem can satisfactory explained a lot of metallic systems, its limitation is obvious. It fails in the low-dimensional cases, by now, we know that this is invalid in one dimension. [158, 159] However, the question of its validity in two dimensions is more subtle. As many of the interesting systems, their physics are believed to be largely dictated by the correlation in two dimensions. This is also the main battlefield of strongly correlated systems. In one dimension, various rather accurate numerical methods are available; in higher dimensions, mean field theory is presumably a good starting point. At two dimension, there is usually no good control on analytical calculations, and numerical methods for lattice models are hindered by the minus sign problem in Quantum Monte Carlo or the lack of good renormalization of the Hilbert space. Another clear deficit of the Fermi liquid theory is its inability to describe the quantum criticality, It is obvious that the criticality involves collective behaviors. [160] An independent particle picture is doomed to fail in describing critical points. Further increase the interaction, the independent particle picture in the momentum space has to be replaced by the Mott picture in the real space, and there is no adiabatic continuation which can tune the single particle metallic system into a real space picture of Mott insulator. [161]

Thus, a systematic, even though approximated, method is sought for the study of strong correlation. The method which can describe the physics from metallic, to critical, to insulating phase as the interaction is cranked up is crucial for the study of strongly correlated systems. We will discuss in the following that the

dynamical mean field method and its cluster extension, dynamical cluster approximation, fulfilled such a request. [32, 162] Combining with the density functional theory, semi-quantitative numerical results can sometimes be obtained for strongly correlated materials.

In a sense, from the above discussion, the term strong correlation refers to the behavior of electrons that cannot be well-described by simple one-electron theories. Materials which naturally have a tendency to strong correlation are those involved unfilled d or f orbitals. This is due to the small orbital radius of d and f orbitals and so does the overlap of orbitals. This reduces the kinetic energy of the system, the electrons are said to be living in the narrow band, and thus the interaction becomes more important. Many interesting materials discovery which host a range of interesting experimental observations, the most prominent example is the high-temperature conductivity, pseudogap behaviors, quantum criticality, in the past few decades involve transition metal or inner transition metal. This is precisely because of the effective strong correlation due to the d or f orbitals.

Understanding those exotic behaviors is extremely challenging. There is a major obstacle. Once the single particle picture fails, there is no good starting point for many perturbative calculations. It renders into a regime almost all analytical methods will fail. In the following we will discuss two majors progress in attacking correlated systems. The first one is density functional theory (DFT)[163, 164] , a many-site single particle treatment; and the other one is dynamical mean field theory (DMFT) [30, 32, 165], a single site many-body treatment.

5.2 Numerical Approaches in Strongly Correlated Materials

Numerical calculation in strongly correlated fermion systems is a major challenge in condensed matter physics. In real world, materials consist of 10^{23} interacting particles, which is impossible to solve at first glance. Fortunately, not all the par-

ticles contribute to the property of materials. For example, in a metal only the electrons close to the Fermi level can be excited and contribute ,e.g., to the transport and magnetic properties. In a lattice, lattice excitations are few at low T, but they are responsible for inelastic neutron scattering. Even though, the remaining problem is still hard.

Density functional theory (DFT)[163, 164] provides a framework to solve the electron structure problem. Using this theory, the properties of a many-electron system can be described by a functional of election density. Combined with approximations that address the exchange-correlations such as the local density approximation (LDA) [166], density functional theory produces satisfactory data that agrees well the experiments for many cases. However, despite the success in weakly correlated materials, there are still difficulties in applying this method to other cases, such as systems with strongly correlation [167]. The accuracy of density functional theory has seen gradually improve over the year, the continual development in functional to include correction from beyond local density plays an important part. However, it seems to be quite difficult to handle such strong correlation by simple improvements of the density functional theory. A promising direction is to combine other methods which can treat strong correlation with the density functional theory method. A popular choice is to employ the dynamical mean field theory[30, 32] (DMFT) to include the effect from the electron-electron correlation on top of the single particle dispersion obtained from the density functional theory.

Dynamical mean field theory has been widely used on a range of strongly correlated systems. It is a method well suited for strongly correlated systems, in particular, it captures the Mott transition, a hallmark of strong correlation, of the Hubbard model. In this approach, the solution of the lattice model is mapped to a quantum impurity model with self-consistency conditions. A quantum impurity

problem describes an atom embedded in a host medium. The impurity consists of a set of orbitals with different parameters, populated with electrons that interact with each other. The orbitals are hybridized to bath orbitals representing the degrees of freedom of the host materials. The solution of impurity problem can be obtained in a few different ways. We will focus on the different variants of Monte Carlo methods.

A commonly used technique is the Hirsch-Fye method [168], in which a Hubbard-Stratonovich transformation is used to decouple the interaction part, leading to determinants which give the weights associated with the configurations of the auxiliary fields, which are then sampled by a Monte Carlo procedure. One issue is that Hirsch-Fye cannot be easily applied to complicated interaction that includes more than just density-density interaction, due to the lack of simple ansatz to decouple the interacting terms. The matrix size scales to the interaction as well as the inverse of temperature, which makes the calculation inefficient at low temperatures. This method also requires discretization of the imaginary time interval, which introduce systematic errors, and may not be optimal for the multi-orbital case with complicated off-diagonal couplings.

The Trotter error in Hirsch-Fye algorithm can be eliminated by using the Continuous Time Monte Carlo algorithms. For example, one can solve the problem exactly in non-interacting limit, and treat the interaction with a Taylor-series expansion. By doing stochastic sampling of diagrams in the weak-coupling expansion of partition function, the interaction expansion (CT-INT) algorithm [169] provides a discretization error free alternative Hirsch-Fye algorithm. Still, in the CT-INT algorithm, it is difficult to treat non-Hubbard-type interactions. Also, the size of the matrix used in the CT-INT method grows quickly with the interaction, making the calculation very time-consuming at very strong interactions.

Another way to treat the impurity problem is the hybridization expansion (CT-HYB) approach [170–173]. The fact that the order of expansion decreases with increasing interaction makes this method favorable for strong interaction systems. The algorithm is also found to work at very low temperatures and is applicable to a wider class of impurity models including those with complicated off-diagonal couplings since the local problem is treated exactly.

5.3 Algorithm

5.3.1 Hybridization Expansion CTQMC Algorithm

A quantum impurity model may be represented as a Hamiltonian H_{QI}

$$H_{\mathrm{QI}} = H_{\mathrm{loc}} + H_{\mathrm{bath}} + H_{\mathrm{hyb}} \tag{5.1}$$

$$\begin{aligned} H_{\mathrm{loc}} &= H^0_{\mathrm{loc}} + H^I_{\mathrm{loc}} = \sum_{ab} E^{ab} d^\dagger_a d_b + \sum_{pqrs} I^{pqrs} d^\dagger_p d^\dagger_q d_r d_s & (5.2)\\ H_{\mathrm{bath}} &= \sum_{k\alpha} \varepsilon_{k\alpha} c^\dagger_{k\alpha} c_{k\alpha} & (5.3)\\ H_{\mathrm{hyb}} &= \sum_{k\alpha b} (V^{\alpha b}_k c^\dagger_{k\alpha} d_b + \mathrm{h.c.}) & (5.4) \end{aligned}$$

H_{loc} describes the "impurity" (a system with a finite (typically small) number of degrees of freedom), H_{bath} describes the non-interacting system, and H_{hyb} gives the coupling between the impurity and bath.

The Anderson impurity model describes a localized electronic level, subject to a local Coulomb interaction, which is coupled to a band of non-interacting conduction electrons. In the single-impurity single-orbital case, its Hamiltonian is given by

$$H_{\mathrm{AIM}} = \underbrace{\sum_{k\sigma} \varepsilon_k c^\dagger_{k\sigma} c_{k\sigma}}_{H_{\mathrm{bath}}} + \underbrace{\sum_{\sigma} \varepsilon_0 d^\dagger_\sigma d_\sigma + U n_\uparrow n_\downarrow}_{H_{\mathrm{loc}}} + \underbrace{\sum_{k\sigma} \left(V_k c^\dagger_{k\sigma} d_\sigma + h.c. \right)}_{H_{\mathrm{hyb}}} \tag{5.5}$$

In Hybridization Expansion Continuous Time Quantum Monte Carlo (CT-HYB) [170–173], we take the hybridization term as a perturbation, i.e.

$$H_b = H_{\rm hyb} = \sum_{pj} (V_p^j c_p^\dagger d_j + \sum_{pj} V_p^{j*} d_j^\dagger c_p), \tag{5.6}$$

and thus the partition function becomes

$$\begin{aligned} Z = \sum_{k=0}^{\infty} \int_0^\beta d\tau_1 \dots \int_{\tau_{k-1}}^\beta d\tau_k \int_0^\beta d\tau_1' \dots \int_{\tau_{k-1}'}^\beta d\tau_k' \sum_{\substack{j_1,\cdots j_k \\ j_1',\cdots j_k'}} \sum_{\substack{p_1,\cdots p_k \\ p_1',\cdots p_k'}} V_{p_1}^{j_1} V_{p_1'}^{j_1'*} \cdots V_{p_k}^{j_k} V_{p_k'}^{j_k'*} \\ \times {\rm Tr}_d \left[T_\tau e^{-\beta H_{\rm loc}} d_{j_k}(\tau_k) d_{j_k'}^\dagger(\tau_k') \cdots d_{j_1}(\tau_1) d_{j_1'}^\dagger(\tau_1') \right] \\ \times {\rm Tr}_c \left[T_\tau e^{-\beta H_{\rm bath}} c_{p_k}^\dagger(\tau_k) c_{p_{k'}}(\tau_k') \cdots c_{p_1}^\dagger(\tau_1) c_{p_1'}(\tau_1') \right]. \end{aligned} \tag{5.7}$$

The bath partition function could be integrated out:

$$Z_{\rm bath} = {\rm Tr} e^{-\beta H_{\rm bath}} = \prod_\sigma \prod_p (1 + e^{-\beta \varepsilon_p}). \tag{5.8}$$

With the anti-periodic hybridization function $\mathbf{\Delta}$,

$$\Delta_{lm}(\tau) = \sum_p \frac{V_p^{l*} V_p^m}{e^{\varepsilon_p \beta} + 1} \times \begin{cases} -e^{-\varepsilon_p(\tau-\beta)}, & 0 < \tau < \beta \\ e^{-\varepsilon_p \tau}, & -\beta < \tau < 0 \end{cases}, \tag{5.9}$$

and by separating the contributions from each spin, we obtain

$$\begin{aligned} Z = Z_{\rm bath} \\ \times \prod_j \sum_{k_j=0}^{\infty} \int_0^\beta d\tau_1^j \dots \int_{\tau'^j_{k_j-1}}^\beta d\tau'^j_{k_j} \\ \times {\rm Tr}_d \Big[T_\tau e^{-\beta H_{\rm loc}} d_j(\tau_{k_j}^j) d_j^\dagger(\tau_{k_j}'^j) \dots d_j(\tau_1^j) d_j^\dagger(\tau_1'^j) \Big] \det \mathbf{\Delta}_j, \end{aligned} \tag{5.10}$$

where

$$\Delta = \begin{bmatrix} \Delta(\tau_0' - \tau_0) & \Delta(\tau_0' - \tau_1) & \cdots & \Delta(\tau_0' - \tau_n) \\ \Delta(\tau_1' - \tau_0) & \Delta(\tau_1' - \tau_1) & \cdots & \Delta(\tau_1' - \tau_n) \\ \cdots & \cdots & \cdots & \cdots \\ \Delta(\tau_n' - \tau_0) & \Delta(\tau_n' - \tau_1) & \cdots & \Delta(\tau_n' - \tau) \end{bmatrix}. \tag{5.11}$$

We can sample the partition function above using Monte Carlo method. To determine the weight of each configuration, we need to calculate the contribution from the local part (the trace) and the hybridization part (the determinant). We would discuss both in the following sections 5.3.2, 5.3.3, and introduce scalable algorithms in 5.6.3 and 5.6.2.

5.3.2 Evaluation of the Trace Using the Segment Picture

The trace factor represents the impurity with particles hopping in and out at imaginary times τ' and τ, and the determinant sums up all compatible hybridization events with the bath. In the impurity basis ($|0\rangle$,$|\uparrow\rangle$,$|\downarrow\rangle$, $|\uparrow\downarrow\rangle$) the Hubbard Hamiltonian is diagonal, and the creation and annihilation operators for given spin have to alternate for the trace to be finite. This allows the configuration of the operators to be represented in a segment picture, where each pair of neighboring creation/annihilation operators are represented by a segment on the imaginary time axis, as shown in Figure 5.1.

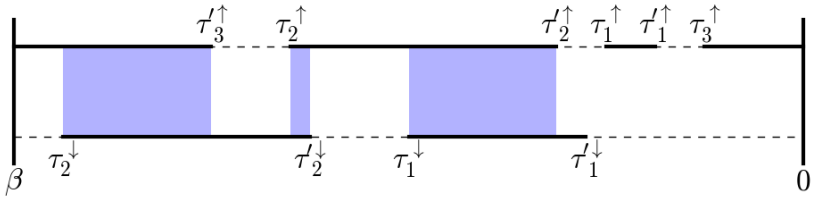


FIGURE 5.1. A segment picture showing a possible configuration of the two spin channels. The blue section indicates the overlap between two spin channels, l_{overlap} in Equation 5.12.

In the said basis, the contribution of the local Hamilton can be evaluated as:

$$W_{\text{loc}} = s_{\uparrow} s_{\downarrow} e^{\mu(l_{\uparrow}+l_{\downarrow})-U l_{\text{overlap}}}, \tag{5.12}$$

where l_σ is the total length of segments on spin channel σ, l_{overlap} is the total overlap between two spin channels. The sign s_σ is -1 when one of the segments winds around from β to 0, and $+1$ otherwise.

5.3.3 Evaluation of the Trace for Non-diagonal Hamiltonian

In models such as Dynamical Hubbard Model, the local part of Hamiltonian is not diagonal in the occupation number basis, and, therefore, we cannot use the segment picture to evaluate the trace easily. In this case, we need to evaluate the trace of this term:

$$e^{-H*(\beta-t_n)}F_{t_n}e^{-H*(t_n-t_{n-1})}F_{t_{n-1}}\dots F_{t_0}e^{-Ht_0}, \tag{5.13}$$

where H is the Hamiltonian, and F_{t_i} is a Fermion operator at time t_i.

To evaluate the exponential terms, we diagonalize the Hamiltonian with

$$H = UVU^T$$

where V is a diagonal matrix with eigenvalues of H, each column of U is an eigenvector of H. Using

$$UU^T = I,$$

we have

$$e^{-Ht} = e^{-UVU^Tt} = Ue^{-Vt}U^T,$$

and the term 5.13 becomes

$$Ue^{-V*(\beta-t_n)}U^TF_{t_n}Ue^{-V*(t_n-t_{n-1})}\dots F_{t_0}Ue^{-Vt_0}U^T.$$

Define

$$D_t = U^TF_tU,$$

the term is then

$$Ue^{-V*(\beta-t_n)}D_{t_n}e^{-V*(t_n-t_{n-1})}\dots D_{t_0}e^{-Vt_0}U^T. \tag{5.14}$$

We can then evaluate the full trace of the matrix above, using a series of matrix multiplications.

5.3.4 Monte Carlo Sampling Procedure

To sample the configuration space with Monte Carlo procedures, one can propose a new configuration by:

- Adding a new segment to the existing configuration;
- Removing a segment to the existing configuration;
- Shifting an end of a segment in the existing configuration;

To satisfy the detailed balance condition, we must make sure that

$$W_{AB}/W_{BA} = W[B]W[A] \tag{5.15}$$

where A and B are two configurations, and W_{AB} is the transition probability from configuration A to configuration B, and vice versa.

In the case of adding a segment (Figure 5.2), one can first randomly choose a starting point for a segment, say τ' in $(\beta, 0]$. If τ' falls on one of the existing segments, the proposal is rejected. Otherwise if τ' is located between two segments τ_j and τ'_{j+1}, we pick the end point from $[0, l_{\max}]$, where $l_{\max} = \mathrm{mod}(\tau'_{j+1} - \tau + \beta, \beta)$.

Assuming there is an infinitesimal grid with grid size $d\tau$ on the imaginary time axis, the proposal probability can be found as:

$$P_{\text{prop:k}\to\text{k+1}} = \frac{d\tau^2}{\beta l_{\max}}. \tag{5.16}$$

In the case of removing a segment (Figure 5.3), one can randomly pick a segment from k_σ existing segments, and thus the proposal probability is

$$P_{\text{prop:k}\to\text{k}-1} = \frac{1}{k_\sigma}. \tag{5.17}$$

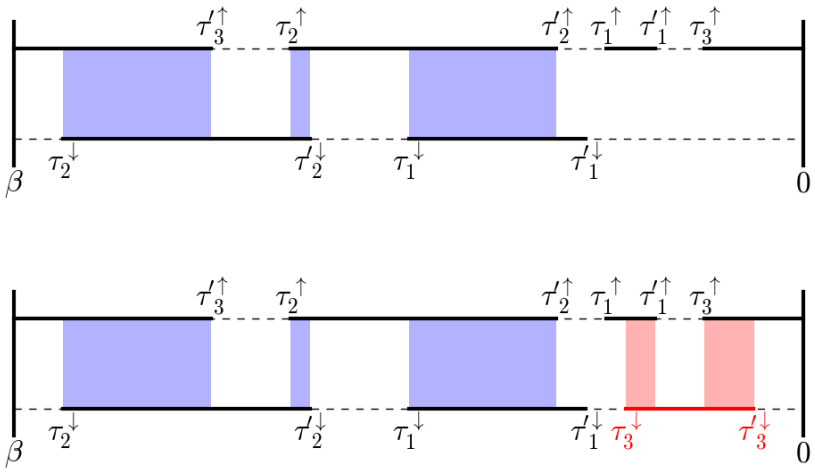


FIGURE 5.2. A segment picture showing the addition of a segment (τ_3', τ_3) (red line) to the $|\downarrow\rangle$ channel. The red shade shows the change of overlap from this addition.

Combined with the detailed balance condition and previous weight calculations, we have the probability of accepting an addition as:

$$P_{\text{accpt:k}\to\text{k+1}} = \min\left(1, \text{sign}(\tau - \tau')\frac{\beta l_{\max}}{k_\sigma + 1}\frac{\det \Delta^{k+1}}{\det \Delta^k} e^{\mu l} e^{-U\Delta l_{\text{overlap}}}\right), \tag{5.18}$$

where l is the length of segment to be added, and $\Delta l_{\text{overlap}}$ is the change in overlap between two spin channels.

Similarly, the acceptance ratio for removing a segment is:

$$P_{\text{accpt:k}\to\text{k-1}} = \min\left(1, \text{sign}(\tau - \tau')\frac{k_\sigma}{\beta l_{\max}}\frac{\det \Delta^{k-1}}{\det \Delta^k} e^{-\mu l} e^{U\Delta l_{\text{overlap}}}\right). \tag{5.19}$$

The probability for accepting a shifting move (Figure 5.4) is:

$$P_{\text{accpt:k}\to\text{k}} = \min\left(1, \text{sign}(\tau_k - \tau_k')\text{sign}(\tau_{k'} - \tau_{k'}')\frac{\det \Delta^k_{\text{new}}}{\det \Delta^k} e^{\mu\Delta l} e^{-U\Delta l_{\text{overlap}}}\right). \tag{5.20}$$

In the case where one or more channels have no segments, we need to evaluate the trace from the Hamiltonian. Here k refers to the number of segments on the channel where the move is proposed, and k' refers to the number of segments on the channel with opposite spin direction.

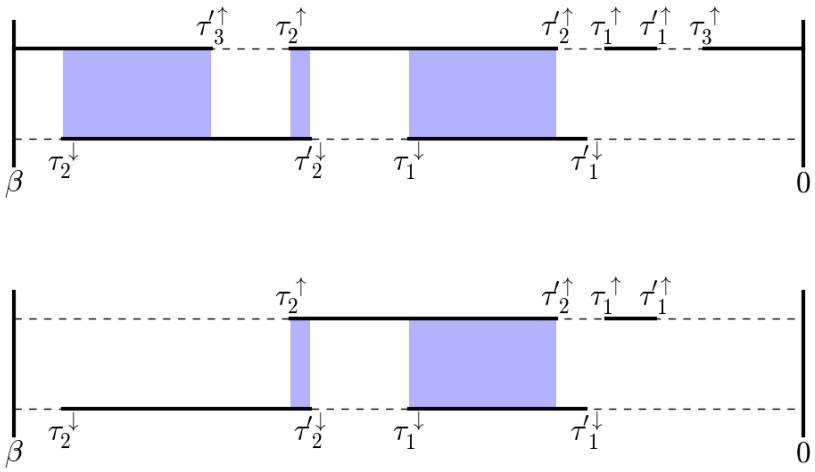


FIGURE 5.3. A segment picture showing the removal of a segment (τ_3', τ_3) (red line) from the $|\uparrow\rangle$ channel.

1. Adding a segment to channel σ.

$$P_{\text{accpt:k}\to\text{k+1}} = \min\left(1, \text{sign}(\tau - \tau')\frac{\beta l_{\max}}{k_\sigma + 1}\frac{\det \Delta^{k+1}}{\det \Delta^k} * Q\right), \tag{5.21}$$

where Q is the ratio of the new trace verses the old trace.

(a) $k = 0, k' \neq 0$

$$Q = \frac{e^{-l'_\sigma \epsilon_f} e^{-l_{ov} U}}{1 + e^{-\beta\epsilon_f} e^{-l_{\bar\sigma} U}}, \tag{5.22}$$

where l'_σ is the length of the segment to be added, l_{ov} is the overlap between the new segment and the segments on channel $\bar\sigma$, and $l_{\bar\sigma}$ is the total length of segments on channel $\bar\sigma$.

(b) $k \neq 0, k' = 0$

$$Q = \frac{e^{-l'_\sigma \epsilon_f}(1 + e^{-\beta\epsilon_f} e^{-(l_\sigma + l'_\sigma)U})}{1 + e^{-\beta\epsilon_f} e^{-l_\sigma U}}, \tag{5.23}$$

where l'_σ is the length of the segment to be added, and l_σ is the total length of segments on channel σ.

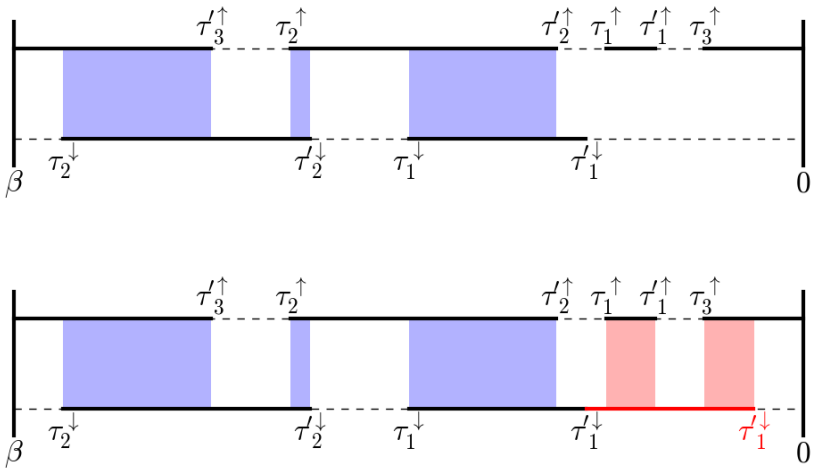


FIGURE 5.4. A segment picture showing shifting the end of a segment (τ_1', τ_1) (red line) on the $|\downarrow\rangle$ channel. The red shade shows the change of overlap from this shift.

(c) $k = 0, k' = 0$

$$Q = \frac{e^{-l'_\sigma \epsilon_f}(1 + e^{-\beta\epsilon_f} e^{-l'_\sigma U})}{1 + 2e^{-\beta\epsilon_f} + e^{-\beta(2\epsilon_f + U)}}, \tag{5.24}$$

where l'_σ is the length of the segment to be added.

2. Removing a segment.

$$P_{\text{accpt:k}\to\text{k}-1} = \min\left(1, \text{sign}(\tau - \tau')\frac{k_\sigma}{\beta l_{\max}}\frac{\det \Delta^{k-1}}{\det \Delta^k} * Q^{-1}\right), \tag{5.25}$$

where Q is calculated in Equations 5.22 $\sim$ 5.24, in corresponding situations.

3. Shift a segment. In the case of shifting, k cannot be zero. Therefore the only special case is $k \neq 0, k' = 0$. In this case,

$$Q = \frac{e^{-(l'_\sigma - l_\sigma)\epsilon_f}(1 + e^{-\beta\epsilon_f} e^{-l'_\sigma U})}{1 + e^{-\beta\epsilon_f} e^{-l_\sigma U}}, \tag{5.26}$$

where l_σ is the length of the chosen segment before the shift operation, l'_σ is the length after the operation.

When adding/removing/shift a segment on a channel, the hybridization matrix needs to be updated accordingly.

1. Shifting

   One can shift either the beginning or the end of a segment, which corresponds to modifying a column/row in the hybridization matrix. For example, when shifting the end of segment i from τ_i' to τ_{new}', the ith row of the matrix is updated:

$$\begin{bmatrix} \Delta(\tau_0'-\tau_0) & \Delta(\tau_0'-\tau_1) & \cdots & \Delta(\tau_0'-\tau_n) \\ \Delta(\tau_1'-\tau_0) & \Delta(\tau_1'-\tau_1) & \cdots & \Delta(\tau_1'-\tau_n) \\ \cdots & \cdots & \cdots & \cdots \\ \Delta(\tau_i'-\tau_0) & \Delta(\tau_i'-\tau_1) & \cdots & \Delta(\tau_i'-\tau_n) \\ \cdots & \cdots & \cdots & \cdots \\ \Delta(\tau_n'-\tau_0) & \Delta(\tau_n'-\tau_1) & \cdots & \Delta(\tau_n'-\tau) \end{bmatrix} \Rightarrow$$

$$\begin{bmatrix} \Delta(\tau_0'-\tau_0) & \Delta(\tau_0'-\tau_1) & \cdots & \Delta(\tau_0'-\tau_n) \\ \Delta(\tau_1'-\tau_0) & \Delta(\tau_1'-\tau_1) & \cdots & \Delta(\tau_1'-\tau_n) \\ \cdots & \cdots & \cdots & \cdots \\ \Delta(\tau_{new}'-\tau_0) & \Delta(\tau_{new}'-\tau_1) & \cdots & \Delta(\tau_{new}'-\tau_n) \\ \cdots & \cdots & \cdots & \cdots \\ \Delta(\tau_n'-\tau_0) & \Delta(\tau_n'-\tau_1) & \cdots & \Delta(\tau_n'-\tau) \end{bmatrix} \tag{5.27}$$

2. Adding Adding a segment (τ, τ') between the original $i-1$ and ith segment corresponds to adding a row and column in the hybridization matrix.

$$\begin{bmatrix} \Delta(\tau_0'-\tau_0) & \cdots & \Delta(\tau_0'-\tau_{i-1}) & \Delta(\tau_0'-\tau) & \cdots & \Delta(\tau_0'-\tau_n) \\ \Delta(\tau_1'-\tau_0) & \cdots & \Delta(\tau_1'-\tau_{i-1}) & \Delta(\tau_1'-\tau) & \cdots & \Delta(\tau_1'-\tau_n) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \Delta(\tau_{i-1}'-\tau_0) & \cdots & \Delta(\tau_{i-1}'-\tau_{i-1}) & \Delta(\tau_{i-1}'-\tau) & \cdots & \Delta(\tau_{i-1}'-\tau_n) \\ \Delta(\tau'-\tau_0) & \cdots & \Delta(\tau'-\tau_{i-1}) & \Delta(\tau'-\tau) & \cdots & \Delta(\tau'-\tau_n) \\ \Delta(\tau_i'-\tau_0) & \cdots & \Delta(\tau_i'-\tau_{i-1}) & \Delta(\tau_i'-\tau) & \cdots & \Delta(\tau_i'-\tau_n) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \Delta(\tau_n'-\tau_0) & \cdots & \Delta(\tau_n'-\tau_{i-1}) & \Delta(\tau_n'-\tau) & \cdots & \Delta(\tau_n'-\tau_n) \end{bmatrix} \tag{5.28}$$

3. Removing Removing a segment (τ_i, τ_i') also removes the corresponding row and column in the hybridization matrix.

5.4 Measurement

5.4.1 Single Particle Green's Function

The imaginary time green's function can be found by:

$$G(\tau) = -\left\langle \frac{1}{\beta} \sum_{ij}^{k} \left(\Delta^{(k)}\right)^{-1}_{ji} \delta(\tau, \tau_i' - \tau_j) \right\rangle_{MC} = -\left\langle \frac{1}{\beta} \sum_{ij}^{k} M^{(k)}_{ji} \delta(\tau, \tau_i' - \tau_j) \right\rangle_{MC}. \tag{5.29}$$

To reduce noise and save memory, we split β in to fine grids `N_TAU` and bin data.

We can also find the Matsubara frequency Green's function by Fourier transform:

$$G(i\omega) = -\langle \frac{1}{\beta} \sum_{i,j} \exp^{i\omega(\tau_j' - \tau_j)} M_{ji} \rangle_{MC} \tag{5.30}$$

Note that for a small number of `N_TAU`, the Fourier transform back to Matsubara frequency may be inaccurate.

5.4.2 Susceptibilities

The charge susceptibility is defined as

$$\chi_{\mathrm{c}}(\tau) = \langle [n_\uparrow + n_\downarrow](\tau)[n_\uparrow + n_\downarrow](0) \rangle = \frac{1}{\beta} \int_0^\beta d\tau_0 [n_\uparrow + n_\downarrow](\tau + \tau_0)[n_\uparrow + n_\downarrow](\tau_0). \tag{5.31}$$

Here $[n_\uparrow + n_\downarrow](\tau)$ stands for $n_\uparrow + n_\downarrow$ at the imaginary time τ.

The spin susceptibility is defined as

$$\chi_{\sigma}(\tau) = \langle [n_\uparrow - n_\downarrow](\tau)[n_\uparrow - n_\downarrow](0) \rangle = \frac{1}{\beta} \int_0^\beta d\tau_0 [n_\uparrow - n_\downarrow](\tau + \tau_0)[n_\uparrow - n_\downarrow](\tau_0). \tag{5.32}$$

In CY-HYB, the terms can be easily evaluated by shifting all of the segments on one channel, and measure the overlap between the shifted channel and the other channel.

5.5 DMFT Loop

A commonly used method to solve lattice problems is to use the dynamical mean field theory (DMFT) to approximate the original problem by an impurity problem plus a self-consistency condition.

The impurity problem is solved by the CT-HYB impurity solver described above, and an impurity Green's function $G_f(i\omega)$ is obtained. The impurity Green's function is used to produce the lattice Green's function $\mathcal{G}(i\omega)$ by the coarse-graining process. The impurity Green's function and the lattice Green's function should obey Dyson's equation:

$$G_f(i\omega) = \frac{1}{\mathcal{G}^{-1}(i\omega) + \Sigma(k, i\omega)}. \tag{5.33}$$

For next-neighbor hopping t on the Bethe lattice with density of states

$$\rho_{Bethe}(\epsilon) = \begin{cases} \frac{\sqrt{4t_*^2 - \epsilon^2}}{2\pi t_*^2} & \text{for} |\epsilon| \leq 2|t| \\ 0 & \text{otherwise} \end{cases}, \tag{5.34}$$

the self-consistency equation yields a simple relation

$$\mathcal{G}(i\omega) = i\omega + \mu - t_*^2 G_f(i\omega) \tag{5.35}$$

or

$$\Delta(i\omega) = t_*^2 G_f(i\omega). \tag{5.36}$$

This also allows us to do the Fourier transform from Matsubara frequency to imaginary time easily.

Overall, the DMFT loop is implemented as following (Figure 5.5):

1. Initialize the hybridization function $\Delta(\tau)$.
2. Call the impurity solver and get $G_f(i\omega)$.
3. Obtain the new hybridization function by $\Delta'(i\omega) = t_*^2 G_f(i\omega)$.
4. Linearly mix the new hybridization function with the old by

$$\Delta(i\omega) = m * \Delta'(i\omega) + (1 - m)\Delta_{\text{old}}(i\omega). \tag{5.37}$$

5. Fourier transform $\Delta(i\omega)$ to $\Delta(\tau)$.
6. Goto 2, and iterate till converge.

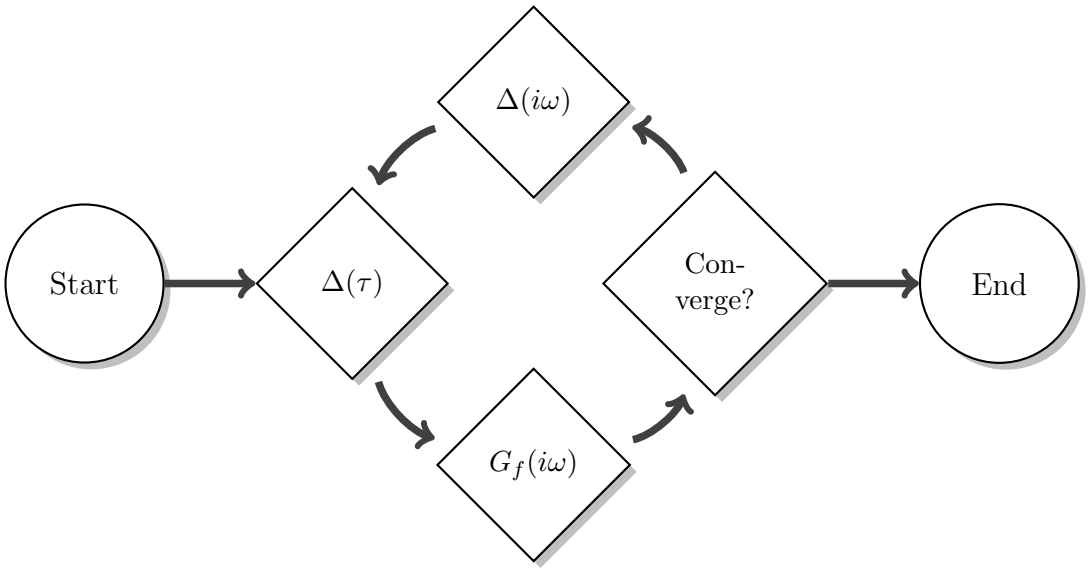


FIGURE 5.5. A diagram for the DMFT loop.

5.6 Implementation

We implement an impurity solver based the CT-HYB algorithm for Intel Many Integrated Core Architecture, or Intel MIC. Intel MIC is an x86-compatible multiprocessor architecture that can utilize existing parallelization software tools, such as OpenMP, OpenCL, etc. The x86 compatibility makes it easy to execute the program on coprocessor with little code modification. The Xeon Phi 7120P is capable of 1.2 teraFLOPS of double precision floating point instructions with 352 GB/sec memory bandwidth at 300W. The current top supercomputer on TOP500 list, Tianhe-2, uses Intel Ivy Bridge processors and Xeon Phi coprocessors to achieve 33.86 petaFLOPS.

5.6.1 OpenMP Parallelization

We use a straightforward OpenMP approach to parallelizing our code. By deploying multiple Markov chains on each processor/coprocessor, we have an embarrassingly parallel program, where each of the processes is independent of another, thus, no communication overhead is required. Next, we discuss how we speed up the computation and optimize our performance.

5.6.2 Fast Matrix Update

The most time-consuming part of the algorithm is the update of the hybridization matrix. For each proposed update, the determinant of the new matrix is required to compute the accepting probability. A straight forward determinant computation scales to k^3, where k is the expansion order. Due to the fact that in each update move, only one row and one column are changed, one can use the Sherman-Morrison formula to update the matrix determinant and the inverse matrix in k^2, thus making the update process much more efficient, especially at low temperatures.

Suppose $\mathbf{A}$ is an invertible square matrix, and $\mathbf{u}$ and $\mathbf{v}$ are column vectors that describe the update to the matrix, then the determinant of the new matrix is

$$\det(\mathbf{A} + \mathbf{u}\mathbf{v}^T) = (1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}) \det(\mathbf{A}). \tag{5.38}$$

To update the inverse of the matrix, one can use

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}. \tag{5.39}$$

In practice, the update of the inverse includes a few more steps, such as inserting/removing empty rows/columns, to ensure the size of the matrix reflects the change in the number of segments. Each Monte Carlo move requires two updates to the matrix and its inverse, one for the column, and one for the row.

With the vectorization on Intel CPUs and MIC coprocessors, this procedure can be done efficiently on both platforms.

5.6.3 Krylov Method

When the Hamiltonian is not diagonal, the update of the trace can be very expensive. The complexity of the method described in 5.3.3 is $O(m^3 n)$, where m is the size of the matrix, and n is the number of fermion operators in the series. Since m scales exponentially with the number of orbitals, this can be very expensive

even for a moderate number of orbitals (say 5). Instead, we can use the Krylov method[172] to find the trace.

First, we find the few lowest eigenstates of the Hamiltonian $|i\rangle$, since they are usually more relevant at low temperatures. Then the trace is approximately

$$\sum_i \langle i|e^{-H*(\beta-t_n)}F_{t_n}e^{-H*(t_n-t_{n-1})}F_{t_{n-1}}\dots F_{t_0}e^{-Ht_0}|i\rangle. \tag{5.40}$$

Then each of the term in the summation become of a series of the following operations:

- $e^{-Ht}|v\rangle$,
- $F|v\rangle$.

The second operation is $O(m^2)$, so we'll ignore it for now. For the first term, we can generate a Krylov space using the following method:

1. $v_1 = v/||v||$,
2. Iteration: do $j = 1, 2, \dots, k$
   (a) $w = Hv_j$
   (b) Iteration: do $i = 1, 2..., j$
      i. $h_{i,j} = w \cdot v_i$
      ii. $w = w - h_{i,j}v_i$
   (c) $h_{j+1,j} = ||w||$, $v_{j+1} = w/h_{j+1,j}$

With these iteration, we generate a orthonormal basis $V_k = [v_1, v_2, \dots, v_k]$ and a $k \times k$ matrix H_k, where $H_k(i, j) = h_{i,j}$.

The exponential term can be just evaluated by:

$$e^{-Ht}v \approx ||v||V_m e^{-H_k t}e_1, \tag{5.41}$$

where $e_1 = [1, 0, 0, \ldots 0]^T$. The complexity of this operation is $O(k^3 + mk^2 + m^2k)$. Usually a small value (~ 3) of k is needed, thus, the complexity of the computation is reduced. Overall the complexity scales as $O(m^2kn)$.

5.6.4 Using Legendre Polynomials for Measurement

To reduce high-frequency noise in the measurements of Green's function, one can use a set of Legendre polynomials[174] as basis and measure the coefficients:

$$G_l = \sqrt{2l+1} \int_0^\beta d\tau P_l(x(\tau)) G(\tau). \tag{5.42}$$

In CT-HYB:

$$G_l = -\frac{\sqrt{2l+1}}{\beta} \left\langle \sum_{ij} M_{ji} \tilde{P}_l(\tau_i' - \tau_j) \right\rangle_{MC}, \tag{5.43}$$

where

$$\tilde{P}(\tau) = \begin{cases} P_l(x(\tau)) & \tau > 0 \\ -P_l(x(\tau+\beta)) & \tau < 0 \end{cases} \tag{5.44}$$

and

$$x(\tau) = 2\tau/\beta - 1. \tag{5.45}$$

To restore $G(\tau)$ or $G(i\omega)$ from the measured set of coefficients,

$$G(\tau) = \sum_{l \geq 0} \frac{\sqrt{2l+1}}{\beta} P_l(x(\tau)) G_l \tag{5.46}$$

and

$$G(i\omega) = \sum_{l \geq 0} G_l \frac{\sqrt{2l+1}}{\beta} \int_0^\beta \exp^{i\omega_n \tau} P_l(x(\tau)) = \sum_{l \geq 0} T_{nl} G_l, \tag{5.47}$$

where

$$T_{nl} = (-1)^n i^{l+1} \sqrt{2l+1} j_l \left(\frac{(2n+1)\pi}{2} \right) \tag{5.48}$$

and $j_l(z)$ are spherical Bessel functions. Note that in the procedure, no model-guided Fourier transform is used.

By setting an appropriate cut-off at the number of Legendre series, high-frequency noise is filtered. One can measure the error in the Legendre polynomials to determine where the cutoff should be.

5.6.5 Optimization

We use Intel Vtune Amplifier to benchmark the code and identify the bottlenecks. The Intel Vtune Amplifier provides a set of performance insight into CPU and Xeon Phi performance, threading performance, etc. We use Command Line Interface (CLI) of Intel Vtune Amplifier, and inspected metrics such as:

- Walltime of application
- Hotspot, tells the time consumption
- Cycles per instruction, or CPI rate
- L1 Hit Ratio
- Estimated Latency Impact
- Vectorization Intensity
- L1 Compute to Data Access Ratio
- L2 Compute to Data Access Ratio

Several techniques are used to eliminate the bottlenecks and improve overall performance, including:

- Random number generators The Monte Carlo sampling technique we use requires multiple random numbers generated on each thread with every update step, and we use pseudo-random generators (random number generator) to produce them.

  The most commonly used random number generator in C is the `rand()` function, which returns a pseudo-random integer using a hidden state. The `srand()` set its argument as the seed for a new sequence of pseudo-random

number to be returned by `rand()`. The function rand() is not thread-safe, since the hidden state it uses is modified on each call. When called from multiple threads, this cause congestion, and the performance penalty is even worse for a massively parallel platform like Xeon Phi comparing to CPUs.

In order to get thread-safe behavior in our multi-threaded application, we can use the function `rand_r()`. Unlike `rand()`, `rand_r()` uses a pointer to an unsigned int to store state between calls, so each different thread can use its own state. Therefore, `rand()` calls should be replaced with thread-safe `rand_r()` to avoid performance penalty (Figure 5.6) and ensure code correctness.

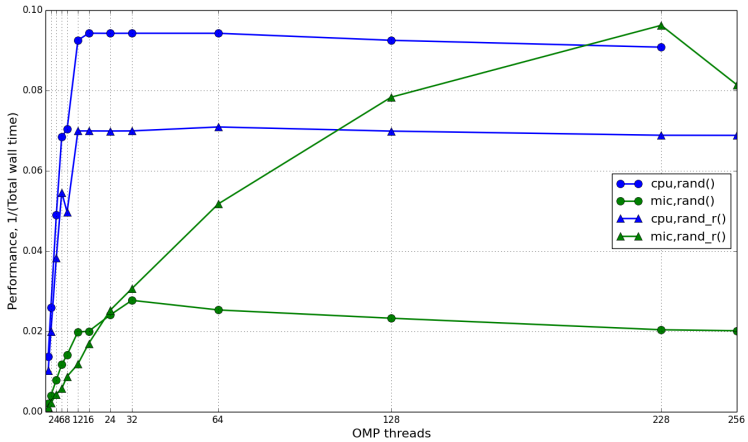


FIGURE 5.6. Performance using different random number generators.

We also tested other more advanced random number generators in our code, such as Mersenne Twister, and PCG. They offer better statistical quality in the pseudo-random number sequence, and also have great multi-threading efficiency. However, the overall performance gain is not significant, since the thread-safe random number generator itself is not very time consuming comparing to other parts of the code.

- Move memory allocation out of the OpenMP region In our implementation, the hybridization matrix needs to be updated for each accepted Monte Carlo move, and the size of the matrix depends on the number of segments in each channel. In addition to the old and updated matrices, temporary storage for the intermediate matrices used in the fast update process are also required.

  We find that the repeated allocation and free of memory used for the said matrices poses a huge performance penalty. A better practice would be allocating all the space for the matrices before the OpenMP region, and free after the threads join. This sets a limit on the max size of the matrices. For a low temperature, more segments are expected, which means the size of the matrices would be bigger, and thus a larger space needs to be assigned (Figure 5.7).

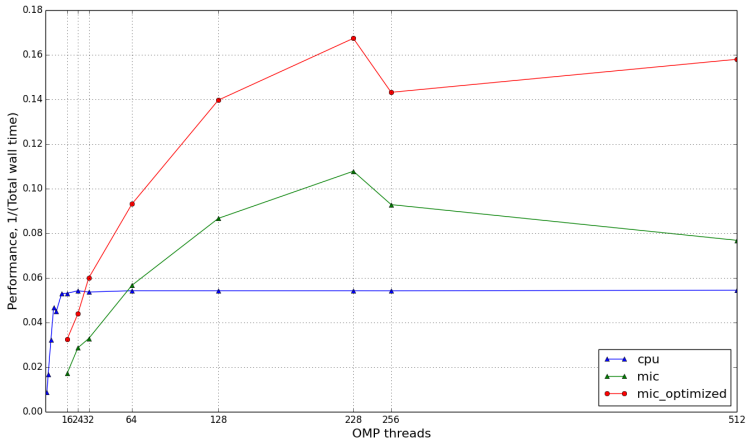


FIGURE 5.7. Performance before and after memory allocation optimization for MIC.

- Improve data access pattern in kernel by
  - Separating loops to avoid cache bank conflict
  - Interchanging loops to guarantee Unit-Stride Access
  - Aligning elements in array

– Enhancing matrix storage

With these optimization methods, we improved CPI rate, L1 hit ratio, and vectorization intensity. We achieved more than two times speedup comparing to original CPU code.

5.7 Preliminary Results and Discussion

We benchmark our CTQMC solver with another Hirsch-Fye code and compare the results, as shown in Figure 5.8. This verifies that we can match the results of other impurity solvers.

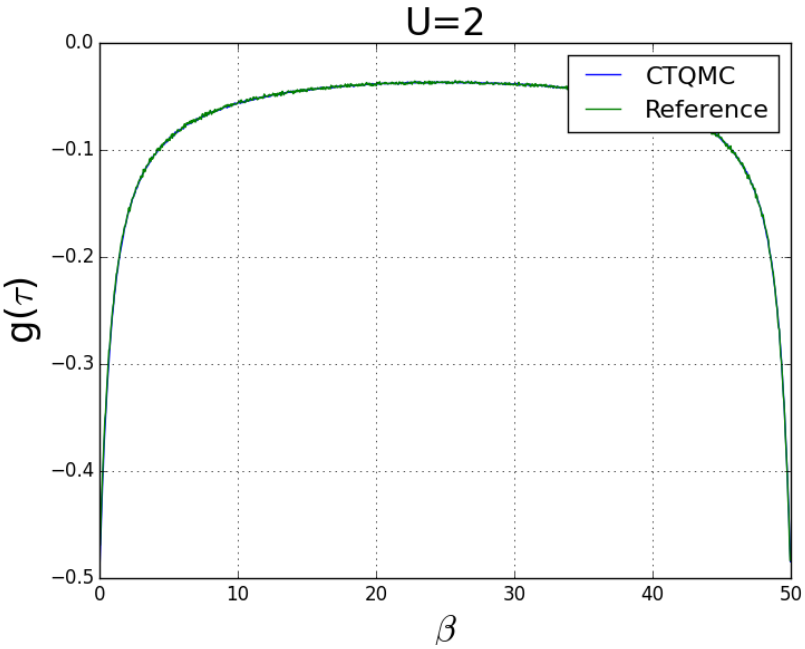


FIGURE 5.8. Results for $G(\tau)$ on Anderson Model, from our code comparing with weak coupling results. Here $\beta = 50$, $U = 2$, $\mu = 1$, $v^2 = 0.156$, $D = 1$, $\Delta = v^2(log(i\omega + D)log(i\omega D))$

Although the formalism shown in 5.3 is derived for single impurity Anderson model, the CT-HYB algorithm can be extended to more orbitals, or other models easily, by changing the H_{loc} term in the Hamiltonian and treating the term exactly. For example, we can easily extend the code to two or more orbitals, and use exact

diagonalization to calculate the contribution of the local term. For more orbitals, a Krylov solver[172] can be used to reduce the cost of trace computation, as discussed in section 5.6.3.

One of the possible application of the solver is the Dynamical Hubbard Models. The Dynamical Hubbard models contain the idea of break the electron-hole symmetry, due to the insufficiency of conventional Hubbard model in describing some scenarios in real materials, where important physics happens in transport and other processes. With the new methods applied, we hope one can answer the question about superconductivity in the Dynamical Hubbard model.

Chapter 6
Conclusion

In this dissertation, the work on two projects is covered.

In the Three-Dimensional Edwards-Anderson Model project, we developed an efficient GPU implementation of Monte Carlo simulation with parallel tempering and multispin coding technique. We achieved world-leading performance in GPU implementation on this model. We then used the code to study the model in an external field. Our results show that susceptibilities are not normally distributed and mean is dominated by rare events. As a result, a huge number of disordered samples must be included in the average. With the current method and computing power, we cannot gain a definitive answer on the nature of the spin glass phase.

In the Hybridization Expansion Continuous Time Monte Carlo Solver, we delivered an impurity solver on the Intel Xeon Phi platform, using the fast update procedures. We showed that this code is twice as fast than our original CPU implementation, and can be easily extended to include more orbitals and complicated interactions. In collaboration with Roozbeh Karimi and Prof Koppelman, we developed a Krylov solver for systems with more orbitals. This impurity solver combined with the density functional theory and dynamical mean field theory can be used for calculations on multi-orbital systems, such as cuprate and iron-based superconductors.

References

[1] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, et al. (LIGO Scientific Collaboration and Virgo Collaboration), Phys. Rev. Lett. **116**, 061102 (2016), URL http://link.aps.org/doi/10.1103/PhysRevLett.116.061102.

[2] K. Binder and A. P. Young, Rev. Mod. Phys. **58**, 801 (1986).

[3] J. Brooke, D. Bitko, T. F., Rosenbaum, and G. Aeppli, Science **284**, 779 (1999), ISSN 0036-8075, http://science.sciencemag.org/content/284/5415/779.full.pdf, URL http://science.sciencemag.org/content/284/5415/779.

[4] J. Brooke, T. F. Rosenbaum, and G. Aeppli, Nature **413**, 610 (2001), URL http://dx.doi.org/10.1038/35098037.

[5] S. Ghosh, T. F. Rosenbaum, G. Aeppli, and S. N. Coppersmith, Nature **425**, 48 (2003), URL http://dx.doi.org/10.1038/nature01888.

[6] D. M. Silevitch, D. Bitko, J. Brooke, S. Ghosh, G. Aeppli, and T. F. Rosenbaum, Nature **448**, 567 (2007), URL http://dx.doi.org/10.1038/nature06050.

[7] D. H. Reich, B. Ellman, J. Yang, T. F. Rosenbaum, G. Aeppli, and D. P. Belanger, Phys. Rev. B **42**, 4631 (1990), URL http://link.aps.org/doi/10.1103/PhysRevB.42.4631.

[8] W. Wu, D. Bitko, T. F. Rosenbaum, and G. Aeppli, Phys. Rev. Lett. **71**, 1919 (1993), URL http://link.aps.org/doi/10.1103/PhysRevLett.71.1919.

[9] V. Cannella and J. A. Mydosh, Phys. Rev. B **6**, 4220 (1972), URL http://link.aps.org/doi/10.1103/PhysRevB.6.4220.

[10] S. F. Edwards and P. W. Anderson, Journal of Physics F: Metal Physics **5**, 965 (1975), URL http://stacks.iop.org/0305-4608/5/i=5/a=017.

[11] S. Kirkpatrick and D. Sherrington, Phys. Rev. B **17**, 4384 (1978), URL http://link.aps.org/doi/10.1103/PhysRevB.17.4384.

[12] D. Sherrington and S. Kirkpatrick, Phys. Rev. Lett. **35**, 1792 (1975).

[13] G. Parisi, Journal of Physics A: Mathematical and General **13**, 1101 (1980).

[14] G. Parisi, Journal of Physics A: Mathematical and General **13**, 1887 (1980).

[15] G. Parisi, Journal of Physics A: Mathematical and General **13**, L115 (1980), URL http://stacks.iop.org/0305-4470/13/i=4/a=009.

[16] A. B. Harris, T. C. Lubensky, and J.-H. Chen, Phys. Rev. Lett. **36**, 415 (1976).

[17] H. Tasaki, Journal of Statistical Physics **54**, 163 (1989).

[18] J. E. Green, M. A. Moore, and A. J. Bray, Journal of Physics C: Solid State Physics **16**, L815 (1983).

[19] D. S. Fisher and D. A. Huse, Journal of Physics A: Mathematical and General **20**, L1005 (1987).

[20] D. S. Fisher and D. A. Huse, Phys. Rev. B **38**, 386 (1988).

[21] A. P. Young and H. G. Katzgraber, Phys. Rev. Lett. **93**, 207203 (2004).

[22] F. Barahona, Journal of Physics A: Mathematical and General **15**, 3241 (1982).

[23] R. N. Bhatt and A. P. Young, Phys. Rev. Lett. **54**, 924 (1985).

[24] H. G. Ballesteros, A. Cruz, L. A. Fernández, V. Martín-Mayor, J. Pech, J. J. Ruiz-Lorenzo, A. Tarancón, P. Téllez, C. L. Ullod, and C. Ungil, Phys. Rev. B **62**, 14237 (2000), URL http://link.aps.org/doi/10.1103/PhysRevB.62.14237.

[25] R. A. Baños, A. Cruz, L. A. Fernandez, J. M. Gil-Narvion, A. Gordillo-Guerrero, M. Guidetti, D. Iñiguez, A. Maiorano, E. Marinari, V. Martin-Mayor, et al., Proceedings of the National Academy of Sciences **109**, 6452 (2012), http://www.pnas.org/content/109/17/6452.full.pdf, URL http://www.pnas.org/content/109/17/6452.abstract.

[26] T. J. Collaboration, in *Proceedings of the Future HPC Systems: the Challenges of Power-Constrained Performance* (ACM, New York, NY, USA, 2012), FutureHPC '12, pp. 2:1–2:11, ISBN 978-1-4503-1453-4.

[27] D. Vollhardt, Rev. Mod. Phys. **56**, 99 (1984), URL http://link.aps.org/doi/10.1103/RevModPhys.56.99.

[28] M. Imada, A. Fujimori, and Y. Tokura, Rev. Mod. Phys. **70**, 1039 (1998), URL http://link.aps.org/doi/10.1103/RevModPhys.70.1039.

[29] R. O. Jones and O. Gunnarsson, Rev. Mod. Phys. **61**, 689 (1989), URL http://link.aps.org/doi/10.1103/RevModPhys.61.689.

[30] A. Georges and G. Kotliar, Phys. Rev. B **45**, 6479 (1992), URL http://link.aps.org/doi/10.1103/PhysRevB.45.6479.

[31] M. Jarrell, Phys. Rev. Lett. **69**, 168 (1992), URL http://link.aps.org/doi/10.1103/PhysRevLett.69.168.

[32] A. Georges, G. Kotliar, W. Krauth, and M. J. Rozenberg, Rev. Mod. Phys. **68**, 13 (1996), URL http://link.aps.org/doi/10.1103/RevModPhys.68.13.

[33] M. J. Rozenberg, X. Y. Zhang, and G. Kotliar, Phys. Rev. Lett. **69**, 1236 (1992), URL http://link.aps.org/doi/10.1103/PhysRevLett.69.1236.

[34] A. Georges and W. Krauth, Phys. Rev. Lett. **69**, 1240 (1992), URL http://link.aps.org/doi/10.1103/PhysRevLett.69.1240.

[35] A. Fujimori, I. Hase, H. Namatame, Y. Fujishima, Y. Tokura, H. Eisaki, S. Uchida, K. Takegahara, and F. M. F. de Groot, Phys. Rev. Lett. **69**, 1796 (1992), URL http://link.aps.org/doi/10.1103/PhysRevLett.69.1796.

[36] X. Y. Zhang, M. J. Rozenberg, and G. Kotliar, Phys. Rev. Lett. **70**, 1666 (1993), URL http://link.aps.org/doi/10.1103/PhysRevLett.70.1666.

[37] A. Georges and W. Krauth, Phys. Rev. B **48**, 7167 (1993), URL http://link.aps.org/doi/10.1103/PhysRevB.48.7167.

[38] M. J. Rozenberg, G. Kotliar, H. Kajueter, G. A. Thomas, D. H. Rapkine, J. M. Honig, and P. Metcalf, Phys. Rev. Lett. **75**, 105 (1995), URL http://link.aps.org/doi/10.1103/PhysRevLett.75.105.

[39] A. Y. Matsuura, H. Watanabe, C. Kim, S. Doniach, Z.-X. Shen, T. Thio, and J. W. Bennett, Phys. Rev. B **58**, 3690 (1998), URL http://link.aps.org/doi/10.1103/PhysRevB.58.3690.

[40] S.-K. Mo, J. D. Denlinger, H.-D. Kim, J.-H. Park, J. W. Allen, A. Sekiyama, A. Yamasaki, K. Kadono, S. Suga, Y. Saitoh, et al., Phys. Rev. Lett. **90**, 186403 (2003), URL http://link.aps.org/doi/10.1103/PhysRevLett.90.186403.

[41] NVIDIA, *TESLA K80 GPU ACCELERATOR Board Specification* (NVIDIA, 2015).

[42] J. Nickolls, I. Buck, M. Garland, and K. Skadron, Queue **6**, 40 (2008).

[43] J. Jeffers and J. Reinders, *Intel Xeon Phi Coprocessor High Performance Programming* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2013), 1st ed., ISBN 9780124104143, 9780124104945.

[44] X. Liao, L. Xiao, C. Yang, and Y. Lu, Frontiers of Computer Science **8**, 345 (2014), ISSN 2095-2236, URL http://dx.doi.org/10.1007/s11704-014-3501-3.

[45] www.top500.org, *Top500 list - november 2015* (2015), URL http://www.top500.org/list/2015/11/.

[46] www.green500.org, *The green500 list* (2015), URL http://www.green500.org/.

[47] www.green500.org, *The green500 list - november 2015* (2015), URL http://www.green500.org/news/green500-list-november-2015.

[48] L. Dagum and R. Menon, Computational Science & Engineering, IEEE **5**, 46 (1998).

[49] J. E. Stone, D. Gohara, and G. Shi, IEEE Des. Test **12**, 66 (2010), ISSN 0740-7475, URL http://dx.doi.org/10.1109/MCSE.2010.69.

[50] *Openacc home*, URL http://www.openacc.org.

[51] J. Tholence and E. Wassermann, Physica B+C **86**, 875 (1977), ISSN 0378-4363, URL http://www.sciencedirect.com/science/article/pii/0378436377907240.

[52] E. F. Wassermann and J. L. Tholence, AIP Conference Proceedings **29**, 237 (1976), URL http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.30603.

[53] R. W. Kline, A. M. de Graaf, L. E. Wenger, and P. H. Keesom, AIP Conference Proceedings **29**, 169 (1976), URL http://scitation.aip.org/content/aip/proceeding/aipcp/10.1063/1.30565.

[54] O. S. Lutes and J. L. Schmit, Phys. Rev. **134**, A676 (1964), URL http://link.aps.org/doi/10.1103/PhysRev.134.A676.

[55] J. Tholence, Solid State Communications **35**, 113 (1980), ISSN 0038-1098, URL http://www.sciencedirect.com/science/article/pii/0038109880902252.

[56] J. Tholence and R. Tournier, Journal de Physique Colloques **35**, C4 (1974), URL https://hal.archives-ouvertes.fr/jpa-00215633.

[57] F. Holtzberg, J. L. Tholence, and R. Tournier, *Amorphous Magnetism II* (Springer US, Boston, MA, 1977), chap. The Remanent Magnetization of Spin Glasses and the Dipolar Coupling, pp. 155–167, ISBN 978-1-4613-4178-9, URL http://dx.doi.org/10.1007/978-1-4613-4178-9_17.

[58] G. Nieuwenhuys and J. Mydosh, Physica B+C **86**, 880 (1977), ISSN 0378-4363, URL http://www.sciencedirect.com/science/article/pii/0378436377907264.

[59] L. E. Wenger and P. H. Keesom, Phys. Rev. B **13**, 4053 (1976), URL http://link.aps.org/doi/10.1103/PhysRevB.13.4053.

[60] S. Nagata, P. H. Keesom, and H. R. Harrison, Phys. Rev. B **19**, 1633 (1979), URL http://link.aps.org/doi/10.1103/PhysRevB.19.1633.

[61] H. A. Katori and A. Ito, Journal of the Physical Society of Japan **62**, 4488 (1993), http://dx.doi.org/10.1143/JPSJ.62.4488, URL http://dx.doi.org/10.1143/JPSJ.62.4488.

[62] C. A. M. Mulder, A. J. van Duyneveldt, and J. A. Mydosh, Phys. Rev. B **25**, 515 (1982), URL http://link.aps.org/doi/10.1103/PhysRevB.25.515.

[63] F. Holtzberg, T. L. Francavilla, C. Y. Huang, and J. L. Tholence, Journal of Applied Physics **53**, 2229 (1982), URL http://scitation.aip.org/content/aip/journal/jap/53/3/10.1063/1.330780.

[64] R. H. Swendsen and J.-S. Wang, Phys. Rev. Lett. **57**, 2607 (1986).

[65] K. Hukushima and K. Nemoto, Journal of the Physical Society of Japan **65**, 1604 (1996), URL http://jpsj.ipap.jp/link?JPSJ/65/1604/.

[66] E. Marinari and G. Parisi, EPL (Europhysics Letters) **19**, 451 (1992), URL http://stacks.iop.org/0295-5075/19/i=6/a=002.

[67] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Science **284**, 779 (1999), URL http://science.sciencemag.org/content/284/5415/779.

[68] W. Wang, J. Machta, and H. G. Katzgraber, Phys. Rev. E **92**, 063307 (2015), 1508.05647.

[69] W. Janke, Physica A Statistical Mechanics and its Applications **254**, 164 (1998).

[70] Y. Fang, S. Feng, K.-M. Tam, Z. Yun, J. Moreno, J. Ramanujam, and M. Jarrell, Computer Physics Communications **185**, 2467 (2014), ISSN 0010-4655, URL http://www.sciencedirect.com/science/article/pii/S0010465514001854.

[71] S. Monaghan, Journal of VLSI signal processing systems for signal, image and video technology **6**, 139 (1993), ISSN 0922-5773.

[72] A. T. Ogielski and I. Morgenstern, Phys. Rev. Lett. **54**, 928 (1985).

[73] A. T. Ogielski, Phys. Rev. B **32**, 7384 (1985).

[74] A. Cruz, J. Pech, A. Tarancón, P. Téllez, C. Ullod, and C. Ungil, Computer Physics Communications **133**, 165 (2001), ISSN 0010-4655.

[75] J. H. Condon and A. T. Ogielski, Review of Scientific Instruments **56**, 1691 (1985).

[76] M. Taiji, N. Ito, and M. Suzuki, Review of Scientific Instruments **59**, 2483 (1988).

[77] T. Jörg, H. G. Katzgraber, and F. Krzakala, Phys. Rev. Lett. **100**, 197202 (2008).

[78] M. A. Moore, Journal of Physics A: Mathematical and General **38**, L783 (2005).

[79] T. Temesvári, Phys. Rev. B **78**, 220401 (2008).

[80] H. G. Katzgraber, Journal of Physics: Conference Series **95**, 012004 (2008).

[81] M. Sasaki, K. Hukushima, H. Yoshino, and H. Takayama, Journal of Magnetism and Magnetic Materials **310**, 1514 (2007), ISSN 0304-8853, Proceedings of the 17th International Conference on Magnetism.

[82] M. Sasaki, K. Hukushima, H. Yoshino, and H. Takayama, Phys. Rev. Lett. **99**, 137202 (2007).

[83] D. Larson, H. G. Katzgraber, M. A. Moore, and A. P. Young, Phys. Rev. B **87**, 024414 (2013).

[84] H. G. Katzgraber, T. Jörg, F. Krzkala, and A. K. Hartmann, Phys. Rev. B **86**, 184405 (2012).

[85] H. G. Katzgraber, D. Larson, and A. P. Young, Phys. Rev. Lett. **102**, 177205 (2009).

[86] L. Leuzzi, G. Parisi, F. Ricci-Tersenghi, and J. J. Ruiz-Lorenzo, Phys. Rev. Lett. **103**, 267201 (2009).

[87] K. A. Hawick, A. Leist, and D. P. Playne, Int. J. Parallel Prog. **39**, 183 (2011).

[88] B. Block, P. Virnau, and T. Preis, Computer Physics Communications **181**, 1549 (2010), 1007.3726.

[89] T. Preis, P. Virnau, W. Paul, and J. J. Schneider, J. Comput. Phys. **228**, 4468 (2009), ISSN 0021-9991.

[90] M. Weigel, International Journal of Modern Physics C **23**, 1240002 (2012).

[91] M. Weigel, J. Comput. Phys. **231**, 3064 (2012), ISSN 0021-9991.

[92] M. Baity-Jesi, R. A. Baños, A. Cruz, L. A. Fernandez, J. M. Gil-Narvion, A. Gordillo-Guerrero, M. Guidetti, D. Iñiguez, A. Maiorano, F. Mantovani, et al., The European Physical Journal Special Topics **210**, 33 (2012), ISSN 1951-6401, URL http://dx.doi.org/10.1140/epjst/e2012-01636-9.

[93] J. R. L. de Almeida and D. J. Thouless, Journal of Physics A: Mathematical and General **11**, 983 (1978), URL http://stacks.iop.org/0305-4470/11/i=5/a=028.

[94] A. J. Bray and M. A. Moore, Phys. Rev. Lett. **41**, 1068 (1978).

[95] G. Parisi, Phys. Rev. Lett. **43**, 1754 (1979), URL http://link.aps.org/doi/10.1103/PhysRevLett.43.1754.

[96] M. Mézard, G. Parisi, N. Sourlas, G. Toulouse, and M. Virasoro, Phys. Rev. Lett. **52**, 1156 (1984), URL http://link.aps.org/doi/10.1103/PhysRevLett.52.1156.

[97] G. Parisi, eprint arXiv:cond-mat/0205387 (2002), cond-mat/0205387.

[98] M. Talagrand, Annals of Mathematics **163**, pp. 221 (2006).

[99] F. Guerra, Communications in Mathematical Physics **233**, 1 (2003).

[100] R. Zorn, H. Herrmann, and C. Rebbi, Computer Physics Communications **23**, 337 (1981).

[101] J. Houdayer, The European Physical Journal B - Condensed Matter and Complex Systems **22**, 479 (2001), ISSN 1434-6028, URL http://dx.doi.org/10.1007/PL00011151.

[102] S. Liang, Phys. Rev. Lett. **69**, 2145 (1992), URL http://link.aps.org/doi/10.1103/PhysRevLett.69.2145.

[103] T. Jörg, Progress of Theoretical Physics Supplement **157**, 349 (2005), http://ptps.oxfordjournals.org/content/157/349.full.pdf+html, URL http://ptps.oxfordjournals.org/content/157/349.abstract.

[104] E. Bittner, A. Nußbaumer, and W. Janke, Phys. Rev. Lett. **101**, 130603 (2008), URL http://link.aps.org/doi/10.1103/PhysRevLett.101.130603.

[105] H. G. Katzgraber, S. Trebst, D. A. Huse, and M. Troyer, Journal of Statistical Mechanics: Theory and Experiment **2006**, P03018 (2006), URL http://stacks.iop.org/1742-5468/2006/i=03/a=P03018.

[106] S. Trebst, M. Troyer, and U. H. E. Hansmann, The Journal of Chemical Physics **124**, 174903 (2006), URL http://scitation.aip.org/content/aip/journal/jcp/124/17/10.1063/1.2186639.

[107] A. Nguyen, N. Satish, J. Chhugani, C. Kim, and P. Dubey, in *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis* (IEEE Computer Society, Washington, DC, USA, 2010), SC '10, pp. 1–13, ISBN 978-1-4244-7559-9.

[108] K. Datta, M. Murphy, V. Volkov, S. Williams, J. Carter, L. Oliker, D. Patterson, J. Shalf, and K. Yelick, in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing* (IEEE Press, Piscataway, NJ, USA, 2008), SC '08, pp. 4:1–4:12, ISBN 978-1-4244-2835-9.

[109] F. Belletti, M. Cotallo, A. Cruz, L. A. Fernández, A. Gordillo, A. Maiorano, F. Mantovani, E. Marinari, V. Martín-Mayor, A. Muñoz-Sudupe, et al., Computer Physics Communications **178**, 208 (2008), 0704.3573.

[110] M. Creutz, L. Jacobs, and C. Rebbi, Phys. Rev. Lett. **42**, 1390 (1979).

[111] NVIDIA, *CUDA CURAND Library*, NVIDIA Corporation, Santa Clara, CA, USA (2010).

[112] J. K. Salmon, M. A. Moraes, R. O. Dror, and D. E. Shaw, in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis* (ACM, New York, NY, USA, 2011), SC '11, pp. 16:1–16:12, ISBN 978-1-4503-0771-0.

[113] M. Manssen, M. Weigel, and A. K. Hartmann, ArXiv e-prints (2012), 1204.6193.

[114] A. M. Ferrenberg and D. P. Landau, Phys. Rev. B **44**, 5081 (1991), URL http://link.aps.org/doi/10.1103/PhysRevB.44.5081.

[115] H. G. Katzgraber, M. Körner, and A. P. Young, Phys. Rev. B, 73, 224432 (2006) **73**, 224432 (2006), arXiv:cond-mat/0602212.

[116] J. Mydosh, *Spin Glasses: An Experimental Introduction* (Taylor & Francis Group, 1993).

[117] H. Diep, *Frustrated Spin Systems* (World Scientific, 2004).

[118] D. Thouless, P. Anderson, and R. Palmer, Philosophical Magazine **35**, 593 (1977).

[119] A. J. Bray and M. A. Moore, Journal of Physics C: Solid State Physics **13**, L469 (1980).

[120] W. L. McMillan, Journal of Physics C: Solid State Physics **17**, 3179 (1984).

[121] A. Bray and M. Moore, in *Heidelberg Colloquium on Glassy Dynamics*, edited by J. Hemmen and I. Morgenstern (Springer Berlin Heidelberg, 1987), vol. 275 of *Lecture Notes in Physics*, pp. 121–153.

[122] N. Hatano and J. E. Gubernatis, Phys. Rev. B **66**, 054437 (2002).

[123] E. Marinari, C. Naitza, F. Zuliani, G. Parisi, M. Picco, and F. Ritort, Phys. Rev. Lett. **81**, 1698 (1998).

[124] E. Marinari, C. Naitza, F. Zuliani, G. Parisi, M. Picco, and F. Ritort, Phys. Rev. Lett. **82**, 5175 (1999).

[125] H. Bokil, A. J. Bray, B. Drossel, and M. A. Moore, Phys. Rev. Lett. **82**, 5174 (1999).

[126] M. A. Moore, H. Bokil, and B. Drossel, Phys. Rev. Lett. **81**, 4252 (1998).

[127] C. Monthus and T. Garel, Phys. Rev. B **88**, 134204 (2013).

[128] G. Hed, A. P. Young, and E. Domany, Phys. Rev. Lett. **92**, 157201 (2004).

[129] P. Contucci, C. Giardinà, C. Giberti, G. Parisi, and C. Vernia, Phys. Rev. Lett. **99**, 057206 (2007).

[130] M. Palassini and A. P. Young, Phys. Rev. Lett. **85**, 3017 (2000).

[131] M. Palassini and A. P. Young, Phys. Rev. Lett. **85**, 3333 (2000).

[132] T. Aspelmeier, M. A. Moore, and A. P. Young, Phys. Rev. Lett. **90**, 127202 (2003).

[133] E. Marinari and G. Parisi, Phys. Rev. Lett. **86**, 3887 (2001).

[134] J. Houdayer and O. C. Martin, Phys. Rev. Lett. **82**, 4934 (1999).

[135] E. Marinari, G. Parisi, and F. Zuliani, Phys. Rev. Lett. **84**, 1056 (2000).

[136] G. Kotliar, P. W. Anderson, and D. L. Stein, Phys. Rev. B **27**, 602 (1983).

[137] H. G. Katzgraber and A. P. Young, Phys. Rev. B **67**, 134410 (2003).

[138] H. G. Katzgraber and A. P. Young, Phys. Rev. B **68**, 224408 (2003).

[139] L. Leuzzi, Journal of Physics A: Mathematical and General **32**, 1417 (1999).

[140] H. G. Katzgraber and A. K. Hartmann, Phys. Rev. Lett. **102**, 037207 (2009).

[141] L. Leuzzi, G. Parisi, F. Ricci-Tersenghi, and J. J. Ruiz-Lorenzo, Phys. Rev. Lett. **101**, 107203 (2008).

[142] M. A. Moore, H. Bokil, and B. Drossel, Phys. Rev. Lett. **81**, 4252 (1998).

[143] G. Migliorini and A. N. Berker, Phys. Rev. B **57**, 426 (1998).

[144] J. C. Ciria, G. Parisi, F. Ritort, and J. J. Ruiz-Lorenzo, Journal de Physique I (France) **3**, 2207 (1993).

[145] Y. Fang, S. Feng, K.-M. Tam, Z. Yun, J. Moreno, J. Ramanujam, and M. Jarrell, ArXiv e-prints (2013), 1311.5582.

[146] 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, et al. (Janus Collaboration), Phys. Rev. B **88**, 224416 (2013).

[147] F. Krzakala, J. Houdayer, E. Marinari, O. C. Martin, and G. Parisi, Phys. Rev. Lett. **87**, 197204 (2001).

[148] M. Hasenbusch, A. Pelissetto, and E. Vicari, Phys. Rev. B **78**, 214205 (2008).

[149] F. Ritort, Phys. Rev. B **50**, 6844 (1994).

[150] L. A. Fernandez, V. Martin-Mayor, G. Parisi, and B. Seoane, EPL (Europhysics Letters) **103**, 67003 (2013).

[151] H. G. Katzgraber and F. Krząkała, Phys. Rev. Lett. **98**, 017201 (2007).

[152] R. A. Baños, A. Cruz, L. A. Fernandez, A. Gordillo-Guerrero, J. M. Gil-Narvion, M. Guidetti, A. Maiorano, F. Mantovani, E. Marinari, V. Martin-Mayor, et al., Journal of Statistical Mechanics: Theory and Experiment **2010**, P05002 (2010).

[153] C. Monthus and T. Garel, Phys. Rev. B **88**, 134204 (2013), URL http://link.aps.org/doi/10.1103/PhysRevB.88.134204.

[154] M. Baity-Jesi, R. A. Baños, A. Cruz, L. A. Fernandez, J. M. Gil-Narvion, A. Gordillo-Guerrero, D. Iniguez, A. Maiorano, F. Mantovani, E. Marinari, et al., ArXiv e-prints (2014), 1403.2622.

[155] X.-L. Qi and S.-C. Zhang, Rev. Mod. Phys. **83**, 1057 (2011), URL http://link.aps.org/doi/10.1103/RevModPhys.83.1057.

[156] L. D. Landau, JETP **3**, 920 (1957).

[157] L. D. Landau, JETP **5**, 101 (1957).

[158] J. M. Luttinger, Journal of Mathematical Physics **4**, 1154 (1963), URL http://scitation.aip.org/content/aip/journal/jmp/4/9/10.1063/1.1704046.

[159] S.-i. Tomonaga, JETP **5**, 544 (1950).

[160] J. A. Hertz, Phys. Rev. B **14**, 1165 (1976), URL http://link.aps.org/doi/10.1103/PhysRevB.14.1165.

[161] N. F. Mott and R. Peierls, Proceedings of the Physical Society **49**, 72 (1937), URL http://stacks.iop.org/0959-5309/49/i=4S/a=308.

[162] T. Maier, M. Jarrell, T. Pruschke, and M. H. Hettler, Rev. Mod. Phys. **77**, 1027 (2005), URL http://link.aps.org/doi/10.1103/RevModPhys.77.1027.

[163] P. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964), URL http://link.aps.org/doi/10.1103/PhysRev.136.B864.

[164] W. Kohn and L. J. Sham, Phys. Rev. **140**, A1133 (1965), URL http://link.aps.org/doi/10.1103/PhysRev.140.A1133.

[165] M. Jarrell, Phys. Rev. Lett. **69**, 168 (1992), URL http://link.aps.org/doi/10.1103/PhysRevLett.69.168.

[166] S. Lundqvist and S. H. M. (eds.), *Theory of the Inhomogeneous Electron Gas* (Plenum Press, New York, 1983).

[167] V. I. Anisimov, A. I. Poteryaev, M. A. Korotin, A. O. Anokhin, and G. Kotliar, Journal of Physics: Condensed Matter **9**, 7359 (1997), URL http://stacks.iop.org/0953-8984/9/i=35/a=010.

[168] J. E. Hirsch and R. M. Fye, Physical Review Letters **56**, 2521 (1986).

[169] A. N. Rubtsov, V. V. Savkin, and A. I. Lichtenstein, Physical Review B **72**, 035122 (2005), cond-mat/0411344.

[170] E. Gull, A. J. Millis, A. I. Lichtenstein, A. N. Rubtsov, M. Troyer, and P. Werner, Rev. Mod. Phys. **83**, 349 (2011), URL http://link.aps.org/doi/10.1103/RevModPhys.83.349.

[171] K. Haule, Phys. Rev. B **75**, 155113 (2007), URL http://link.aps.org/doi/10.1103/PhysRevB.75.155113.

[172] A. M. Läuchli and P. Werner, Phys. Rev. B **80**, 235117 (2009), URL http://link.aps.org/doi/10.1103/PhysRevB.80.235117.

[173] P. Werner and A. J. Millis, Phys. Rev. B **74**, 155107 (2006), URL http://link.aps.org/doi/10.1103/PhysRevB.74.155107.

[174] L. Boehnke, H. Hafermann, M. Ferrero, F. Lechermann, and O. Parcollet, Phys. Rev. B **84**, 075145 (2011), 1104.3215.

[175] K. Fischer and J. Hertz, *Spin Glasses*, Cambridge Studies in Magnetism (Cambridge University Press, 1993), ISBN 9780521447775.

ChapterAppendix A
Notes on Covariance Matrix Spectrum

A.1 Covariance Matrix Spectrum

The covariance matrix is defined as

$$M_{\alpha,\beta} = \frac{1}{N}\sum_i S_i^{(\alpha)} S_i^{(\beta)} - \frac{1}{N}\sum_i S_i^{(\alpha)} \frac{1}{N}\sum_j S_j^{(\beta)}$$

where α, β are replicas of the same disorder configuration, but from different Markov chains.

One can then examine the spectrum use Singular Value Decomposition, since the covariance is always a positive definite matrix. We then calculate the average of the singular values over different disorder configurations. Figure A.1 shows the singular value calculated with this method, using the data we gathered on Shelob.

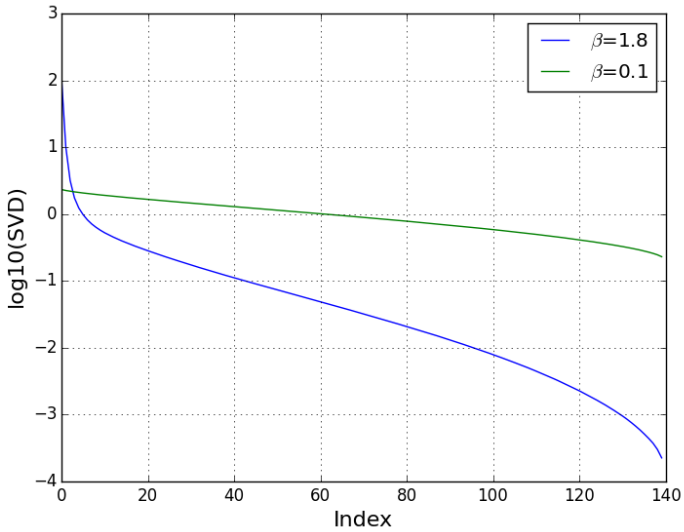


FIGURE A.1. Singular values of covariance matrix for 3D EA model, h=0, L=8, averaged over 1000 disorder realizations. Each matrix is formed from 140 samples of the same disorder realizations.

A.2 Different Phases/Scenarios

Here we discuss a few possible scenarios of this model.

1. High temperature or paramagnetic phase

   The diagonal elements in the covariance matrix would be

   $$M_{\alpha,\alpha} = 1 - (\frac{1}{N}\sum_i s_i^{(\alpha)})^2 \sim 1 - O(N^{-1})$$

   the off diagonal elements would be

   $$M_{\alpha,\beta\neq\alpha} = \frac{1}{N}\sum_i S_i^{(\alpha)} S_i^{(\beta)} - \frac{1}{N}\sum_i S_i^{(\alpha)} \frac{1}{N}\sum_j S_j^{(\beta)} \sim O(N^{-1/2})$$

   In general the matrix is not singular, and the distribution should look like the $\beta = 0.1$ line in Figure A.1.

2. Ferromagnetic phase

   In the ferromagnetic limit, the system has two symmetric degenerate states. All the replicas should then fall into one of these states. The vacuum cancels the contribution from the overlap, and therefore all elements in the matrix would be close to zero.

3. Droplet picture

   There is only a pair of degenerate ground states in the Droplet picture. In this case, the matrix would be composed with only $+1$ and -1, since the vacuum term is zero on average. The matrix is singular and has only 1 finite singular value (Figure A.2).

4. Replica Symmetry Breaking (RSB) picture

   In RSB, there is a tree-like hierarchy in the structure of phase space. We can assume the overlap also follows a tree structure (Figure A.3).

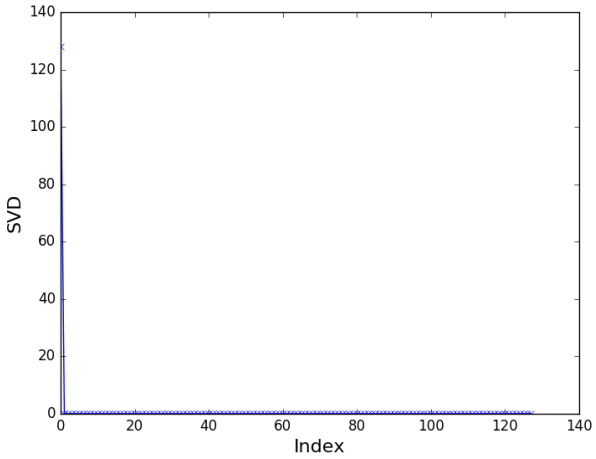


FIGURE A.2. Theoretical distribution for the eigenvalues of the Droplet model, N=128.

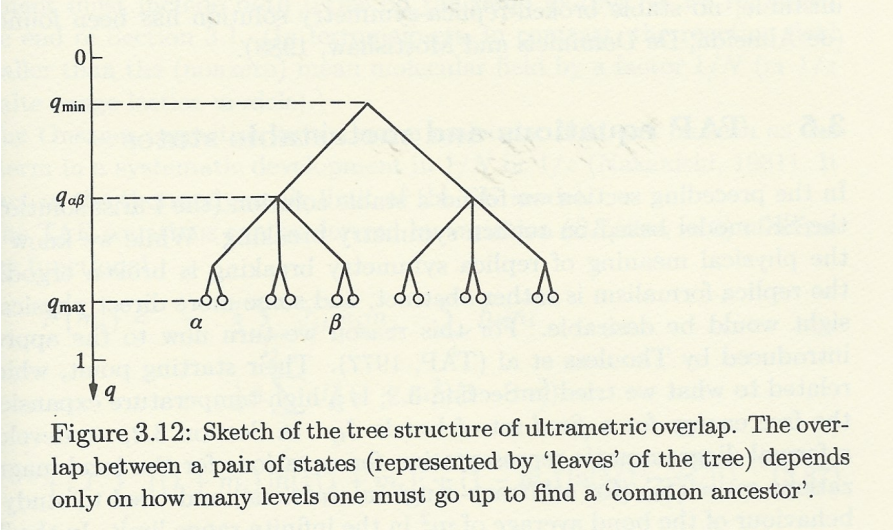


Figure 3.12: Sketch of the tree structure of ultrametric overlap. The overlap between a pair of states (represented by 'leaves' of the tree) depends only on how many levels one must go up to find a 'common ancestor'.

FIGURE A.3. Illustration of the tree structure. From Fischer and Hertz [175], page 93

For simplicity, we used a binary tree to represent the structure of states. In equilibrium, all replicas should fall onto one of the leaves of the binary tree.

The depth of the tree determines the degree of degeneracy of ground state. A larger depth is favorable since RSB predicts an infinite number of degenerate ground states in the thermal dynamic limit. We used 32, which corresponds to 2^{32} ground states. The distribution of q levels shows the similarity of ground states. We start with a power-law distribution. We can then use this

tree structure to investigate the behavior of covariance matrix in the RSB picture.

To mimic the simulation, we generate multiple sets of samples. Each set contains 140 samples. Each sample is a ground state, i.e, a random leaf of the binary tree. We can then find the overlap between each pair of samples from their position on the tree, form the covariance matrix and calculate the singular values. We average over 2000 sets of samples (Figure A.4).

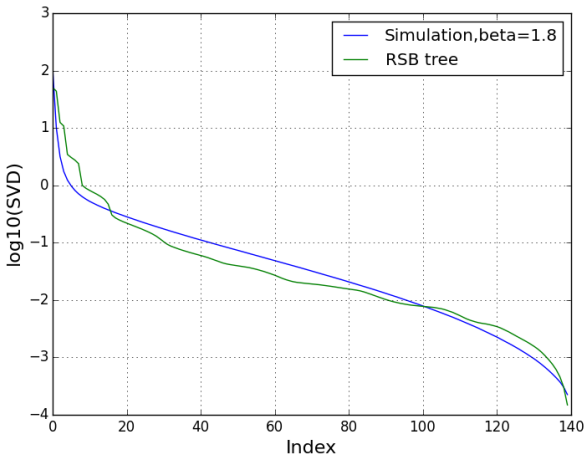


FIGURE A.4. Comparison between the simulation data and the RSB tree test. For the RSB tree, we used a tree depth of 32, $q_{max} = 1.0$, $q_{min} = 0.2$, power-law distributed q levels.

We see some interesting similarities in the distribution between the singular values we generated using this method, and the singular values we calculated from real simulation data.

A.3 Distribution of Eigenvalues

To understand the spectrum better, we look at some of the distribution of a few eigenvalues (Figure A.5 and A.6). Here λ_i stands for the ith largest eigenvalue.

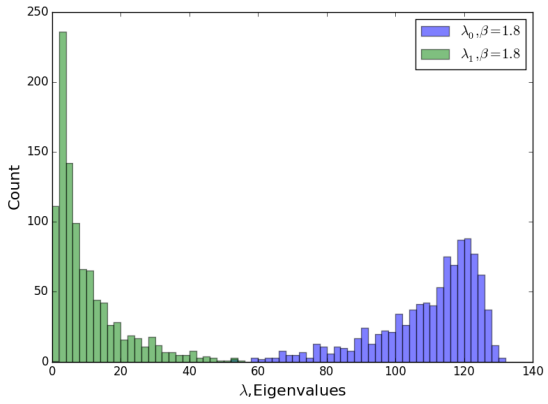
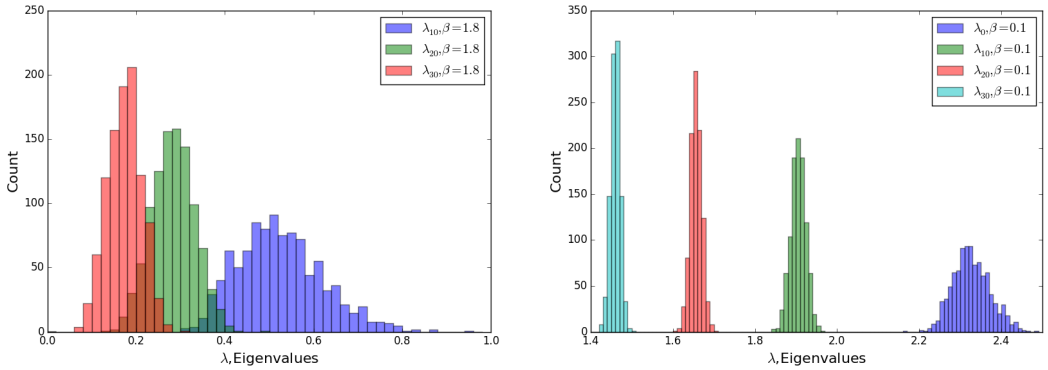


FIGURE A.5. Histogram of the largest two eigenvalues, $\beta = 1.8$. The distribution is not symmetrical and non-Gaussian.



(a) $\beta = 1.8$ (b) $\beta = 0.1$

FIGURE A.6. Histogram of the three eigenvalues, λ_{10},λ_{20}, λ_{30}, for $\beta = 1.8$ and $\beta = 0.1$

From the distribution, we see that the distribution of first few largest eigenvalues at $\beta = 1.8$ (Figure A.5) is not Gaussian. All the other data are almost normally distributed (Figure A.6).

Take λ_1, $\beta = 1.8$ for example. It seems that for a portion of realizations, λ_1 is much larger than the smaller eigenvalues, i.e. there are more than one large eigenvalues. For other realizations λ_1 is on the same order and has similar distribution with the smaller eigenvalues, i.e. there is only one large eigenvalue.

The disorder realizations can be very different from each other. Previously for the droplet model, we only looked at one disorder configuration, and this may not

be enough. We need to do more realizations and examine the distribution/average to see if all the realizations have only 1 large eigenvalue.

The largest eigenvalue of the covariance matrix is the variance in the direction that the data varies the most. The second largest eigenvalue is the the greatest variance among the directions that are orthogonal to the first eigenvector, and so on. So if there is only one large eigenvalue, this means there is only one pair of states that are dominant, since the system only varies in that one direction. In any other direction, the energy penalty to make a move is much higher and thus the variance is much smaller, hence a smaller eigenvalue.

Counting degenerating ground states is possible for some toy systems. For example, for a system of three spins in a triangle that have nearest neighbor interaction. If all interaction are ferromagnetic, then the system is ferromagnetic, and has only two degenerate states. This corresponds to the case with only one dominant eigenvalue.

Consider the same system with all anti-ferromagnetic bonds. Then the system has six degenerate ground states, with a set of four possible values of overlap between each other. The corresponding covariance matrix has two finite eigenvalues. Another example is a system of four spins in a square, with three anti-ferromagnetic bond and one ferromagnetic bond. Then the system has eight degenerate ground states, with a set of nine possible values of overlap between each other. The corresponding covariance matrix has three finite eigenvalues.

A.4 Data from Droplet Model

We ran the model with a weakly correlated random disorder (van Hemmen model), where

$$J_{ij} = \eta_i \zeta_j + \zeta_i \eta_j$$

The distribution of q is shown in Figure A.7. The distribution of eigenvalues is

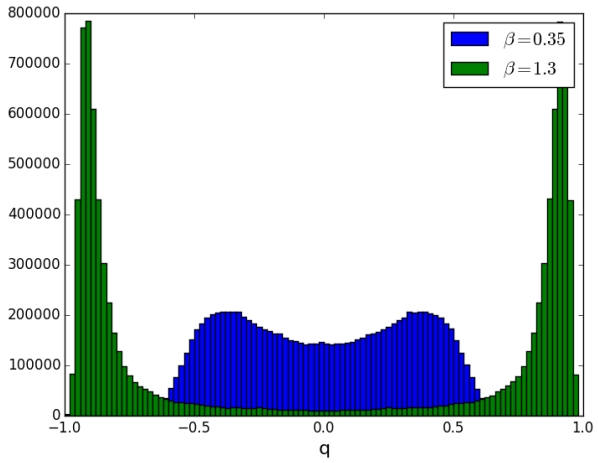
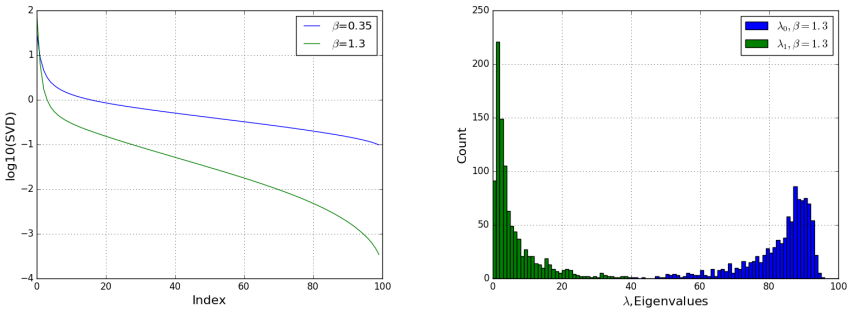


FIGURE A.7. The distribution of q in the droplet model.

shown in Figure A.8(a). The histogram of the largest few eigenvalues is shown in Figure A.8(b).



(a) (b)

FIGURE A.8. Singular values of covariance matrix, and the distribution for 3D droplet model, h=0, L=8, averaged over 1000 disorder realizations. Each matrix is formed from 100 samples of the same disorder realizations.

Comparing the result from Droplet model and EA model, we do not see much difference. Even there is a difference between the Droplet model and EA model, it seems the system size we use is not large enough to show it.

A.5 Data from RSB Model

We use the Sherrington-Kirkpatrick Model here:

$$H = -\frac{1}{\sqrt{N}} \sum_{1 \le i \le j \le N} J_{ij}\sigma_i\sigma_j$$

This model has an equilibrium phase transition at $T_c = 1$. The distribution of eigenvalues is shown in Figure A.9. At low temperature, the data seems very similar

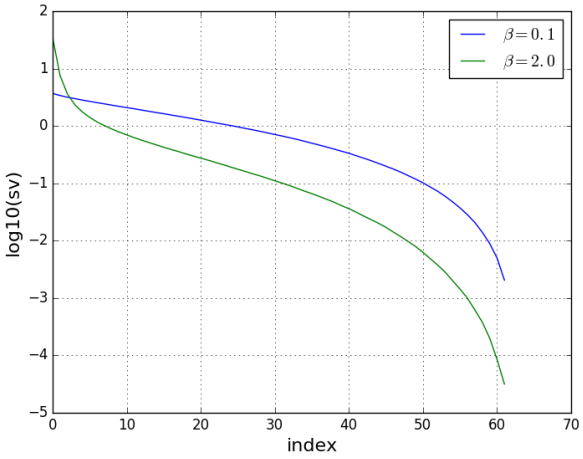


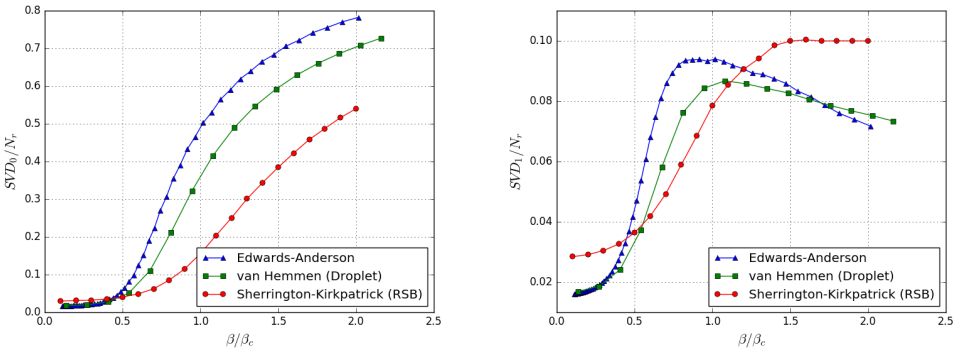
FIGURE A.9. Singular values of covariance matrix for SK model, h=0, N=64, averaged over 280 disorder realizations. Each matrix is formed from 64 samples of the same disorder realizations.

to that of EA model and Droplet model.

A.6 Comparison Among Three Models

We further inspect the temperature dependency of the eigenvalues for different models,by plotting the (second largest eigenvalue of the covariance matrix / linear size of the covariance matrix) vs (β/β_c) for three different models, where β_c is the critical beta for each model, as shown in Figure A.10.

At first glance, the second largest eigenvalue reveals a big difference among the three models. The assumption is that the droplet picture has a simple structure in its phase space, thus, all eigenvalues except the largest will converge to zero at zero



(a) Largest (b) Second Largest

FIGURE A.10. Temperature dependency of the largest eigenvalue and second largest eigenvalue, for EA, droplet and RSB model.

temperature, while the RSB picture has a rich structure in the phase space and the second largest eigenvalue will persist. From the figure, it may seem the data for EA model is more droplet-like. They both have a peak near the critical beta and goes down at the lower temperature. However, when going into lower temperatures, we see that the second largest eigenvalue for SK model shows a similar trend, by decreasing at lower temperatures (Figure A.11).

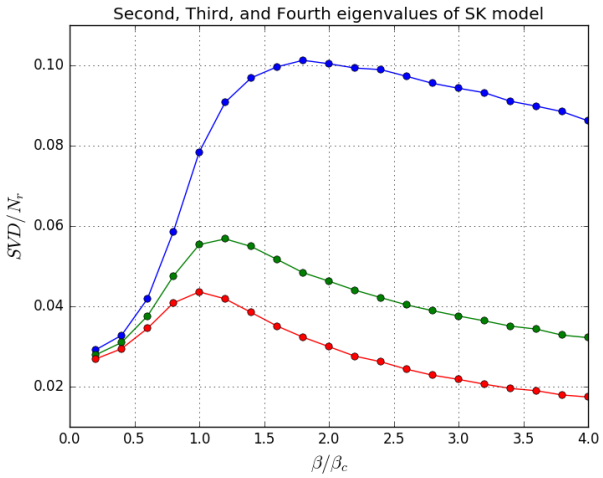


FIGURE A.11. The second, third and fourth largest eigenvalues for RSB model vs β/β_c.

In conclusion, the data from the study on the covariance is not conclusive to discriminate the models and decide the nature of the spin glass phase.

Vita

Sheng Feng is born and raised in Wuxi, China. He went to University of Science and Technology of China for his bachelor's degree in physics, and graduated in 2010. After that he has been studying in Louisiana State University, in pursuit of his Ph.D. degree in physics. He is expected to graduate in May 2016.