Coventry University

BSc(Hons) Mathematics and Physics

331MP Final Year Thesis

Monte Carlo Simulation of the Ising Model

Author
Ubaid Khan

Supervisor
Dr Nikolaos Fytas

A Report Presented in the School of Computing, Electronics and Mathematics, Coventry University, for the degree of Bachelor of Science in Mathematics and Physics, April 2021

Contents

Acknowledgements

Throughout the writing of this dissertation, I have received substantial support, guidance and assistance.

Firstly, I would like to thank my dissertation supervisor, **Dr. Nikolaos Fytas**, for being a constant source of guidance and for always pushing me to achieve my best. The meetings and conversations we shared throughout the year were immeasurably helpful and insightful, and I will forever be grateful for the knowledge that you shared. None of this would have been possible without you.

I would also like to thank Dr. Thierry Platini for his enjoyable lectures throughout the course of my degree, and for always being supportive.

I also express my thanks to my close friends and family for their unconditional and unwavering support, patience and help.

Abstract

The aim of this dissertation, *Monte Carlo Simulation of the Ising Model*, is to explore an extremely important and useful area of statistical physics and thermodynamics. In classical mechanics, many questions and problems can be sufficiently answered analytically by the use of equations and calculations. It is normally not overly challenging to find exact solutions for properties of finite-sized systems, and to a great level of detail. However, certain finite systems require a different approach: statistical mechanical methods. To summarise the contents of this dissertation, Monte Carlo methods will be applied to the 2D Ising model, with the use of the Metropolis algorithm. Firstly, a literature review of relevant work will be completed, which will outline the fundamental theory of the subject. The subsequent chapters will provide an expansion on the fundamental concepts, and then give an analysis of some actual simulations completed through MATLAB. Leading up to the application in MATLAB, the foundations of the most important concepts will be laid.

It is plausible to ask the following questions upon simply reading the title of the thesis: What are Monte Carlo simulations and what is the 2D Ising model? A Monte Carlo simulation is an iterative method used to calculate the expectation value; the probability of a system being in a certain state. It is used to calculate the partition function on a large lattice model (in this case the Ising Model), by using many features, such as probability distributions, the weight (which is the probability of a system being in a particular state μ at time t) and many other equations and features that will be expanded upon within the thesis. The 2D Ising model is a square lattice of sites (or spins) at each vertex, where each site can be in one of two states. Each site is labelled with an index i, the two states being -1 and +1. To say that the i^{th} site is in the state -1, we write $\sigma_i = -1$. Generally, it is said that the "spins" are pointing either up or down. The Ising Model is the most commonly used model in statistical physics, and helps us to visualise properties relating to phase transitions. When the number of particles in a thermodynamic system increases, the number of possible states increases exponentially, so it is helpful that the Ising Model is very simple and easy to understand, and it is also one of the only exactly solvable models, thus allowing the analysis of the properties of system to be executed with great detail.

MATLAB is a programming platform that was specifically designed for the analysis and design of complex systems that transform the world of engineering and science. It is generally used for:

1. Analysing data

2. Developing algorithms

3. Creating models and applications

These primary purposes perfectly fit the criteria for the investigation in this thesis, and by the end I aim to combine a wide array of relevant knowledge in order to execute effective Monte Carlo simulations in MATLAB, in the hoped of producing legible and conclusive results.

1 Introduction

1.1 Motivations and Preliminary Remarks

The use of statistical physics has given modern physicists a deeper and broader understanding of a wide range of scientific concepts. Statistical methods have provided a ‘bridge’ between the gap in the microscopic level (typically unobservable properties at an internal molecular level) and the macroscopic level (e.g., easily observed properties such as volume and pressure) of thermodynamic systems. The focus throughout this thesis will be on phase transitions, with various statistical mechanical methods and derivations being explored. Furthermore, through the use of the MATLAB programming language, the distribution of particles at different energy levels as functions of temperature and system size will be observed, with the equilibrium state configurations being considered and analysed. Various questions will be addressed, such as “What effect does the temperature/system size have on the Energy and Magnetisation of the lattice in the Monte Carlo simulation?”, “Why can we not use direct analytical methods to perform calculations instead?”, and “How can our findings be represented in graphs?”, as well as many more theoretical thought-provoking questions. The Ising model and its relevance to magnetism (and ferro-magnetism) and its purpose will be discussed as well as with relation to Monte Carlo calculations. A variety of links between the theoretical statistical mechanics and the real-world applications will be established, in order to answer a wide range of questions on the subject. By the end of the thesis, we hope to reach solid conclusions from the simulations and calculations, and also to provide factual and detailed information and answers about the topic of Monte Carlo simulations of the Ising Model and statistical mechanics (and thermodynamics) in general.

1.2 Outline of Structure

The dissertation has been divided into four main chapters, each covering a particular component to the larger investigation. The first chapter is to provide preliminary concepts, and to introduce the thesis and to provide context and background information. In the second chapter, a literature review of related and similar published works will be completed, alongside a detailed analysis in relation to the thesis. This will be done to a high standard to show my understanding of the subject and to demonstrate my knowledge. Following this in the third chapter, foundational knowledge and the theoretical approach towards the problem will be addressed, demonstrated with viable explanations and mathematical information. Within this chapter the Ising Model and Monte Carlo simulations will be explained, as well as various equations for the probability distributions, in order to provide the base knowledge required for the application to programming. The final main chapter is focused on the programming and technical approach. Various codes and results will be published and explained, with step-by-step explanations of lines of code (given as MATLAB comments). To conclude the report, final judgements and findings will be published, with an aim of providing sufficient clarification and summarisation of the various chapters, combined with recommendations for improvements and further work that can be done.

2 Literature Review

2.1 : Motivations and aims for this chapter

In this chapter of the thesis, a literature review will be completed. The following chapters have been selected (by myself and Dr. Fytas) for review: G. T. Barkema, M. E. J. Newman, Monte Carlo Methods in Statistical Physics (Chapters 1-4), Oxford University Press, Oxford, 1999. The purpose of this literature review is to establish familiarity of the subject (Monte Carlo simulations of the Ising Model) and to demonstrate a solid understanding of current and previous research in this particular field. In general, the main purposes of a literature review (within a thesis) are to summarise previous research, highlight any clashing or opposing theories and to identify any considerable gaps that may be present in the field of interest. The literature was provided by the dissertation supervisor Nikolaos Fytas, who strongly recommended and advised its study before starting the report. The main aim of this chapter is to demonstrate suitable knowledge and understanding of Monte Carlo simulations and statistical physics in general, and to provide strong foundations and fundamental theory for the remainder of the thesis.

2.2 : Literature review of Barkema, Chapters 1-4

2.2.1 - Chapter 1: Introduction

In the first chapter, the book is introduced by stating that the focus will be on Monte Carlo methods (for solving problems in statistical physics). This is an introductory chapter for the book, so it is expected to lay foundations for statistical mechanics and provide the most important equations along with some derivations. The first section of this chapter ("1.1 Statistical mechanics") explains the reasons and benefits of utilising statistical methods rather than explicit calculations. It provides the reader with a key visualisation of a condensed matter system to demonstrate that "it is not feasible to solve Hamilton's equations for these systems because there are simply too many equations". (An expansion of the Hamiltonian and its principles will be expanded upon in the next chapter of the thesis). The section shows the need to use probabilistic statements and averages, as they are accurate enough. After providing some additional theory and examples, the "master equation" and the mathematical approach is introduced. One important value that is given is the weight, $w_\mu(t)$, which is representative of the probability that a particle is in a certain state (in this case μ) at a time t. Additionally, the master equation (equation 1.1) is given, which makes this a strong introductory section for the book. The equations that are initially seemingly complicated are dissected to show that a knowledge of some basic principles of statistics is required to understand them. Section 1.1 then goes on to discuss the nature and importance of expectation values for the quantities in the systems and envisaging them as "a time average of the quantity", instead of making single instantaneous measurements.

The next section of the first chapter is titled: "1.2 Equilibrium". Here, Barkema *et al.* discuss the relevance of a system reaching a state of equilibrium. They describe the equilibrium of a system to be when the weights (the probability of a particle/system being in a certain state) take constant values for the rest of time, i.e., the system is fixed in one state with no possibility of a phase transition. It is also noted that all systems that are governed by the probabilistic equations (given in the previous section) will inevitably reach equilibrium in the end. Proceeding this, additional equations/functions are defined in order to better demonstrate the concept and principles of equilibrium. One such function is the partition function Z, which is a crucial component in statistical mechanics and "can tell us virtually everything we might want to know about the macroscopic behaviour of the system". It is also said that the probability distribution is a Boltzmann distribution. Towards the end of this section, more equations and derivations are given to provide a better understanding of the relevance of the variables.

After this, a subsection "1.2.1 Fluctuations, correlations and responses" is commenced, which provides a detailed summary (both mathematical and explanatory) of fluctuations (etc.) and the effects that they have on the calculations. The partition function is manipulated and an equation for the mean square deviation is given in equations 1.16 to 1.19. The results are further developed to establish that in the limit of a very large system; which is referred to as the thermodynamic limit; the fluctuations can be ignored. Later on in the subsection, the linear response theorem and correlation functions are defined and explained, which deal with susceptibility and the measure of correlation between simultaneous fluctuations on multiple sites.

Another very important subsection is defined as "1.2.2 An example: the Ising model", which specifically introduces the Ising model (the most thoroughly researched model in the entirety of statistical physics) and applies it to the thermodynamic system that has been discussed up to this point. The Ising model is one that postulates a simple magnetic (often cubic) lattice, which uses the fact that the magnetisation of a (bulk) material is composed of a combination of the magnetic dipole moments of many atomic "spins" within it. The model can be viewed as a cubic lattice with a spin (a magnetic dipole moment) on each vertex, which are referred to as "sites". The Ising model is highly regarded due to its simplicity and effectiveness, most notably due to the model only taking the values ±1, which represents either a so-called "up" spin or a "down" spin. These spin states are made more realistic by developing and including the terms from the Hamiltonian (discussed in the first section). Additional equations are constructed towards the end of this subsection, most notably a Hamiltonian function to include an external magnetic field B, and J which represents the interactions on each site (i.e., an interaction parameter); see equations (1.30) and (1.38).

In "1.3 Numerical methods", Barkema et al. investigate a numerical approach in the calculation of the partition function. It is also given that the thermodynamic limit and such computations have been completed in two-dimensional models, however in three dimensions results are yet to be found. An ingenious method is given in an example of calculations on a 5x5 two-dimensional square lattice system, where the number of spin states that are needed to be considered has been essentially cut in half, by using the simple logic of spin symmetry. As we have an infinite number of terms from the partition function, it is seemingly wasteful to perform such calculations on finite-sized lattices, however by looking at the results from the two-dimensional case, a non-analytic critical point is defined as kT=2.3J (this critical temperature will have an important feature on the MATLAB code in chapter 4), and by moving the temperature away from this critical point on smaller lattices returns more accurate results. The section is ended by stating that these numerical methods for the calculations take an incredibly long time, even for small lattices such as the 5x5 example (8 hours on a moderately powerful computer), and as the system size increases, the processing time increases exponentially. This introduces the following subsection, "1.3.1 Monte Carlo simulation".

The brief subsection of 1.3.1 provides an introduction to Monte Carlo simulations. It is started by stating that it is the only known method for the calculation of the partition function on a large lattice (such as the Ising model). "The basic idea behind Monte Carlo simulation is to simulate the random thermal fluctuation of the system from state to state over the course of an experiment". The system is simulated to create a theoretical Ising model (lattice system) and perform phase transitions in a way the weight $w_\mu(t)$ (from section 1.1) is equivalent to that of a real-world system. The remainder of this subsection gives some benefits of the Mone Carlo simulation, e.g., the fact that it is not necessary to sample a large number of states of the system in order to achieve accurate estimations.

The final section of chapter 1 is "1.4 A brief history of the Monte Carlo method". This section is relatively unimportant for knowledge on the calculations; however, it is still useful to know about the origins of the Monte Carlo method and the logic behind it. A few graphical representations are given, as well as some mathematical justification. One interesting fact given in this chapter is that the (somewhat unfortunate) reason that the Monte Carlo method gained popularity amongst the physics community was due to the Second World War, for the construction of the Hydrogen bomb. Various physicists have been mentioned and credited throughout this section, in order to properly contextualise the time-frame and motivations behind the development of Monte Carlo. The review of this section has been kept short, as the section's purpose is just to provide some contextual knowledge to the reader.

In summary of chapter 1, it is sufficient to say that a suitable amount of background knowledge has been provided and explored, through the use of mathematical expressions as well as pure logic and explicit explanation. One possible gap that may be considered is three-dimensional lattices and some propositions on how to approach them, as this area was not developed here. However, the first chapter has given the reader a solid foundation to continue through the rest of the book, and to progress on to the technically advanced approach to the problem. Barkema *et al.* has given a clear and concise introduction to an intricate and thought-provoking area of physics and thermodynamics.

2.2.2 - Chapter 2: The principles of equilibrium thermal Monte Carlo simulation

The purpose of this chapter is to provide more detail about the ideas behind Monte Carlo simulations and the principles of thermal equilibrium as previously discussed. This chapter is crucial in order to understand the algorithms that will be explored later in the book. The separate sections within this chapter are kept short, as they provide compact explanations on the various components of Monte Carlo. Chapter 2 is initiated with "2.1 The estimator", which looks at the estimator Q_M of a quantity Q. Equation (2.2) shows the reader that as the sample size M increases, the accuracy of the estimation also increases; $\lim_{M\to\infty} Q_M = \langle Q\rangle$. However, it is unrealistic to select an infinite sample size for a simulation, so one must choose a suitable M to give an accurate estimation for $\langle Q\rangle$. A useful example is provided: a small $10 \times 10 \times 10$ cubic Ising system would have $2^{1000} \simeq 10^{300}$ states, and a few hours on a computer with a strong processor would sample $\sim 10^8$. This leaves Q_M as a very inaccurate estimate of $\langle Q\rangle$. This example provides a good transition into the next section.

The goal of the previous example was to demonstrate to the reader that there must be a method for one to obtain the states that give the most important contributions to the estimation, and this is discussed in "2.2 Importance Sampling". It is given that a litre of gas (at room temperature and atmospheric pressure) would take about $10^{10^{23}}$ times the lifetime of the universe to pass through every possible state, and this is obviously impossible to perform calculations on. The most common form of importance sampling is by using the Boltzmann probability distribution (as the system is sampling the states according to this, rather than sampling all states with equal probability), and taking a sample of the states that is proportional to its Boltzmann weight. A new equation for the estimator is given, Equation (2.4):

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

; this is a considerably more simple equation than the ones given previously (notably Eq(2.3)). This is followed by subsection “2.2.1 Markov processes”, which deals with how the states are chosen to match with the correct Boltzmann probability; Markov processes act as the generating engine for the set of states that are used. A Markov process works in a random fashion (such that the same new state will not be generated each time the initial state μ is given), and the transition probability $P(\mu \to \nu)$ is the probability of the Markov process generating the state ν when given μ. Furthermore, the transition probabilities satisfy the conditions that they do not vary over time, and that they are dependant only on the properties of the current states (i.e., only on μ, ν). As a Monte Carlo simulation is repetitive, Markov processes are repeated to obtain Markov chains of states, and eventually the generated states reach the Boltzmann distribution (also referred to as reaching equilibrium). Two additional conditions are assigned to the Markov process: Ergodicity (explored in subsection “2.2.2 Ergodicity”) and Detailed Balance (which is analysed in the subsequent “2.2.3 Detailed Balance”). Ergodicity is a condition that requires the possibility for the Markov process to be able to reach any state of the system from any other state, i.e., there must be at least one path of non-zero transition probabilities between any two states. Detailed Balance is the condition that verifies the Boltzmann distribution is being reached, rather than another distribution. Barkema et al. provide multiple equations and proofs to support this condition.

The penultimate section of the second chapter is “2.3 Acceptance ratios”; a small section that looks at the flexibility of the choice of algorithm, the selection probability and various equation and quantities to support the explanations of such theories. Equation (2.16) $P(\mu \to \nu) = g(\mu \to \nu)A(\mu \to \nu)$ shows that the transition probability can be separated into two parts: the selection probability (g) and the acceptance ratio/probability (A). An acceptance ratio of zero would result in the system never leaving the state μ (which would not be done in practice), and a ratio value of 1 would result in the opposite. This gives the freedom to choose any value (between zero and one) for the acceptance ratio, to suit the needs of the calculation. To summarise this, $g(\mu \to \nu)$ is the probability set that is used to generate random new states, and $A(\mu \to \nu)$ is used to determine whether the new state is accepted or rejected.

Chapter 2 is concluded with “2.4 Continuous time Monte Carlo”. Here, the BKL algorithm is expanded upon, which gives the Markov process additional freedom in the way that the states are chosen. A useful example is provided: considering a system at a low temperature, one can assume that the movement within the system is very slow. The case could very easily arise that the system remains in the ground state for many consecutive time steps, and even move up to the first excited state for one time step and then drop back down, and this is clearly a waste of time and computer processing power, as no useful results would be generated. The BKL algorithm would deal with a scenario of this nature with the following steps:

- $P(\mu \to \nu)$ is calculated. These are the probabilities of all the states that μ can transition into within one Monte Carlo step. A new state ν, with probability proportional to $P(\mu \to \nu)$ and the state of the system is changed accordingly.
- The time interval Δt is calculated by using P from above. The time interval must be calculated at each step, as it will generally change between steps.

- The time t is incremented by Δt, as this imitates Δt Monte Carlo steps. t is used to keep count of the equivalent Monte Carlo steps that the simulation has gone through

However, the first step presents an obvious drawback in that for larger systems, the calculation for the accessibility to all states from the initial state (μ) may take a long time. Nevertheless, it is still a lot more efficient to perform the calculations with this method rather than without, because the set of transition probabilities generally does not change a lot between steps.

To briefly conclude chapter 2, one can observe the efficiency and effectiveness of the discussed principles and methods. With each drawback from a method, a new one was introduced to counteract the problem. This chapter has provided the required tools for a deeper understanding of Monte Carlo. After reading and contemplating on the first two chapters of the book, the reader would possess sufficient knowledge to progress on to the programming application of Monte Carlo methods and the Ising Model.

2.2.3 - Chapter 3: The Ising Model and the Metropolis Algorithm

In this third chapter of the book, Barkema et al. present the Metropolis algorithm, and links into the ideas (about the Ising model) discussed in the first chapter. Many crucial areas of the problem are addressed, such as the metropolis algorithm itself and the logic behind it, the importance and relevance of equilibration, the measurement and presence of errors, as well as many graphical representations and mathematical derivations. It is noted that this chapter is very lengthy in comparison to the preceding chapters, which is somewhat symbolic of the importance and technicalities within this chapter.

The first section of this chapter is introduced by stating that the Metropolis algorithm is the most famous and widely used Monte Carlo algorithm, created by Nicolas Metropolis et al. in 1953. The algorithm is used to illustrate concepts such as equilibration, the measurement of expectation values, calculating errors. The Metropolis algorithm is derived by following the logic given in "2.3 Acceptance Ratios", whereby $g(\mu \to \nu)$ (a set of selection probabilities satisfying the condition of ergodicity) is chosen such that $\exists$ one possible transition from one state to another (in this case $\mu \to \nu$), along with $A(\mu \to \nu)$ (a set of Acceptance probabilities) such that detailed balance is satisfied (for Equation (2.7)). The algorithm is repeated by choosing a new state (ν) and then randomly either accepting or rejecting it with the use of A (the system is changed to the new state if accepted, or it stays in the original state otherwise), and this process is repeated. The idea of single-spin-flip dynamics is given: as the system spends most of its time in states with a narrow range of energies (and rarely transitions into states that significantly alter the system's energy), it is best to only consider the states that differ from the current one by the "flipping" a singular spin. "*Using single-spin-flip dynamics guarantees that the new state ν will have an energy E_ν differing from the current energy E_μ by at most $2J$ for each bond between the spin we flip and its neighbours*". The lattice coordination number z is also defined as the number of neighbours that each site on the lattice has. Furthermore, single-spin-flip dynamics obey ergodicity (as it is clearly possible to reach any state by individually flipping the spins within a finite lattice).

Chapter 3.2 Equilibration defines the equilibration time τ_{eq}, which is the period of time required to run the simulation until the system has reached equilibrium at the desired temperature. Figure 3.2 shows a 100×100 2D square Ising lattice coming to equilibrium at $T = 2.4$. After this, more graphs and theory of equilibration of systems are provided to a great level of detail.

Chapter 3.3 Measurements addresses the more technical concepts of the subject. Important subsections within this chapter are 3.3.1 Correlation times, and 3.3.2 Correlation times and Markov matrices. Within this chapter, a lot of the important mathematics behind the logic of the algorithm is given, with a multitude of results (and plots). Chapter 3.4 addresses the calculation of errors in the algorithm, most notably exploring the Blocking method, the Bootstrap method and the Jackknife method. The rest of chapter 3 provides more important ideas, e.g., about entropy, and then concludes by performing an actual step-by-step calculation.

The final section of the chapter addresses a very important subject: **3.7.2 Critical fluctuations and critical slowing down**. This shows the main weaknesses of the Metropolis algorithm. The critical region is defined as being the region close to T_c. As the system approaches the critical temperature (from above), large clusters of spins pointing in the same direction (either up or down), and these clusters have a significant contribution on the energy and magnetisation of the system (such fluctuations in E and m are referred to as critical fluctuations). The statistical errors that are measured in quantities such as m and E are proportional to the size of the critical fluctuations, thus increasing as the system approaches T_c. This gives the Metropolis algorithm its inaccuracy around T_c. This perfectly introduces the next chapter of the book, which looks at other algorithms that do not possess this weakness of the Metropolis.

2.2.4 - Chapter 4: Other Algorithms for the Ising Model

Chapter 4 of this book addresses alternative methods for the investigation of the Ising Model. This chapter addresses the weaknesses of the Metropolis algorithm, and proposes an alternative approach that counteracts these issues.

Despite the fact that the Metropolis algorithm is seen as an extremely useful and important algorithm in statistical physics, it is not perfect. The Wolff algorithm has a particular advantage over the Metropolis, as it takes clusters of spins instead of just considering single-spin-flip dynamics, which saves a lot of time. However, this is slightly more complex and harder to comprehend compared to the Metropolis algorithm. (Further details of the Wolff algorithm and its strengths will be adapted Chapter 5 of the dissertation).

This chapter gives a lot of theory surrounding these so-called “cluster” algorithms, with the aid of various equations and graphs, most notable Figure 4.2, demonstrating the logarithmic scale of the increase in the value of z (the dynamic exponent) with the system size.

After the strengths of the Wolff and Swendsen-Wang algorithms were introduced, various other benefits and techniques are addressed in this chapter, however this content falls outside the scope of this dissertation. It is worth noting the weaknesses of the Metropolis algorithm and the presence and benefits of alternative approaches, however a further analysis of these methods is unnecessary here. As aforementioned, a more detailed explanation (relevant to this thesis) is given later, precisely in Section **5.4 Alternative Approaches**.

2.3 Summary of Literature Review

In conclusion of this section, the literature that Barkema *et al.* has produced here is extraordinary. The amount of content covered in such a small space is remarkable. I would consider this book to be essential for all those who are interested in the basics of statistical mechanical methods, and thermodynamics in general. The conciseness of the book makes the subject easy to comprehend for the reader, and within this chapter of the thesis I aimed to provide a strong summary. The fundamental theory has been covered, and the required tools have been given in order to understand the rest of the dissertation.

3 Fundamental Theory

3.1 Introduction and Context

This chapter will be introduced by discussing statistical physics, and the importance and needs of these statistical methods. Prior to the exploration of any theories and ideas, it is important to define a phase transition. Phase transitions are the physical shifts between the fundamental states of matter (solid, liquid, gas, (and plasma)). All elements and substances can experience phase transitions from one state to another, given specific combinations of temperature and pressure are sustained. The most basic example of this phenomenon is a solid reaching a certain temperature (and the environment being of a certain pressure) and melting, producing a liquid. For example, when ice is heated to 0°C (and maintained at standard atmospheric pressure (1 atm)), it will undergo a phase transition and shift from a solid state to a liquid state. Refer to **Fig.1** (Appendices) for a basic diagram of phase transitions. It is worth noting that the phase of a substance/system is dependent on both the temperature and pressure.

The following example is given to aid the visualisation of statistical mechanics and to provide some basic context to the reader. Assume one was to perform calculations on the properties of a given condensed matter system, specifically the changes of states (i.e., the phase transitions), e.g., a room full of gas. One can immediately notice the intricacies and the size of the problem simply by considering the number of atoms in the system. Even though these atoms would typically follow basic equations of motion and the mathematics itself would not be overly challenging, by considering the sheer quantity of atoms one can see that it is virtually impossible to perform these calculations exactly. A typical living room in the UK roughly holds $51m^3$ of gas ($\sim 17.092m^2 \times \sim 3m = 51m^3 = 51000$ litres) [**1**], and for a litre of air $1L = 5.64 \times 1021$ atoms [**2**]. It is clear that the problem we are faced with is enormous and almost impossible to comprehend in the small space of a living room. Furthermore, if the size of the system were to increase (e.g., the whole earth instead of just a room) the problem leaves our scope of understanding. Even if computer programs were utilised, the number of equations of the Hamiltonian system would take far too long to compute. Statistical physics treats the properties of these enormous systems in a probabilistic manner. Mostly, one would take a sample size of the system that they want to measure and extrapolate the results to fit the larger scale. This works because the equations of motion that the particles follow are relatively simple, and there is not much room for any discrepancies. Although this is not as accurate as taking exact measurements, it is accurate enough to provide conclusive results, as well as giving us a realistic way to perform calculations.

3.2 Further Developments and Logic

Perhaps the most practical and understandable way to imagine the systems (from a theoretical stance) would be through a Hamiltonian function (H). One can use the Hamiltonian to determine the total energy of a system at any given state. However, the sole use of a Hamiltonian system gives us the problem of the system constantly remaining in one energy state without many notable changes at all; due to the conservation of energy within Hamiltonian systems. In order to counteract this issue and to progress with our calculations, we must introduce an additional external system: a thermal reservoir. This reservoir maintains the constant exchange of energy with the Hamiltonian system and as a result is always working to create differences in the temperature; it can be thought of as a perturbation on the Hamiltonian. These changes in temperature regularly shift the system to different energy levels and allows us to consider the rate of transition for the system. Barkema denotes this transition rate as $R(\mu \to \nu)$ as the rate of transition from a state μ to a state ν. We assume that the transition rate is timeindependent, however if we were to consider the probability of the system being in a particular state, we use $R(\mu \to \nu)dt$. However, with just this transition rate function, we are given quite limited information. The general nature of the transitions is very frequent and rapid (without the consideration of systems with unusually low temperatures), so we only need to wait a short amount of time until the system shifts into another state (there are a very large number of possible states that the system could transition to). Thus, a further component must be introduced: a set of weights $w_\mu(t)$. This set of weights exhibits the probability of the system being in a state at a time t. These weights are essential to our statistical mechanical approach to the problem, and they give us wider knowledge about the system's state. A master equation can be constructed [**3**]:

$$\frac{dw_\mu}{dt} = \sum_\nu [w_\nu(t)R(\nu \to \mu) - w_\mu(t)R(\mu \to \nu)]$$

It is easily noticeable that the first term within the summation represents the system transitioning from μ to ν and the second term vice-versa. It must be noted that the sum rule must be obeyed (as we are working with probabilities, and we must assume that the system cannot be "stateless"):

$$\sum_\mu w_\mu(t) = 1$$

A fundamental component of this investigation is the equilibration of a system. When the rate of change dw_μ/dt "vanishes" (equating to 0), we say the system is in a state of equilibrium, as the weights would take constant values for the rest of time. The weights w_μ are restricted to fall between zero and one, thus prohibiting an exponential growth to the solutions of the master equation. This shows that the systems following these equations must eventually reach a state of equilibrium.

$$p_\mu = \lim_{t\to\infty} w_\mu(t) = \frac{1}{Z}e^{-E_\mu/kT}$$

represents the equilibrium occupation probabilities, which give the equilibrium values for the weights w_μ of the system. Here, Z is the partition function; if we were to know the variation of Z with any parameters affecting the system (e.g., temperature or volume), we have almost every detail of the system's macroscopic behaviour. Also, E_μ is the energy of the given state μ (as given in the previous chapter), and k is the Boltzmann constant.

3.3 Introducing Monte Carlo and the Ising Model

In statistical physics, calculations and demonstrations and performed by considering probabilities. The nature of the problem that we are faced with points towards adopting "Monte Carlo" calculations/simulations. A Monte Carlo simulation is a statistical mathematical technique, used in a wide array of areas. This method was named after the resort town in Monaco (home to world-renowned casinos) and was first introduced by the scientists working on the atomic bomb in the second World War. In the real world, Monte Carlo methods are used to model many physical and hypothetical systems, as well as measuring the risk factors involved in quantitative analyses and various decision-based situations. Monte Carlo simulations are performed by taking a system with an associated uncertainty (in our case a thermodynamic condensed matter system) and performing iterative calculations with a probability distribution. Calculations will be iterated by taking a different set of random values (within the range of the probability distribution) each time, eventually resulting in a distribution of possible outcome values. Due to the probabilistic nature of the phase transitions and interactions between the particles, the use of Monte Carlo simulations is seemingly the most efficient and viable way to approach the problem. As established earlier, trying to calculate the problem exactly is almost impossible and is impractical due to the high number and random nature of the interactions of particles occurring each second. Using a Monte Carlo simulations presents many benefits, most notably the ability to estimate and account for the standard error, as even the sample values themselves can be used to roughly obtain an idea of the error. Monte Carlo simulations typically iterate for large values of n (sample size), thus minimising the error and producing more accurate results. The error and the factors that affect it will be later analysed in chapter 4, displayed on the Figures from the MATLAB simulations. Furthermore, Monte Carlo simulations provide legible and easily understandable results (i.e., they are generally easily understood by non-mathematicians and non-physicists) and are also very flexible and easy to develop (as required by the user). However, Monte Carlo simulations are not perfect, as they also present some drawbacks. The most obvious one would be the fact that a larger sample (number of required repeated runs/iterations) is needed for higher accuracy, and this requires more processing power and computer time. The fact that Monte Carlo simulations analyse probability distributions means that the outputs are never exact (although the highly accurate outputs are normally sufficient, nothing would be more accurate than analytical calculations).

However, it is also useful to be able to visualise the problem and to think about it in a more practical sense. There are a multitude of commonly used “models” to illustrate physical problems. The most widely used and convenient model is the Ising Model, and this will be the focus throughout the dissertation. The Ising model is the simplest way to represent a thermodynamic system undergoing phase transitions (of the first order). It is a lattice model that was first proposed to aid the explanation of ferromagnetism in materials, in 1925 **[4]**. The model uses the property of “spin” (spins carry magnetic moments and are present in elementary particles). **[5]** A basic example of a lattice for the Ising model would be a that of a cubic lattice with atoms at each vertex. Each atom has a spin (a finite magnetic moment) which points in a certain direction. This can be mathematically represented with the following Hamiltonian:

$$H = -\sum_{i,j} J_{ij}\sigma_i\sigma_j + h\sum_i \sigma_i$$

As previously stated, the Hamiltonian describes the total energy of the system. σ represents the spins (can either be equal to $\sigma = -1$ or $\sigma = +1$), J_{ij} is representative of the interaction between spins on the sites i and j (i.e., at each vertex) and h is the external applied magnetic field.

An important note is that the Ising model has the property of a “critical temperature”, T_c, which was found exactly with the aid of the Kramers-Wannier duality, **[7]**. This value was found to be $T_c = 2.2692J$, or as used in the MATLAB code in the next section, $2/log(1+\sqrt{2})$. At this temperature, the model exhibits a second order phase transition (and for a first order transition, an external field must be added).

3.4 Building up to the Metropolis Algorithm

In order to acquire all of the necessary tools to progress with the investigation and apply the Metropolis algorithm, a selection of necessary conditions must be upheld. These such conditions are named “ergodicity” and “detailed balance”. However, prior to explaining these conditions, the following pieces of Monte Carlo must bbe established:

- Markov processes: This is the fundamental logic behind the iterations in the Monte Carlo simulation, and they are reliant on the Markov chain as the generating engine from one state to the next. The Markov process is the mechanism that generates the new state ν from μ, where the transition probability is $P(\mu \to \nu)$.

- Importance Sampling: used to obtain the states that give the most important contributions the estimation. This prevents a waste of computing time from the consideration of arbitrary contributions. To achieve this, we sample the states according to the Boltzmann probability distribution, rather than sampling all states with equal probability.

- Acceptance Ratios: the set of probabilities used in the simulation to generate a random new state: $P(\mu \to \nu) = g(\mu \to \nu)A(\mu \to \nu)$ is used to determine whether the new state is either accepted or rejected. If it is accepted, the system is now in the new state, and if it is rejected the system stays in the initial state and performs the calculations again – thus giving an iterative nature to Monte Carlo. g represents the selection probability, being the probability of the algorithm generating a new state ν from the initial state μ. A is the acceptance probability, stating that the system should be changed to ν for a fraction of time, and μ otherwise; the choice of this acceptance probability is free for the user (between zero and one). Choosing it as zero would fix the system in one state, never leaving μ (however this is not an ideal situation).

The Markov chains discussed above must follow two particular requirements: ergodicity and detailed balance. Ergodicity requires the Markov process to be able to reach any state in the system from any other state (if the simulation is run for a sufficient amount of time). This condition is a necessity in reaching the Boltzmann distribution for the generation of states. Detailed balance states that after the system has come to equilibrium, it must be the Boltzmann distribution that is reached, and not another one.

Furthermore, a mathematical representation of equilibration is given:

$$p_\mu = \sum_\nu p_\nu P(\nu \to \mu)$$

which, when summarised, means that the rate at which the system performs transitions into and out of any state μ must be equal.

4 Implementation into MATLAB

4.1 Introduction and Foundation of the Metropolis Algorithm

The Metropolis algorithm (first introduced by Metropolis et al., 1953) is generally the most widely used Monte Carlo algorithm. The Metropolis algorithm can be utilised to effectively exemplify many of the general concepts involved in a real Monte Carlo calculation. Examples of the most notable concepts are equilibrium, expectation values and the calculations for the presence of errors. In the following MATLAB code, iterative procedures are applied to represent final configurations and averages of magnetisation of energy. The Metropolis Algorithm generates a new configuration by taking the previous one and computing a transition probability, which is dependent on the fluctuation in energy, ΔE. However, the new configuration must satisfy a (chosen) acceptance probability in order to progress. Depending on ΔE, the new configuration would either overwrite the previous one and be considered as the "current" configuration, or it would be discarded and the system would remain unchanged, thus repeating the algorithm, i.e., a Monte Carlo simulation.

However, the Metropolis algorithm is not a perfect one, and comes with a unique set of drawbacks. One of the the main weaknesses of the algorithm would be the analysis of properties of the system at temperatures approaching the critical (T_c). Around this temperature, large critical fluctuations have a noticeable effect on the results, and the errors within the quantities increase massively. Furthermore, there is nothing that can be done to counteract the critical fluctuations, as they are a property of the model near the phase transition. (Propositions of alternative methods were briefly discussed in Chapter 2, and will be expanded upon in Chapter 5).

Finally, before starting the coding process, it is worth briefly discussing the finite-size scaling method: used to extract values for critical exponents (in this case the dynamic exponent z). Such extractions are achieved through the observation of the variation of certain quantities as the system size changes. This method features heavily in the Ising model by measuring τ, the correlation time for a multitude of different system sizes. (Refer to section **5.4** for further development of this idea).

4.2 Code Construction

Prior to writing any code, it is always worth exploring the motivations behind the code and what we require the code to do. In this thesis, we are presented with a multitude of questions, each requiring a unique and robust approach, with relevant application into programming. The programming application has been divided into three parts, with the objective of fully exploring Monte Carlo simulations of the Ising model and to show a wide range of results and conclusions.

4.2.1 Process before simulations

After reading relevant material (recommended by the project supervisor) and understanding the fundamental concepts, I was provided with the foundations of the MATLAB code. Dr. Fytas provided the following four codes: "ising.m", "isingenergy.m", "isingplot.m", and "metropolis.m". Upon receiving these codes, I was tasked with running and understanding them, along with changing certain lines of code and variables in order to fit the requirements of the investigation within the thesis. In the following weeks, I developed the four foundational codes to an adequate standard, and I created various MATLAB codes from scratch, in order to compute the averages and standard error of the data, and to plot and properly represent the findings of the simulations. These new .m files are given in the Appendices: Listing 5, Listing 6, Listing 7. Furthermore, the codes have been well-commented to show the purpose of particular lines and sections.

4.2.2 Structure and Hypotheses of Simulations

"Part (a)" focuses on the properties of the system size in relation to the critical temperature. The sample size will remain fixed for this part of the simulations, i.e., "*n_samples=1*", as this is not one of the dependant factors for this part. However, the system size and critical temperature shall be adjusted: for each system size N=24, N=48, N=64 and N=96, typical spin configurations will be produced for temperatures ranging across the critical temperature. Three different temperatures will be considered for each system size, $kT = 1$ (a temperature well below the critical temp.), kT=2.31 (very close to the critical temp.) and kT=4 (well above the crit. temp.). This will provide results for the investigation on how the changes in system size and temperature effect the displayed spin configurations produced by the Monte Carlo simulation. For low temperatures, we expect to see very little movement as the system has less energy (as discussed in the previous two chapters of the thesis), so the spin configurations would be mostly uniform (e.g., one solid colour). When the system is close to the critical temperature, we would expect to see clusters of spins together, due to the nature of the Ising model itself. When the system has a higher temperature, a somewhat random assortment of spins is expected, due to the system having a lot of energy, and therefore lots of movement (phase transitions).

The succeeding part of the programming section of the thesis ("part (b)") explores the changes in Energy and Magnetisation of the system as a function of the temperature. The system size shall be fixed to N=64 and the temperature of the system will be incremented. The first temperature will be kT=1.5, and the temperature will be incremented in steps of 0.05 ($\Delta t = 0.05$), until kT=3. This will produce 31 plot points to be represented in a graph, which will be sufficient to properly display the results. Furthermore, the sample size will be fixed at "$n\_samples = 30$", which will give 30 different spin configurations, and thus 30 different values for the Energy and Magnetisation at each temperature step, which are to be saved in separate text files. From these values, the average and standard error (for the error bars on the graph) will be deduced, and this process will be repeated for each increment in temperature. Finally, a plot for the Energy will be produced, as well as a plot for the Magnetisation. By examining the equations and logic given in chapters 2 and 3 of this thesis, one would expect to observe an increase in energy with an increase in temperature, and the opposite for the magnetisation. These observations are expected because generally when a system (or particle) is subject to higher temperatures, its energy increases, resulting in faster movements and a higher transition rate. The magnetisation is expected to follow an opposing trend.

Finally, in "part (c)", the variation in Energy and Magnetisation as a function of the system size will be investigated. This will be done with a similar method to part (b) (with the calculations of averages and standard deviations (error) for the plots). The temperature will be fixed to the critical temperature: $kT = 2/log(1 + \sqrt{2})$, and the sample size of the simulation will be set to 100. Following these conditions, the Monte Carlo simulations will be performed with the following set of system sizes: N= 16,24,32,48,64,96, as this gives a wide range of results to be considered and will result in a clear and coherent plot. Two plots will be produced, as in part (b), as well as "fitting" line to better represent the trend of the data.

A selection of MATLAB .m files have been constructed to follow these three "parts", along with detailed comments to demonstrate good understanding of the application. The codes have been developed extensively and approved with the help of the project supervisor, Dr Nikolaos Fytas. The raw “.m” files will be given in the appendices, along with the relevant plots and figures from each part.

4.3 Results from Simulations

4.3.1 "Part (a)"

As outlined above, the first part of the simulation investigated the properties of the final configuration, depending on the system size and temperature. The results from this part are published in Chapter 6 (Appendices), from Figures 2 - 13. Configurations from three temperatures for each system size (for 4 distinct sizes) have been considered, thus resulting in 12 configurations to be given in the Appendices.

The first observation is a somewhat obvious one in that a larger system size would give more clear results, purely due to the nature of the configurations (i.e., the system size and "accuracy" are proportional to one another). The next observation showed that the hypothesised results (from the previous subsection) were true, e.g.,

- at $kT = 1$, i.e., a low temperature, very little movement occurred within the system, and as a result fewer phase transitions. This is represented in the final configurations as a screen with mostly one colour, as many of the sites remained in their initial spin state (due to a lack of transitions). (Evidence of these results are given in the following figures: **Fig.2, Fig.5, Fig.8**, and **Fig.11**)
- near the critical temperature, $kT = 2.31$, we see "clusters" of spins forming together in the final configurations. This is represented by groups of "dots" of the same colour (sites with the same spin state). As hypothesised above, this is due to the nature of the Ising model's behaviours at the critical temperature.This is given as Figures: **Fig.3, Fig.6, Fig.9**, and **Fig.12**.
- When the system is set to a higher temperature well above the critical point ($kT = 4$), the sites experience a lot of movement and phase transitions. This is represented on the configuration diagrams as a somewhat random assortment of dots on the screen. Refer to **Fig.4, Fig.7, Fig.10**, and **Fig.13** for these findings.

All the plots of the spin configurations are in agreement to the theoretical picture, that is as follows: At temperatures well below the critical the system is at its ordered state with almost all spins being aligned, so that the absolute value of the magnetisation of the system is on average $|m| = 1$. This way the ferromagnetic system lowers its energy. On the other hand as we increase the temperature, thermal fluctuations increase and as we arrive close to the critical temperature, then fluctuations become important at all length scales and we see the onset of a phase transition. Geometrically this is nicely shown by the formation of clusters of $+1$ spins (black coloured dots) in the sea of the -1 spins (white coloured dots), or the other way round. If we keep increasing the temperature well above the critical, then the system is disordered, in its paramagnetic phase, meaning that on average half of the spins will point up and half of the spins will point down giving a magnetisation $|m| \approx 0$. In all cases, minor differences are attributed to the fact that we study finite systems. We expect this theoretical picture to be fully verified at the infinite limit size volume.

4.3.2 "Part (b)"

For the results from Part (b), please see the following figures within the appendices: **Fig.16, Fig.17**.

For this part of the programming, the sample size was fixed at 30, thus for each iteration of the temperature, 30 simulations and data points were recorded. Statistical sampling was then completed in order to obtain the averages and standard error, as seen in the MATLAB codes **Listing 5**, and **Listing 6**. *(It is also worth noting that the system size was fixed at N=64).*

From **Fig.16**, there is a clear positive correlation between the Energy and Temperature, as shown by the shape of the plot line. One reason for this result would be from the hypothesis given in the previous subsection, such that higher temperatures result in more movement of particles within the system, and thus higher energy. The same result also corresponds in Part (a) with the higher temperature lattice configurations, as the somewhat random assortment of spins demonstrates more movement (and more phase transitions) within the system. Furthermore, by observing the graph there is a clear "jump" in energy at around the critical temperature, which is an interesting result, and further confirms the theory discussed previously within the thesis; the nature of a 2^{nd} order phase transition occurring at around the critical point.

Fig.17 represents the average magnetisation as a function of the temperature. Here, an opposing correlation is observed, compared to the previous plot line for the energy. These results imply that as the system is subjected to a higher temperature, the (average) magnetisation falls. Also, at temperatures around the critical point (~ 2.31), a sharp decrease in the magnetisation is seen. This result unsurprisingly agrees with the theoretical ideas provided in earlier sections of the thesis. One possible proposition for improvement of accuracy would be a larger sample size (for greater statistical sampling and calculations), and perhaps further investigation of the properties of the system around the critical temperature, as this area of the plot seems to be inconsistent with the rest (i.e., going from a slower, steadier decline to sharp one, then resuming the steadiness).

4.3.3 "Part (c)"

For the results from Part (c), please refer to **Fig.18** and **Fig.19** from the Appendices.

For this final section of the implementation in MATLAB, the sample size was fixed at 100, giving 100 data points for Energy and Magnetisation at each simulated system size. The temperature of the system was also fixed at the exact critical temperature, $kT = 2/log(1+\sqrt{2})$. Following the same processes as the previous part, statistical sampling was completed and values for Average Energy and Magnetisation (with standard error) were obtained.

By analysing the plot in **Fig.18** (the change in Average Energy as a function of the system size), a positive trend is seen. As the size of the system increases, the Average Energy of the system also increases. Another important observation to be made is that the size of the error bars decrease as the system gets larger, which suggests that a larger system is a generally a more accurate one. This could explain the two instances where the Average Energy drops (instead of rising), at N=24 and N=48. The error bars are relatively larger for these values, which highlights an element of inaccuracy. A method of obtaining more accurate results would be more extensive statistical sampling, perhaps with a larger sample size, as more data would provide more realistic Averages (and a smaller margin of error). However, the method completed here is sufficient for the requirements of this thesis.

Finally, **Fig.19** represents the variation in Average Magnetisation as a function of the system size (fixed at the critical temperature). This plot, as opposed to the previous one, shows a negative correlation, which implies that as the system increases in size, the average magnetisation will be reduced. The magnetisation at the critical temperature T_c is expected to scale as $m \sim \mathrm{L}^{\beta/\nu}$ with the system size. As shown in the figure, a fitting of this form was produced, represented by the solid (purple) line. The obtained result is $\beta/\nu = 0.148(25)$, which agrees within errors with the expected value 0.125 of the 2D Ising ferromagnet. The fitting quality was also acceptable with χ^2 / DOF = 0.29 / 2 (where “DOF” denotes the degrees of freedom). It is worth noting that the larger system sizes, $L = 64$ and $L = 96$, were excluded from the fitting procedure, as it appears that more extensive statistical averaging was needed for these sizes and perhaps longer equilibration times (which were beyond the scope of the current thesis).

5 Conclusions and Final Statements

5.1 Final Thoughts on Dissertation

The aim of this dissertation was to highlight and demonstrate the importance of statistical mechanical methods (i.e., Monte Carlo methods). A multitude of ideas were explored with great detail and explanations. The introduction chapter provided a suitable foundation for the thesis, and the literature review identified many of the key concepts in this area of study. The subsequent chapter gave an adequate expansion on the theory, and provided the reader with all of the necessary tools to understand the Metropolis algorithm and the programming section. On a personal note, the importance and weight of this area of physics was brought to my full attention during the study of this dissertation. Overall, the contents within the thesis have covered a wide range of the principle ideas of the Monte Carlo method and the Ising model.

5.2 Conclusions and Final Remarks on Simulations

In general, it suffices to say that the results from the performed simulations were in agreement to the hypotheses provided from the theory. The simulations have highlighted both the benefits and shortcomings of using an environment such as MATLAB. I believe that I executed my role to a high standard, as the MATLAB codes functioned well as a collective. I adapted the initial files to suit the needs of the investigation, and created suitable additional files (from scratch) to complete the findings and represent them appropriately. The overall construction of the code (as well as the debugging process) itself was not overly challenging, and clear, legible results were found. On the contrary, the need for more extensive statistical sampling and more processing time (with greater computer power) was apparent; "part (c)" in particular spotlighted this (due to the time for the simulation of 100 samples being inconveniently long), and also by the omission of the two largest system sizes from the fitting line. As a collective across the three parts, the data and results were exactly what was expected, and provided relevant answers for the investigation of the thesis. Although improvements could have been made (e.g., more statistical sampling and longer simulation times), the figures (and relevant data) were sufficiently conclusive for the thesis in general.

5.3 Improvements and Recommendations

Due to the nature of statistical physics (and statistical sampling in general), there is always room for improvement. A somewhat obvious recommendation (as discussed above) would be greater sample sizes, which require more processing power and computer time. As the technological industry is constantly striving to produce better resources and more efficient and powerful computers, these improvements are definitely within the realm of reality. Simulations of a higher speed and size are inevitable, resulting in the output of increasingly accurate results. Expectations and predictions would indeed become close enough to exact values.

Personally, I would have improved the investigation in this dissertation by increasing the sample sizes (specifically in parts "(b)" and "(c)"). This would have certainly resulting in smoother plots, and significantly reduced the presence of errors and anomalies within the data. However, as indicated above, the requirements for this thesis were met to a sufficient level, and the results generally followed what was expected.

5.4 Alternative Approaches

As briefly mentioned in the above chapters, there are alternative ways to perform the calculations and reach the desired results in a more efficient and accurate manner. Although the Metropolis algorithm is the most commonly used one, and the way in which it simulates the Ising Model using single-spin-flip dynamics is impressive, there are some significant weaknesses. One of the most notable drawbacks of the Metropolis algorithm is the critical losing down exponent, z (not to be confused with the lattice coordination number given in the previous chapters, which is also given as z). As the system approaches T_c, the simulations require longer times to reach equilibration, and this is clearly inconvenient for many reasons. It is worth noting that the magnitude of this z value is not universal; it differs depending on the chosen algorithm. For the 2D Ising Model running the Metropolis algorithm, (refer to *Barkema, Figure 4.2*), the correlation time τ is shown on a logarithmic scale, showing the exponential growth in τ with the system size. This high value of z highlights the imperfection of the Metropolis algorithm, and introduces the need for a "better" algorithm (in terms of a lower dynamic exponent).

In order to oppose this high z value of ~ 2, so-called "cluster" algorithms were created. Examples of such cluster algorithms would be the Wolff algorithm and the Swendsen-Wang algorithm. These significantly reduce the critical slowing down component, resulting in a sharp increase in simulation speeds. The basic logic of these alternative algorithms is that instead of flipping one spin at a time (much like the Metropolis algorithm), they flip clusters of spins simultaneously, thus improving the dynamics. If one desired to simulate larger systems and to undertake a more refined analysis of the 2D Ising Model, such cluster algorithms would be of greater use. As aforementioned, the value of the dynamic exponent z for the (more simple) Metropolis algorithm is close to 2, whereas the value of z for these cluster algorithms is close to 0.47, which is a drastic improvement. It is important to note that this improvement in z unsurprisingly comes at the expense of some convenience for the user, as the cluster algorithms are relatively more complex and require a lot more processing power and/or computing time.

In summary, the reason for the slowing down exponent of the Metropolis algorithm being quite large is not very difficult to conceptualise. The fundamental reason for the large z value is due to the presence of critical fluctuations near the phase transition (along with the divergence of the length of correlation). Large regions of spins pointing in the same direction (also referred to as "domains") form when the correlation length becomes large close to the critical temperature T_c. It is rather time-consuming for the algorithm to flip such a region, as the nature of the Metropolis algorithm is to execute in a spin-by-spin fashion, thus giving each move a higher probability of being rejected (due to the ferromagnetic interactions that occur between neighbouring spins). The Wolff algorithm (as aforementioned) flips clusters of spins rather than singularly, thus being more effective in such a scenario; however it can be slowed down near the critical temperature (due to the formation of larger domains near T_c)

In conclusion, the benefits of using Metropolis algorithm for the 2D Ising model outweigh the drawbacks, as the system sizes that were simulated were not too large. The methods that were adopted in this investigation proved to be sufficient and acceptable, as evidenced in the results that were produced.

6 Appendices

6.1 Phase Transitions Diagram

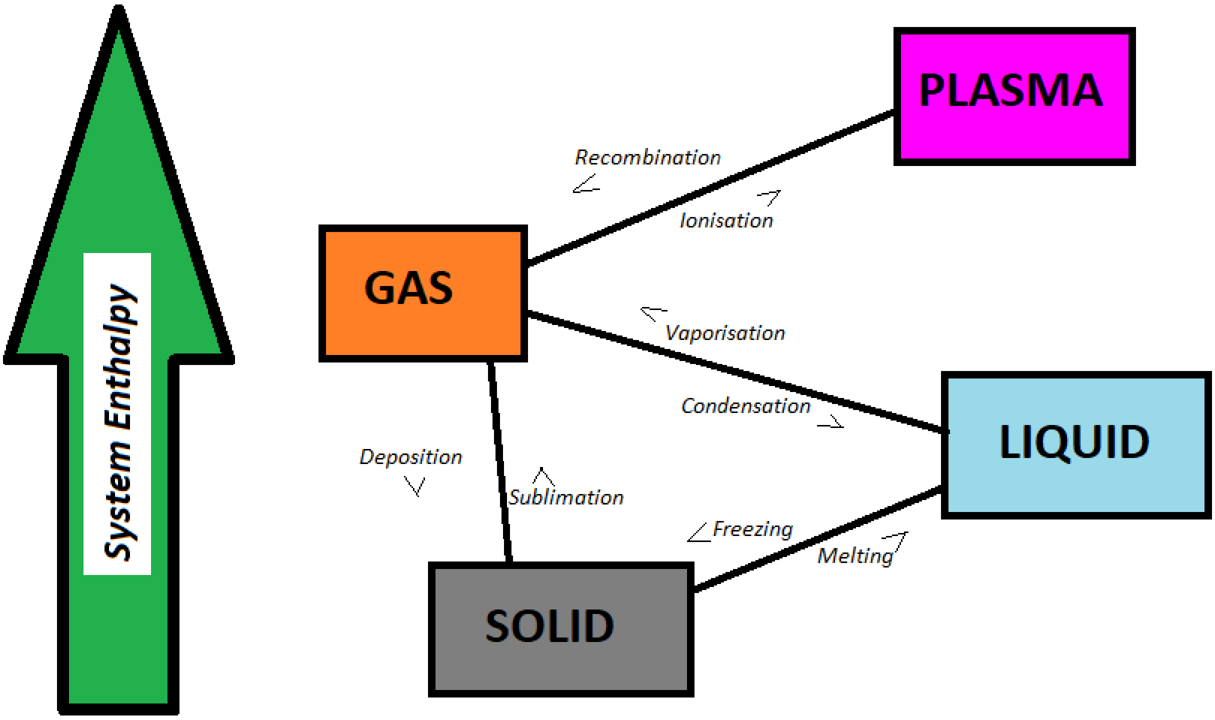


Figure 1: A basic diagram to represent phase transitions

6.2 Full MATLAB Codes

```
% System sizes: N = 6, 8, 12, 16, 24, 32, 48, 64, 96.
% Simulation times:
% For N < 24 use: t = 1000*N^2 and t_equil = 500*N^2.
% For 24 < N < 64 use: t = 1500*N^2 and t_equil = 800*N^2.
% For N = 96 use: t = 750*N^2 and t_equil = 400*N^2.

% System size
N = 96;

% Temperature
kT = 4;

% strength of the coupling (this remains fixed)
J = 1;

% n_samples is the number of samples to be simulated at a given temperature
% in order to take average values and increase statistics
n_samples = 1;

file_M = fopen('magnetization.txt','w');
file_E = fopen('energy.txt','w');

```

```
for isample=1:n_samples

% Generate a random initial configuration.
% Comment out to keep sampling with the previous configuration.
p=.5; % average proportion of initial +1 spins
grid = sign(p-rand(N)); % random initial configuration

% Run the Metropolis algorithm and return a matrix of spin values:
t = 750*N^2; % choose t update steps (use big multiple of N^2; increasing as N increases)
t_equil = 400*N^2; % take measurements for average values for t > t_equil
grid = metropolis(N,kT,J,t,t_equil,grid);

% Compute final magnetization density and energy density
% M = sum(sum(grid))/numel(grid);
 M = sum(abs(sum(grid)))/numel(grid); %definition of absolute magnetization
fprintf(file_M,'%d %f\n',isample,M);
E = isingenergy(grid,J);
fprintf(file_E,'%d %f\n',isample,E);
% plot final configuration
isingplot(grid,N,J,kT,M,E);

end


fclose(file_M);
fclose(file_E);
```

Listing 1: MATLAB Code for the main "ising.m" file which is used to run the simulations

```
function energy = isingenergy(grid,J)

neighbors = circshift(grid,[0 1]) + circshift(grid,[0 -1]) + ...
            circshift(grid,[1 0]) + circshift(grid,[-1 0]);
energy = -J*sum(sum(grid.*neighbors))/numel(grid);

end
```

Listing 2: "isingenergy.m" file, which deals with the energy component of the simulations

```
function isingplot(grid,N,J,kT,M,E)
% Display the current state of the system
image((grid+1)*128);
title(sprintf('2D Ising model with %0.4g by %0.4g lattice',N,N));
xlabel(sprintf('J=%0.2f, kT = %0.2f, M = %0.3f, E = %0.3f',J,kT,M,E));
axis square; colormap bone; set(gca,'XTickLabel',[],'YTickLabel',[]);
drawnow;
end
```

Listing 3: "isingplot.m", which is used for the plot i.e., displaying the configurations on the screen to the user

```
function grid = metropolis(N,kT,J,t,t_equil,grid)
% Metropolis sampling for the 2D Ising model
% plot initial configuration
M = sum(sum(grid))/numel(grid);
E = isingenergy(grid,J);
isingplot(grid,N,J,kT,M,E);
```

```

% precompute the indicies adjacent to each spin index
adj = neighbors(1:N^2,N);

% Pick a sequence of random spins (with a linear index)
spin = randi(N^2,t,1);

% Evolve the system for a fixed number of steps
for i=1:t,
    % location of ith spin
    s = spin(i);
    % Calculate the change in energy of flipping s
    dE = 2*J*grid(s)*sum(grid(adj(s,:)));
    % Calculate the transition probability
    p = exp(-dE/kT);
    % Decide if a transition will occur
    if rand <= p, grid(s) = -1*grid(s); end
    % Refresh display of current spin configuration every N^2 trials
    if ((i>t_equil) & (mod(i,N^2)==0))
        % Sum up our variables of interest and plot
        M = sum(sum(grid))/numel(grid);
        E = isingenergy(grid,J);
       % isingplot(grid,N,J,kT,M,E);
    end
end

end

function adj = neighbors(s,N)
% take a list of linear indices s and return the linear indices of the
% neighbors of s on an N by N grid with periodic boundary conditions.

s = s-1; % index by zero
adj = zeros(length(s),4);

% s = r*N+c;
r = floor(s/N);
c = rem(s,N);

adj(:,1) = mod(r+1,N)*N+c;   %down
adj(:,2) = mod(r-1,N)*N+c;   %up
adj(:,3) = r*N+mod(c+1,N);   %right
adj(:,4) = r*N+mod(c-1,N);   %left

adj = adj+1; % index by one again

end
```

Listing 4: MATLAB Code for the ”metropolis.m” file, for the Metropolis algorithm implemented in the Monte Carlo simulations

```
%Code to quickly calculate the Average and Standard Error for the data
%obtained by running the simulations

%Loading the data from the simulation, stored in the energy.txt file:
Edata=load('energy.txt');
%Taking only the second column into consideration:
X=Edata(:,2);
%Energy Average and Standard Deviation:
```

```
Eav=mean(X)
Esd=std(X)

%The same logic applies:
Mdata=load('magnetization.txt');
Y=Mdata(:,2);
Mav=mean(Y)
Msd=std(Y)
```

Listing 5: "Averages and stddevs.m", which is used to quickly calculate the Average and Standard Deviation for the data stored in the text files

```
%ENERGY
figure(1)
%Range of Energy values considered:
xe=1.5:0.05:3;
%Energy Average Data:
ye=[-3.8932 -3.8341 -3.8393 -3.7914 -3.7768 -3.6997 -3.7016 -3.6708...
    -3.6167 -3.5802 -3.4664 -3.3961 -3.2987 -3.2146 -3.1745 -2.8667 ...
    -2.7477 -2.6276 -2.3867 -2.3146 -2.1898 -2.1336 -2.0380 -2.0076 ...
    -1.9417 -1.8943 -1.8258 -1.7461 -1.6854 -1.6995 -1.6172];
%Standard Error for the error bars:
olderre=[0.0857 0.1409 0.0985 0.1161 0.0877 0.1563 0.1424 0.0785 0.0822 ...
    0.0865 0.1102 0.1272 0.1075 0.1363 0.1765 0.2108 0.1827 0.1814...
    0.1773  0.1804 0.1434 0.1198 0.1611 0.1371 0.1412 0.1280 0.1245 ...
    0.1437 0.1236 0.1083 0.1188];
%From the error bar formula given as Eq(4.A), we must divide by sqrt(30):
erre=olderre/sqrt(30);
errorbar(xe,ye,erre,'c')
xlabel('kT')
ylabel('Average Energy')
title('Change in Avg Energy as a function of the Temperature (kT)')

%MAGNETISATION
%The same logic applies as in the first figure:
figure(2)
xm=1.5:0.05:3;
ym=[0.9836 0.9487 0.9523 0.9492 0.9631 0.9375 0.9467 0.9469 0.9386...
    0.9309 0.9090 0.8862 0.8548 0.8354 0.8230 0.6697 0.6146 0.4964...
    0.4048 0.3374 0.3169 0.2994 0.2939 0.2982 0.2805 0.2753 0.2469 ...
    0.2465 0.2320 0.2421 0.2256];
olderrm=[0.0229 0.1141 0.1452 0.1144 0.0314 0.0776 0.0531 0.0150 0.0180 ...
    0.0188 0.0259 0.0415 0.0467 0.0502 0.0581 0.1457 0.1406 0.1730...
    0.1437 0.1240 0.1073 0.0658 0.0805 0.0785 0.0635 0.0713 0.0480 ...
    0.0420 0.0408 0.0432 0.0409];
%divide all sds by root(n) = root(30):
errm=olderrm/sqrt(30);
errorbar(xm,ym,errm,'m')
xlabel('kT')
ylabel('Average Magnetisation')
title('Change in Avg Magnetisation as a function of the Temperature (kT)')
```

Listing 6: MATLAB Code for the "plot_for_part_b.m" file

```
%ENERGY
figure(1)
%Values of considered System Sizes:
x=[16,24,32,48,64,96];
```

```
%Data for Average Energy:
y=[-2.8838 -2.9069 -2.8545 -2.8618 -2.8195 -2.7574];
%Standard error for error bars:
er=[0.3080 0.2701 0.1803 0.1281 0.1194 0.0606];
%From the error bar formula given as Eq(4.A), we must divide by sqrt(100):
err=er/10;
%err=[0.0308 0.02701 0.01803 0.01281 0.01194 0.00606];
errorbar(x,y,err,'r')% ,'linestyle','none')
xlabel('System Size')
ylabel('Average Energy')
title('Change in Average Energy as a function of the System Size')

%MAGNETISATION
figure(2)
x=[16,24,32,48,64,96];
y=[0.7409 0.7002 0.6636 0.6312 0.5553 0.3696];
er=[0.1323 0.1543 0.1375 0.1344 0.1666 0.1203];
%err=[0.01323 0.01543 0.01375 0.01344 0.01666 0.01203];
err=er/10;
errorbar(x,y,err,'k','linestyle','none')
hold on
%Fitting function to be plotted in conjunction with the data points:
fitting=1.11882*x.^(-0.14862);
plot(x,fitting,'color',[rand rand rand])
xlabel('System Size')
ylabel('Average Magnetisation')
title('Change in Average Magnetisation as a function of the System Size')
```

Listing 7: MATLAB Code for the ”plot_for_part_c.m” file

6.3 Configurations from "Part (a)"

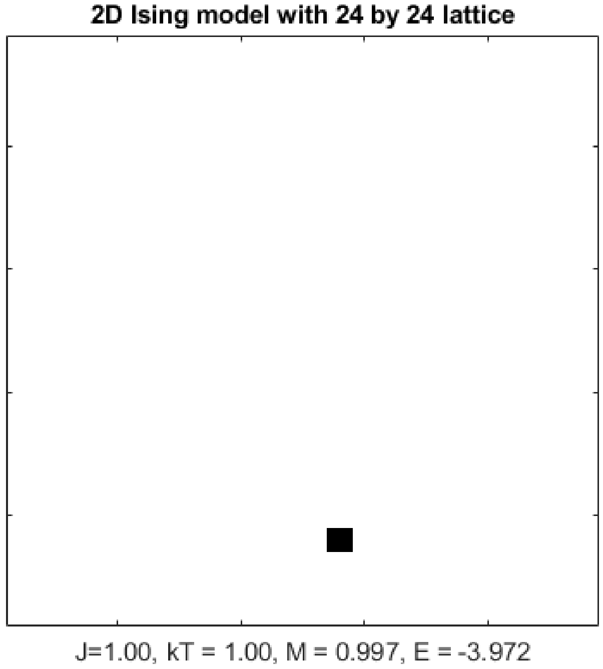


Figure 2: Example of a configuration for N=24 at kT=1

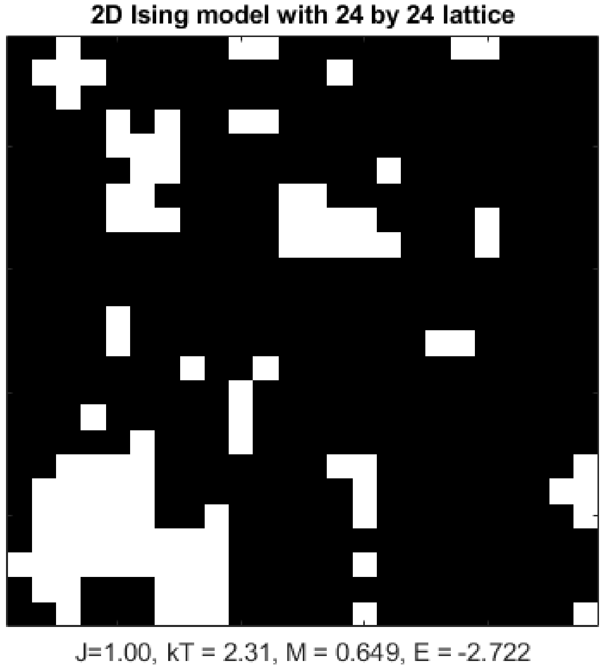


Figure 3: Example of a configuration for N=24 at kT=2.31

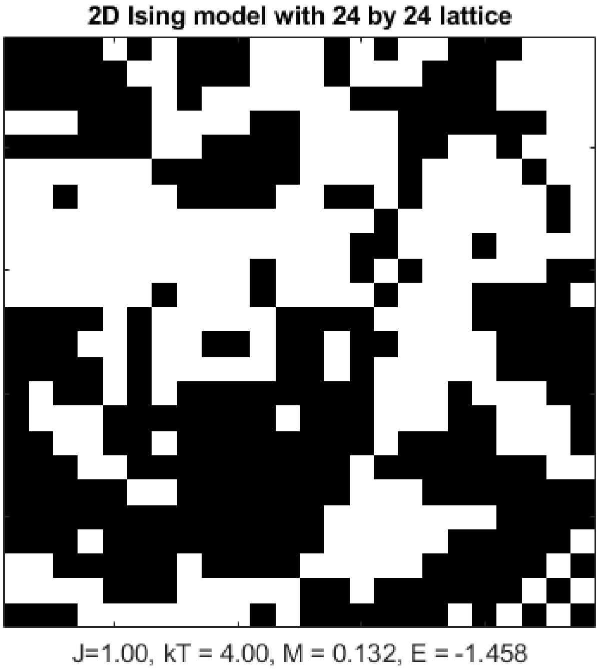


Figure 4: Example of a configuration for N=24 at kT=4

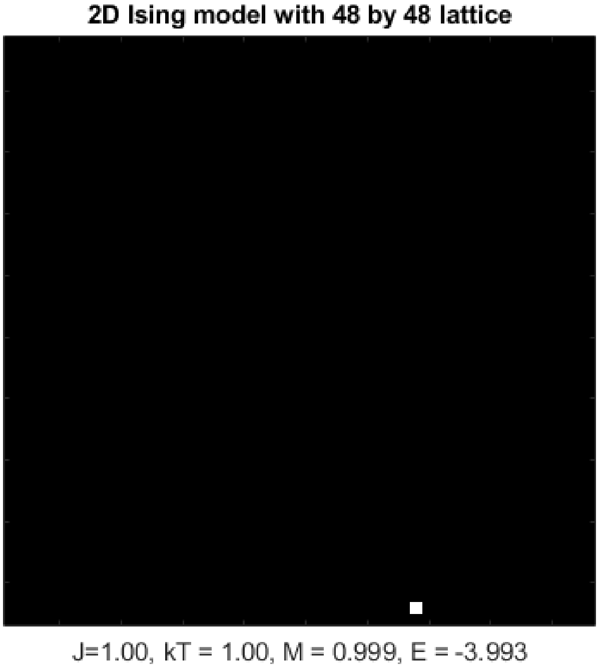


Figure 5: Example of a configuration for N=48 at kT=1

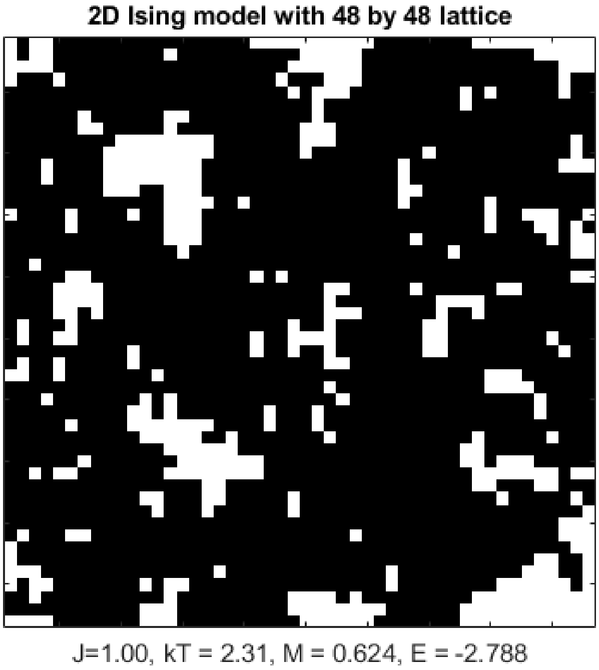


Figure 6: Example of a configuration for N=48 at kT=2.31

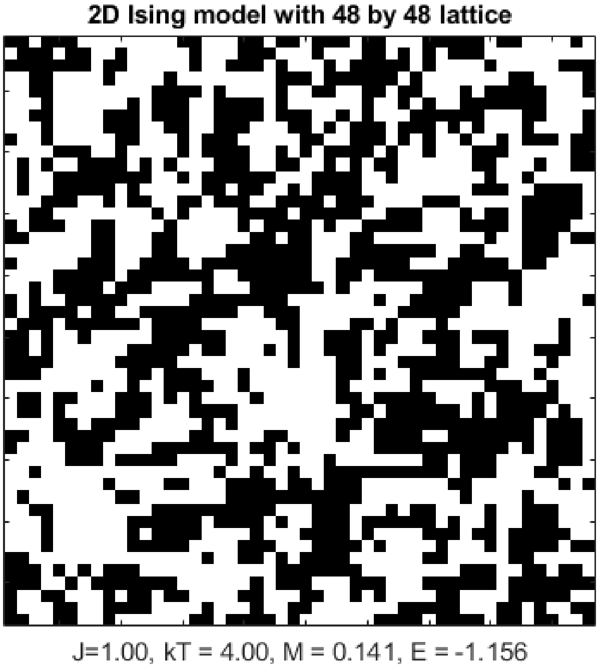


Figure 7: Example of a configuration for N=48 at kT=4

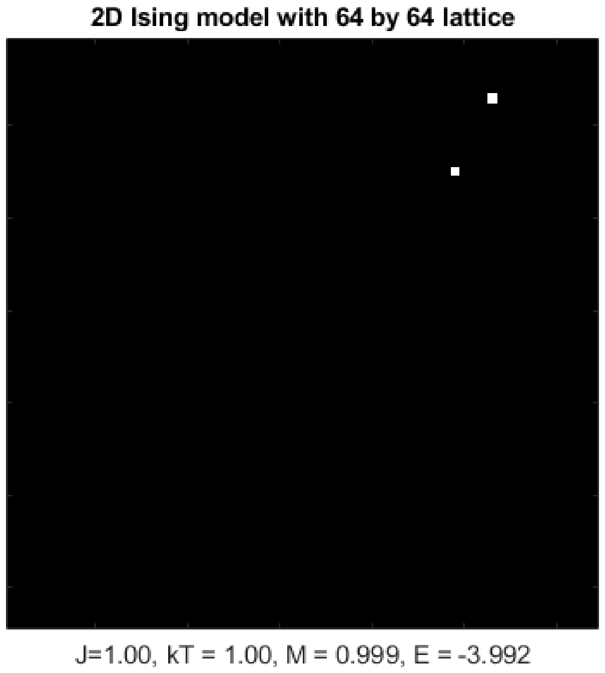


Figure 8: Example of a configuration for N=64 at kT=1

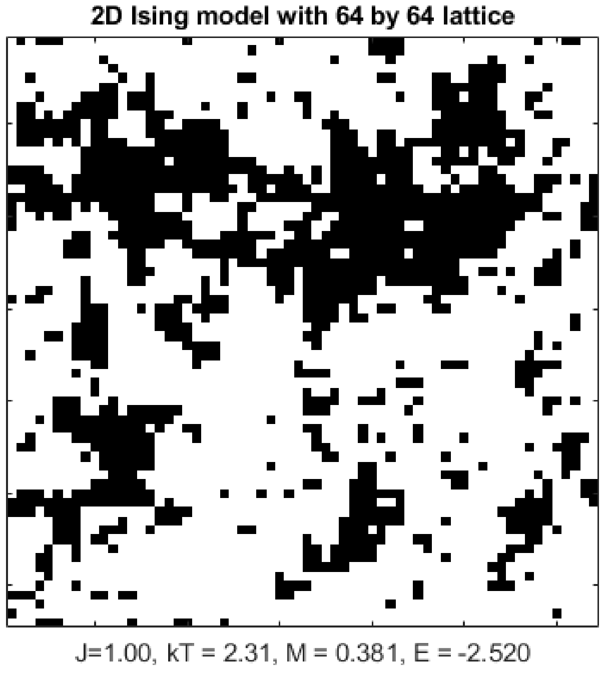


Figure 9: Example of a configuration for N=64 at kT=2.31

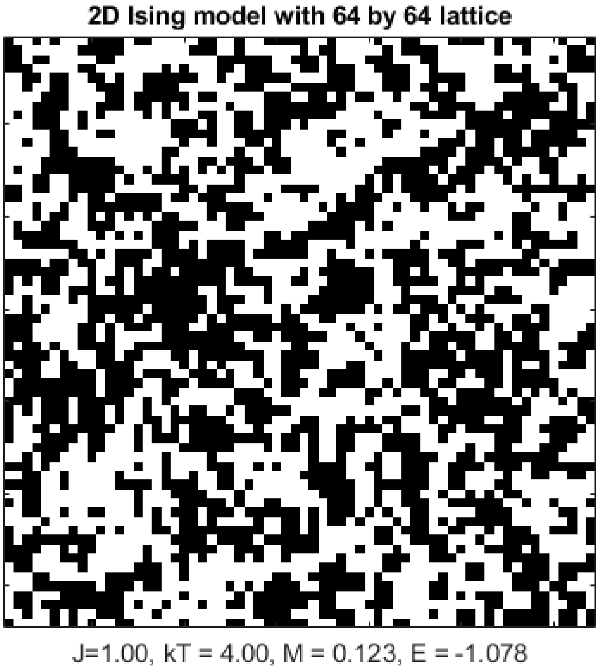


Figure 10: Example of a configuration for N=64 at kT=4

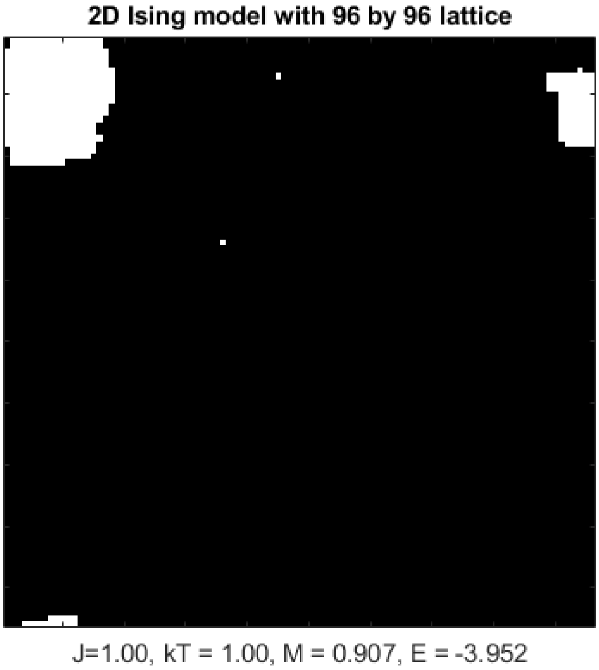


Figure 11: Example of a configuration for N=96 at kT=1

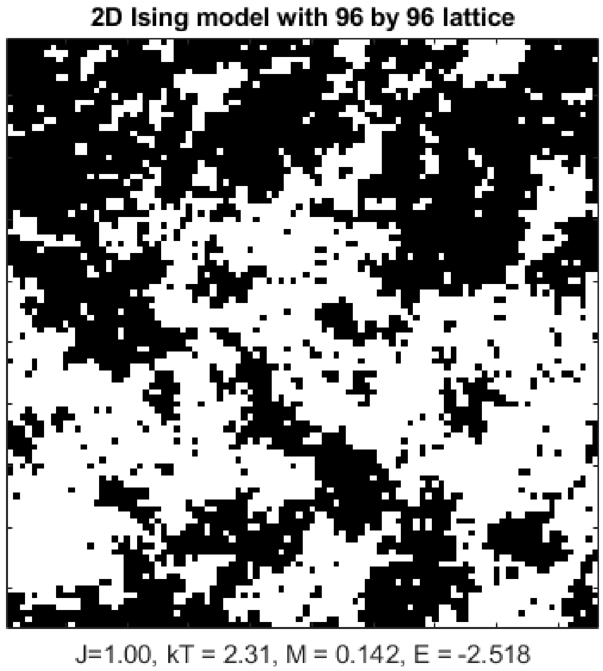


Figure 12: Example of a configuration for N=96 at kT=2.31

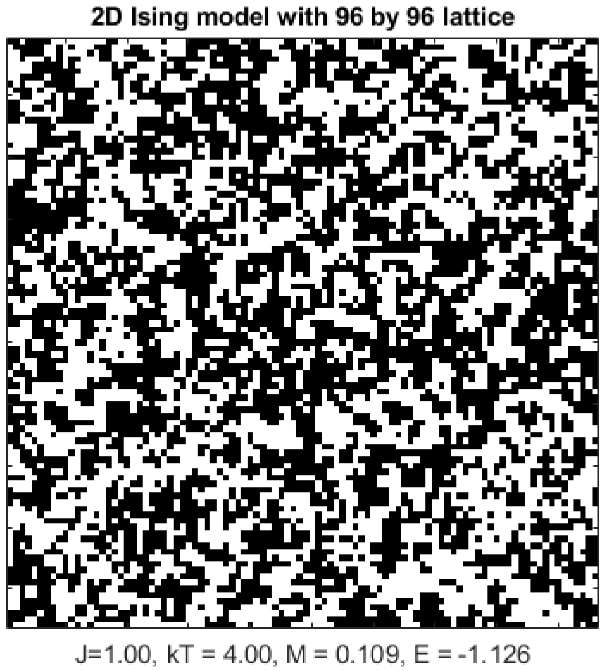
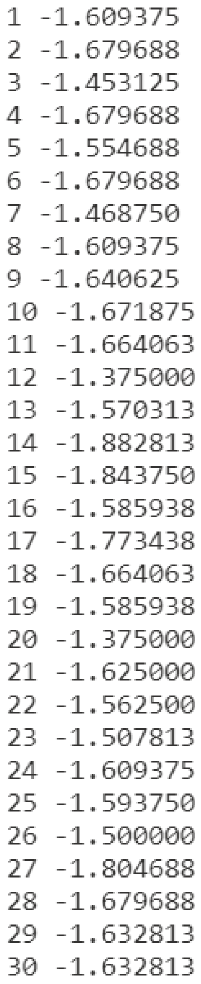


Figure 13: Example of a configuration for N=96 at kT=4

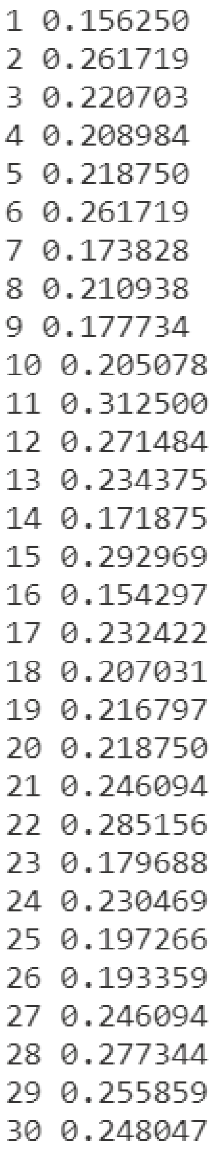
6.4 Figures from “Part (b)” and “Part (c)”

6.4.1 Examples of Data Points obtained from the simulations



```
1 -1.609375
2 -1.679688
3 -1.453125
4 -1.679688
5 -1.554688
6 -1.679688
7 -1.468750
8 -1.609375
9 -1.640625
10 -1.671875
11 -1.664063
12 -1.375000
13 -1.570313
14 -1.882813
15 -1.843750
16 -1.585938
17 -1.773438
18 -1.664063
19 -1.585938
20 -1.375000
21 -1.625000
22 -1.562500
23 -1.507813
24 -1.609375
25 -1.593750
26 -1.500000
27 -1.804688
28 -1.679688
29 -1.632813
30 -1.632813
```

Figure 14: Example of data obtained through the running of simulations, in this case one example of the energy.txt file is displayed



```
1 0.156250
2 0.261719
3 0.220703
4 0.208984
5 0.218750
6 0.261719
7 0.173828
8 0.210938
9 0.177734
10 0.205078
11 0.312500
12 0.271484
13 0.234375
14 0.171875
15 0.292969
16 0.154297
17 0.232422
18 0.207031
19 0.216797
20 0.218750
21 0.246094
22 0.285156
23 0.179688
24 0.230469
25 0.197266
26 0.193359
27 0.246094
28 0.277344
29 0.255859
30 0.248047
```

Figure 15: Example of data obtained through the running of simulations, in this case one example of the magnetisation.txt file is displayed

6.4.2 Final Plots

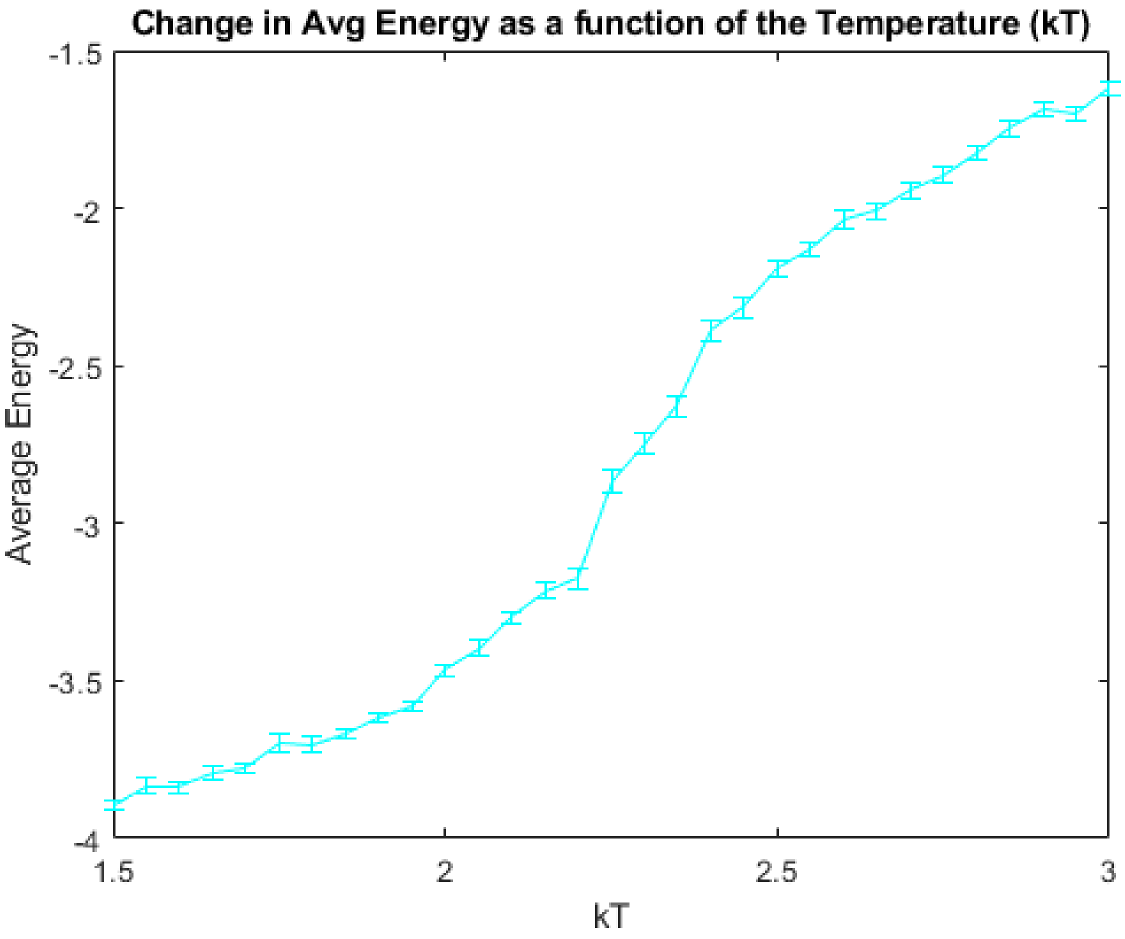


Figure 16: Plot of the data obtained from statistical sampling, demonstrating the proportionality of the Avg. energy of a system and its temperature. **The system size in this case was N=64**

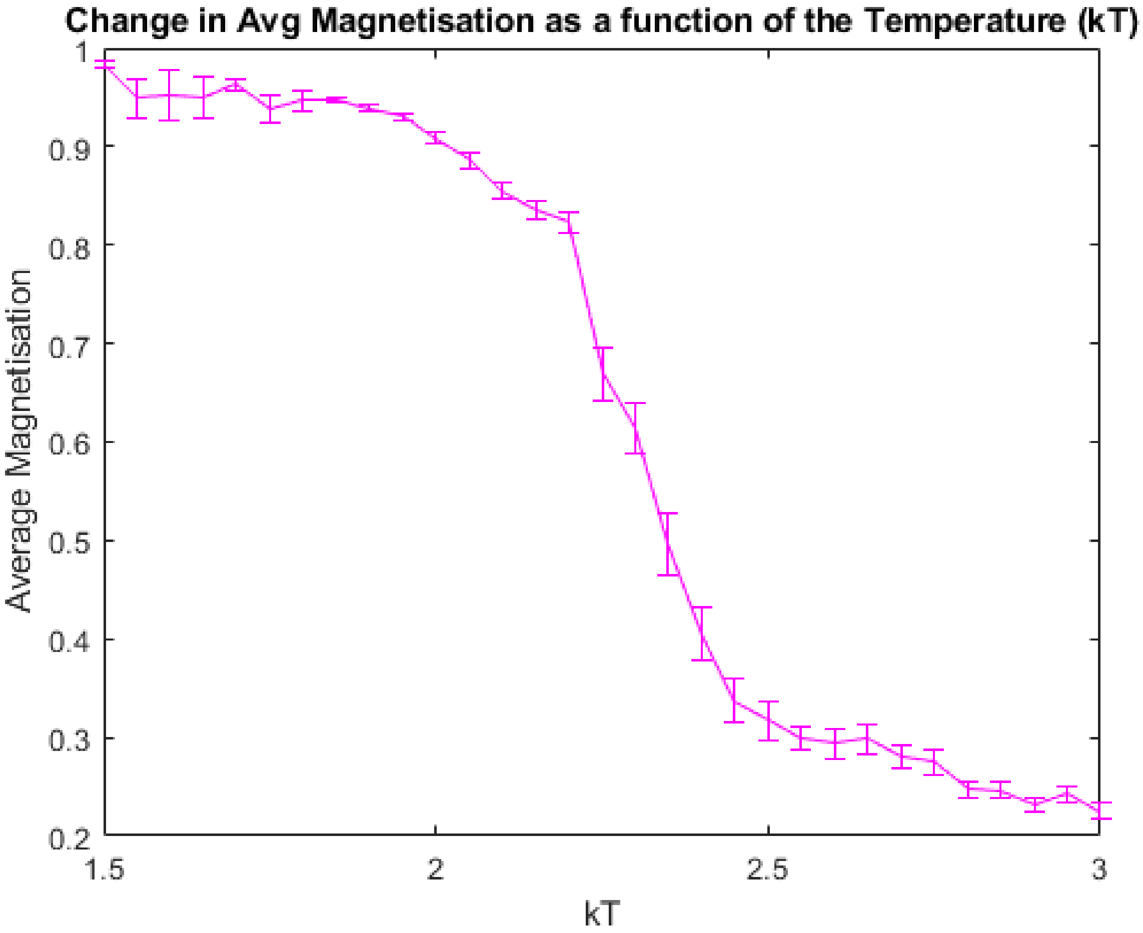


Figure 17: Plot of the data obtained from statistical sampling, demonstrating the inverse proportionality of the Avg. magnetisation of a system and its temperature. **The system size considered here was N=64**

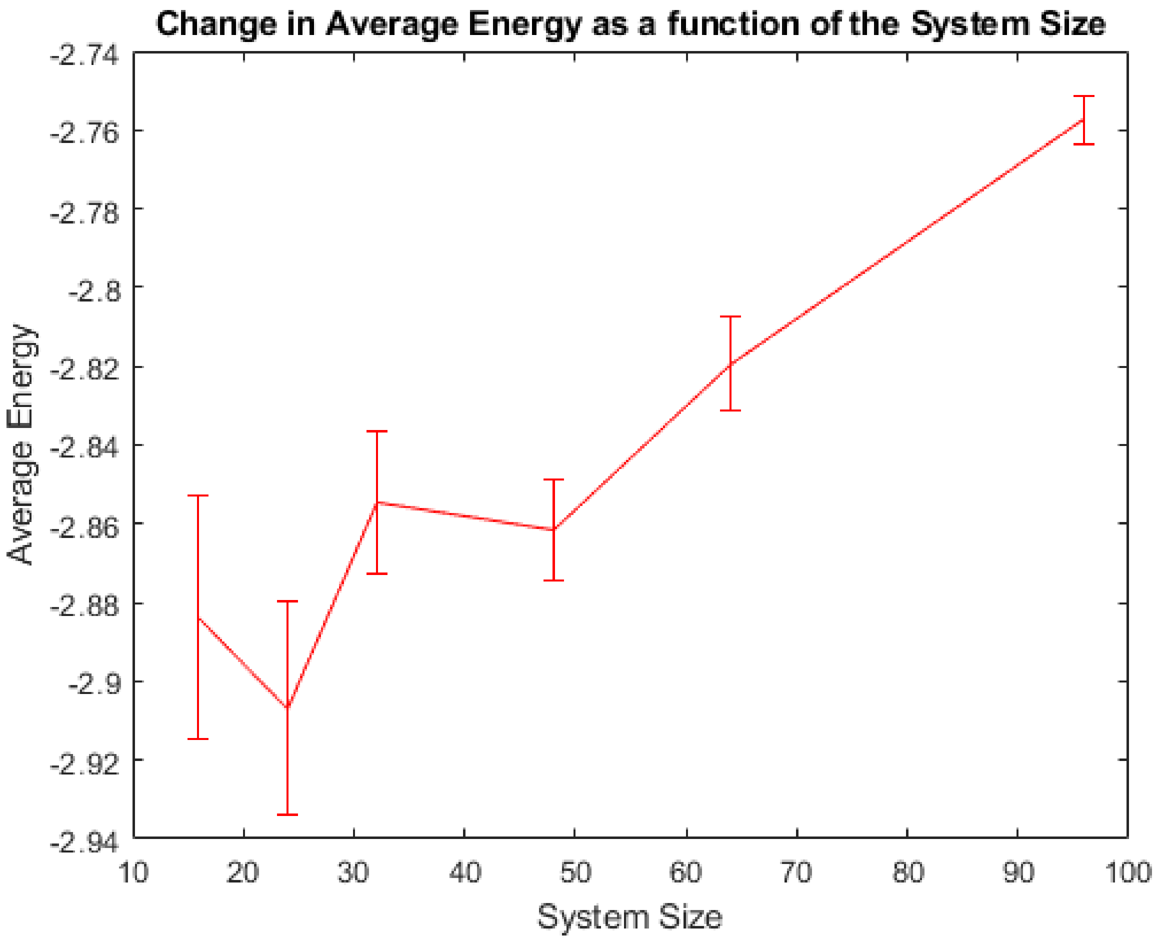


Figure 18: Plot for the data with 6 different system sizes is shown, **at the critical temperature,** $T_c = 2/log(1 + \sqrt{2})$

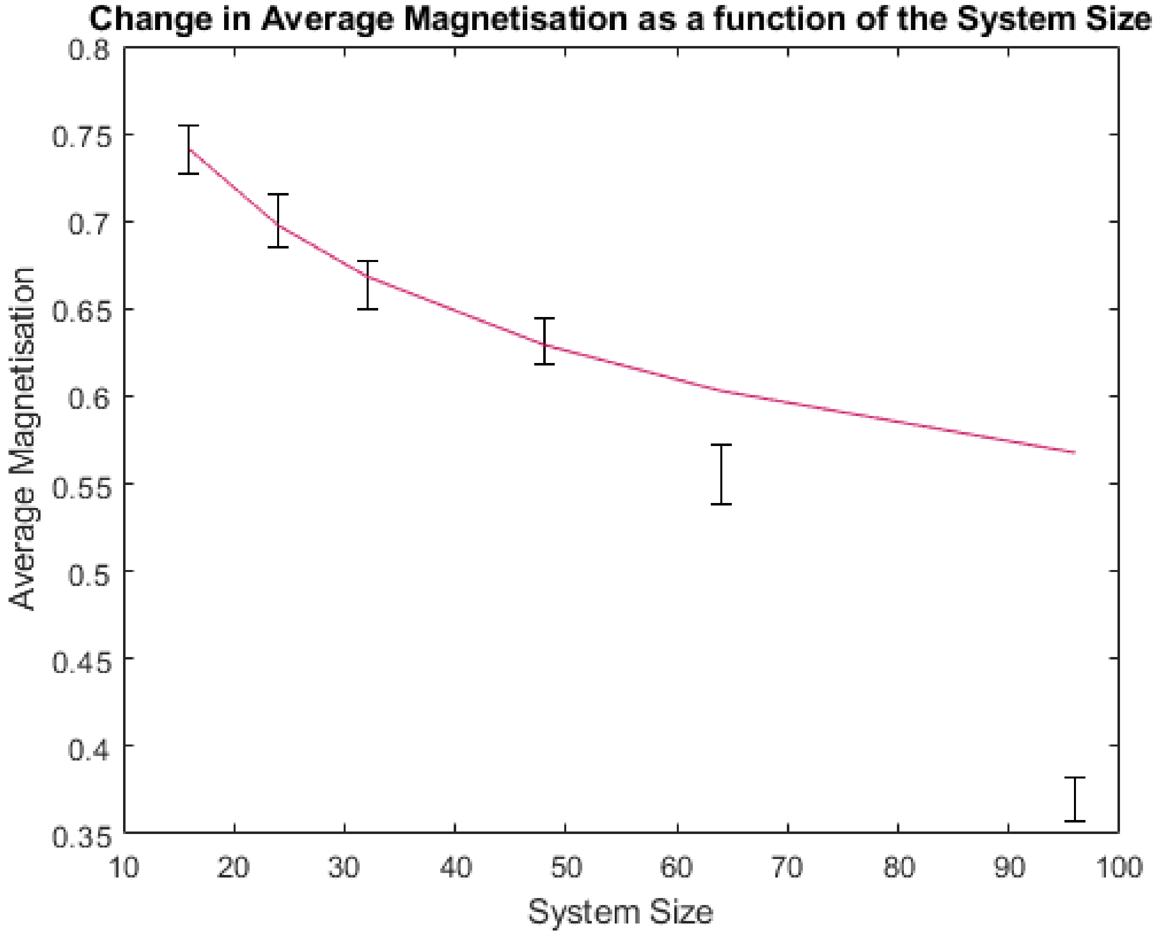


Figure 19: The original plot is shown only with the error bars, and the fitting function has been displayed on the same figure. **This was also fixed at the critical temperature,** $T_c = 2/log(1+\sqrt{2})$. The magnetisation at the critical temperature is expected to scale as $m \sim L^{\beta/\nu}$ with the system size. Here, a fit of this form is given, as shown by the solid line. Note that the larger sizes $L = 64$ and $L = 96$ were excluded from the fits, as it appears that more extensive statistical averaging was needed for these sizes and perhaps longer equilibration times

6.4.3 Project Diary

23/09/20 : First Meeting with project supervisor. The thesis was introduced, along with very basic and preliminary concepts. Structure of meetings was set-up, being either weekly or bi-weekly (depending on schedules).

28/09/20 : Material was sent by the supervisor to understand the fundamental theory, in the form of Barkema chapters 1-4, and Hjorth-Jensen notes chapter 13. Barkema's Chapter 1.1 was recommended, and any questions and ideas were to be presented in the next meeting.

06/10/20 : Discussed the progression from the last meeting. I presented the supervisor with some basic questions, e.g.: "Real world example for some context?" and "What does $w_\mu(t)$ represent?", along with some questions about the logistics of the report. Was given the task of reading up to Barkema 1.4, and discussing any questions in the following meeting.

19/10/20 : The meeting commenced with a question of mine: "What was the reason to introduce magnets into the problem of phase transitions?", to which Dr. Fytas responded that it is the simplest method and it is easy to understand, and explained the basics of ferromagnetism. Discussed that some systems have higher probabilities: "important states", which introduced importance sampling and acceptance ratios. I was told to read chapter 2 of Barkema for the next meeting.

04/11/20 : The last meeting was ended with Dr. Fytas recommending the reading of Chapter 2. Some questions from me were as follows: "I wanted to start making a rough plan for some progress on the report over the Christmas period", and "We always want it to reach the Boltzmann Distribution, what happens if it reaches another?", to which the replies were that other distributions were simply not allowed, and that statistical physics is built upon the Boltzmann distribution; it is a necessary condition. A brief discussion about planning the report and balancing my other modules also occurred. Dr. Fytas recommended to read up to 3.4 of Barkema, along with some of the addition Jensen chapter 13 notes.

20/11/20 : The material that I read throughout the weeks were discussed, most notably the Jackknife, blocking and bootstrap method, along with single-spin-flip dynamics. T=0 is an ordnance state and T_{max} is a disordnance state, and somewhere in the middle, "weird" things happen. The initial m files would be sent after the next meeting, and I was told to explore more theory from Barkema and Jensen notes.

08/12/20 : The initial MATLAB files were sent, and the basics of the simulations and the configurations were discussed. I changed the value of T_c from $2 \times 2/log(1 + sqrt(2))$ to $2/log(1 + sqrt(2))$ in the ising.m code for the critical temperature value. We also discussed the progress that I hoped to achieve over the Christmas period, and to run some simulations for part (a) and (b) and obtain some results.

11/01/21 : Discussed briefly about my exams before the Christmas break, results due on 20/01. After running the files, I was happy with the process, and had no questions in particular. Dr. Fytas tasked me with plotting Energy and Magnetisation as a function of Temperature for all system sizes, and to take screenshots of final configurations of part (a).

08/02/21 : Email correspondence was regular throughout the month, so there was no need for a meeting for a few weeks. The results and plots were discussed, with some ideas for slight improvements. The history of the Ising model was discussed, as well as some advie for the literature review section of the dissertation report. I would send a draft within the next week of what I had so far.

10/03/21 : After the data and results were produced, we analysed them together, and found what we initially expected. The draft was discussed, and minor changes were made. Overall a brief meeting, due to regular email correspondence and not many questions from me.

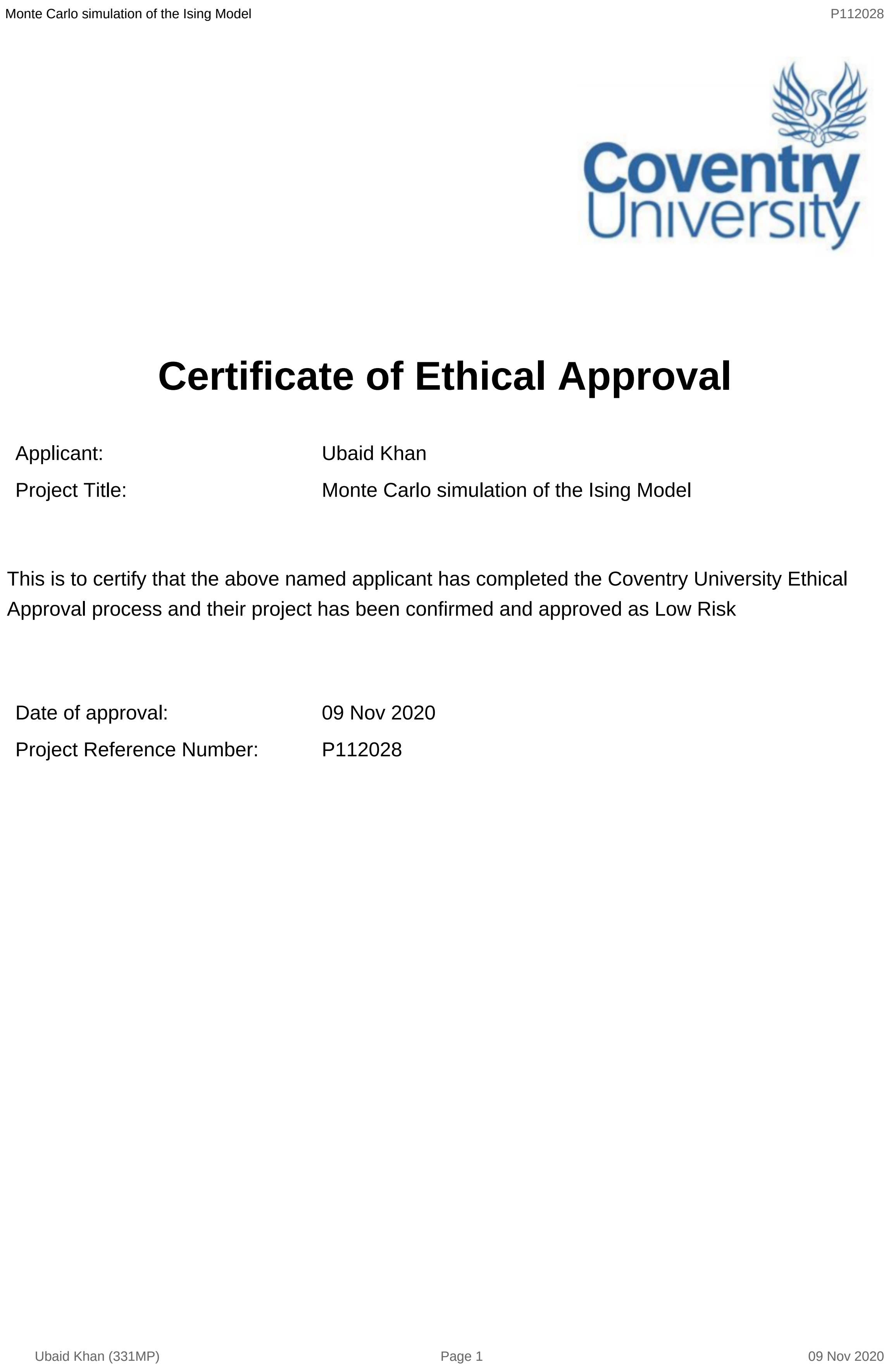
26/03/21 : The fitting for part (c) was completed together, and another draft of the report was sent. I was given advice on the preparation of my upcoming final exams, and how to prepare for the thesis presentation.

17/04/21 : Final exams from other modules completed, focus fully shifted to the completion of the project and the presentation. Dr. Fytas provided me with some recommendations on what to include in the conclusion section of the thesis.

22/04/21 : Thesis presentation given to Dr. Fytas and Dr. Taras Yavorskyi.

30/04/21 : Deadline for submission, final version sent to supervisor beforehand to approve: all parties satisfied with the standard of work. Small final recommendations were made on additional information in the figures for part (b) and (c).

6.4.4 CU Ethics Approval Certificate



Monte Carlo simulation of the Ising Model P112028

Certificate of Ethical Approval

Applicant:	Ubaid Khan
Project Title:	Monte Carlo simulation of the Ising Model

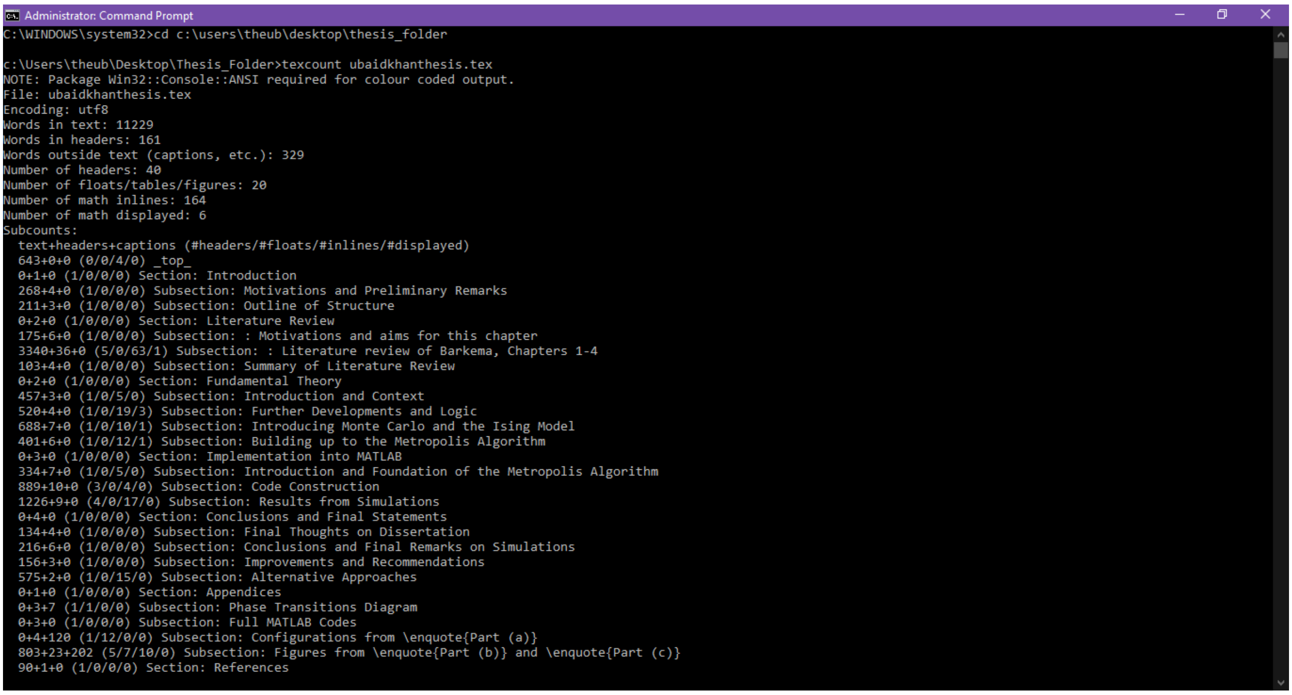
This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:	09 Nov 2020
Project Reference Number:	P112028

Ubaid Khan (331MP) Page 1 09 Nov 2020

The total word count for this dissertation was accurately deduced by using the *texcount.pl* Perl script.

The final word count is: **11229**.
A screenshot of the *cmd* window with the texcount script is given:



```
C:\WINDOWS\system32>cd c:\users\theub\desktop\thesis_folder

c:\Users\theub\Desktop\Thesis_Folder>texcount ubaidkhanthesis.tex
NOTE: Package Win32::Console::ANSI required for colour coded output.
File: ubaidkhanthesis.tex
Encoding: utf8
Words in text: 11229
Words in headers: 161
Words outside text (captions, etc.): 329
Number of headers: 40
Number of floats/tables/figures: 20
Number of math inlines: 164
Number of math displayed: 6
Subcounts:
  text+headers+captions (#headers/#floats/#inlines/#displayed)
  643+0+0 (0/0/4/0) _top_
  0+1+0 (1/0/0/0) Section: Introduction
  268+4+0 (1/0/0/0) Subsection: Motivations and Preliminary Remarks
  211+3+0 (1/0/0/0) Subsection: Outline of Structure
  0+2+0 (1/0/0/0) Section: Literature Review
  175+6+0 (1/0/0/0) Subsection: : Motivations and aims for this chapter
  3340+36+0 (5/0/63/1) Subsection: : Literature review of Barkema, Chapters 1-4
  103+4+0 (1/0/0/0) Subsection: Summary of Literature Review
  0+2+0 (1/0/0/0) Section: Fundamental Theory
  457+3+0 (1/0/5/0) Subsection: Introduction and Context
  520+4+0 (1/0/19/3) Subsection: Further Developments and Logic
  688+7+0 (1/0/10/1) Subsection: Introducing Monte Carlo and the Ising Model
  401+6+0 (1/0/12/1) Subsection: Building up to the Metropolis Algorithm
  0+3+0 (1/0/0/0) Section: Implementation into MATLAB
  334+7+0 (1/0/5/0) Subsection: Introduction and Foundation of the Metropolis Algorithm
  889+10+0 (3/0/4/0) Subsection: Code Construction
  1226+9+0 (4/0/17/0) Subsection: Results from Simulations
  0+4+0 (1/0/0/0) Section: Conclusions and Final Statements
  134+4+0 (1/0/0/0) Subsection: Final Thoughts on Dissertation
  216+6+0 (1/0/0/0) Subsection: Conclusions and Final Remarks on Simulations
  156+3+0 (1/0/0/0) Subsection: Improvements and Recommendations
  575+2+0 (1/0/15/0) Subsection: Alternative Approaches
  0+1+0 (1/0/0/0) Section: Appendices
  0+3+7 (1/1/0/0) Subsection: Phase Transitions Diagram
  0+3+0 (1/0/0/0) Subsection: Full MATLAB Codes
  0+4+120 (1/12/0/0) Subsection: Configurations from \enquote{Part (a)}
  803+23+202 (5/7/10/0) Subsection: Figures from \enquote{Part (b)} and \enquote{Part (c)}
  90+1+0 (1/0/0/0) Section: References
```

Figure 20: cmd window of texcount.pl Perl script, showing the breakdown of the word count

7 References

[**1**] LABC Blog post (2018), *What is the average house size in the UK?*

[**2**] Physics stack exchange (2013), *How many molecules touch you in your lifetime?*

[**3**] G.T. Barkema, M.E.J Newman, *Monte Carlo Methods in Statistical Physics*, Oxford University Press, Oxford, 1999: Equation (1.1)

[**4**] Barry McCoy (2010) *Ising model: exact results.*

[**5**] Hjorth-Jensen, M (2013) *Computational Physics*, University of Oslo, Oslo

[**6**] Boyle, P. P. (1977). *Options: A Monte Carlo approach*, Journal of Financial Economics

[**7**] H. A. Kramers, G. H. Wannier (1941). *"Statistics of the two-dimensional ferromagnet"*