Negative temperatures and fluid simulations of a modified 1D Ising model

Kevin Oon Zhi Jian

Supervisor: Professor Wang Jian-Sheng

A thesis submitted in partial fulfillment for the degree of Bachelor of Science with Honours

Department of Physics, Faculty of Science,
National University of Singapore, 2014/15

Acknowledgements

I would like to thank Professor Wang Jian-Sheng for his invaluable guidance and patience throughout this final year project. This project has given me a deeper insight into the fields of statistical and computational physics.

I would also like to express my appreciation towards the teaching faculty in National University of Singapore for sharing their deep knowledge and understanding of subject matter in the various modules I have read throughout my four years of study. The skills and knowledge I have aquired in NUS will serve as a solid basis for the next step in my adult life.

I would like to thank my peers in NUS Physics 2014/15 who have been present throughout my undergraduate life; without their support I would not have been able to handle the stress and rigour of a Physics course.

Last but not least, I would like to express my eternal gratitude to my parents for their unconditional love and support from the very beginning of my existance.

Abstract

Boltzmann's formula for entropy in statistical mechanics can give rise to negative absolute temperatures. A recent paper by Dunkel and Hilbert[1] claims that Boltzann's formula is inconsistent with thermostatistics, and hence negative absolute temperatures do not exist. We show, by means of an alternative formulation to obtain the absolute temperature scale of a spin paramagnet, that Boltzmann's formula for entropy remains the correct definition of entropy instead of the "Gibbs" volume entropy proposed by Dunkel and Hilbert.

We simulate a modified one dimensional Ising model using the Metropolis algorithm in MATLAB to investigate the system's behaviour in response to the variation of some governing parameters. We find evidence to suggest the existance of negative total pressure for systems in positive temperature, in contrast to a paper by Braun et al[2] that negative pressure only exists in system with negative temperature.

Contents

Part I

Introduction

In thermodynamics, entropy S is a measure of order and disorder in a system and was developed to explain experimental observations of the loss of functional energy in combustion reactions. Scientists that contributed to the theory of entropy in thermodynamics include Rudolf Clausius (1822-1888) and Sadi Carnot (1796-1832). The statistical definition of entropy was developed by Ludwig Boltzmann (1844-1906) in 1877 as part of his formulation on statistical mechanics. In Section 1, we explain how Boltzmann's formulation of entropy gives rise to the existance of negative temperature, and also briefly cover recent papers in academia regarding the existance (or lack thereof) of negative temperatures.

1 Boltzmann entropy to negative temperatures

Boltzmann's entropy formula in statistical mechanics is an equation that relates the entropy S of a system in thermodynamic equilibrium to the number of microstates Ω that corresponds to a macrostate:

$$S_B = k_B \ln \Omega, \tag{1}$$

where k_B is the Boltzmann's constant. Absolute temperature can be defined from the Boltzmann entropy function as follows:

$$\frac{1}{T} = \left(\frac{\partial S}{\partial U}\right)_{V,N}. \tag{2}$$

One of the earliest papers that postulates the existance of negative temperature due to Boltzmann's equation was N.F Ramsay[3]. He postulates that the only requirement for the existance of negative temperature is that the entropy S should not be a monotonically increasing function of the internal energy U. We can show this with a simple application of Boltzmann's formula to a two-level spin system.

1.1 Two-level system

We can consider a system of N particles which can take two spin values: spin up with energy $+\epsilon$ or spin down with energy 0. If there are N_+ particles with spin up, the total energy of the system is

$$E = \epsilon N_+. \tag{3}$$

We apply Boltzmann's formula by counting the number of microstates in the system to obtain a function for entropy:

$$S_B\left(N_+, N\right) = k_B \ln\left(\frac{N!}{N_+!\left(N-N_+\right)!}\right). \tag{4}$$

We can plot $S_B\left(N_+, N\right)$ for some particle number N to observe the shape of the entropy function in Figure 1. The lowest possible energy state is achieved when all particles are spin down, which is clearly a highly ordered state and corresponds to $S = 0$. When the energy of the system (proportional to N_+) is increases, it is clear from Figure 1 that at a certain energy level, there exists more spin up particles than spin down particles (population inversion) which leads to a decrease in the number of microstates for that energy. Similarly the highest energy state with all particles spin up is also a highly ordered state, $S = 0$.

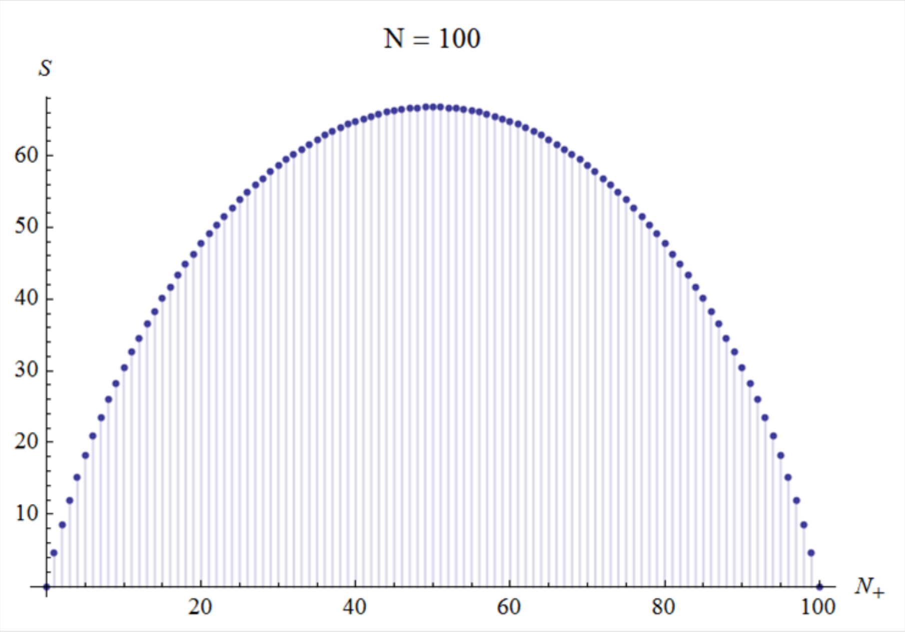

Figure 1: S_B for a two level spin system, $N = 100$. For $N_+ > 50$ the energy decreases with increasing N_+.

Following the definition of temperature in Eq. (2) we obtain the temperature scale as a function of energy and we see in Figure 2 that the region with decreasing entropy corresponds to a negative temperature.

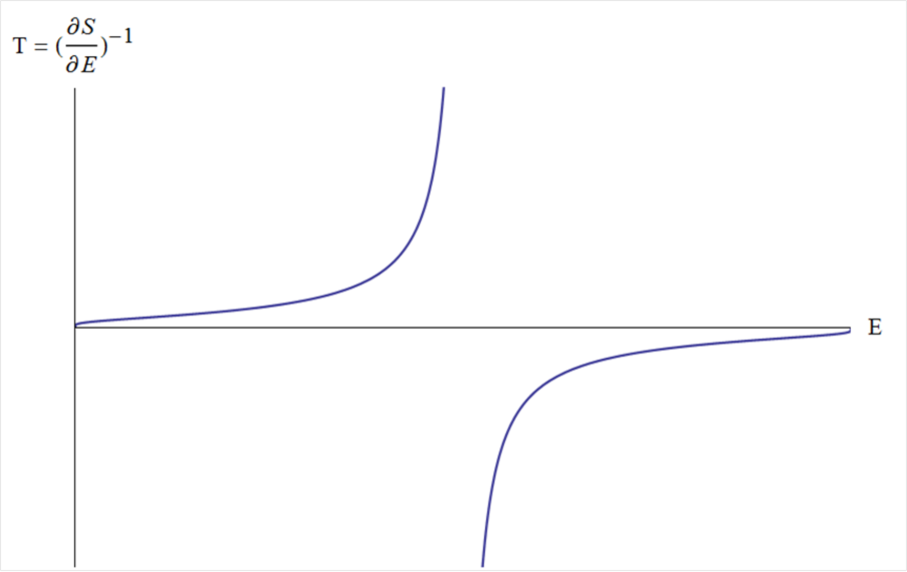

Figure 2: Temperature scale according to Boltzmann's formula. The system in a negative temperature state has more energy.

1.2 Properties of negative absolute temperatures

From the definition of absolute temperature in Eq. (2) we have a numerical scale of absolute temperature[3]:

$$+0^\circ\,\mathrm{K} < +300^\circ\,\mathrm{K} < +\infty^\circ\,\mathrm{K} < -\infty^\circ\,\mathrm{K} < -300^\circ\,\mathrm{K} < -0^\circ\,\mathrm{K}. \tag{5}$$

Essentially this means the negative temperatures are always "hotter" than positive temperatures; if two objects of negative temperature and positive temperature come into contact, heat will flow from the negative temperature object to the positive temperature object. We can intuitively see how this is consistent with the two-level system in Section 1.1, as negative temperatures will correspond to systems with higher energy than positive temperatures.

Negative temperatures cannot exist for system where the upper bound of energy is unbounded. This means that most real world system, such as the classical ideal gas, do not have negative temperatures as kinetic motion of particles usually do not have an upper energy bound. Negative temperatures have been realized in localized spin systems[4–6] where an upper bound of the system energy exists due to its finite, discrete spectrum. However, a recent paper by Braun *et al*[7] in 2013 also established the real world existance of negative temperatures in motional degrees of freedom.

1.3 Dunkel & Hilbert: Negative temperatures do not exist

A paper was recently published by Jörn Dunkel and Stefan Hilbert[1] that made the claim that the Boltzmann definition of entropy is inconsistent both mathematically and thermodynamically. The implications of such a claim would be that negative temperatures are invalid and do not exist. Since the Boltzmann formula not the correct thermodynamic formula, Dunkel and Hilbert proposed an alternate "Gibbs" volume entropy formula (not to be confused with the well-known Gibbs formula, $S = -k_B \sum_i p_i \ln p_i$):

$$S_G\left(E,N\right) = k_B \ln\left(\sum_{i\leq E} \omega\left(i,N\right)\right). \tag{6}$$

The difference between the "Gibbs" formula (S_G) and Boltzmann formula (S_B) is that while S_B at a particular energy E is determined by counting the number of microstates at that energy E, the entropy in S_G is determined by counting the number of microstates at that energy E and all other energies below that. If we revisit the two-level system and plot the entropy again, using the proposed Gibbs formula (Figure 3), we see that entropy is now a strictly monotonically increasing function of energy; negative temperature no longer exists following the definition in Eq. (2).

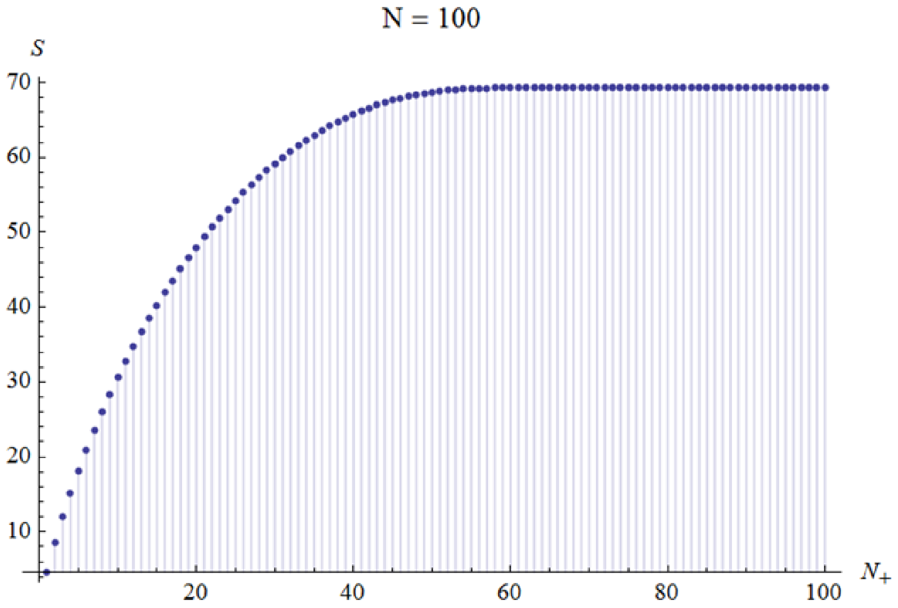

Figure 3: S_B for a two-level spin system, $N = 100$. For $N_+ > 50$ the energy decreases with increasing N_+.

The question now raised is whether this "Gibbs" formula is indeed the correct formula to use to describe entropy, or does Boltzmann formula remain the correct, consistent thermodynamic formula to use? We attempt to answer this question by looking at how entropy and temperature is defined. We have been starting with the definition of entropy in order to define our absolute temperature scale, but is there a way to define our absolute temperature scale without refering to any definition of entropy? It turns out that according to a formulation of thermodynamics by A. B. Pipard[8], we can define the absolute temperature scale by the construction of Carnot cycles.

Part II

Boltzmann entropy and Carnot cycle

Pipard's formulation of thermodynamics outlines a method to obtain an absolute temperature scale via the construction of Carnot cycles. In order to do so, one must define isotherms and adiabat curves of the system, so as to be able to construct a Carnot cycle consisting of isothermal and adiabatic processes. In Section 2, we apply Pipard's formulation to a paramagnet to obtain an absolute temperature scale of the system without reference to the Boltzmann formula for entropy.

2 Absolute temperature scale from Pipard formulation

2.1 Empirical temperature: An Ising thermometer

We start by considering how to first obtain some form of empirical temperature of a system. Empirical temperatures are based on the measurement of physical properties one can observe from the system. For example, using an ideal gas thermometer, we can measure the volume or the pressure of the ideal gas when it is in thermal equilibrium contact with the system in question. With the empirical temperature scale, one obtains the absolute temperature scale by a matter of a fixed reference point (for ITS-90[9], 273.16 K defined at the triple point of water).

2.1.1 One dimensional (1D) Ising model

In our thesis, we use a 1D Ising chain as our empirical thermometer. The 1D Ising model consists of a number of spins N_B, each with a spin value $\sigma_i = \pm 1$ attached, which can either be spin up or spin down. Each spin interacts with its neighbour such that the energy of the system depends of each spin-spin configuration:

$$E_B = -J \sum_{i}^{N_B} \sigma_i \sigma_{i+1} - h' \sum_{i}^{N_B} \sigma_i, \tag{7}$$

where conventionally periodic boundary conditions are applied to the first and last spin: $\sigma_{N_B+1} = \sigma_1$. J represents the interaction strength and h' represents the strength of an external magnetic field. For this part we assume the absence of an external magnetic field ($h' = 0$). We can define an observable property of the Ising model that is a representation of the average energy of the system:

$$\theta \equiv \langle \sigma_1 \sigma_2 \rangle_{E_B}, \tag{8}$$

where the ensemble average $\langle\theta\rangle_{E_B}$ means that we take the mean of the quantity θ over all possible configurations while keeping the energy constant at E_B. If we write out all possible spin configurations of the Ising model for a small value of N_B and compute θ and E_B for each configuration, we can see that θ is a function of the average energy $\langle E_B\rangle$ of the system:

$$\langle E_B\rangle = -JN_B\theta. \tag{9}$$

This can justify our definition of the empirical temperature. We note the range of the empirical temperature: $-1 \leq \theta \leq 1$. Also, we note that the energy of the Ising model is related to the number of domain walls[1] i present in the configuration: $E_B(i) = -JN_B + 2Ji$. This can be illustrated in Figure 4: counting 6 domain walls we obtain an energy of $E_B\,(6) = -10J + 12J = +2J$, which is our expected result if we calculated the energy using Eq. (7).

+ + + ▯ − ▯ + ▯ − ▯ + + + ▯ −▯

Figure 4: 10 particle Ising model. Energy is a function of the number of domain/domain walls (represented as ▯).

2.1.2 Paramagnet system

The system of which we want to obtain the absolute temperature scale of is the spin paramagnet. The paramagnet consists a number of spins N, each with a spin value $\alpha_i = \pm 1$. The energy of the paramagnet system is simply the sum of its spin value:

$$E_P = -h\sum_{i}^{N}\alpha_i. \tag{10}$$

If we define the number of spin up particles in the paramagnet to be N_+ and spin down N_- we have an alternate expression for the energy of the paramagnet:

$$E_P = -hN_+ + hN_- = -hM, \tag{11}$$

with the magnetization defined as $M \equiv N_+ - N_-$. We also note that the energy of the paramagnet is related to the number of spin up particles present in the configuration: $E_P = h\,(N - 2N_+)$.

2.1.3 Heat and work in the paramagnet system

We would like to define heat and work in a paramagnet system. From the First Law of Thermodynamics, the internal energy E of a system can be increased by either letting

[1] A domain wall exists between two neighbouring particles with opposite spins.

the system absorb heat Q or do work W:

$$dE = \delta Q + \delta W. \tag{12}$$

For the paramagnet with energy $E_P = -hM$, its total differential is

$$dE_P = -hdM - Mdh. \tag{13}$$

We define work done on the paramagnet[10]: $\delta W \equiv -Mdh$. Correspondingly, heat flow into the system is $\delta Q \equiv -hdM$. An adiabatic process ($\delta Q = 0$) of the paramagnet is hence the change in the system energy as a result of changing the external magnetic field h, while keeping the magnetization M constant.

2.1.4 Coupled system: Ising thermometer and paramagnet

Our next objective is to obtain the paramagnet isotherm curves by using the Ising model as our thermometer. The Ising thermometer has an empirical property known to be a function of its average energy (Eq. (8)), and we would like to use it to measure the unknown empirical temperature of the paramagnet system which is at some energy E_{P1} (Figure 5).

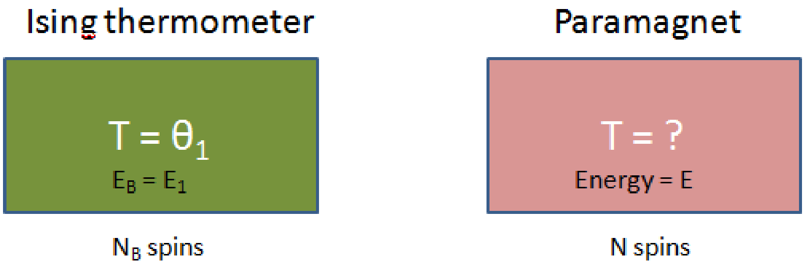

Figure 5: The Ising thermometer has a measurable empirical temperature θ_1. There is no heat transfer between the systems.

We now allow heat transfer to occur between the Ising thermometer and the paramagnet (coupling the systems together) and observe the empirical temperature of the Ising thermometer, θ. Measuring the empirical temperature of the Ising themometer is the same (summing over the spins of the Ising thermometer only), but now the total energy of the two systems E is kept constant:

$$\theta = \left\langle \frac{1}{N_B} \sum_{i=1}^{N_B} \sigma_i \sigma_{i+1} \right\rangle_{E=E_B+E_P}. \tag{14}$$

If it turns out that θ at some value (e.g. θ_2) does not change before and after allowing heat transfer between the systems, we can say that the Ising thermometer and the paramagnet are in thermal equilibrium (Figure 6). At this point we can conclude that the paramagnet possesses the same temperature θ_2 as the Ising model.

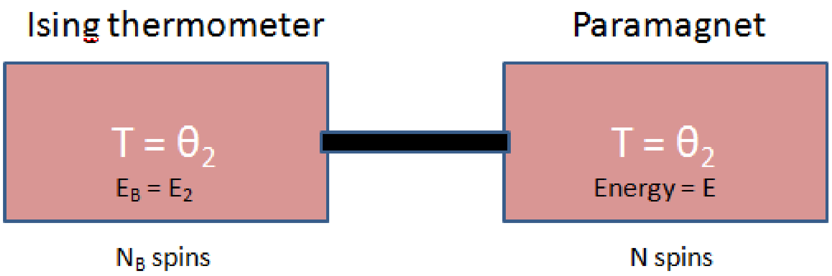

Figure 6: Allowing heat transfer between the two systems; if no heat transfer is observed the paramagnet is in thermal equilibrium with the Ising thermometer.

All possible configurations of a coupled system, at some energy values of the thermometer and the paramagnet, can be written in the form of combinatorial formulas.The average energy of the Ising thermometer in the coupled system is then:

$$\langle E_B\left(i\right)\rangle = \frac{\sum_i E_B\left(i\right) n^B\left(i\right) n\left(N_+\right)}{\sum_i n^B\left(i\right) n\left(N_+\right)}. \tag{15}$$

The number of possible configurations the Ising thermometer has at some energy: $n^B\left(i\right) = \frac{N_B!}{i!(N_B-i)!}$, $i = 0, 2, ...N_B$, the number of possible configurations the paramagnet has at some energy: $n\left(N_+\right)=\frac{N!}{N_+!N_-!}$,

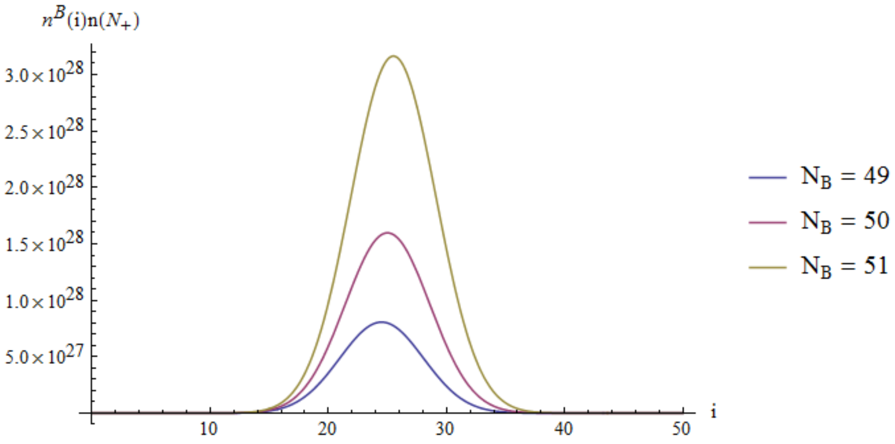

Figure 7: $n^B\left(i\right) n\left(N_+\right)$ for various N_B, where $N_+ = 25$

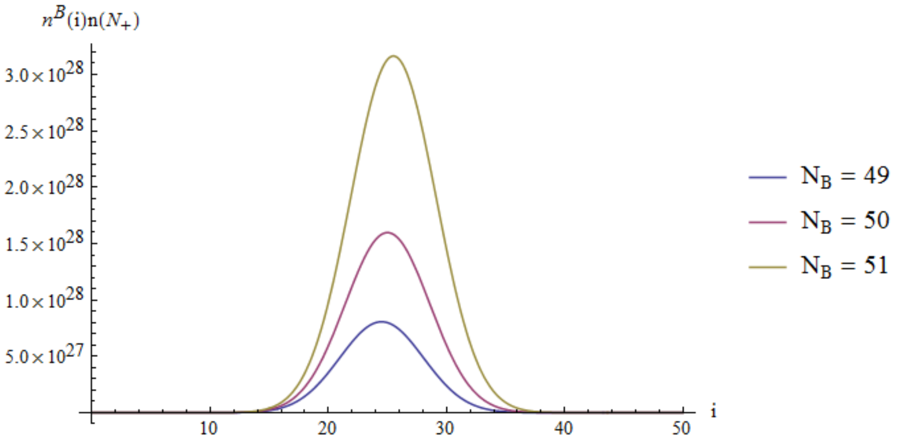

Figure 8: $n^B\left(i\right)n\left(N_+\right)$ for various N, where $N_+ = 25$

Figures 7 and 8 show the effects of varying particle numbers N and N_+ on the number of possible states $n^B\left(i\right)n\left(N_+\right)$. We first note that the peaks of the curves, which is the largest value of $n^B\left(i\right)n\left(N_+\right)$, represent the most probable values of i of the system. We see that as the particle number of either the bath N_B or the paramagnet N increase, the number of possible states increases exponentialy, while the widths of the curves do not increase.

If we consider the thermodynamic limit ($N^B \to \infty$, $N \to \infty$) we can see that the curves would "sharpen" similar to a dirac delta curve: the largest weight (and hence the most probable value) that contributes to the ensemble average $\langle E_B\left(i\right)\rangle$ comes from the largest value of $n^B\left(i\right)n\left(N_+\right)$. Hence, we need only consider the most probable value of $E_B\left(i\right)$ to determine the ensemble average $\langle E_B\left(i\right)\rangle$. We can take the log of $n^B\left(i\right)n\left(N_+\right)$ as we are not concerned with actual values, allowing us to simplify using Stirling's approximation:

$$\begin{aligned}\ln\left(n^B\left(i\right)n\left(N_+\right)\right) =& \ln\left(\frac{N_B!}{i!\left(N_B-i\right)!}\frac{N!}{N_+!N_-!}\right)\\ =&N_B\ln N_B + N\ln N - i\ln i\\ &-\left(N_B-i\right)\ln\left(N_B-i\right)\\ &-N_+\ln N_+ + N_+ - N_-\ln N_- + N_-.\end{aligned} \tag{16}$$

We take its derivative and set it equal to 0, corresponding to a state where $n^B\left(i\right)n\left(N_+\right)$ is a maximum:

$$\begin{aligned}\frac{d}{di}\left(\ln\left(n^B\left(i\right)n\left(N_+\right)\right)\right) = 0 = &-\ln i + \ln\left(N_B-i\right)\\ &-\frac{dN_+}{di}\ln N_+ - \frac{dN_-}{di}\ln N_-.\end{aligned} \tag{17}$$

Rearranging, we have:

$$\ln\left(\frac{i}{N_B-i}\right) = \frac{J}{h}\ln\left(\frac{N_-}{N_+}\right). \tag{18}$$

Using the relations $\theta = 1 - \frac{2i}{N_B}$ and $N\,(or\,M) = N_+ \pm N_-$ we obtain an equation of state for the paramagnet system:

$$\frac{M}{N} = \tanh\left(\frac{h}{J}\tanh^{-1}\theta\right). \tag{19}$$

This equation of state describes the isothermal curves of the paramagnet system using the empirical temperature scale of the Ising thermometer θ.

2.2 Absolute temperature: Carnot cycles

2.2.1 Carnot cycle of the paramagnet

A Carnot cycle is constructed by first considering two isotherms of a system which correspond to temperatures θ_1 and θ_2. Two adiabatic[2] lines are constructed which cut the θ_1 line at points A and B, and the θ_2 line in C and D. The Carnot cycle then consists of an isothermal process along temperature θ_1, changing the state of the system from A to B, an adiabatic process from B to C (which changes the temperature from $\theta_1$to θ_2), an isothermal process along θ_2 from C to D and finally an adiabatic process from D to A. If during the isothermal change from A to B the system takes in an amount of heat Q_1 from the reservoir at θ_1 and along the isothermal process C to D the heat Q_2 is taken in from the reservoir at θ_2, with the completion of the Carnot cycle the system must have performed $Q_1 + Q_2$ amount of work.

Pipard uses Kelvin's law[8] to show that for given values of θ_1 and θ_2 the ratio of $-\frac{Q_1}{Q_2}$ is the same for all Carnot cycles. More specifically, he justifies the introduction of the absolute temperature scale $T(\theta)$ having the property:

$$-\frac{Q_1}{Q_2} = \frac{T(\theta_1)}{T(\theta_2)}, \tag{20}$$

for any Carnot cycle. Once the absolute temperature scale is defined for the system the entropy can be simply found using the Clausius theorem:

$$\int dS = \int \frac{\delta Q_{rev}}{T}. \tag{21}$$

Going back to the coupled Ising-paramagnet system, we can use the equation of state of the paramagnet derived earlier to draw isothermal curves. We can redefine the empirical temperature θ in order to make the plot simpler:

$$\tilde{\beta} \equiv \frac{1}{J}\tanh^{-1}\theta. \tag{22}$$

[2] From our definition of heat and work in Section 2.1.3, the adiabatic curves are straight lines in a M-h graph where the magnetization M is held constant and the external magnetic field h changes.

Keeping in mind that $\tilde{\beta}$ is still the empirical temperature measured from the Ising thermometer. Since $-1 \leq \theta \leq 1$, we have $-\infty \leq \tilde{\beta} \leq +\infty$. The equation of state of the paramagnet is now:

$$M = N \tanh\left(\tilde{\beta} h\right). \tag{23}$$

We construct the Carnot cycle of the paramagnet system using two arbitrary empirical temperatures of the system (Figure 9)

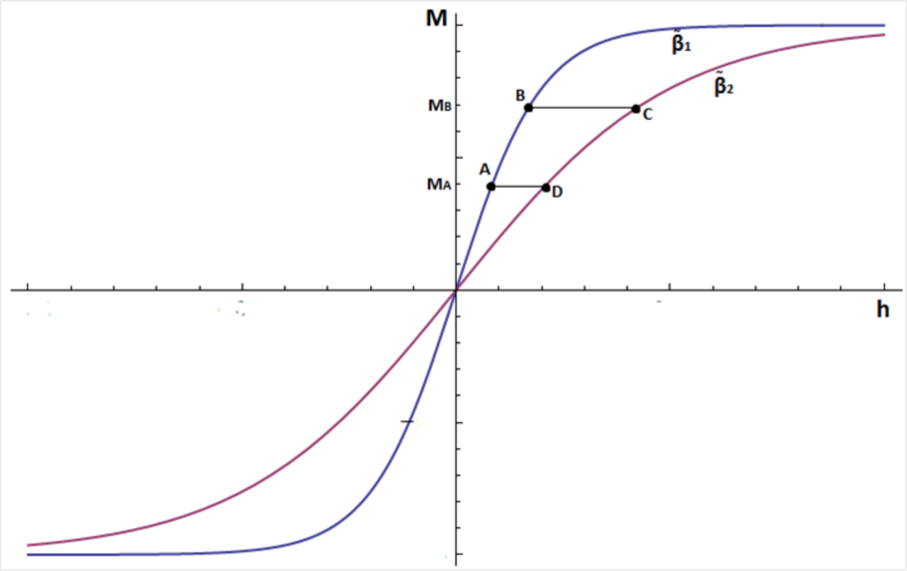

Figure 9: Carnot cycle (ABCD) of the paramagnet at two empirical isotherms, $\tilde{\beta}_1$ and $\tilde{\beta}_2$.

From the isotherm curves we have the following relationships:

$$\tilde{\beta}_1 h_A = \tilde{\beta}_2 h_D = \tanh^{-1}\left(\frac{M_A}{N}\right), \tag{24}$$

$$\tilde{\beta}_1 h_B = \tilde{\beta}_2 h_C = \tanh^{-1}\left(\frac{M_B}{N}\right). \tag{25}$$

We can compute the heat flow along the isotherm $\tilde{\beta}_1$ from A to B:

$$Q_1 = \int_A^B -h dM = -hM|_{h_A,M_A}^{h_B,M_B} + \int_{h_A}^{h_B} M dh = -h_B M_B + h_A M_A + \frac{N}{\tilde{\beta}_1} \ln \frac{\cosh\left(\tilde{\beta}_1 h_B\right)}{\cosh\left(\tilde{\beta}_1 h_A\right)},$$

which can be simplified using Eqs. 24 and 25:

$$Q_1 = \frac{1}{\tilde{\beta}_1}\left(-M_B \tanh^{-1}\frac{M_B}{N} + M_A \tanh^{-1}\frac{M_A}{N} + \frac{N}{\tilde{\beta}_1}\ln\frac{\cosh\left(\tanh^{-1}\frac{M_B}{N}\right)}{\cosh\left(\tanh^{-1}\frac{M_A}{N}\right)}\right). \tag{26}$$

Similarly, the heat flow along the isotherm $\tilde{\beta}_2$ from C to D:

$$Q_2 = -\frac{1}{\tilde{\beta}_2}\left(-M_B \tanh^{-1}\frac{M_B}{N} + M_A \tanh^{-1}\frac{M_A}{N} + \frac{N}{\tilde{\beta}_1}\ln\frac{\cosh\left(\tanh^{-1}\frac{M_B}{N}\right)}{\cosh\left(\tanh^{-1}\frac{M_A}{N}\right)}\right). \tag{27}$$

We can compute the ratio:

$$-\frac{Q_1}{Q_2} = \frac{\tilde{\beta}_2}{\tilde{\beta}_1} = \frac{\frac{1}{J}\tanh^{-1}\theta_2}{\frac{1}{J}\tanh^{-1}\theta_1}. \tag{28}$$

We can then introduce the absolute scale according to Pipard:

$$-\frac{Q_1}{Q_2} = \frac{\frac{1}{J}\tanh^{-1}\theta_2}{\frac{1}{J}\tanh^{-1}\theta_1} \equiv \frac{T(\theta_1)}{T(\theta_2)}, \tag{29}$$

where the absolute temperature scale,

$$T(\theta) = K\frac{1}{\tanh^{-1}\theta}, \tag{30}$$

is a function of the emprical temperature θ with an arbitrary constant K. K can be determined (according to ITS-90) by requiring that the absolute temperature be $T(\theta_{tp}) = 273.16\,\mathrm{K}$ at the empirical temperature of the triple point of water θ_{tp}. With the absolute temperature scale of the paramagnet defined we can then determine the entropy function. From the absolute temperature scale (Eq. (30)) we have the relation:

$$T\tilde{\beta} = \frac{K}{J}. \tag{31}$$

The entropy can be determined using Clausius theorem (Eq. (21)). If we consider a reversible process along the isotherm from A to B,

$$S_B - S_A = \frac{1}{T_1}\left(-h_B M_B + h_A M_A + \frac{N}{\tilde{\beta}_1}\ln\frac{\cosh\left(\tilde{\beta}_1 h_B\right)}{\cosh\left(\tilde{\beta}_1 h_A\right)}\right). \tag{32}$$

We see that entropy is a function:

$$S_{pipard} = -\frac{hN}{T}\tanh\left(\frac{h}{\frac{J}{K}T}\right) + \frac{J}{K}N\ln\cosh\left(\frac{h}{\frac{J}{K}T}\right) + \text{const.} \tag{33}$$

We can compare this with the entropy function of the paramagnet system derived from canonical ensemble (see Appendix A):

$$S_{canon} = -\frac{hN}{T}\tanh\left(\frac{h}{k_BT}\right) + k_BN\ln\cosh\left(\frac{h}{k_BT}\right) + k_BN\ln 2. \tag{34}$$

We see that the entropy functions obtained have the same form, and they are equivalent if the arbitrary constants are equal, $k_B \equiv \frac{J}{K}$. We also see that the canonical derivation of paramagnet entropy gives a value of the unknown constant obtained following Pipard's formulation. In the thermodynamic limit, the microcanonical ensemble and the canonical ensemble are equivalent. Appendices A and B show that we obtain the same expression for energy of the paramagnet when we consider the system in both the canonical and microcanonical ensembles.

2.2.2 Carnot cycle in negative and positive temperature reservoirs

This formulation also works if we consider a Carnot cycle between a positve temperature reservoir and a negative temperature reservoir (Figure 10). Consider an isotherm curve of the paramagnet at temperature $\tilde{\beta}_2' \equiv -\beta_2$ which is of opposite sign to $\tilde{\beta}_2$. It is clear that $h_{C'} = -h_C$ and $h_{D'} = -h_D$.

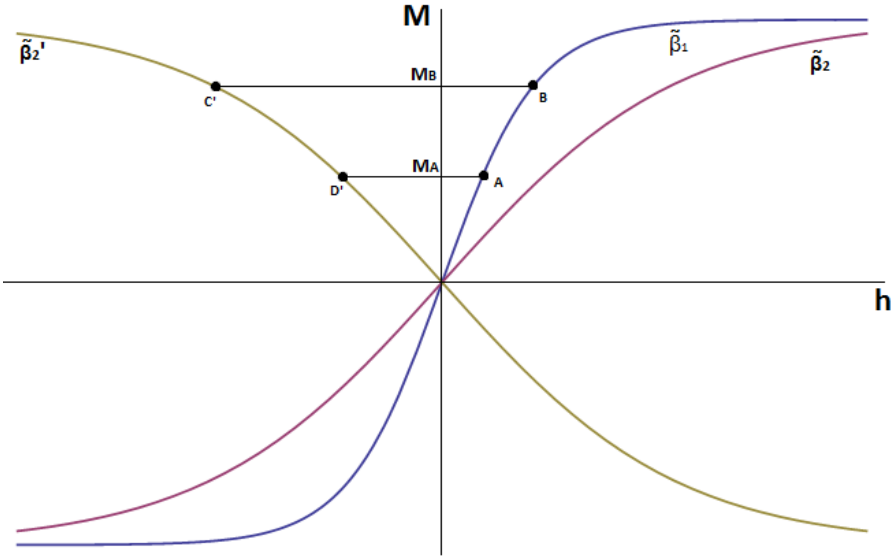

Figure 10: Carnot cycle (ABC'D') with negative and positive temperature reservoirs, $\tilde{\beta}_1$ and $\tilde{\beta}_2'$.

The heat flow along the isotherm $\tilde{\beta}_2'$ from C' to D' is then:

$$Q_2' = -\frac{1}{\tilde{\beta}_2'}\left(-M_B\tanh^{-1}\frac{M_B}{N} + M_A\tanh^{-1}\frac{M_A}{N} + \frac{N}{\tilde{\beta}_1}\ln\frac{\cosh\left(\tanh^{-1}\frac{M_B}{N}\right)}{\cosh\left(\tanh^{-1}\frac{M_A}{N}\right)}\right) = -Q_2. \tag{35}$$

Since heat flows into the system for both isothermal curves, the left term of Eq. (20) does not have a negative sign. The derivation onwards is equivalent to the rest of Section 2.2.1.

2.2.3 A note on Carnot efficiency

The Carnot efficiency equation places a limit to the thermal efficiency of heat engines:

$$\eta_{carnot} = \frac{W}{Q_H} = 1 - \frac{T_L}{T_H}, \tag{36}$$

where Q_H is the heat absorbed by the system at the hot reservoir, T_L and T_H are the temperatures of the cold and hot reservoirs respectively, and W the total work done by the system. One notes that if a Carnot cycle is performed with negative temperature $T_H < 0$ and positive temperature $T_L > 0$, substituting the values into Eq. (36) results in a Carnot efficiency of more than 1. However, one should not confuse this as a proof of free energy generation, and this is explained in a reply by Daan Frenkel and Patrick Warren[11] (summarized below).

Consider a Carnot cycle of a spin system between a "hot" negative temperature reservoir and a "cold" positive temperature reservoir. The Carnot cycle consists of two isothermal processes and two adiabatic processes. We first consider the spin system in an inverted population state (thus having negative temperature), and connect it to the "hot" reservoir. We then perform an isothermal process that decreases the entropy of the spin system by some value ΔS . Since the spin system is an inverted population, by Figure 1 we see that heat must flow from the "hot" reservoir into the spin system to decrease its entropy. Next we remove contact from the reservoir and perform an adiabatic process that inverts the spin population to a normal populatation state (thus having positive temperature) whilst maintaining no change in entropy of the spin system. We then connect the spin system to a "cold" reservoir. In order to be considered a cycle there must be no net change in entropy; we hence perform an isothermal process that increases the entropy of the spin system by the same value in the first step: ΔS. Here we note that since the spin system (at positive temperature) is not an inverted population, by Figure 1 we see that heat must also flow from the "cold" reservoir into the spin system to increase its entropy.

The definition of the Carnot efficiency in Eq. (36) is calculating the net work done per heat absorbed by the system, but we see that the heat absorbed by the system is no longer Q_H only, as the heat also flows into the system during the isothermal process at the cold reservoir. A more correct definition of the efficiency of a Carnot cycle between positive and negative temperature would be:

$$\eta_{carnot} = \frac{W}{Q_H + Q_C}. \tag{37}$$

3 Conclusion

We obtained an expression of entropy for a paramagnet system without using the Boltzmann's formula for entropy. Instead, we construct an empirical temperature scale and used the general property of Carnot cycles to define the absolute temperature scale, from which we use Clausius theorem to obtain the entropy function S_{pipard}. We compared it with the entropy obtained from the canonical ensemble S_{canon} and note that they have the same form. Since the canonical ensemble and the microcanonical ensemble are equivalent in the thermodynamic limit, we claim that our entropy function S_{pipard} is consistent with the entropy function S_{boltz} of the microcanonical ensemble (which was obtained using Boltzmann's formula). As the Pipard formulation of entropy is consistent with the Boltzmann formulation of entropy, this provides support that the Boltzmann entropy formula remains the correct entropy formula to use, instead of the Dunkel Hilbert "Gibbs" volume entropy.

Part III

Random walk Ising model

In this next part, we investigate a modifed Ising model by means of computational simulation in the canonical ensemble. We would like to investigate the properties of a modified Ising model; in which particles are allowed to move freely (i.e. as a fluid), with their energy also dependant on the distance between neighbours. In particular, we will focus on the concept of "pressure" that can be calculated if we model the Ising model as a fluid (Section 6.2). We can also look at how the canonical ensemble behaves when the system has negative temperature. The computer simulation will be carried out in MATLAB using the Metropolis algorithm.

In Section 4 we introduce the properties of the Metropolis algorithm and its relevance in computational statistical mechanics. In Sections 5 and 6 we cover the realization of the modified Ising model as a computer simulation. Section 7 shows the results obtained from the simulations and we talk about any interesting phenomena observed.

4 Markov Chain Monte Carlo: Metropolis algorithm

In the canonical ensemble in statistical mechanics, the partition function Z describes the statistical properties of a system in thermodynamic equilibrium. For a discrete system we have

$$Z = \sum_{\{s\}} e^{-\frac{E}{k_B T}}, \tag{38}$$

where we sum over all the possible microstates $\{s\}$ of the system, each having energy E_s. However when we study systems with many particles or systems with many degrees of motion, it is clear the number of possible configurations become extremely large very quickly. Furthermore, for continous systems (for example ideal gas) the partition function becomes a multidimensional integral. It is usually impossible to evaluate such integrals analytically or computationally. To approximate the partition function we hence use Markov chain Monte Carlo (MCMC) methods[12]. MCMC is a computing technique that allows for sampling of a Markov chain whose stationary probability distribution state is the desired probabilty. MCMC is used particularly when the desired probability distribution is difficult to sample directly or too complex to decompose into simple (few variable) probability functions.

4.1 Markov Chain and the Metropolis algorithm

A Markov chain[13] is a stochastic process that undergoes transition from one state to another, and the probability of transition from one state to another only depends on the current state (i.e. memoryless). By constructing a transition matrix P that represents the probabilities of all possible state transitions, it is possible to calculate the long time probability distribution of the states. We can use a simple weather model[14] as an example of a Markov chain. Figure 11 shows the probabilities of weather conditions (either rainy or sunny), given the weather on the preceding day.

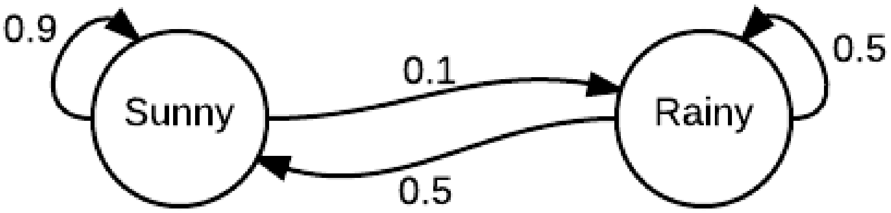

Figure 11: A simple weather model as an example of a Markov chain.

The simple weather system can also be represented as a transition matrix P:

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.5 & 0.5 \end{bmatrix}. \tag{39}$$

The probability distribution of sunny and rainy weather on the next day x_n, can be calculated from the previous day x_{n-1}:

$$x_n = x_{n-1}P. \tag{40}$$

The probability on the nth day x_n, given the weather of the first day x_1 can hence be calculated using P:

$$x_n = x_1 P^{n-1}. \tag{41}$$

To obtain the steady state of the weather, we can solve P as an eigenvalue problem to obtain the steady state solution:

$$\begin{bmatrix} q_1 & q_2 \end{bmatrix} = \begin{bmatrix} 0.833 & 0.167 \end{bmatrix}, \tag{42}$$

which means that in the long term, 83.3% of the days are sunny.

The Metropolis[15] algorithm is a MCMC method that uses random walk to obtain random samples from a Markov chain whose long term probability distribution is equal to the desired probability distribution. The Metropolis algorithm can draw samples from any probability distribution $P(x)$, if the value of a function $f(x)$ that is proportional to the density of P can be computed. The only requirement of the proportionality of $f(x)$ makes the Metropolis algorithm useful as the actual values of $P(x)$ (whose normalization factor may be difficult to compute) need not be computed. The Metropolis algorithm generates a sequence of sample values whose distribution converges to $P(x)$. At each iteration, the next sample is dependant only on the current sample (Markov chain property).

The Metropolis algorithm is outlined below:

1. Start at an initial state x_0.

2. For each iteration t:

   (a) Generate a candidate x',

   (b) Calculate the acceptance ratio $\alpha = \frac{f(x')}{f(x)}$, where $f(x)$is proportional to the probability density,

      i. If $\alpha \geq 1$ accept the new state and set current state to the candidate state: $x_t = x'$

      ii. If $\alpha < 1$ accept the new state with the probability α: If the result is rejected the current state remains the same: $x_t = x_0$.

When Metropolis algorithm randomly moves about the sample space, it sometimes accepts the move and moves to a new sample state and sometimes it rejects the move and remains in the current state. As the acceptance ratio α is an indication of how probable the new candidate state is compared to the current state, the algorithm will tend to stay in high probability density regions and occasionally visit low probability density regions. If the algorithm is allowed to iterate over a long enough period, the Markov chain distribution enters a steady state where the resultant sampling of the Markov chain is equal to the desired probability distribution.

5 The modified 1D Ising model

The 1D Ising model[16] was invented by the physicist Wilhelm Lenz (1888-1957) and solved by Ernst Ising (1900-1998) in his 1920 thesis. In this project, we modify the 1D Ising model to include distance-dependant interation between particles. The modified 1D Ising model consists of a set of particles N in a box of length L, where each particle possess a spin: $\sigma_i = \pm 1$, as well as a position: $0 \leq x_i \leq L$. The system also has an external magnetic field h interacting with it. The particles are allowed to move about the length of the box subject to the random walk conditions specified in the program (see ...). Figure 12 shows an example of the modified 1D Ising model for $N = 5$.

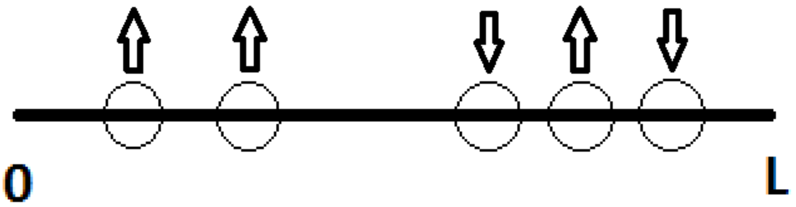

Figure 12: The modified 1D Ising model for $N = 5$ particles in a box of length L.

For each particle-particle interaction not only is the interaction dependant on their spins $\sigma_i \sigma_{i+1}$, but now it is also dependant on the distances between each neighbouring particle, $V(r)$. The energy of the modified 1D Ising model is hence:

$$E = -\sum_{i}^{N} \sigma_i \sigma_{i+1} V\left(|r_i - r_{i+1}|\right) - h \sum_{i}^{N} \sigma_i. \tag{43}$$

We note that the interaction parameter J present in the original Ising model (Eq. (7)) can be subsumed under the potential function $V\left(|r_i - r_{i+1}|\right)$.

5.1 Periodic boundary conditions

To reduce edge effects and better observe bulk effects we can implement periodic boundary conditions: the $(n+1)$th particle is equivalent to the 1st particle. Normally, for the 1D Ising model we need only consider the spins: $\sigma_{N+1} = \sigma_1$; however since the particles now have positions we need also consider the distance between the first and last particle.

This can be easily determined if we consider the box to be infinitely repeating end-to-end (illustrated in Figure 13).

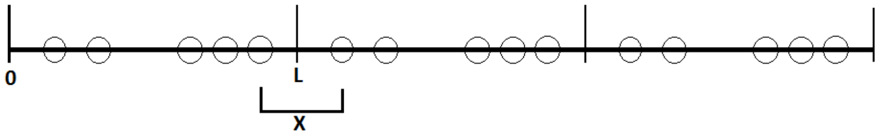

Figure 13: Periodic boundary conditions in a 5 particle modified 1D Ising model. The distance between the first and last particle can be calculated as X.

5.2 Distance dependant potential, $V(r)$

The interaction between the particles in this simulation is modeled as a parabola with a cutoff:

$$V(r) \equiv \begin{cases} \frac{(r-h_-)(r-h_+)}{(h-h_-)(h-h_+)}k, & h - dp \leq r \leq h + dp, \\ 0, & \text{otherwise}, \end{cases} \tag{44}$$

where $h_- \equiv h - dp$, $h_+ \equiv h + dp$.

The parameters h, k, dp, govern the shape of the parabola $V(r)$. The stationary point of $V(r)$ is centered at h and has a value of $V(h) = k$. Langrangian interpolation ensures that $V(r)$ passes through three points: $V(h) = k$, $V(h_-) = 0$ and $V(h_+) = 0$. The cutoff distance of the parabola refers to the variable dp: if r falls outside the region bounded by (h_-, h_+), the potential curve evaluates to 0; there is no interaction between the particles. This cutoff variable hence governs the range of the particle-particle interactions. Figure 14 illustrates $V(r)$ and $V'(r)$ for $h = 1$, $k = -1$, $dp = 1$.

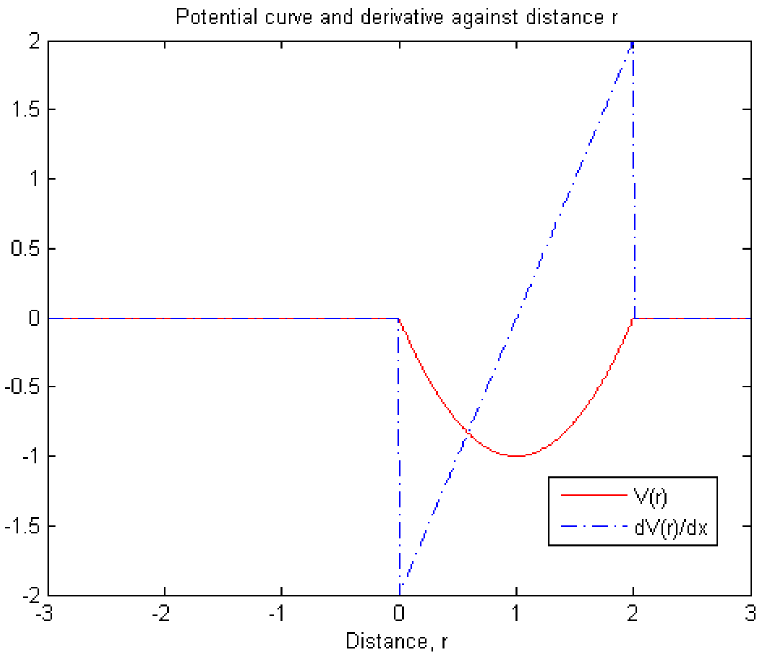

Figure 14: Potential curve $V(r)$ is a parabola whose peak is located at (h, k) with a cutoff distance dp such that $V(h \pm dp) = 0$.

5.2.1 Why implement a cutoff distance?

The decision to implement a cutoff distance was due to our choice of the potential interation curve as a parabola; the large distances r the curve of the parabola means that energy interaction exponentially increases as distance increases. Also, even inverting the potential curve we expect no difference to the system as the particles can simply flip their spins to minimize energy. Without a cutoff distance implemented, quick simulations showed that the spins tend to cluster together to maximize distance between domain walls and spins orient between large distance spin-spin pairs to minimize energy.

Figures 15 and 16 show the behavior of the particle movements with and without a cutoff distance in the potential curve $V(r)$. We can observe that without a cutoff distance, the particles tend to drift together to maximize the distances between two anti-parallel spins so as to minimize the total energy of the system. This is illustrated in Figure 15 as large spaces appearing for $t > 6000$, suggesting that the particles formed two distinct clusters. Implementing a cutoff distance means that the distance dependant interaction between neighbouring spins is relatively short ranged compared to the size of the box.

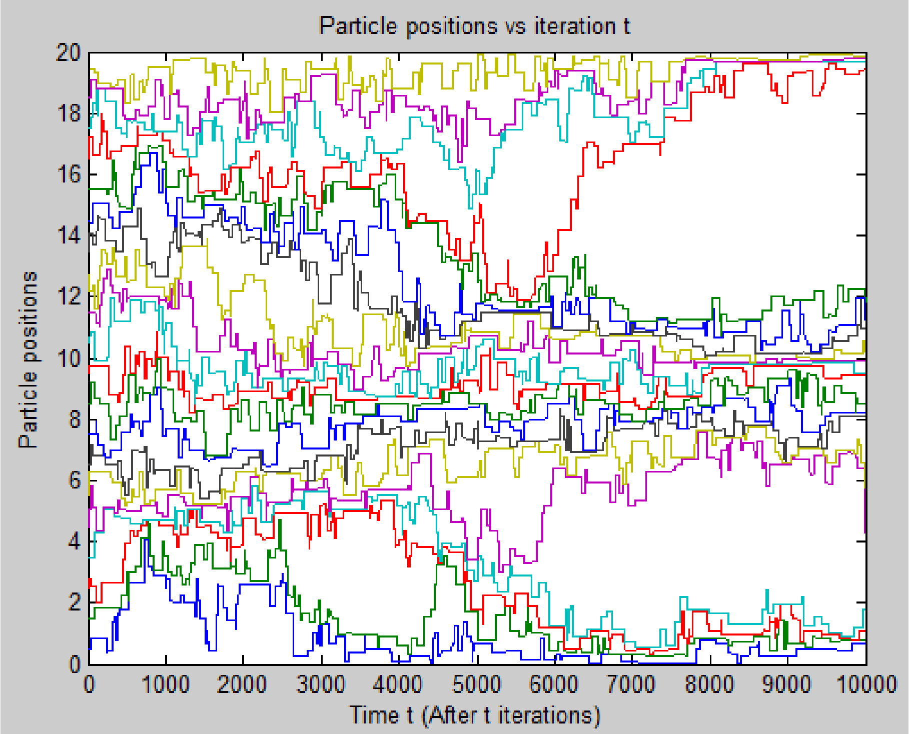

Figure 15: Tracking particle positions for each iteration up till $t = 10000$. Without a cutoff distance, particles tend to stick together to minimize energy.

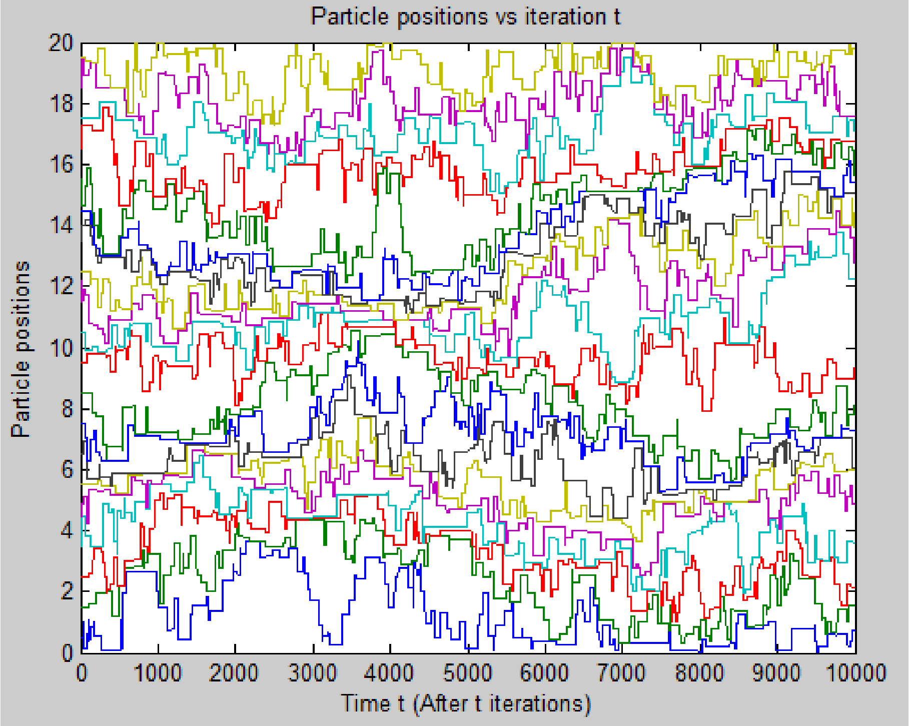

Figure 16: Tracking particle positions for each iteration up till $t = 10000$. With a cutoff, particles no longer appear to stick together.

5.3 Metropolis algorithm in the modified 1D Ising model

We can simulate our modified 1D Ising model in the canonical ensemble as a MCMC, using the Metropolis algorithm. In this case, there is now two possible "moves" the system can make during the random walk process. We can either: 1. Flip the spin, or 2. Move the particle. The pseudo-code for the Metropolis algorithm is then as such:

```
1. Initialize the Ising model at a state $x_0$ according to program parameters: Each
   particle $i$ is assigned a spin $\sigma_i = \pm 1$ and a position $0 \leq x_i \leq L$.

2. For each iteration $t$:

   (a) Do either a spin flip or a position move. In all simulations run we alternate
       between a spin move and a position move.

         i. Spin flip: Generate a candidate state $x'$ by flipping the spin of a random
            particle.
        ii. Position move: Generate a candidate state $x'$ by moving a random particle
            a small distance no larger than $dx$.

   (b) Calculate the new energy of the system, $E'$ and calculate the acceptance ratio
       $\alpha = e^{-\left(\frac{E'-E}{k_B T}\right)}$.

         i. If $\alpha \geq 1$ accept the new state and set current state to the candidate state:
            $x_t = x'$.

            A. If $\alpha < 1$ accept the new state with the probability $\alpha$: If the result is
               rejected the current state remains unchanged: $x_t = x_{t-1}$.
```

5.4 MATLAB implementation

The modified 1D Ising model was simulated in MATLAB. The program is seperated into multiple .m script files, each with a specific function for ease of readability and debugging. A list of parameters that govern the Ising system can be found in Table 1. In addition, the MATLAB code in Appendix C is able to iterate over an array of values *var* containing a range of values to observe the varying of certain parameters (e.g. temperature, length of box, magnetic field strength).

Parameter	Description
type	Spin initialization type (see initialize.m)
N	Number of particles
L	Size of 1D box
T	Temperature
h, k, dp	Potential curve parameters (see Section 5.2)
bf	Strength of magnetic field (h in Eq. (43))
dx	Random walk limit (see move.m)
t	Number of iterations

Table 1: Parameters governing the modified 1D Ising model.

main.m

Main script that performs initialization of the system and iteration of the Metropolis algorithm by calling the following listed functions. The Metropolis algorithm is run for t iterations, and the first 50% of sampled observables are discarded (resulting in $\frac{t}{2}$ samples), assuming that the system enters a steady state equilibrium by then.

initialize.m

Initializes the modified 1D Ising model. Inputs: Number of particles N, Length of box L, type of spin initialization (1, 2 or 3). Output: 1. A position array consisting of N elements, each element p_i corresponding to the x coordinate of the ith particle. The positions are initialized such that each particle is equidistance from its neighbours. 2. A spin array consisting of N elements, each element $s_i = \pm 1$ corresponding to the spin of the ith particle. The spins are initialized depending on the *type* variable specified (1: Random, 2: Alternating, 3: All spin up,+1).

energy.m

Calculates the total energy sum of the system by summing spin-spin interaction and spin-magnetic field interaction (Eq. (43)) by calling hamil.m function. Inputs: Position array, spin array, potential curve parameters h, k, dp. Magnetic field strength bf, Box size L. Output: total energy E.

hamil.m

Calculates the energy between A-th spin and B-th spin. Inputs: Position array, spin array, A, B,h, k, dp, bf, L. Output: Energy between Ath spin and Bth spin.

metropolis.m

Generates a pass/fail on the candidate state based on the Metropolis algorithm by calculating acceptance ratio α and drawing a random number. Inputs: Initial energy $initE$, Final energy $finalE$, Temperature T. Output: 0 if the candidate state is to be rejected, 1 if the candidate state is accepted.

move.m

Moves the nth particle to a new position by choosing a uniformly distributed random value between $(-dx, dx)$. The first and last particles are not allowed to move beyond the box limits $[0, L]$. The particles are also not allowed to pass through one another. Inputs: Position array, index of particle to be moved n, movement limit dx and box size L. Output: Returns -1 if the new position is invalid, else returns the new proposed position.

flip.m

Flips the spin of the ath particle. Inputs: Spin array, index of particle to be moved a. Output: The updated spin array.

potential.m

The potential curve representing distance dependent interaction between neighbour particles; in this simulation a parabola with a cutoff distance (Eq. (44)) is used. Inputs: Distance between two spins r (x in the code), parabola parameters h, k, dp. Output: The value of the potential curve at distance r.

dpotential.m

The derivative of the potential curve in potential.m. (Eq. (48)). Inputs: Distance between two spins r (x in the code), parabola parameters h, k, dp. Output: The first derivative of the potential curve at distance r.

virial.m

Calculates the virial term of the pressure equation (2nd term of Eq. (47) below). Inputs: Position array, spin array, potential curve parameters h, k, dp. Box size L. Output: value of the virial term of the system.

plotterv3.m

Script that assists in plotting the various observables of the system.

6 Observables

6.1 Heat capacity and magnetic susceptibility of the system

The observables in the system measurable in the MCMC are $\langle E\rangle$, $\langle E^2\rangle$,$\langle M\rangle$ and $\langle M^2\rangle$. From these we can also determine the heat capacity C_V and magnetic susceptibility χ of the system:

$$C_V = \frac{\partial \langle E \rangle}{\partial T} = \frac{1}{k_B T^2} \left(\left\langle E^2 \right\rangle - \langle E \rangle^2 \right), \tag{45}$$

$$\chi = \frac{\partial M}{\partial h} = \frac{1}{k_B T} \left(\left\langle M^2 \right\rangle - \langle M \rangle^2 \right). \tag{46}$$

6.2 Pressure

As we model the Ising model as a fluid (the particles are free to move about[3]) we can also measure pressure if we keep temperature constant and vary the size of the box L (See Appendix D for derivation):

$$\langle P \rangle = \left\langle - \frac{\partial F}{\partial V}\bigg|_{T,N} \right\rangle = \frac{N k_B T}{L} - \frac{1}{L} \left\langle \sum_{i=1}^{N-1} \sigma_i \sigma_{i+1} r_i V'(r_i) \right\rangle_{T,N}, \tag{47}$$

where $r_i = |x_i - x_{i+1}|$, and $V'(r)$ is the 1st derivative of the potential function $V(r)$; for the parabola potential this becomes:

$$V'(r) \equiv \begin{cases} -\frac{k(r_- + r_+ - 2r)}{(h - r_-)(h - r_+)}, & h - dp \leq r \leq h + dp, \\ 0, & \text{otherwise.} \end{cases} \tag{48}$$

Eq. (47) consists of two terms: an ideal gas term and a "virial" term, which can be measured from observable values in our simulation. We are interested in the behavior of $\langle P \rangle$ and the virial term as we vary system parameters; in particular we would like to know whether $\langle P \rangle$ can be negative even for a system having a positive temperature.

Figure 17 shows an example calculated pressure of the system. We can see that the total (red) pressure consists of the ideal gas term (blue) and the virial term (green).

[3] Not completely free due to coding, see Section 8.1

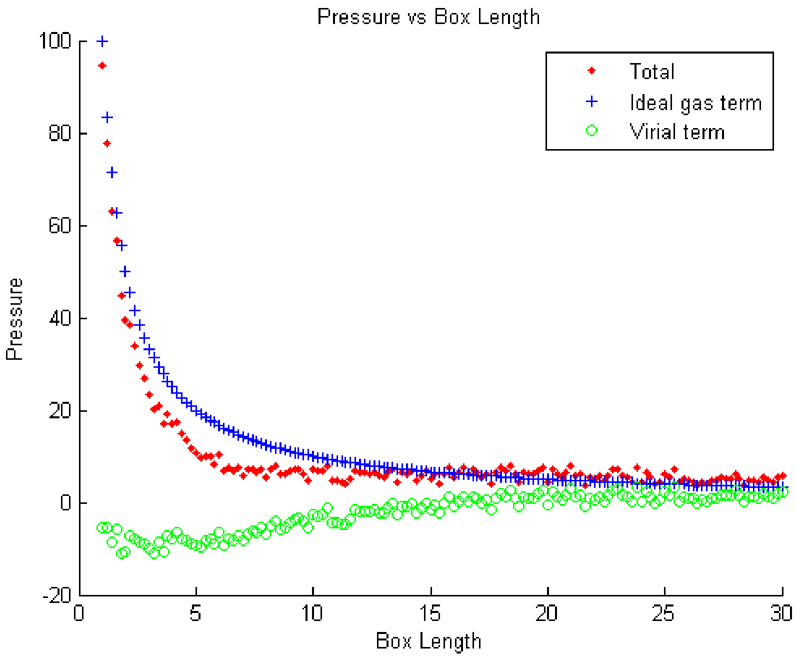

Figure 17: The total pressure (red) of the system is the sum of the ideal gas term (blue) and the virial term (green) (Eq. (58)).

7 Results and observations

7.1 Default parameters

The behavior of the system is governed by a large number of variables (Table 1). To observe the behavior of the system we choose to fix some of the less interesting parameters and vary others parameters. Table 2 lists the default values used for simulations when they are not explicitly stated and also when they are not being varied.

Parameter	Description	Default value
$type$	Spin initialization type (see initialize.m)	3 (all spin up)
N	Number of particles	20
L	Size of 1D box	20
T	Temperature	5
h, k, dp	Potential curve parameters (see Section 5.2)	1, -1, 1
bf	Strength of magnetic field (h in Eq. (43))	0 (no field)
dx	Random walk limit (see move.m)	1
t	Number of iterations	$>$10000

Table 2: Default parameters used in simulation of the modified 1D Ising model.

Figure 18 shows a sample result of a MATLAB simulation, where the system observables are measured as we vary the box length L. Included is also a plot of rejection rates as a result of candidate rejection in the Metropolis algorithm. It is also broken down into various sources of such a candidate rejection:

- Spin Metro: Rejection of a spin flip move due to Metropolis algorithm.
- Random walk: Rejection of a position move due to the new position of the particle being an invalid position (Out of the box or beyond a neighbour particle).
- Position Metro: Rejection of a position move due to Metropolis algorithm.

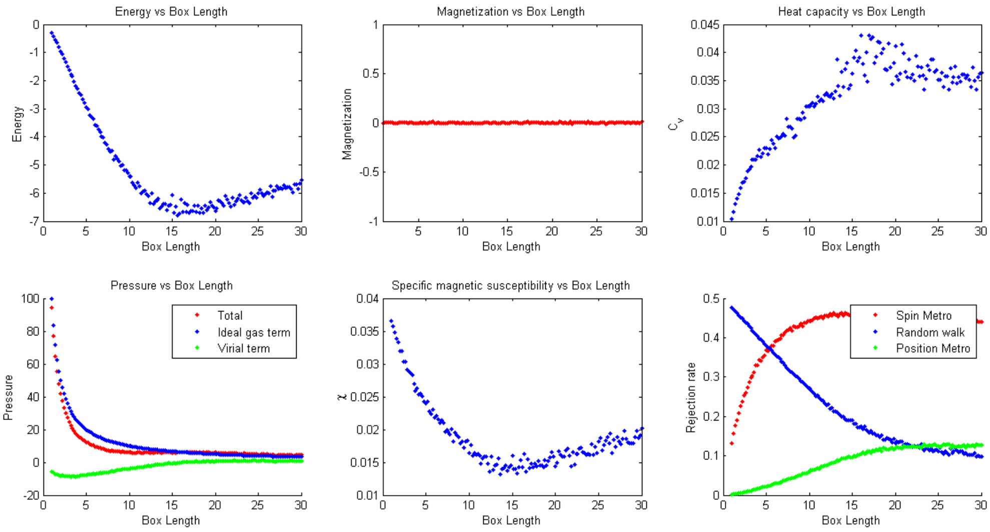

Figure 18: An example result of various observables of the system.

7.2 Checking the MATLAB code with an exact solution

The normal (spin interaction only) 1D Ising model (Eq. (7)) was solved analytically by Ising himself, and we can use the exact solution to check part of our MATLAB code by setting $V(r) = 1$ for all r and setting the Metropolis algorithm to iterate the flipping of spins only.

The exact solution for magnetization of the normal 1D Ising model as a function of external field h and temperature T is known[17]:

$$M(h,T) = \frac{e^{\frac{J}{k_B T}} \sinh\left(\frac{h}{k_B T}\right)}{\sqrt{e^{\frac{2J}{k_B T}} \sinh^2\left(\frac{h}{k_B T}\right) + e^{\frac{-2J}{k_B T}}}} \tag{49}$$

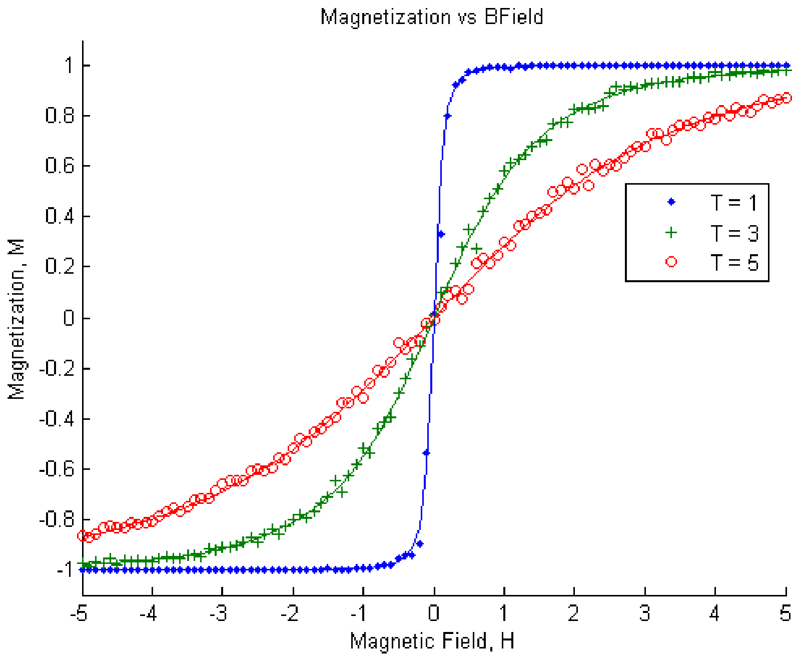

Figure 19: Lines: Theoretical magnetization M as function of magnetic field H. Dots: Magnetization obtained from simulation. For temperatures, $T = 1, 3, 5$.

Figure 19 shows the magnetization M of the system as a function of magnetic field h for various temperatures. The solid lines represent the theoretical values from Eq. (49) while the dots are the values obtained from the MATLAB code.

Similarly, the energy per spin E also has an exact solution[18]:

$$e\left(h,T\right) = -\left(J + \frac{h\sinh\left(\frac{h}{k_BT}\right) + \frac{1}{f}\left(\frac{h}{2}\sinh\left(\frac{2h}{k_BT}\right) - 2J\exp\left(\frac{-4J}{k_BT}\right)\right)}{\cosh\left(\frac{h}{k_BT}\right) + f}\right) \tag{50}$$

where the function f:

$$f = \sqrt{\cosh^2\left(\frac{h}{k_BT}\right) - 2e^{-\frac{2J}{T}}\sinh\left(\frac{2J}{k_BT}\right)} \tag{51}$$

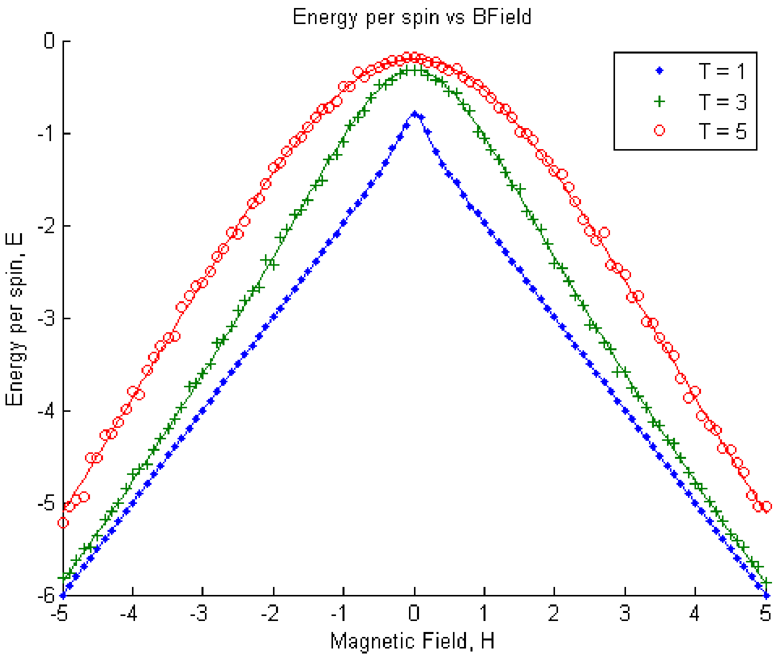

Figure 20: Lines: Theoretical energy per particle e as function of magnetic field h. Dots: Energy per particle obtained from simulation. For temperatures, $T = 1, 3, 5$.

Figure 20 shows the energy per particle e of the system as a function of magnetic field h for various temperatures. The solid lines represent the theoretical values from Eq. (50) while the dots are the values obtained from the MATLAB code. We can see that the observables obtained from the simplified MATLAB code agrees well with theoretical values of magnetization and energy.

7.3 Negative temperatures in the Ising models

As stated in Part I, Boltzmann's entropy formula allows for existance of negative temperatures. We can also set the temperature in our canonical ensemble simulation to a negative value and observe its effects on the system. We simulated the normal 1D Ising model using the simplified MATLAB code in Section 7.2, measuring $\langle E \rangle$ and $\langle M \rangle$ while varying the temperature. The system was initialized in an all-spins-up configuration (ferromagnetic), corresponding to the lowest energy state.

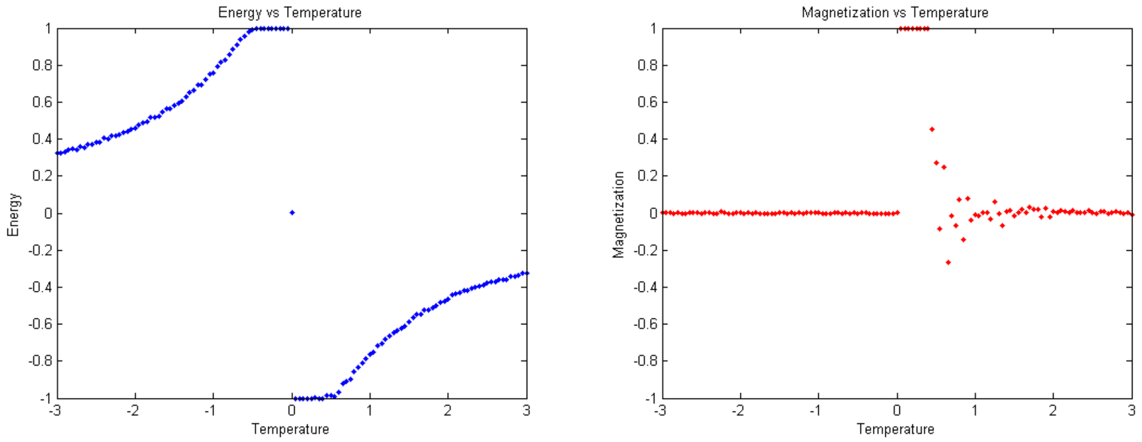

Figure 21: Energy and magnetization vs temperature in the simple 1D Ising model.

Figure 21 shows the energy and magnetization of the system for varying temperatures. We can see that for low positive temperatures, it is ferromagnetic and remains ferromagnetic as it is difficult to transition to a state with higher energy according to the Metropolis algorithm. At higher positive temperatures it is easier to jump to higher energy states, so the system becomes anti-ferromagnetic again.

We see that the system is anti-ferromagnetic for negative temperatures. Calculating the acceptance ratio α in the Metropolis algorithm, a negative temperature value in the canonical ensemble would give more probability weightage to a state of higher energy than lower energy. We can see that for the region of negative temperature, the total energy of the system is larger than the region of positive temperature. This is consistent with our intuitive understanding of negative temperatures outlined in Section 1.2, where a system with negative temperature always possesses more energy than a system with positive temperature.

We can do the same for the modified Ising model. Figure 22 plots the energy and magnetization of the system for varying temperatures. For the modified Ising model to be consistent with the simple Ising model, we choose a positive value for the spin-spin interaction, $k = +1$. This means that according to Eq. (43) a ferromagnetic state corresponds to a higher energy level than an anti-ferromagnetic state, which is consistent with the simple 1D Ising model. We see similar behavior at low positive temperatures; the system remains ferromagnetic (due to being initially ferromagnetic), which then becomes anti-ferromagnetic as the temperature increases.

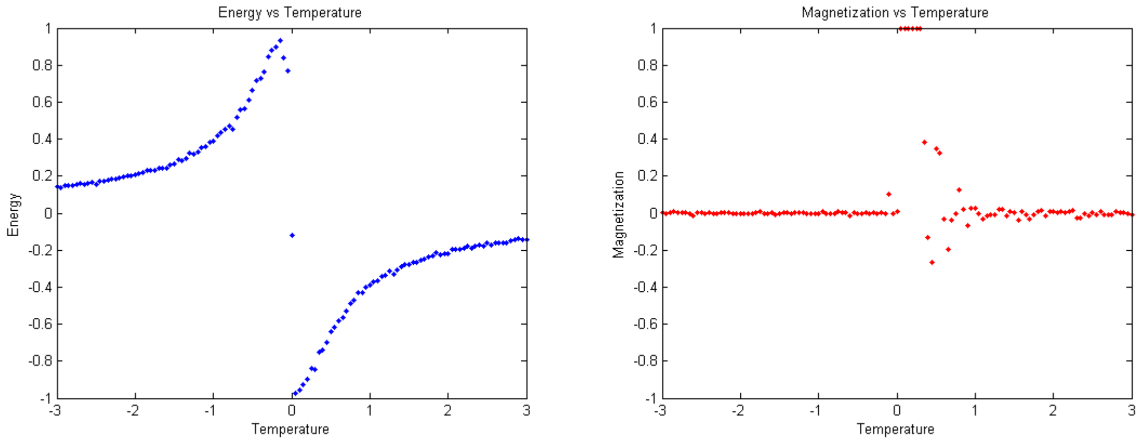

Figure 22: Energy and magnetization vs temperature in the modified 1D Ising model.

7.4 Spin-spin interaction strength k

The parameter k changes the y-coordinate of the peak of the distance potential $V(r)$ (Refer to Figure 14). A larger magnitude of k would increase the interaction strength between neighbour spins due to distance. Simulations are run in the absence of an external magnetic field ($h = 0$). Figure 23 shows the effects of different negative k values on the virial term as we vary the box length, and Figure 24 shows the effects of different positive k values on the virial term as we vary the box length.

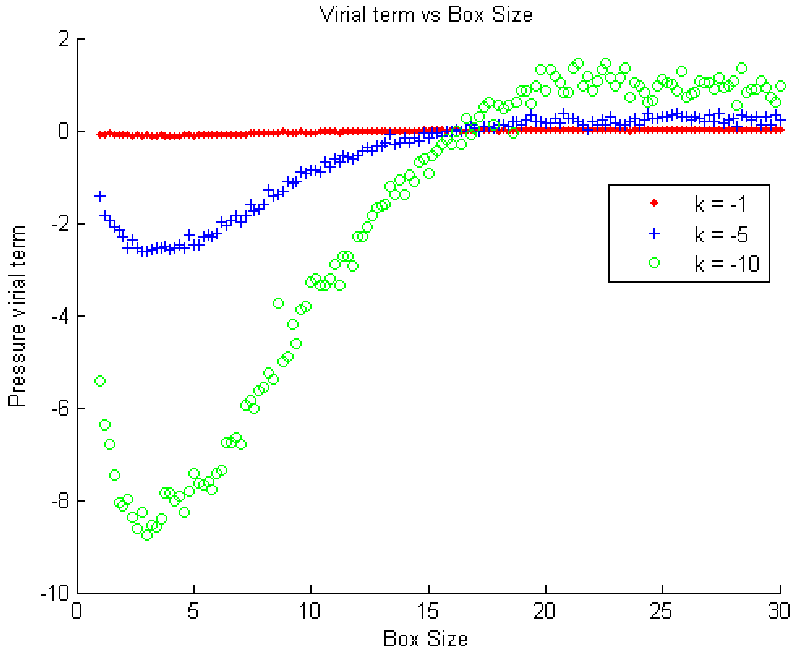

Figure 23: Virial pressure term vs Box size for negative k. A larger k results in a larger virial term.

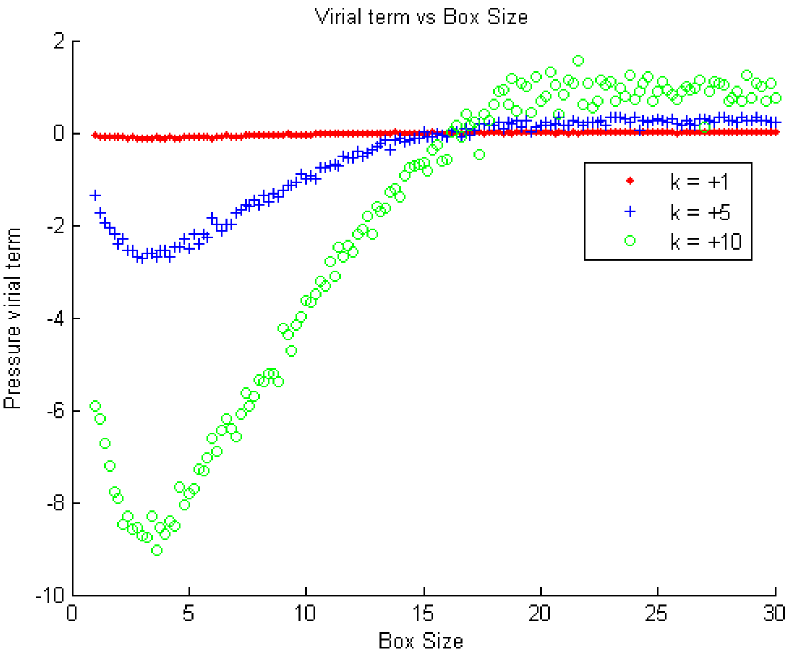

Figure 24: Virial pressure term vs Box size for positive k. The virial term does not change signs.

We note that changing the sign of k does not cause the virial term to change signs[4]. We can observe that a larger $|k|$ would make the virial term of the pressure larger in magnitude. If the interaction strength is large enough, the negative contribution by the virial term to the pressure equation can be larger than the ideal gas term, resulting in a negative value for total pressure (See Figure 25). In the paper by Braun *et al,* they determined that "absolute pressure and temperature necessarily have the same sign in equilibrium, $P/T \geq 0$"[2]. In our simulation of the 1D Ising fluid, however, we can see that we can obtain a negative total pressure even when the system has a positive temperature:

[4] In fact, positive k curves look very similar to negative k curves.

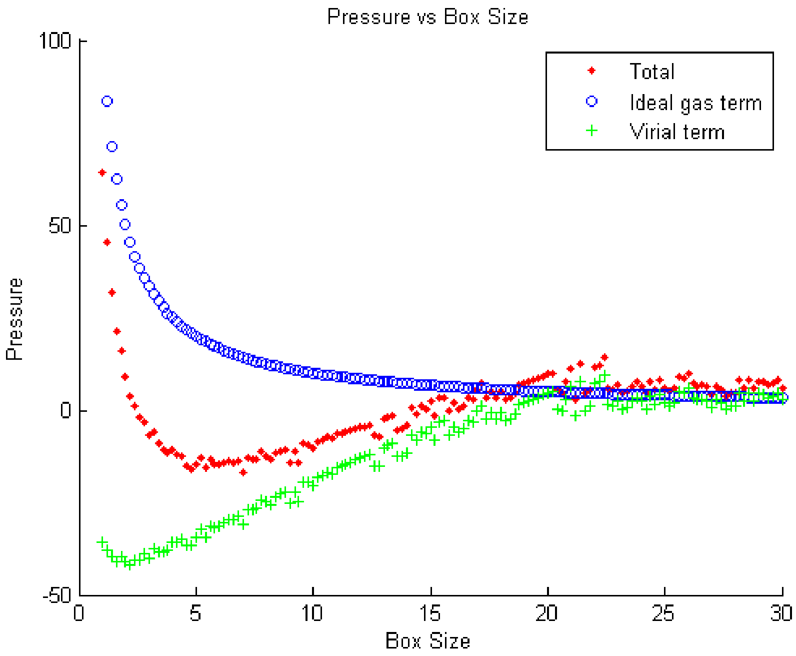

Figure 25: Pressure vs Box length for $k = -30$, $T = 5$. In the region $3 < L < 15$ the total pressure is negative.

7.5 Aligning the particle spins with an external magnetic field

In the previous section, we noted that the virial term is becomes more negative when we decrease box size L and spin interaction strength k in the absence of an external magnetic field. For those results, we also measured the magnetization of the system and realized that the magnetization is constantly near 0, as it seems that anti-parallel spins result in the system having lower energy as a result of Eq. (43). If we now introduce an external magnetic field, we can cause the particles to align in a specific direction. Similar simulations to Section 7.4 are run where the box size is varied, but this time we fix the interaction strength, $k = -10$, and vary the value of the external magnetic field h.

If the external magnetic field h is large enough, one expects that the system would prefer to align as parallel spins as it would be a state with lower energy. Figure 26 shows the effects of an external magnetic field that was large enough to align the particle spins ($|h|$ large enough that magnetization M is ± 1 for all values of L). We see that as the box size decreases, the viral term becomes more positive for both positive and negative magnetic fields. We also observe that the virial curves coincide for both a positive and negative external magnetic field h. This suggests that the system does not distinguish between a postive and a negative external magnetic field.

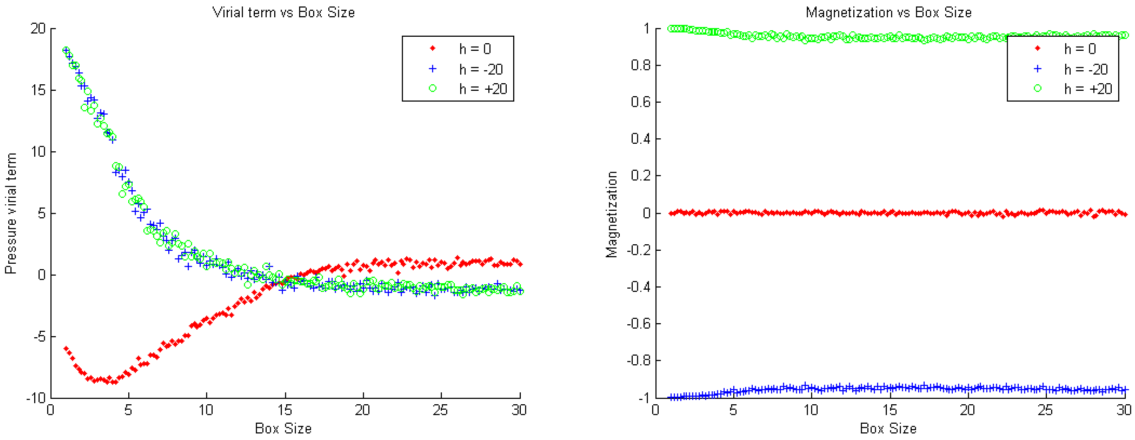

Figure 26: Virial term (left) and magnetization (right) vs Box length for $k = -10$. The virial curves for $+1$ and -1 magnetization coincide.

8 Further work/Issues

8.1 Random walk conditions

To make the code simple we do not allow particles to move past each other during the Metropolis position move: this means that the positions of the spins are all ordered: $x_i < x_{i+1}$. However, this means that there is a "pseudo" infinitely repulsive potential interaction between particles when the distance is 0 (pseudo because flipping spins does not result in infinitely attractive interaction). Similarly, the code does not allow the first and last particles to move beyond the box limits, although ideally periodic boundary conditions would mean that a particle moving past the edges of the box would simply wrap around to the other edge of the box. The illegal moves are considered a rejection of the candidate state according to Metropolis algorithm. This would affect the sampling of observables, as they are rejected not because of a higher energy state according to the Metropolis algorithm.

Further work can be done in the MATLAB code to allow for particles to freely move past one another, where the particle indexing updates itself to reflect the correct neighbours.

8.2 Potential curve alternatives: Linear, Gaussian potential

The potential curve used for the distance dependant interaction was a parabola; this introduced the problem of particles clumping together as mentioned in Section 5.2.1. The implementation of a cutoff distance dp is another parameter that makes the system more complicated to manage. To avoid the need to implement a manual cutoff distance in our code, we can consider different choices of the potential curve.

The parabola potential increases exponentially in value as distance increases, and this is likely the cause of clumping as particles tend to maximize distances between each other. If we choose a linear curve as our potential curve, since the sum distance between particles always equal to box size, particles may no longer seek to maximize distances.

We can also consider using a Gaussian curve; as the curve (and also its derivative) tends to 0 as r is further from the Gaussian peak, this means that the particle interaction becomes negligible for larger distances; there is no need to manually implement a cut off. We can modify the Gaussian width instead to change the range of the particle interactions.

Also, the parabola potential curve is not a realistic potential that represents interactions between actual physical atoms and molecules. Is it possible to simulate this modified Ising model with a more realistic potential curve? The Lennard Jones potential does not seem to be viable as the strength approaches infinity when r approaches 0; in our Ising fluids the particles can change spins to change the sign of the interaction strength.

8.3 Absolute magnetization $|\langle M \rangle|$

In Section 7.4 we saw that the pressure virial curves remain negative for both positive k and negative k values. Also, in Section 7.5 we found that the pressure virial curves of the system coincide when the magnetization of the system is $M = +1$ and $M = -1$. It seems that the modified Ising model does not distinguish between spin-up and spin-down states.

In that case, for simulations where we set the potential curve parameter k to be negative (i.e. anti-ferromagnetic has a lower energy state) we may run into issues with the calculation of average magnetization $\langle M \rangle$, if the temperature of the system allows it to jump between large energy levels (i.e. high positive temperature). For system that favors ferromagnetism, a spin up configuration or a spin down configuration are equally favorable conditions. If the temperature is large enough, the system can jump from a $+1$ magnetization to a -1 magnetization easily if the iteration is large enough. If the system then spends significant time in both $+1$ and -1 configurations, the average magnetzation can end up near zero (indicating anti-ferromagnetism), even though the sample is in reality ferromagnetic.

When doing simulations without an external magnetic field h, the absolute magnetization $|M|$ should hence be sampled instead for a more accurate indication of ferromagnetic behavior.

8.4 Effects of other parameters not investigated in the simulations

The modified 1D Ising model is governed by many parameters. In our simulations, we only varied a few choice parameters (k, bf, T, L) whilst keeping other parameters as default values. For our simulations the graphs plotted are either for varying temperature

T or varying box size L. For future analysis, other parameters of the system such as cutoff distance dp, number of particles N and spin initialization type could also be varied to observe their effect on the system.

8.5 Extending to 2D Ising model

The Ising models simulated in this thesis only involve particles in 1D. Since the spin-only, square lattice Ising model has been solved analytically for both 1D and 2D, we know that there exists a phase transition[17] between the ferromagnetic and paramagnetic phases for a 2D spin-only Ising model. Further work on the modified Ising model could be extended to higher dimensions, where the particles are allowed to freely move in a 2D box. However, one must then consider what meaning nearest neighbour interactions has when particles can move freely in space. In a spin-only 2D Ising model, particles are fixed in a lattice and spin-spin interactions for any one particle is equivalent to considering only nearest neighbour interactions in the lattice. However, if we let particles move freely in 2D, it is likely that two originally neighbouring particles (hence having spin-spin interaction) may drift apart (from random walk) far enough that each particle has completely different nearest neighbours. One idea could be to recalculate the nearest neighbours of every particle for every iteration, which can introduce large orders of computational complexity.

9 Conclusion

In Part III, we constructed a Monte Carlo simulation of a modified version of the 1D Ising model, where particles also interact with a distance dependant potential. The computer simulation was carried out in MATLAB and the code allows for one to describe the system with a number of parameters. The MATLAB code was verified and is consistent with exact known solutions for the simple 1D Ising model. We observed that if distance interaction is strong enough, the system can exhibit negative total pressure even for positive temperatures. This is in contrast to the statement by Braun *et al* that negative pressures only exist for positive temperatures. We also observed systems in the absence of a magnetic field do not distinguish between $+1$ magnetization states and -1 magnetization states. The computer simulation allows for us to obtain a general intuition of the fluid behavior of the modified 1D Ising model. For future work, we could consider investigating the effects of varying other system parameters. We can also consider using other potential curves than the parabola to eliminate the problem of needing a cutoff distance. We can also consider extending this modified 1D Ising model into two or more dimensions.

We simulated a system of spin particles which move freely and have short ranged distance-dependent interaction. This simulation of an Ising fluid is crudely similar to ferrofluids, which are colloidal liquids which consists of particles possessing individual magnetic moments. Indeed, there have been work done in magnetic fluid simulations that use an Ising spin fluid system[19;20] that is similar to the modified 1D Ising model that we have set up. It may be possible in the future, with further work, to be able to simulate the behaviour of real ferrofluids with our code.

References

[1] Dunkel, Jorn; Hilbert, Stefan. "Consistent thermostatistics forbids negative absolute temperatures". *Nature Physics* 10 (2013).

[2] Braun, S., Ronzheimer, J., Schreiber, M., Hodgman, S., Rom, T., Bloch, I., & Schneider, U. (n.d.). “Negative Absolute Temperature for Motional Degrees of Freedom”. *Science*, Supplementary materials (2013).

[3] N. F. Ramsey, “Thermodynamics and statistical mechanics at negative absolute temperatures”. *Phys. Rev.* 103, 20 (1956).

[4] E. M. Purcell, R. V. Pound, “A nuclear spin system at negative temperature”. *Phys. Rev.* 81, 279 (1951).

[5] A. S. Oja, O. V. Lounasmaa, “Nuclear magnetic ordering in simple metals at positive and negative nanokelvin temperatures”. *Rev. Mod. Phys.* 69, 1 (1997).

[6] P. Medley, D. M. Weld, H. Miyake, D. E. Pritchard, W. Ketterle, “Spin gradient demagnetization cooling of ultracold atoms”. *Phys. Rev. Lett.* 106, 195301 (2011).

[7] Braun, S., Ronzheimer, J., Schreiber, M., Hodgman, S., Rom, T., Bloch, I., & Schneider, U. (n.d.). “Negative Absolute Temperature for Motional Degrees of Freedom”. *Science*, 52-55 (2013).

[8] Pippard, A.. “Elements of Classical Thermodynamics for Advanced Students of Physics”. *Cambridge England: University Press* (1957).

[9] M.L. McGlashan. “The international temperature scale of 1990 (ITS-90)”. *The Journal of Chemical Thermodynamics*, 653-663 (1990).

[10] Barrett, M., & Macdonald, A. (n.d.). “The Form of Magnetic Work in a Fundamental Thermodynamic Equation for a Paramagnet”. *American Journal of Physics*, 613-613 (1999).

[11] D. Frenkel & P. B. Warren. “Gibbs, Boltzmann, and negative temperatures”. *American Journal of Physics*, 83, 163 (2015).

[12] Brooks, S. “Handbook of Markov chain Monte Carlo”. *Boca Raton: CRC Press/Taylor & Francis* (2011).

[13] Norris, James R. “Markov chains”. *Cambridge University Press* (1998).

[14] “Examples of Markov chains”. *Wikipedia.* Retrieved 2015, from http://en.wikipedia.org/wiki/Examples_of_Markov_chains.

[15] N. Metropolis et al. “Equation of State Calculations by Fast Computing Machines”. *Journal of Chemical Physics*, 21, 1087-1092 (1953).

[16] B. A. Cipra. “An introduction to the Ising model”. *American Mathemathical Monthly*, 94(10), 939-959 (1987).

[17] Baxter, R.. “Exactly solved models in statistical mechanics”. *London: Academic Press* (1982).

[18] Adler, J. (n.d.). “1D & 2D Simulations Vs Theory”. Retrieved 2015, from http://phycomp.technion.ac.il/~lior/1D&2DVsTheory.html.

[19] M.J.P. Nijmeijer, J.J. Weis. “Simulation of magnetic fluids”. *Annual Review of Computational Physics IV, World Scientific* (1996).

[20] W. Fenz, R. Folk. “Monte Carlo simulation of phase equilibria in Ising fluids and their mixtures”. *Condensed Matter Physics*, 6, 4(36), 675-686 (2003).

[21] Frenkel, D., & Smit, B. “Understanding molecular simulation from algorithms to applications (2nd ed.)”. *San Diego: Academic Press* (2002).

[22] B. Veytsman, M. Kotelyanskii. “Thermodynamic Properties in NVT Ensemble”. Retrieved 2015, from http://borisv.lk.net/matsc597c-1997/simulations/Lecture4/node1.html.

Appendix A Paramagnet in the canonical ensemble

The paramagnet has the Hamiltonian:

$$\mathcal{H} = -h \sum_{j=1}^{N} \sigma_j$$

The partition function of a single spin:

$$\mathcal{Z}_1 = e^{\frac{-h}{k_B T}} + e^{\frac{+h}{k_B T}} = 2\cosh\left(\frac{h}{k_B T}\right)$$

So the combined partition function of N spins:

$$\mathcal{Z} = \left(2\cosh\left(\frac{h}{k_B T}\right)\right)^N \tag{52}$$

We can obtain energy of the system:

$$E = -\frac{\partial}{\partial\beta} \ln \mathcal{Z} = -Nh\tanh(\beta h) \tag{53}$$

And entropy:

$$\begin{aligned} S &= \frac{E}{T} + k_B \ln \mathcal{Z} \\ &= -\frac{hN}{T}\tanh\left(\beta h\right) + k_B N \ln\left(2\cosh\left(\beta h\right)\right) \\ &= -\frac{hN}{T}\tanh\left(\frac{h}{k_B T}\right) + k_B N \ln\cosh\left(\frac{h}{k_B T}\right) + k_B N \ln 2 \end{aligned}$$

Appendix B Paramagnet in microcanonical ensemble

The number of states available in a N-particle paramagnet with N_+ spins up is

$$\Omega\left(N_+\right) = \frac{N!}{N_+!\left(N-N_+\right)} \tag{54}$$

Using Boltzmann's formula for entropy, and dividing Ω by 2^N to account for correct Boltzmann counting we obtain an entropy function:

$$\begin{aligned} S_{boltz} =& k_B \left(\ln\left(N!\right) - \ln\left(N_+!\right) - \ln\left(\left(N-N_+\right)!\right) - N\ln\left(2\right)\right) \\ =& k_B \left(N\ln N - \left(\frac{N}{2}-\frac{E}{2h}\right)\ln\left(\frac{N}{2}-\frac{E}{2h}\right) - \left(\frac{N}{2}+\frac{E}{2h}\right)\ln\left(\frac{N}{2}+\frac{E}{2h}\right) - N\ln 2\right) \end{aligned}$$

(55)

where we have applied Stirling's approximation and made use of the relationships: $N_+ = \frac{N}{2} - \frac{E}{2h}$ and $\left(N-N_+\right) = \frac{N}{2}+\frac{E}{2h}$. Using the definition of absolute temperature in Eq. (2):

$$\begin{aligned} \frac{1}{T} = \frac{\partial S}{\partial E} =& \frac{k_B}{2h}\left[\ln\left(\frac{N-\frac{E}{h}}{N+\frac{E}{h}}\right)\right] \\ =& \frac{k_B}{h}\frac{1}{2}\left[\ln\left(\frac{1-\frac{E}{Nh}}{1+\frac{E}{Nh}}\right)\right] \\ =& \frac{k_B}{h}\tanh^{-1}\left(-\frac{E}{Nh}\right). \end{aligned} \tag{56}$$

We can rearrange this to obtain the energy of the system:

$$E = -Nh\tanh\left(\frac{h}{k_BT}\right) \tag{57}$$

which is equivalent to the energy of the paramagnet obtained via the canonical ensemble in Eq. 53.

Appendix C MATLAB Code

The following pages are the MATLAB code of the modified 1D Ising model as outlined in Section 5.4.

The .m files are listed in the following order:

1. main.m
2. initialize.m
3. energy.m
4. hamil.m
5. metropolis.m
6. move.m
7. flip.m
8. potential.m
9. dpotential.m
10. virial.m
11. plotterv3.m

main.m

```
clear all;clf;clc;

global var                      % global vars for plotting
global varstr
global enmean
global magmean
global chi
global cv
global fail1mean
global fail2mean
global fail3mean
global pressure
global press1
global press2

tic
oldtime = 0;

%%%%%%%%% parameters %%%%%%%%%

type = 1;           % type of initialization:
                    %   1 for random, 2 for alternating, 3 for all spins up

N = 20;             % number of atoms
L = 20;             % length of side of 1D box
T = 5               % temperature in units kbT

h = 1;              % potential curve: parabola whose peak is centered at (h,k).
k = -10             %
dp = 1;             % potential = 0 if distance from potential peak curve is > dp

bf = 0              % external b field applied onto the system

dx = 1;             % maximum move interval for Metropolis

t = 100000;         % no of MC steps per var

var = 1:0.2:30;         % variable to be varied
varstr = 'Box Length';

%%%%%%%% end parameters %%%%%%%%

%%%%%%%% begin simulation %%%%%%

for j = 1:length(var)

    L = var(j);         % variable is set here
    fprintf('%s: %.3f\n',varstr,var(j));

    [position spin] = initialize(N,L,type);         % initialize system

    fail1 = 0;                                      % initialize fail counters
    fail2 = 0;
    fail3 = 0;

    for i = 1:t         % begin MCMC

        switch mod(i,2)         % alternete between spin flip and position move

            case 1              % PARTICLE SPIN FLIP

                initE = energy(position,spin,h,k,dp,bf,L);      % calculate initial energy
                ran = randi(N);
                spin = flip(spin,ran);                          % flip a random spin
                finalE = energy(position,spin,h,k,dp,bf,L);     % calculate new system energy

                pf = metropolis(initE,finalE,T);                % metropolis pass/fail

                if (pf == 0)                                % if fail
                    spin = flip(spin,ran);                  % flip back the spin
                    fail1 = fail1+1;                        % track failure count: SPIN FAIL
                end

            case 0              % PARTICLE POSITION MOVE

                initE = energy(position,spin,h,k,dp,bf,L);  % calculate initial energy

                ran = randi(length(position));              % pick a random spin index
                oldpos = position(ran);                     % store old position of spin
                newpos = move(position,ran,dx,L);           % generate new position of spin
                                                            % if new position invalid returns -1

                if (newpos == -1)                   % if new pos is invalid
                    fail2 = fail2+1;                % track failure count: MOVE FAIL
                else
                    position(ran) = newpos;         % if valid, update position of nth spin
                end

                finalE = energy(position,spin,h,k,dp,bf,L);     % calculate final system energy

                pf = metropolis(initE,finalE,T);    % metropolis algorithm pass/fail

                if (pf == 0)                        % if new system not accepted due to
                metropolis
                    position(ran) = oldpos;         % move back the nth particle
                    fail3 = fail3+1;                % track failure count: MOVE METROPOLIS FAIL
                end


        end

        en(i) = energy(position,spin,h,k,dp,bf,L);      % sample energy
        mag(i) = sum(spin(:));                          % sample magnetization
        vir(i) = virial(position,spin,h,k,dp,L);        % sample virial term

    end
    %%%%%%% processing of sampled values %%%%%%%
    % the means of the sampled values are stored per var (temperature, length etc) value.

    en(1:t/2) = [];                             % discard first 50% of the data
    en = en./N;                                 % energy per atom
    enmean(j) = mean(en);                       % compute mean of the sampled energies
    envar(j) = std(en);                         % variance of sampled energies: <E^2> - <E>^2
    cv(j) = envar(j)./(T.^2);                   % compute heat capacity

    mag(1:t/2) = [];                            % discard
    mag = mag./N;                               % magnetization per atom
    magmean(j) = mean(mag);                     % compute mean of the sampled magnetization
    magvar(j) = std(mag);                       % variance of sampled magnetization: <M^2> -
    <M>^2
    chi(j) = magvar(j)./T;                      % compute magnetic susceptibility


    vir(1:t/2) = [];                            % discard
    virmean(j) = mean(vir);                     % compute mean of the sampled virial term

    press1(j) = (N .* T) ./ L;
    press2(j) = -1*virmean(j) ./ L;
    pressure(j) = press1(j) + press2(j);        % calculate pressure from pressure equation


    fail1mean(j)=fail1./t;                      % compute failure rates
    fail2mean(j)=fail2./t;
    fail3mean(j)=fail3./t;

    % timer
    time = toc;
    expected = ((time-oldtime)*length(var));
    timeleft = (expected - time)/(3600*24);
    fprintf('Time left: %s\n\n',datestr(timeleft,'HH:MM:SS'));
    oldtime = time;

end
%%%%%%%% end simulation %%%%%%

% plotting commands
subplot(2,3,1)
plotterv3(1)
subplot(2,3,2)
plotterv3(2)
subplot(2,3,3)
plotterv3(4)
subplot(2,3,4)
plotterv3(7)
subplot(2,3,5)
plotterv3(5)
subplot(2,3,6)
plotterv3(3)
```

initialize.m

```
function [pos spin] = initialize(N,L,type)
% this 1D function generates N atoms: each having an x coordinate and also a spin value
%
% the positions of the atom are evenly spaced from 0 to L (the length of the box) in two
dimensions
% the spin of the atoms are randomly generated either +1 or -1

pos = zeros(1,N);
spin = zeros(1,N);

c = L./N;                   % "lattice constant"
pos = linspace(0,L,N+1);
pos = pos(1:end-1);         % discard last position
pos = pos+(0.5*c);          % skew by 0.5c to obtain evenly space lattice
                            % according to periodic boundary conditions


switch type

    case 1              % random spin initialization
        for i=1:N
            if (rand() < 0.5)
                spin(i) = +1;
            else
                spin(i) = -1;
            end

        end

    case 2              % checkerboard initialization
        for i=1:N
            if (mod(i,2))
                spin(i) = +1;
            else
                spin(i) = -1;
            end

        end

    case 3              % all spin up initialization
        for i=1:N
            spin = ones(1,N);
        end

end
```

energy.m

```matlab
function out = energy(position,spin,h,k,dp,bf,L)
% returns the total energy sum of the system
% by separately summing the contributions
% due to spin - spin interaction and spin - magnetic field interaction

sum1 = 0;
sum2 = 0;

for i = 1:length(spin)      % sum over all spin-spin interactions: including the 1st and last spin
    sum1 = sum1 + hamil(position,spin,i,i+1,h,k,dp,L);
end

sum2 = -1 .* bf .* sum(spin);               % magnetic field term

out = sum1 + sum2;

end
```

```
function out = hamil(position,spin,a,b,h,k,dp,L)
% returns the interaction energy of the Ath and Bth pair of spin.

n = length(spin);

if (a == length(spin))                          % special case: if measuring hamil
                                                % between 1st and last spin (periodic
                                                conditions),

    dist = position(1) + L - position(end);     % sum the distances of (1st spin and left edge)
                                                % and (last spin and right edge)

    b = 1;                                      % set b to 1 to calculate 1st and last spins
    else
    dist = abs(position(b) - position(a));
end

out = -1 * spin(a) * spin(b) * potential(dist,h,k,dp);

end
```

```
function pf = metropolis(initE,finalE,T)

% check whether a move passes or fails the Metropolis algorithm
% pf = 1 if pass, pf = 0 if fail.

prob = exp(-finalE/T) ./ exp(-initE/T);

if (rand() > prob )                            % accept with probability Zfinal/Zinit.
                                               % this statement = 1 for spin not accepted.
                                               % also if prob > 1 will accept
                                               always
    pf = 0;
else
    pf = 1;
end

end
```

move.m

```
function out = move(position,n,dx,L)
% do a move on the lattice in the box of size L,
% where position is the array of positions,
% n is the selected nth spin,
% dx is the maximum move interval
%
% notes: if n is first or last spin, bef and aft are set to be the box edges respectively
%
% returns the new position of the nth atom (without updating position)
% returns -1 if invalid move


if(n == 1)                       % if move 1st atom, bef limit is L = 0, the left edge of the box
    bef = 0;
    else
    bef = position(n-1);
end

if(n == length(position))        % if move last atom, aft limit is L = L, the right edge of the
box
    aft = L;
    else
    aft = position(n+1);
end

ran = unifrnd(-dx,dx);                       % random movement interval
out = position(n) + ran;

if (out < bef | out > aft)                   % check if new pos is beyond neighbour particle pos
    out = -1;                                % if beyond, it is invalid position: return -1
end




end
```

flip.m

```
function out = flip(spin,a)
% flip spin a.

out = spin;
out(a) = out(a) .* -1;

end
```

potential.m

```
function out = potential(x,h,k,dp)
% returns the potential curve as a parabola centered at (h,k) with cutoff dp
% at position x.

x1 = h-dp;
x2 = h;
x3 = h+dp;
y2 = k;

if ( (x<x1) | (x>x3) )
    out = 0;
    else
    out = (x-x1)*(x-x3)/( (x2-x1)*(x2-x3) ) * y2;
end
```

dpotential.m

```
function out = dpotential(x,h,k,dp)
% returns the differentiation of the potential curve as a parabola centered at (h,k) with
cutoff dp
% at position x.

x1 = h-dp;
x2 = h;
x3 = h+dp;
y2 = k;

if ( (x<x1) | (x>x3) )
    out = 0;
    else
    out = -1 .* ( x1 + x3 - 2.*x ) ./ ( (x2-x1)*(x2-x3) ) * k;
end
```

```
function out = virial(position,spin,h,k,dp,L)
% returns the virial term of the pressure equation

sum = 0;

for i = 1:length(spin)

    a = i;
    b = i+1;

    if (a == length(spin))                          % special case: if measuring virial between
    1st
                                                    % and last spin (periodic conditions),

        r = position(1) + L - position(end);        % sum the distances of (1st spin and left
        edge)
                                                    % and (last spin and right edge)

        b = 1;                                      % set b to 1 to calculate 1st and last spin

        else
        r = abs(position(b) - position(a));
    end

    sum = sum + spin(a) .* spin(b) .* r .* dpotential(r,h,k,dp);

end

out = sum;

end
```

plotterv3.m

```
function plotterv3(type)

global var
global varstr
global enmean
global magmean
global chi
global cv
global fail1mean
global fail2mean
global fail3mean
global pressure
global press1
global press2


switch type

    case 1      % plot energy vs varstr
        plot(var,enmean,'.b','MarkerSize',10)
        xlabel(varstr)
        ylabel('Energy')
        title(['Energy vs ' varstr])


    case 2      % plot magnetization vs varstr
        plot(var,magmean,'.r','MarkerSize',10)
        xlabel(varstr)
        ylabel('Magnetization')
        title(['Magnetization vs ' varstr])
        axis([-inf,inf,-1,1])

    case 3      % plot rejection rates vs varstr
        hold all
        plot(var,fail1mean,'.r','MarkerSize',5)
        plot(var,fail2mean,'+b','MarkerSize',5)
        plot(var,fail3mean,'og','MarkerSize',5)
        xlabel(varstr)
        ylabel('Rejection rate')
        legend('Spin Metro','Random walk','Position Metro')
        hold off

    case 4 % plot heat capacity vs varstr
        plot(var,cv,'.b','MarkerSize',10)
        xlabel(varstr)
        ylabel('C_v')
        title(['Heat capacity vs ' varstr])

    case 5 % plot magnetic susceptibility vs varstr
        plot(var,chi,'.b','MarkerSize',10)
        xlabel(varstr)
        ylabel('\chi')
        title(['Specific magnetic susceptibility vs ' varstr])
    case 7 % pressure
        hold all
        plot(var,pressure,'.r','MarkerSize',5)
        plot(var,press1,'+b','MarkerSize',5)
        plot(var,press2,'og','MarkerSize',5)
        xlabel(varstr)
        ylabel('Pressure')
        title(['Pressure vs ' varstr])
        legend('Total','Ideal gas term','Virial term')
        hold off


end
```

Appendix D Derivation of the pressure observable

As we simulate the Ising model as a fluid, we can obtain an expression of pressure from the canonical ensemble[21;22]:

Starting with the themodynamic variable of pressure:

$$\begin{aligned} P &= -\frac{\partial F}{\partial V}\bigg|_{T,N} \\ &= k_B T\, \frac{\partial \ln Z}{\partial V}\bigg|_{T,N} \\ &= \frac{k_B T}{Z}\, \frac{\partial Z}{\partial V}\bigg|_{T,N} \\ &= \frac{k_B T}{V^N \int_0^1 dr_1 ... \int_0^1 dr_N e^{-\beta U}}\, \frac{\partial \left(V^N \int_0^1 dr_1 ... \int_0^1 e^{-\beta U} \right)}{\partial V}\Bigg|_{T,N}, \end{aligned}$$

where $Z = \int_0^V dr_1' ... \int_0^V dr_N' e^{-\beta U} = V^N \int_0^1 dr_1 ... \int_0^1 dr_N e^{-\beta U}$ and

$$\frac{\partial Z}{\partial V}\bigg|_{T,N} = N V^{N-1} \int_0^1 dr_1 ... \int_0^1 dr_N e^{-\beta U} - \frac{V^N}{k_B T} \int_0^1 dr_1 ... \int_0^1 dr_N e^{-\beta U} \frac{\partial U}{\partial V},$$

and using general dimensions, $V \equiv r^D$ so $\frac{dr_i}{dV} = \frac{1}{VD} r_i$,

$$\begin{aligned} \frac{\partial U}{\partial V} &= \sum_i \frac{dU(r_i)}{dr_i} \frac{dr_i}{dV} \\ &= -\sum_i F_i \frac{1}{VD} r_i \\ &= -\frac{1}{VD} \sum_i F_i \cdot r_i. \end{aligned}$$

If the interaction potential is pairwise, $F_{ij} = -F_{ji}$ we can rearrange:

$$\begin{aligned} \sum_i F_i \cdot r_i &= \frac{1}{2} \sum_i \sum_{j \neq i} (r_i \cdot F_{ij} - r_j \cdot F_{ij}) \\ &= \sum_i \sum_{j > i} r_{ij} \cdot F_{ij} \\ &= -\sum_i \sum_{j > i} r_{ij} \frac{dv(r_{ij})}{dr_{ij}}, \end{aligned}$$

where $r_{ij} = |r_j - r_i|$(This means we need only sum over each pair interaction once).

Combining the terms we can express the average pressure as an ensemble average of the virial term:

$$\langle P\rangle = \frac{Nk_BT}{V} + \frac{1}{VD}\left\langle \sum_i \sum_{j>i} r_{ij}\frac{dv\left(r_{ij}\right)}{dr}\right\rangle . \tag{58}$$