

Lattice Field Theory:  
An Undergraduate Introduction

Leon Hostetler

April 24, 2018

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>The Ising Model</b>	<b>3</b>
2.1	Boltzmann Statistics . . . . .	3
2.2	Ferromagnetism . . . . .	4
2.3	The Ising Model of Ferromagnetism . . . . .	6
2.4	Monte Carlo Simulation . . . . .	8
2.5	Comparison with Lattice Field Theory . . . . .	13
<b>3</b>	<b>The Standard Model</b>	<b>15</b>
<b>4</b>	<b>Quantum Field Theory</b>	<b>17</b>
4.1	Fields . . . . .	17
4.2	Quantum Fields . . . . .	18
<b>5</b>	<b>QFT on the Lattice</b>	<b>20</b>
5.1	Theoretical Background . . . . .	20
5.2	The Lattice Simulation . . . . .	22
<b>6</b>	<b>Lattice Calculations</b>	<b>23</b>
6.1	Mapping the Phase Diagram . . . . .	24
6.2	Lattice Spectroscopy . . . . .	25
<b>A</b>	<b>Appendix</b>	<b>31</b>
A.1	Program 1 . . . . .	31
A.2	Program 2 . . . . .	32

# 1 Introduction

The Standard Model of particle physics describes the electromagnetic, strong, and weak interactions, and classifies the elementary particles. It is based on the framework of relativistic quantum field theory. Unfortunately, many Standard Model calculations are difficult or even impossible to do analytically, and so we turn to computational approaches. One approach is that of lattice field theory, where the smooth Minkowski spacetime of quantum field theory is replaced with a discrete Euclidean lattice. Since lattice space is discrete, computers can be used to numerically solve many problems in the Standard Model that cannot be solved analytically.

Textbooks on lattice field theory typically assume knowledge of quantum field theory. This may be appropriate for the graduate student working in lattice field theory. However, an undergraduate assisting on a lattice project will likely feel lost. Finding expository texts at the appropriate level even for an advanced undergraduate is difficult. This brief introduction was written to help fill that gap. I hope that it will help other undergraduates just stepping into the exciting world of lattice field theory.

## 2 The Ising Model

Many of the concepts in lattice field theory have analogues in statistical physics. This goes beyond shared terminology. Even some of the techniques derived in either of these fields are routinely applied in the other. In fact, the whole idea of putting quantum fields on a lattice was inspired by the Ising Model of statistical mechanics which models ferromagnetism on a lattice. The Ising Model then serves as an introduction to lattice field theory.

### 2.1 Boltzmann Statistics

We start with a brief introduction to some key ideas from statistical mechanics.

The *canonical ensemble* is the set of all possible microstates of a system with a fixed number  $N$  of particles, a fixed volume  $V$ , and a fixed temperature  $T$ . The temperature of the system is fixed by assuming that it is allowed to exchange energy with a large heat reservoir with a heat capacity large enough so that its temperature stays essentially fixed even if our system fluctuates in energy. The set of *microstates* of a system is the set of all possible states that the system may be found in. For example, in an atom, the microstates correspond to the various energy levels. If a system is isolated, then it must have a particular energy, and the system is equally likely to be found in any microstate associated with that particular energy. However, we are not interested in an isolated system at a particular energy. We are interested in a system that is exchanging energy with an external reservoir. In this case, the system of interest, say the atom, could be found in any possible microstate—not just those associated with a particular energy. The total system of atom plus reservoir still has a fixed energy, but energy can flow between the atom and the reservoir. The system of interest could be in any of its possible microstates, however, states with lower energy are

more probable.

Suppose we have a system in which the  $i$ th microstate has energy  $E_i$ . Then the *Boltzmann factor* for that microstate is

$$e^{-\beta E_i},$$

where  $\beta = 1/kT$ ,  $T$  is the temperature, and  $k$  is Boltzmann's constant. The *partition function* of the system is the sum of all the Boltzmann factors for the system

$$Z = \sum_i e^{-\beta E_i}.$$

For a system in thermal equilibrium with an external reservoir as described above, we can derive from thermodynamic principles the probability for the system to be in the  $i$ th microstate

$$P_i = \frac{e^{-\beta E_i}}{Z}.$$

This is called the *Boltzmann distribution*.

To compute the average value of a parameter  $O$  (e.g. energy, spin, magnetization, etc), we compute the weighted sum

$$\langle O \rangle = \sum_i P_i O_i.$$

For example, to compute the expectation value of the energy, we use

$$\langle E \rangle = \sum_i P_i E_i = \frac{1}{Z} \sum_i E_i e^{-\beta E_i}.$$

For energy, we can also derive the formula

$$\langle E \rangle = -\frac{\partial}{\partial \beta} \ln Z.$$

This formula is useful if we have an explicit formula for the partition function  $Z$ .

Given the partition function of a system, we can compute all of its thermodynamic properties. For example, the Helmholtz free energy is  $F = -kT \ln Z$ . From this, we can compute other quantities such as the entropy, pressure, and chemical potential of the system

$$S = -\left(\frac{\partial F}{\partial T}\right)_{V,N}, \quad P = -\left(\frac{\partial F}{\partial V}\right)_{T,N}, \quad \mu = \left(\frac{\partial F}{\partial N}\right)_{T,V}.$$

## 2.2 Ferromagnetism

In electromagnetism, electric charge can be isolated as single particles such as the electron and the proton. Each of these “radiates” an electric field. There are no such single particles for magnetism. Magnets always come with both north and south poles. If you break a bar magnet in half, you

don't get two monopoles, you get two magnets each with both north and south poles. Electric fields come from electric charges. Magnetic fields come from *moving* electric charges, and moving electric charges are also called "currents".

A tiny ideal bar magnet consisting of a north and south pole is called a *magnetic dipole*. A magnetic dipole is characterized by vector called the *magnetic dipole moment*. This vector has a magnitude and a direction, quantifying the strength and the orientation of the magnetic field created by the dipole. A current-carrying wire twisted into a circle generates the same kind of magnetic field as that produced by a bar magnet. In terms of magnetic field, a loop of current loop can also be characterized by a magnetic dipole moment. These are illustrated in the images below, which show the magnetic dipole moment vectors for the bar magnet and current loop as blue arrows.

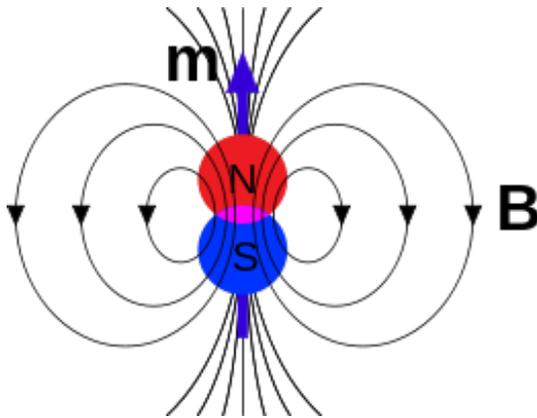


Figure 1: Magnetic dipole (blue arrow) and magnetic field due to a pair of poles.

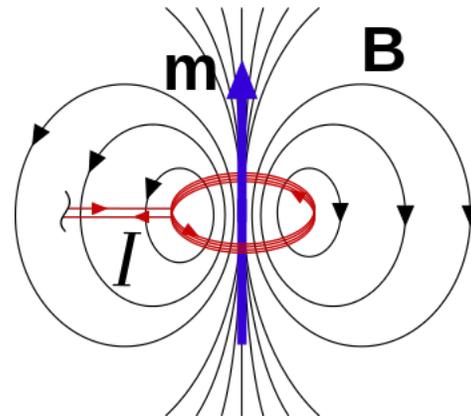


Figure 2: Magnetic dipole (blue arrow) and magnetic field due to a loop of current.

A magnetic dipole has a characteristic magnetic field and a magnetic dipole moment. If a magnetic dipole is placed in an external magnetic field, the interaction between the two fields causes the dipole to rotate such that the dipole moment aligns with the external magnetic field.

Now consider a simple model of an electron in a classical orbit about an atom. The electron is charged, and it is moving in a loop, so its orbit forms a current loop. A loop of current is a magnetic dipole, and so the moving electron produces a magnetic field and a magnetic dipole moment. However, the electron is not just orbiting. It also has spin. A simple model of electron spin is as a sphere of charge that is spinning on its axis. This moving charge also produces a magnetic dipole moment. An electron may not actually be spinning in the classical sense, but its quantum-mechanical spin really does result in a magnetic dipole moment. In fact, the spin of an electron contributes more to the net magnetic dipole moment of an atom than the orbital motion of the electron does. Electrons are spin-1/2 particles. This means an electron's spin, and hence the direction of its magnetic dipole moment can be characterized as "up" or "down".

A material, such as iron, consists of many atoms. Each atom has its own magnetic dipole moment, which comes from its orbiting and spinning electrons. In *ferromagnetism*, a large group of atomic dipole moments can align producing an overall magnetic field much more powerful than

the magnetic fields of the individual atoms.

In a material with atoms that have filled electron shells, the dipoles pair up in the shells. From the Pauli exclusion principle, the pair of electrons in each orbital have opposite spin. So the net dipole moment of a material made up of atoms with filled electron shells is nearly zero since each electron is paired with one of opposite spin. In contrast, materials such as iron with unpaired electrons have a net magnetic dipole moment and thus a net magnetic field. By Hund's rule, the first several electrons in a shell have the same spin, thus magnifying the net magnetic dipole moment of a material containing unpaired electrons.

If a material containing unpaired electrons is exposed to an external magnetic field, the magnetic dipole moments of the atoms try to align with the magnetic field. This is the phenomenon of paramagnetism. In ferromagnetism these atomic dipole moments tend to align spontaneously even when there's no external magnetic field. In iron, for example, this causes the atomic dipole moments in relatively large regions called *magnetic domains* to align with each other. Each domain then has a net magnetic dipole moment.

The magnetic dipole moments in materials like iron align with each other in domains because of a quantum-mechanical interaction between neighboring atoms. When the orbitals of unpaired valence electrons in neighboring atoms overlap in a material like iron, the energy of the system is lower (i.e. it is more favorable) when the electron spins are parallel than when they are not. Thus, you get magnetic domains in the material where all the valence electron spins are aligned. Within each domain, the material is magnetized, however, neighboring domains tend to have opposite magnetizations. Think of each domain as a small magnet, so a material like iron can be thought of as a bunch of small bar magnets. Neighboring bar magnets want to point in opposite directions. This is why a material like iron is generally not noticeably magnetized. However, if such a material is heated and an external magnetic field is applied, the magnetic domains align with the external magnetic field. If it is then cooled while still in the external magnetic field, these domains are locked in and remain aligned even after the external field is removed. You now have a permanent magnet.

If you raise the temperature of a piece of iron, the thermal fluctuations decrease the magnetization of the material. Above the Curie temperature, which is 1043 K for iron, the magnetization becomes zero.

## 2.3 The Ising Model of Ferromagnetism

In short, ferromagnetism occurs because neighboring magnetic dipoles want to align with each other. We can model this behavior with a lattice of atomic dipoles. In our case, we will look at the simpler two-dimensional case. The Ising model gives good results near the Curie temperature, but it does not give accurate predictions at low temperatures.

The two-dimensional Ising model consists of a lattice of dipoles. A dipole in the middle of a lattice has four neighbors—above, below, left, and right. The closest diagonal dipoles are not considered neighbors. Each dipole can only point up or down. Let  $s_i$  be the state, either +1 (up) or -1 (down), of the  $i$ th dipole. The interaction energy of a pair of dipoles is lower when the dipoles

-1	-1	1	-1
1	1	1	1
-1	1	1	1
-1	-1	-1	1

Figure 3: One of the many possible states of a  $4 \times 4$  lattice with a total of 16 dipoles.

are parallel. Let  $-\epsilon$  be the interaction energy when a pair of neighboring dipoles is parallel and  $+\epsilon$  when they are antiparallel. Either way, the interaction energy of a pair of neighboring dipoles is  $-\epsilon s_i s_j$ .

The total interaction energy of the system from all pairs of dipole neighbors is

$$U = -\epsilon \sum s_i s_j,$$

where the sum is over all neighboring pairs  $i, j$ . For the  $4 \times 4$  example shown above, we can calculate the total energy relatively easily since there are only 16 dipoles and each dipole has a maximum of four neighbors. In this case, there are 24 unique neighbor pairs. Calculating for this state gives us  $U = -4\epsilon$ . The lowest possible energy, the ground state, of such a  $4 \times 4$  lattice is when all the dipoles are up or all of them are down. In that case, the total energy of the state is  $U_0 = -24\epsilon$ .

With a finite lattice, we get edge effects since the dipoles along the edges have fewer neighbors than those in the middle. To eliminate this problem, we use periodic boundary conditions. We say that the left edge wraps around to the right edge and the top edge wraps around to the bottom edge. For example, the dipole in the top left corner neighbors the dipole in the bottom left corner as well as the top right corner. Now, every dipole in the lattice has exactly four neighbors. For the  $4 \times 4$  lattice, there are now 34 neighbor pairs. Recalculating, we find that the total interaction energy of the state shown in Fig. (3) is again  $U = -4\epsilon$ . The ground state is now  $U_0 = -34\epsilon$ .

Our system can be treated as one in thermal equilibrium with an external reservoir. We say that it follows a Boltzmann distribution. That is, the probability of finding our lattice in a state with energy  $U$  is

$$P = \frac{e^{-\beta U}}{Z},$$

where

$$Z = \sum e^{-\beta U},$$

is the partition function. Here, the sum is over all possible sets of dipole alignments.

Once we have the partition function of the system, we can calculate things like the average energy as a function of temperature (recall that  $\beta = 1/kT$ )

$$\langle E \rangle = -\frac{\partial}{\partial \beta} \ln Z.$$

We can also look at the magnetization as a function of temperature. The magnetization corresponds to how ordered the lattice is. When all of the dipoles point in the same direction, the system is completely magnetized.

## 2.4 Monte Carlo Simulation

Since the partition function  $Z$  involves a sum over all possible sets of dipole alignments (i.e. all possible microstates), the number of terms in the sum is  $2^N$  where  $N$  is the number of dipoles. So for a tiny  $4 \times 4$  lattice with only  $N = 16$ , the number of terms in  $Z$  is already 65,536. For a medium-size lattice like a  $100 \times 100$  lattice,  $Z$  has  $2^{10,000}$  terms. Adding these by brute force quickly becomes infeasible. We need another approach.

Instead of computing all possible states of the system, we select a small random sample of states. This is an example of a *Monte Carlo* method. The idea is to generate a random sample of states, compute the Boltzmann factors for each state, and then compute the average energy, magnetization, and other thermodynamic quantities from this random sample. This is the *naive* Monte Carlo approach. Unfortunately, for even a modest  $100 \times 100$  lattice, there are  $2^{10,000} \sim 10^{3010}$  possible states. If we randomly select a sample of one billion states, then we would be looking at roughly one in  $10^{3000}$  of the possible states. If, for example, we are interested in the magnetization (when most of the dipoles point in the same direction), then the important states make up such a small fraction of the total, that our random sample would most likely include none of them. So the naive Monte Carlo approach is not good enough.

The phenomena we are interested in studying occurs in the low energy states. So instead of sampling the states randomly, we use the Boltzmann factors to guide our selection process. We start with a random state of the system. Then we choose a random dipole in this state and consider the possibility of flipping it. If the energy of the state would be lower were the dipole flipped, then we go ahead and flip it to generate the next state. If the energy of the state would not be lower, then we choose at random with a probability of  $e^{-\Delta U/kT}$  whether or not to flip it. Here,  $\Delta U$  is the change in energy of the state between flipped and unflipped dipole. Whether or not we flip it, we now repeat the process, by choosing another random dipole. We repeat until every dipole in our system has had many chances to be flipped. This is called the *Metropolis algorithm*, and it generates a sample in which the low energy states (i.e. the states near equilibrium) occur more frequently. Overall, this is an example of a Monte Carlo method with *importance sampling* (as opposed to uniform random selection).

Because of the way it selects whether or not to flip a dipole, the metropolis algorithm advances through statespace (the space of all possible states of the lattice) in such a way that the lattice tends quickly toward a more probable state. What such a state looks like depends on the temperature of the lattice. At high temperatures, the most probable state is a highly disordered one. This is analogous to a real ferromagnet in that at high temperatures, thermal fluctuations oppose the tendency to magnetize. At low temperatures, the thermal fluctuations are minimal and the most probable states are the magnetized, highly-ordered states in which most of the dipoles point in the same direction.

In Appendix (A.1), I give a Python program that implements the Ising model. With it, I simulate a small  $20 \times 20$  lattice (this size can be changed in the program), and each dipole gets, on average, 1000 chances to flip. The temperature  $T$  is a free parameter, and it must be set before running the program. Fig. (4) shows the final lattice state for simulations run at different temperatures.

Notice that at high temperatures such as  $T = 20$  or  $T = 10$ , the final state is highly disordered. In real ferromagnets, this is due to thermal fluctuations which our Monte Carlo simulation models well. At low temperatures such as  $T = 2$  and  $T = 1$ , the final lattice states are highly ordered. This corresponds to a magnetized ferromagnet. Somewhere between  $T = 2.5$  and  $T = 2$ , there is a fairly abrupt transition from fairly disordered (unmagnetized) to fairly ordered (magnetized). Such an abrupt change in a quantity is called a *phase transition*. In fact, the 2D Ising model has been solved exactly, and we know that the critical temperature at which this phase transition occurs is  $T = 2.27$  in our units of  $\epsilon/k$ .

We can quantify the amount of order, i.e. the total magnetization, by the absolute value of the sum of the array. We can do this by adding `print(np.abs(np.sum(s)))` at the end of the program given in Appendix (A.1). Recall that each dipole is represented as a +1 or a -1 in the array. We are defining the total magnetization of our lattice as the magnitude of the sum of all of these. Fig. (5) shows a plot of the total magnetization (M) versus temperature (T). Here we can clearly see the phase transition near  $T = 2.3$ .

In our program, we give each dipole an average of 1000 chances to be flipped. Is this enough to ensure our lattice is in equilibrium? In reality, our lattice only approximates real ferromagnetism if we give each dipole infinite chances to flip. We can't actually give each dipole infinite chances to flip, since that would take infinite time. Fortunately, we don't have to. In the program given in Appendix (A.1), we go through  $1000 \times 20 \times 20 = 400,000$  iterations to give each dipole an average of 1000 chances to be flipped. However, these are far more iterations than we really need. A good indication that our lattice is in equilibrium is when measured quantities stabilize. For example, we can calculate the total magnetization at each iteration to get the plot in the left panel of Fig. (6). In this plot, we see the total magnetization becomes nearly constant (i.e. equilibrium is reached) long before we reach 400,000 iterations. We can improve the efficiency of our program, without losing any useful information, by reducing the iterations. If we reduce the number of iterations by a factor of 10, we get the plot in the right panel of Fig. (6). Now our program runs ten times faster, and we don't lose anything since we are still reaching the equilibrium value of  $M \sim 400$ .

It is also illuminating to modify the program so that it plots the changing states in real-time. Then one sees why this is called a Monte Carlo *simulation* since you can see it starting in an unmagnetized (i.e. random) state and become more and magnetized (i.e. ordered) as random dipoles are flipped.

Note: At low temperature, the simulation may get stuck in a metastable state that is half magnetized in one direction and half magnetized in the other direction. An example is shown in Fig. (7).

We might want to investigate finite size effects. How does the amount of order or disorder in the final state depend on the size of the lattice? With a very large lattice we expect to obtain the

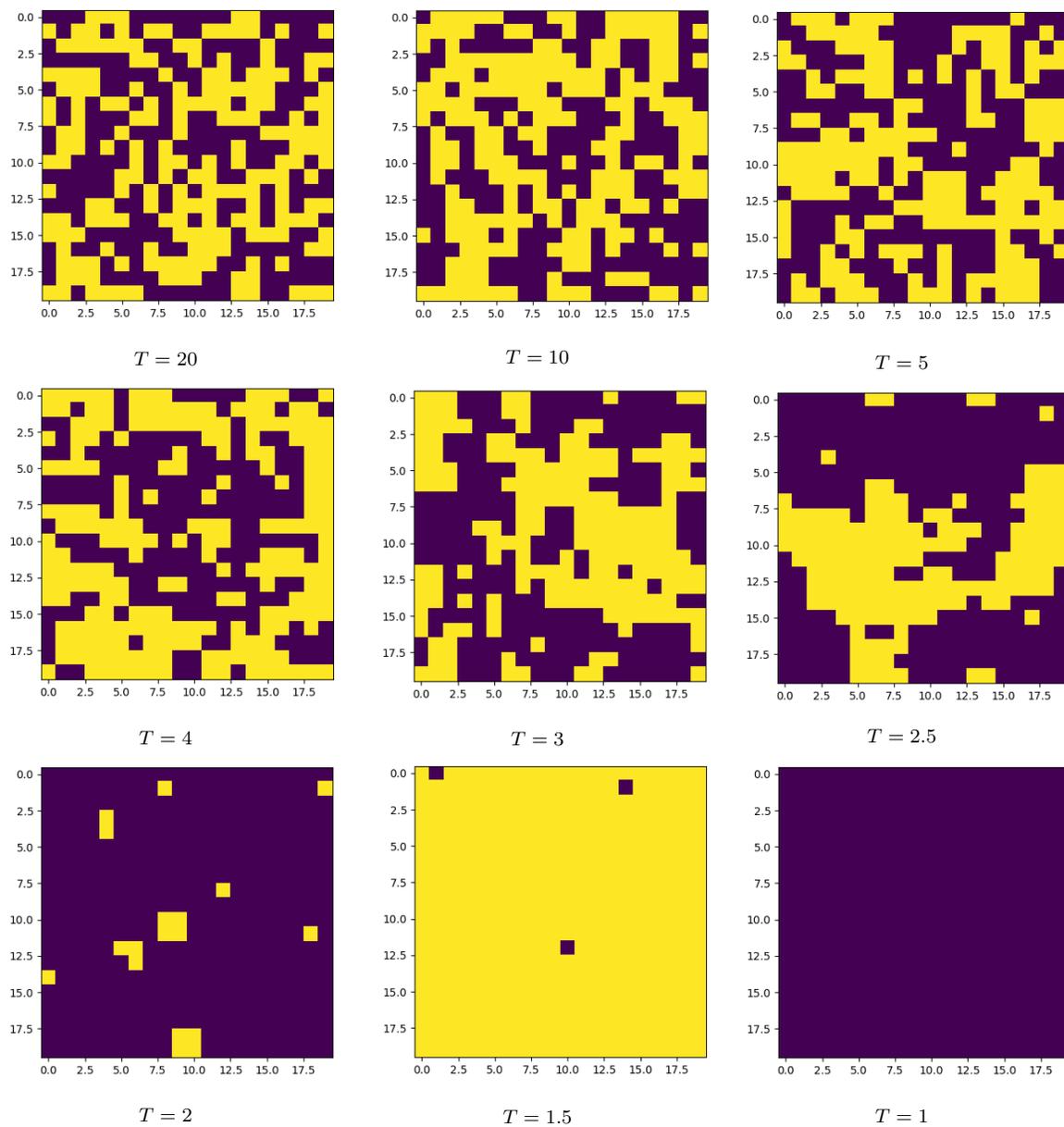


Figure 4: Each panel shows the final lattice state for a simulation run at a given temperature.

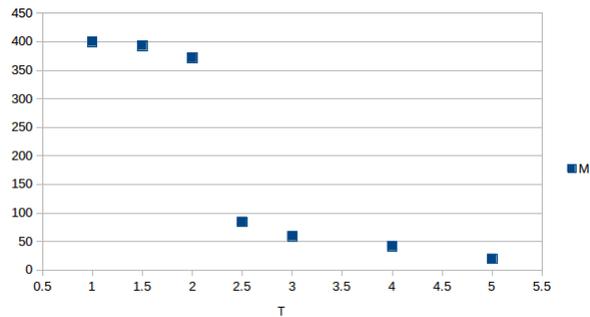


Figure 5: The total magnetization versus temperature. Each point is the average total magnetization taken over five runs at a given temperature on a  $20 \times 20$  lattice. Notice the phase transition near  $T = 2.3$ .

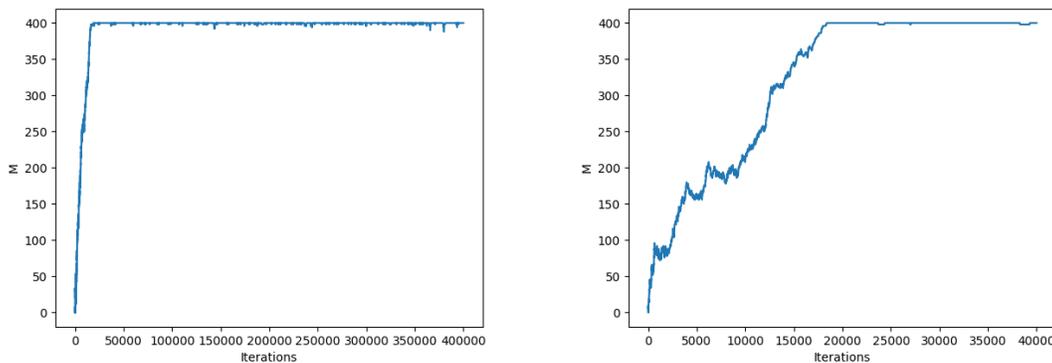


Figure 6: In the left panel, we plot the total magnetization ( $M$ ) of the lattice versus the iteration number. We are using a  $20 \times 20$  lattice with  $T = 1$ . We see that the typical magnetization  $M \sim 400$  is reached quickly. In the right panel, we again plot total magnetization ( $M$ ) versus the iteration number. Now we use 10 times fewer iterations and we still reach equilibrium ( $M \sim 400$ ).

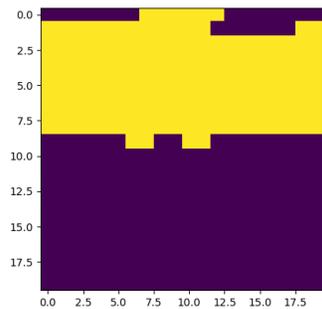


Figure 7: In this case, our lattice became stuck in a metastable state.

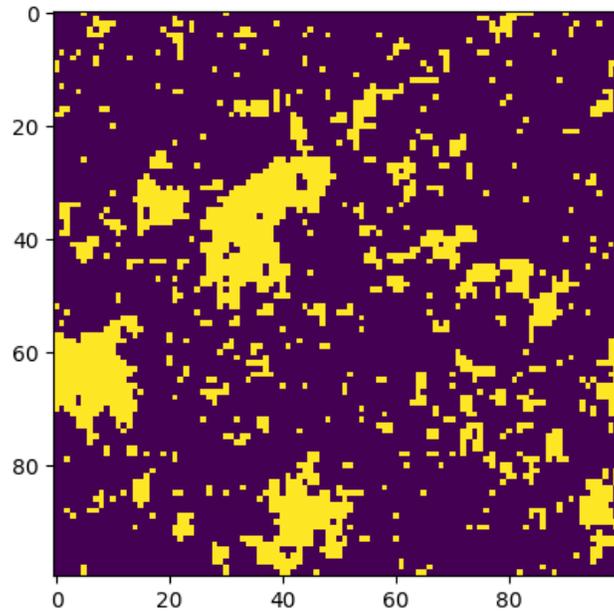


Figure 8: An example of a  $100 \times 100$  lattice with  $10000 \times 100 \times 100$  iterations at the critical temperature of  $T = 2.27$ . This example took about 20 minutes to compute on my personal computer.

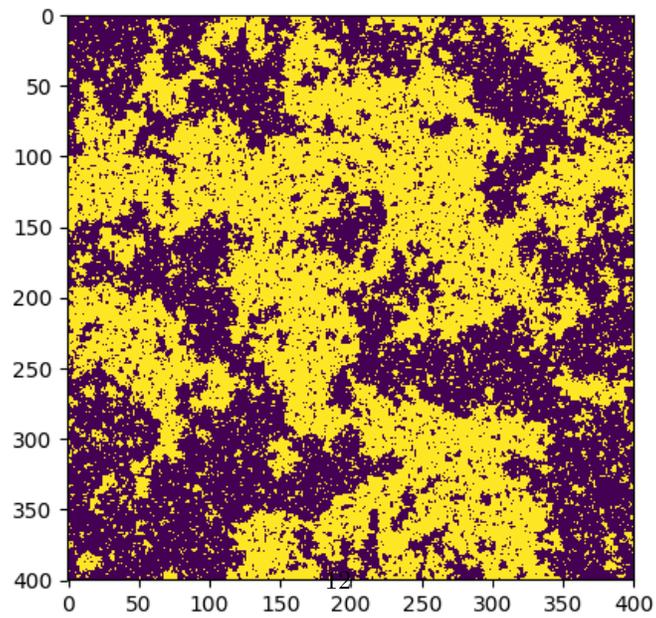


Figure 9: An example of a  $400 \times 400$  lattice with  $1000 \times 400 \times 400$  iterations at the critical temperature of  $T = 2.27$ . This example took about 30 minutes to compute on a personal laptop.

continuum limit. In other words, we expect a large lattice to model actual ferromagnetism quite well. But how large of a lattice do we need? If we use small lattices, what if any corrections should we make to our measurements? To quantify the amount of order in a lattice state, we look at *correlation functions*.

A correlation function quantifies the correlation of the lattice dipoles over a distance. As such, it quantifies the amount of clustering that occurs on a lattice. Given any pair of dipoles, we compute the product of their states  $s_i s_j$ . Then we average this quantity over all pairs of dipoles at the same distance  $r$  from each other. This gives us a measure of how correlated our lattice is at this distance scale. Finally, the overall magnetization is subtracted off by subtracting the square of the average dipole value. We can write our correlation function as

$$c(r) = \overline{s_i s_j} - \overline{s_i}^2.$$

The *correlation length* is the distance over which the correlation function decreases by a factor of  $e$ .

A simple implementation of this correlation function looks only at vertical and horizontal distances. In this case, for each lattice cell, there are exactly four other cells at a distance  $r$ —left, right, above, and below. In Appendix (A.2), there is a Python program that calculates this correlation function for the final state of the lattice. An example for a  $100 \times 100$  lattice with  $10^7$  iterations at  $T = 3$  is shown in Fig. (10). In this case, the correlation function drops quickly to near zero, so we have a small correlation length. In Fig. (11), we repeat the experiment but at  $T = 2$ . Now the correlation function drops off more slowly. The correlation length is much larger indicating larger clusters, and thus, larger magnetization domains.

In these examples, the correlation function is of only the final lattice state. A more useful quantity would be the *average* correlation function taken over many typical states. To do this, one would run the simulation for a while to equilibrate it to a typical state at that temperature. After it has equilibrated, one would continue running the simulation but now compute the correlation function for each subsequent lattice state. Once completed, the average can be taken of all the correlation functions.

## 2.5 Comparison with Lattice Field Theory

In lattice field theory, we transform the 4D Minkowski spacetime of quantum field theories to a 4D finite and discrete lattice. This is analogous to our treatment of a 2D Ising ferromagnet on a lattice.

In the Ising model, each lattice cell represents a dipole. In lattice field theory, the particle fields are defined at the cells and at the connections between the cells. In the Ising model, we start with a random lattice configuration and a set temperature. In lattice field theory, we do the same, but now we have to set multiple parameters. Next, we evolve the lattice using a Monte Carlo method with importance sampling. We let the simulation run for a while to equilibrate it. That is, we run the simulation long enough to reach a lattice state that is typical for the given parameter(s). We are now ready to compute correlation functions of our lattice states. We continue running the

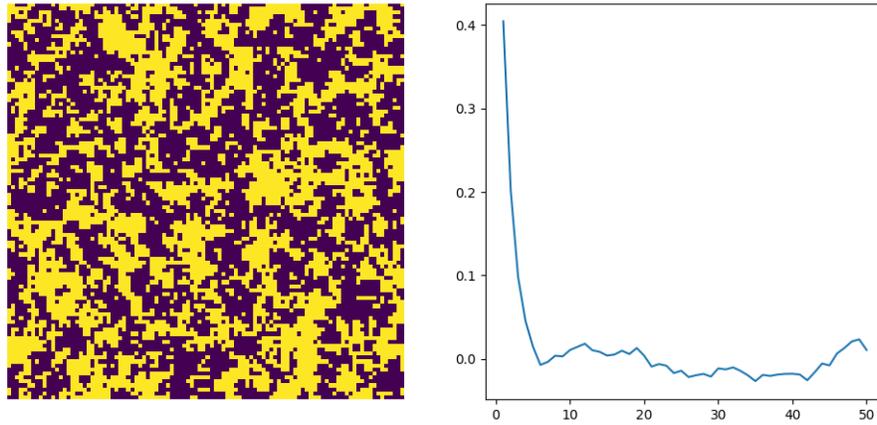


Figure 10: On the left is the final lattice state for a  $100 \times 100$  lattice with  $1000 \times 100 \times 100$  iterations at  $T = 3$ . On the right is the correlation function for this lattice state.

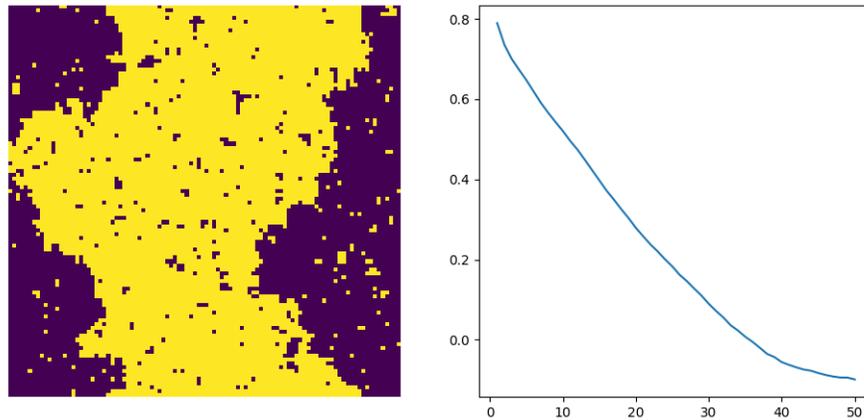


Figure 11: On the left is the final lattice state for a  $100 \times 100$  lattice with  $1000 \times 100 \times 100$  iterations at  $T = 2$ . On the right is the correlation function for this lattice state.

simulation, but now that it's equilibrated, we compute the correlation function for each lattice state in order to obtain the average correlation function for the given set of parameters.

Because of the way a lattice state is generated from the previous state, subsequent lattice states are statistically correlated with each other. To get a more accurate picture of the average correlation function, we need to reduce the correlation between measured states. To do this, we equilibrate our lattice, compute the correlation function, and then run the simulation for a while

before computing the correlation function a second time. Then we run the simulation for a while again before computing the correlation function a third time, and so on. In the end, we construct the average correlation function from which we can estimate, for example, the masses of the particles in the theory.

### 3 The Standard Model

The Standard Model is the theory of particle physics that describes the elementary particles and their interactions. These elementary particles include the leptons, the gauge bosons, the Higgs boson, and the quarks. These particles may interact via the electromagnetic, weak, or strong interactions.

The known particles can be classified in three categories: hadrons, leptons, and field particles. Hadrons are particles that experience the strong nuclear force. Hadrons can be further classified as mesons or baryons. Mesons are bosons with integer spin consisting of two quarks, so they are unaffected by the exclusion principle. Mesons include several pions as well as the eta, rho, and kaon particles. Baryons are fermions with half-integer spins consisting of three quarks, so they must comply with the exclusion principle. Baryons include the proton and neutron as well as the sigma, lambda, and omega particles. Both types of hadrons are composed of quarks, and so they are not elementary particles.

The second category of particles is the leptons. These include the electron, muon, and tau as well as a neutrino associated with each of these. The leptons are all spin-1/2 fermions. They are not composed of quarks and do not interact via the strong force. They are believed to be elementary point particles with no internal structure.

Before we look at the third category of particles—the field particles—we take a look at the forces or “interactions” included in the Standard Model. The one that is familiar to everyone is the electromagnetic interaction, which acts between electrically charged particles. An example of an electromagnetic interaction is an electron-positron annihilation

$$e^- + e^+ \rightarrow \gamma + \gamma,$$

where an electron and positron annihilate each other and create a pair of gamma ray photons. Less commonly known is the weak interaction, which is involved in radioactivity. An example of a weak interaction is the beta decay

$$n \rightarrow p + e^- + \bar{\nu}_e,$$

where a neutron decays into a proton, an electron, and an antielectron neutrino. The third interaction included in the Standard Model is the strong interaction. This interaction acts between quarks, and is what keeps two quarks together in mesons and three quarks together in baryons. It is also the force that binds neutrons and protons together in atomic nuclei. An example of a strong interaction is

$$\pi^+ + \pi^- \rightarrow n + \bar{n}.$$

## Standard Model of Elementary Particles

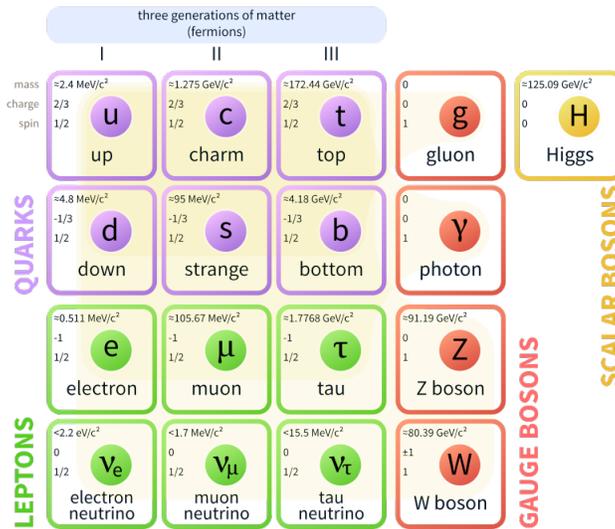


Figure 12: The Standard Model (Wikimedia Commons)

In this interaction, a pair of pions are smashed together creating a neutron and an antineutron. The only fundamental force missing from the Standard Model is gravity, and that is simply because we don't know how to include it yet.

The third category of particles is the field particles. These are the quanta of the different forces. Of these, four are gauge bosons, which are force carriers with integer spin. The photon is the quantum of the electromagnetic field and the force carrier of the electromagnetic force. The  $W^+$ ,  $W^-$ , and  $Z$  bosons are gauge bosons which mediate the weak interaction. The gluons, there are eight of them, are gauge bosons which mediate the strong interaction. Finally, the Higgs boson, which is a scalar (i.e. spin-0) boson rather than a gauge boson, is an excitation of the Higgs field.

The hadrons, if you recall, are not fundamental particles. Rather, they are composed of quarks of which there are six flavors—up, down, strange, charmed, top, and bottom. Quarks have several unique properties. For example, they have fractional charges of  $+\frac{2}{3}e$  and  $-\frac{1}{3}e$ . The baryons are made of three quarks, so given that quarks are all spin-1/2, baryons must also be fermions with odd half-integer spins. Since mesons are composed of a quark and antiquark pair, they must be bosons with integer spin.

Discovered in the 1950s, the  $\Delta^{++}$  particle composed of three identical quarks appears to violate the Pauli exclusion principle. This implies an additional property distinguishing quarks. This property is called color charge, and quarks can have three different values of it—red, green, or blue. Color charge is analogous to electric charge. In quantum electrodynamics (QED), electrically charged particles interact via the electromagnetic force. In quantum chromodynamics (QCD), which is the theory of quarks and the strong force, particles with color charge interact via the strong force.

In QED, the photon mediates the electromagnetic force. In QCD, the gluon mediates the strong force. The photon is uncharged, so it does not experience the electromagnetic force itself. In contrast, gluons carry color charge, so they experience the strong force themselves. This makes QCD calculations much more complicated than QED calculations.

Unlike the electromagnetic force, the strong force does not decrease with distance. This implies that quarks can never be isolated. If you have a pair of quarks bound together by the strong force, no amount of pulling on one quark will allow you to isolate it. If you were to pull a quark-antiquark pair (i.e. a meson) far enough apart, the energy you've inserted would manifest in the creation of two new quarks, so that you now have a quark-antiquark pair in each hand. This phenomenon is called *color confinement*, and it means that a quark will never be observed by itself. We can only observe particles containing multiple quarks.

## 4 Quantum Field Theory

### 4.1 Fields

Quantum mechanics does not allow for the creation or annihilation of particles. Since creation and annihilation of particles is part of the Standard Model, quantum mechanics is insufficient for describing particle interactions. What we need is a field theory. However, a classical field theory, such as classical electrodynamics is insufficient because field excitations are not quantized in such a theory. To describe particle interactions, we need a *quantum* field theory.

In general, a *field* is an object that gives a value to each point in spacetime. For example, a temperature map, which assigns a temperature to each point in the field, is a simple scalar field. A wind velocity map, which assigns a wind speed and direction to each point in the field, is a simple vector field. Other examples of vector fields, which assign a vector to each point in spacetime, include the Newtonian gravitational field and classical electric and magnetic fields.

A *field theory* describes the dynamics of a field. That is, it defines a field and describes how it evolves in time or how it changes with respect to other variables that it depends on. This is typically done by constructing the Lagrangian density of the field. Recall from classical mechanics, that the Lagrangian of a moving particle is defined as the kinetic energy minus the potential energy

$$L = T - V,$$

and in the Lagrangian formulation of classical mechanics, the path that a particle takes is the one that minimizes the action defined as

$$S = \int L dt.$$

In classical mechanics, a single particle moving in a potential of the form  $V = \frac{1}{2}kx^2$ , is an example of a harmonic oscillator. In Lagrangian mechanics, the dynamics of such a particle is specified by a single Lagrangian,  $L = \frac{1}{2}m\dot{x}^2 - \frac{1}{2}kx^2$ . A simple field can be constructed by putting such a harmonic oscillator at each point in spacetime. Now, instead of a single Lagrangian, one talks of

a Lagrangian *density*,  $\mathcal{L}$ . For a scalar field  $\phi$ , the Lagrangian density is a function of the field  $\phi$ , its gradient  $\nabla\phi$ , time derivative  $\frac{\partial\phi}{\partial t}$ , position  $\vec{x}$ , and time  $t$ . The action is now the integral of the Lagrangian density over all spacetime

$$S[\phi] = \int \mathcal{L} d^3\vec{x} dt.$$

## 4.2 Quantum Fields

There are several equivalent formulations of quantum mechanics. Schrodinger's wave mechanics is the formulation most often taught to undergraduates. In this formulation, if we have the wavefunction  $\psi(x)$  of a particle in one dimension, we can calculate the expectation value of an observable as

$$\langle\psi|\hat{O}|\psi\rangle = \int dx \psi^* \hat{O} \psi,$$

where  $\hat{O}$  is the operator associated with the observable. For example, the ground state mean square displacement would be calculated as

$$\langle 0|\hat{x}^2|0\rangle = \int dx \psi_0^* x^2 \psi_0.$$

An alternative is to use Feynman's path integral formulation of quantum mechanics. This is the approach used in constructing quantum field theory. In this formulation, the expectation value  $\langle x^2 \rangle$  is given by

$$\langle 0|\hat{x}^2|0\rangle = \frac{1}{Z} \int Dx x(0)^2 e^{iS[x]},$$

where

$$Z = \int Dx e^{iS[x]},$$

is a normalization factor closely related to the partition function of statistical mechanics. Here,  $S[x]$  is the action, and

$$Dx = \prod_{t=-\infty}^{\infty} dx(t).$$

Here,  $x(t)$  is a possible path of the particle, so the "path" integral given above is an integral over all possible paths, each weighted by a Boltzmann factor  $e^{iS[x]}$ . The "differential"  $Dx$  is an infinite product of differentials  $Dx = dx_1 dx_2 \dots$ , so the path integral is an infinite-dimensional integral.

To generalize to quantum fields, instead of integrating over all possible paths of a quantum particle, one integrates over all possible field configurations. For example, the correlation function  $\langle O_2(t)O_1(0) \rangle$  (or "connected expectation value"), which quantifies the temporal correlations between two operators  $O_1$  and  $O_2$  would have a path integral that looks like

$$\langle O_2(t)O_1(0) \rangle = \frac{1}{Z} \int \mathcal{D}[\Phi] e^{iS[\Phi]} O_2[\Phi] O_1[\Phi],$$

where the integral  $\int \mathcal{D}[\Phi]$  is over all possible configurations of the field  $\Phi$ .

A *quantum* field theory is a field theory in which energy is quantized. In quantum mechanics, quantization is achieved by replacing the observables with operators and enforcing canonical commutation relations. In one dimension, the enforced quantization condition is  $[\hat{x}, \hat{p}] = i\hbar$ . In three dimensions, this becomes  $[\hat{r}_i, \hat{p}_j] = i\hbar\delta_{ij}$ . In quantum field theory, quantization is achieved by replacing the fields with operators, and enforcing similar commutation relations between the degrees of freedom and their conjugate momenta.

There are several problems with the path integral. The most immediate problem is that infinite-dimensional integrals cannot be solved analytically. This problem is circumvented with lattice field theory, where the infinite-dimensional path integrals become finite-dimensional sums that can be evaluated on computers. Another problem with the path integrals given above is that the factor  $e^{iS}$  is a strongly oscillating function. This is resolved by performing a *Wick rotation* to imaginary time where  $t$  is replaced by  $it$ . Now the action  $S$  is replaced by  $iS$  so  $e^{iS} \rightarrow e^{-S}$  is now an exponentially decaying function. The Wick rotation also has the effect of replacing the 4D Minkowski spacetime of quantum field theory with a 4D Euclidean spacetime since, for example, when we replace  $t$  with  $it$ , the spacetime interval  $s^2 = -(ct)^2 + x^2 + y^2 + z^2$  becomes a 4D Euclidean distance function  $s^2 = (ct)^2 + x^2 + y^2 + z^2$ .

In Standard Model calculations, one doesn't apply the whole Standard Model at once. Instead, one works with special cases of the Standard Model, which are powerful theories in their own right. For example, quantum electrodynamics (QED), which is the quantum field theory of electromagnetism, is a special case of the Standard Model. QED consists of two quantum fields—a fermion field  $\psi$  (electrons and positrons) and a gauge field  $A_\sigma$  (photons). Its Lagrangian density looks like

$$\mathcal{L}_{QED} = i\hbar c \bar{\psi} \gamma^\sigma (\partial_\sigma - ieA_\sigma) \psi - mc^2 \bar{\psi} \psi - \frac{1}{4\mu_0} F_{\mu\nu} F^{\mu\nu},$$

where  $F_{\mu\nu}$  is the electromagnetic field tensor, and  $A_\sigma$  is the electromagnetic four-potential. QED gives us a complete description of the interactions of electrons and positrons via the exchange of photons.

The electroweak theory combines the electromagnetic and weak interactions in a single quantum field theory. Its Lagrangian density looks like

$$\mathcal{L}_{EW} = \sum_{\psi} \bar{\psi} \gamma^\mu \left( i\partial_\mu - g' \frac{1}{2} Y_w B_\mu - g \frac{1}{2} \vec{\tau}_L \vec{W}_\mu \right) \psi - \frac{1}{4} W_a^{\mu\nu} W_{\mu\nu}^a - \frac{1}{4} B^{\mu\nu} B_{\mu\nu}.$$

In electroweak theory, the bosons  $W^+$ ,  $W^-$ ,  $Z^0$ , and the photon  $\gamma$  come from the three-component  $\vec{W}_\mu$  gauge field and the  $B_\mu$  gauge field.

Quantum chromodynamics (QCD) is the theory of the strong interaction, which occurs between quarks and gluons. The fields of QCD consist of six quark fields  $\psi^{(f)}(x)$  where  $f \in \{u, d, s, c, t, b\}$  are the six quark flavors, and the gluon fields  $A_\mu^a$ . The Lagrangian density has the form

$$\mathcal{L}_{QCD} = \sum_f \bar{\psi}_f (\gamma^\mu (\partial_\mu + iA_\mu) + m_f) \psi_f - \frac{1}{4g^2} \sum_{a=1}^8 G_{\mu\nu}^a G_a^{\mu\nu}. \quad (1)$$

Here,  $G_{\mu\nu}^a$  is the gluon field strength tensor and contains the gluon fields  $A_\mu^a$ , where  $a = 1, \dots, 8$  is the gluon color index, and  $\mu = 0, \dots, 3$  is the Lorentz spacetime index.

These quantum field theories can all be thought of as special cases of the Standard Model, and they are all amenable to lattice calculations. The general idea is to start with the Lagrangian density of a quantum field theory, calculate the action from the Lagrangian density, and then perform a Wick rotation to Euclidean space and discretize for the lattice.

## 5 QFT on the Lattice

### 5.1 Theoretical Background

To formulate a quantum field theory on a lattice, we discretize the continuum Euclidean action. Recall that the Euclidean form of the action is obtained from the Minkowski (i.e. relativistic) form by performing a Wick rotation. To show an example that includes both fermion and gauge fields, I will briefly sketch this process for formulating QCD on the lattice.

Recall that the action is the integral of the Lagrangian density over all spacetime. For QCD, the action has the form

$$S = \int dx^4 \mathcal{L}_{QCD},$$

where  $\mathcal{L}_{QCD}$  is given by Eq. (1). To simplify things, we will break the action into a fermionic part

$$S_F = \sum_f \int dx^4 \bar{\psi}_f (\gamma^\mu (\partial_\mu + iA_\mu) + m_f) \psi_f,$$

and a gauge part

$$S_G = \frac{1}{4g^2} \sum_{a=1}^8 \int dx^4 G_{\mu\nu}^a G_a^{\mu\nu}.$$

In QCD, the fermion action is invariant under certain gauge transformations, which will not be discussed here. However, when discretizing the action, it must be ensured that the resulting lattice action satisfies the same symmetries.

In the continuum, the fermion fields are described by Dirac 4-spinors

$$\psi^{(f)}(x), \quad \bar{\psi}^{(f)}(x).$$

Think of the 4-spinor  $\psi^{(f)}(x)$  as a 4D vector whose elements are special functions. Such a vector sits at each point  $x$  in 4D spacetime. However, if we include the quark flavors  $f = 1, \dots, 6$  and colors  $c = 1, 2, 3$ , then at each point  $x$  in spacetime, the spinor  $\psi^{(f)}(x)$  really has 12 independent components, so rather than a vector, think of it as a 3D array sitting at each point in spacetime.

Next, we introduce a 4D Euclidean lattice. Each point  $n$  in this lattice is labeled with four components  $n = (n_x, n_y, n_z, n_t)$ . The size of a lattice is typically written like  $12^3 \times 64$ , where  $12^3$

means the lattice is a  $12 \times 12 \times 12$  cube extending 64 units in time. Now, instead of a spinor  $\psi^{(f)}(x)$  at each point  $x$  in continuous spacetime, we have a spinor at each point  $n$  in our lattice. The distance between adjacent lattice points is called the *lattice spacing* and is denoted  $a$ . So a physical spacetime point has the form  $(an_x, an_y, an_z, an_t)$ . We define the lattice to be periodic in all four dimensions. For example, on an  $N^4$  lattice, if you go  $N + 1$  units in any of the four orthogonal directions, you end up back where you started. Such a spacetime is often called a *torus*.

In the fermion action  $S_F$ , we have both an integral  $\int dx^4$  and a partial derivative  $\partial_\mu$ . In elementary calculus, integration is introduced as the limit of a Riemann sum and differentiation is introduced as the limit of a finite difference. We can naively discretize an integral or a derivative simply by doing the reverse—converting the integral back into a Riemann sum and the derivative back into a finite difference. The naive discretization doesn't work for the integrals in lattice field theory. For a  $12^3 \times 64$  lattice, there are  $12^3 \times 64 \approx 10^5$  lattice sites. There are 24 quark degrees of freedom for each flavor and 32 degrees of gluon flavor. So even if we only consider 2 quark flavors, our discretized path integrals still have roughly  $10^5 \times (2 \times 24 + 32) \approx 8 \times 10^6$  dimensions. Numerically solving such an integral using a Riemann sum is infeasible. Instead, we have to use more sophisticated Monte Carlo methods with importance sampling.

As noted earlier, when constructing the lattice action, we have to preserve certain gauge transformations. To do this for the fermion action, we have to introduce gauge fields. The gauge fields, instead of being arrays attached to each lattice point or *site* are arrays that must be attached to the *links* connecting the sites. These arrays are called *link variables*. For example,  $U_\mu(n)$  is the link variable attached to the link extending in the  $\mu$  direction from the lattice site  $n$ . On the lattice, a *plaquette* is the smallest closed loop. It is the smallest square with a lattice site at each vertex. The *plaquette variable*  $U_{\mu\nu}(n)$  is defined as the product of the four link variables about a single plaquette.

We are now ready to write down the lattice action  $S = S_F + S_G$  without derivation. The fermion part of the action is

$$S_F[\psi, \bar{\psi}, U] = a^4 \sum_{f=1} \sum_{n \in \Lambda} \left( \bar{\psi}^{(f)}(n) \sum_{\mu=1}^4 \gamma_\mu \frac{U_\mu(n) \psi^{(f)}(n + \hat{\mu}) - U_{-\mu}(n) \psi^{(f)}(n - \hat{\mu})}{a} + m^{(f)} \bar{\psi}^{(f)}(n) \psi^{(f)}(n) \right),$$

and the gauge part of the action is

$$S_G[U] = \frac{2}{g^2} \sum_{n \in \Lambda} \sum_{\mu < \nu} \text{Re tr} [1 - U_{\mu\nu}(n)].$$

Now that we have the action, we can write down the correlation function in terms of the lattice path integral as

$$\langle O_2(t) O_1(0) \rangle = \frac{1}{Z} \int \mathcal{D}[\psi, \bar{\psi}] \mathcal{D}[U] e^{-S_F[\psi, \bar{\psi}, U] - S_G[U]} O_2[\psi, \bar{\psi}, U] O_1[\psi, \bar{\psi}, U],$$

where the partition function is

$$Z = \int \mathcal{D}[\psi, \bar{\psi}] \mathcal{D}[U] e^{-S_F[\psi, \bar{\psi}, U] - S_G[U]},$$

and

$$\mathcal{D}[\psi, \bar{\psi}] = \prod_{n \in \Lambda} \prod_f d\psi^{(f)}(n) d\bar{\psi}^{(f)}(n), \quad \mathcal{D}[U] = \prod_{n \in \Lambda} \prod_{\mu=1}^4 dU_{\mu}(n).$$

We now have a discretized, Euclidean version of the QCD action and partition function, and these can be computed during a lattice simulation much like the partition function of the Ising Model is computed during a lattice simulation. In the end, we recover the physics of continuous spacetime by taking the infinite volume limit  $N \rightarrow \infty$  and the continuum limit  $a \rightarrow 0$ .

## 5.2 The Lattice Simulation

Our goal is to evaluate the vacuum expectation value of an observable in our quantized Euclidean field theory. However, because the path integrals are impossible to evaluate analytically, we use a Monte Carlo approach.

As with the Ising model, configurations (i.e. lattice states) are selected using a Markov Chain Monte Carlo (MCMC) method like the Metropolis algorithm<sup>1</sup>. As with the Ising model, the Metropolis algorithm is an *importance sampling* Monte Carlo method. When we applied the Metropolis algorithm to the Ising model, we choose a random dipole and considered the possibility of flipping it. If the energy of the resulting configuration is lower, we go ahead and flip the dipole. If the energy of the resulting state is not lower, we choose at random with a probability  $e^{-U/kT}$  whether or not to flip it. In lattice field theory, instead of looking at the change in energy  $\Delta U$ , we look at the change in the action  $\Delta S$ . Instead of  $e^{-U/kT}$ , we use  $e^{-S[U]}$ . Instead of a dipole with value  $\pm 1$  at each cell, we have more complicated objects with more possible values not only at each site in our lattice but also on the connections between the sites. As the algorithm moves from one configuration to another, by design, it tends toward the equilibrium configuration for the given parameters. For the Ising model, we had a single parameter—temperature, and we had to run the simulation for a while before it reached one of the many states near equilibrium for that temperature. In lattice field theory, the process is similar.

Before calculating observables, some large number of equilibrating Monte Carlo steps are performed so that the configuration is near equilibrium before measurements are taken. To say that we have reached an equilibrium configuration for the given set of parameters is to say that we have reached a realistic configuration. A configuration picked at random is probably not a realistic one, hence the MCMC method designed to start from any random configuration and find more realistic ones, which we are interested in measuring.

A lattice MCMC simulation typically has the following flow:

1. **Initialize:** Start with some initial lattice configuration.
2. **Equilibrate:** Run the simulation for some number,  $n_{equi}$ , of equilibration sweeps. In this process, new configurations are accepted according to the MCMC algorithm. When the entire

---

<sup>1</sup>In reality, the Metropolis algorithm is inefficient in lattice field theory, and typically, modified versions such as the *Heatbath* algorithm or *Overrelaxation* methods are used.

lattice has had a chance to be updated, one “sweep” has occurred. The number of equilibrating sweeps must be sufficiently large so that the final configuration is near equilibrium.

**3. Loop over iterations:**

- (a) Once the lattice is in an equilibrium configuration, observables can be evaluated. In other words, a “measurement” can be taken. At this point, the configuration is saved in a data file.
  - (b) Continue updating the lattice configuration for some number of sweeps. These are discarded. Configurations close to each other are correlated with each other, and measurements of successive configurations would be correlated with each other. Therefore, we discard a bunch of configurations between the ones that are saved for measuring in order to reduce the correlations between our measurements.
  - (c) Repeat this until you have some number,  $n_{meas}$  of saved configurations.
- 4. Measurements:** Now we “measure” each saved configuration for our observable(s) (e.g. the mass of the W boson). This gives us a total of  $n_{meas}$  measurements of each observable. Note: Sometimes this step is combined with the previous step.
- 5. Analysis:** Now, from our  $n_{meas}$  measurements of each observable, we can compute our final estimate (and error bar) of the observable by taking the average of our many measurements.

In this description, I skip a lot of details. For example, in the measurements and analysis steps, it is often not this straightforward. In my undergrad thesis project, we measured the masses of W and Higgs bosons. This was not just a simple matter of computing some number from each saved configuration and then computing an average. In reality, a mass is measured by performing an exponential fit to a Euclidean correlator obtained from a saved configuration. However, for any given configuration, the Euclidean correlator is fluctuating too much to obtain a mass estimate. Therefore, a Euclidean correlator is extracted from each saved configuration, and in the end, the correlators from all  $n_{meas}$  saved configurations are combined to get a smoothed, average correlation function. Only for this final correlation function is the exponential fit applied and the boson mass finally estimated.

The computational resources required for such a lattice simulation are enormous. As such, a simulation is typically parallelized to spread it out over multiple processors. Even so, a typical simulation that I did during my undergrad project took from several days to several weeks to complete.

## 6 Lattice Calculations

Here are a few of the things that people are doing in lattice field theory:

- Study phase transitions and map the phase diagram of a quantum field theory
- Calculate the mass/energy spectra of the particles in a quantum field theory
- Calculate particle decay constants
- Calculate the QCD running coupling  $\alpha_s$
- Study the quark-gluon plasma

Here, I will discuss two types of lattice calculations—mapping phase diagrams, and lattice spectroscopy. There are many other topics, but these are the ones I am most familiar with, and they give you a taste of the kind of calculations that can be done in lattice QFT.

## 6.1 Mapping the Phase Diagram

In quantum field theories, one finds phase transitions analogous to the phase transitions of classical statistical physics and thermodynamics. For example, in QCD, there is a confined phase at low temperature where quarks are bound together in hadrons. At high temperature and chemical potential there is a deconfined phase where quarks and gluons exist in a quark-gluon plasma. Identifying the location and properties of the phase transitions separating such different phases of matter is an active area of research in high energy physics.

Here, I will briefly discuss one approach for studying phase transitions in a lattice Higgs model. The lattice Higgs model I will look at is simple relative to lattice QCD, and computational results are much easier to obtain than in lattice QCD. I am using this model here simply because it is the one I did my undergraduate work on, and so I am most familiar with it.

The lattice action for this Higgs model is

$$S = \beta \sum_{\square} \left( 1 - \frac{1}{2} \text{tr } V_{\square} \right) - \kappa \sum_{x,\mu} \text{tr } V_{\mu}(x).$$

The  $V$  are gauge invariant link variables related to the  $U_{\mu}$  gauge link variables. The  $\beta$  and  $\kappa$  are two free parameters in this model<sup>2</sup>. The symbol  $\square$  denotes a single *plaquette*. So the first sum is a sum over all plaquettes in our lattice, and the second is a sum over all link variables. That is, it sums over all lattice sites  $x$  and over all link directions  $\mu$  at each site. Remember, a link is the connection between adjacent lattice sites.

The phase diagram of this Higgs model is believed to look like that given in Fig. (13). The region above the transition line is typically called the “Higgs phase,” and the region below is typically called the “confinement phase.”

In statistical mechanics, one can locate phase transitions by finding where susceptibilities (the quantification of how one quantity changes in response to a change in another quantity) diverge. In the lattice Higgs model, we can use the same approach. Since we are on a finite lattice, our susceptibilities don’t blow up to infinity at a phase transition. Nevertheless, we expect them to have a peak at a phase transition.

The quantity or *observable* we measure is the average link  $L$  defined as<sup>3</sup>

$$L = \left\langle \frac{1}{2} \text{tr } V_{\mu}(x) \right\rangle.$$

---

<sup>2</sup>In many lattice Higgs models discussed in the literature, there is a third free parameter  $\lambda$  in the action. The action shown here is what we obtain when we set  $\lambda = \infty$  in those models. Our simpler model is then called a *frozen* Higgs model.

<sup>3</sup>There are plenty of other observables we could define instead or in addition, such as the average plaquette or the average action per site, but the average link value serves as a representative quantity.

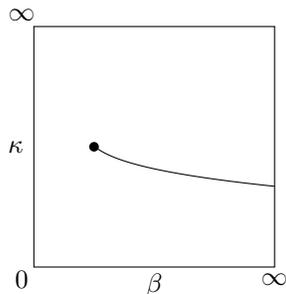


Figure 13: This is what the phase diagram of our Higgs model is believed to look like. In many lattice Higgs models discussed in the literature, there is a third free parameter  $\lambda$  in the action. The phase space of such a theory is now a three dimensional volume, and what we show here is a 2D slice of that phase volume.

When we run our simulations, we automatically calculate quantities like  $\text{tr } V_\mu(x)$  in order to calculate the action  $S$ . During a simulation, these quantities are saved to data files for later analysis. So calculating the observable  $L$  is a simple matter of extracting the elements from the data file, putting them together in the manner specified by the definition of the observable, and taking their average. What we really want, however, is not the observable  $L$ , but rather the susceptibility of  $L$ .

In general, the susceptibility  $\chi$  of a quantity  $X$  is the difference between the squared mean and the mean squared

$$\chi = \langle X^2 \rangle - \langle X \rangle^2.$$

There may be a factor out front, but it is not really important, since we don't care about the actual value of the susceptibility. We only care where it has a maximum.

Recall that  $\beta$  and  $\kappa$  are free parameters in our model. When we run a simulation, we do so by picking some pair of  $\beta$  and  $\kappa$ . Calculating the  $L$  susceptibility for various simulations (on  $12^3 \times 64$  lattices) and plotting them versus  $\kappa$  gives us the plot shown in Fig. (14). Each colored group is a set of simulations run at a set  $\beta$  value.

In this plot, we can easily identify peaks which suggest the location of a phase transition. Each peak corresponds to a single point on the phase transition line. The four peaks shown in the susceptibilities, thus give us four points on this transition line, and we have the beginning of a phase diagram as shown in Fig. (15). However, this is a very preliminary sketch. One would need to repeat the simulations at larger lattices to identify finite size effects, and remove them before one could start sketching the true phase diagram.

## 6.2 Lattice Spectroscopy

Spectroscopy, in this context, is the measurement of particle masses on the lattice. The first step in such a measurement is to identify lattice operators that couple to the particles of interest. Next, a lattice simulation is run and correlation functions constructed with these particle operators are calculated over a large number of lattice configurations. In the end, particle masses can be extracted

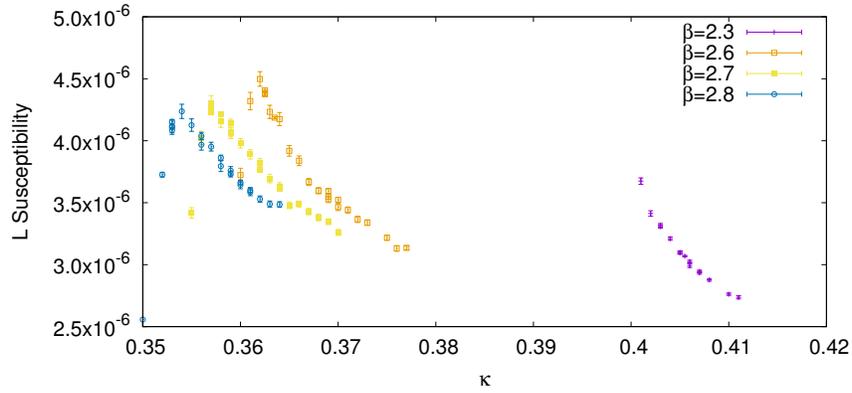


Figure 14: Plots of the  $L$  susceptibility versus  $\kappa$  performed at different  $\beta$  values. The peaks in the susceptibility suggest the location (in  $\kappa$ ) of the phase transition line for that value of  $\beta$ .

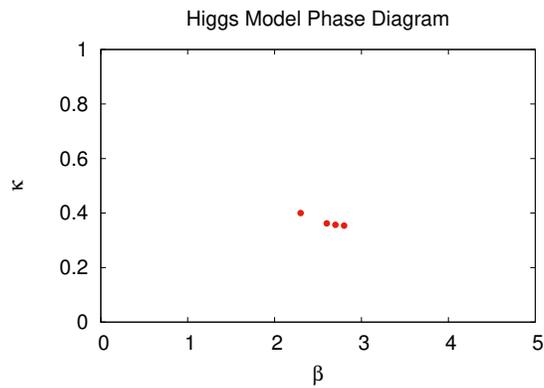


Figure 15: A few preliminary points on the phase transition line of our Higgs model. Each point corresponds to a peak in the  $L$  susceptibility. These four points are the result of about 60 lattice simulations—each which took about one to two weeks to run on a university cluster. Each simulation was run on a  $12^3 \times 64$  lattice, with  $\beta = 2.3, 2.6, 2.7, 2.8$  and each of those at about 15 different  $\kappa$  values.

from those correlation functions by applying various curve-fitting techniques.

After a *particle operator*, also known as a *particle interpolator* or an *annihilation operator*  $\mathcal{O}$  is identified, you obtain its conjugate  $\tilde{\mathcal{O}}$  known as the *creation operator*. These operators are chosen such that the corresponding Hilbert space operators  $\hat{\mathcal{O}}$  and  $\hat{\mathcal{O}}^\dagger$  annihilate and create the particle states we want.

For example, in [4] the Higgs and W boson operators

$$h \equiv \text{tr} [V_\mu(x)], \quad \text{and} \quad w_{r\mu} \equiv \text{tr} [\tau_r V_\mu(x)],$$

are presented. Then, for example, for the Higgs operator, we define the conjugate operator [1, 2]

$$\tilde{h}(\vec{p}, n_t) = \sum_{\vec{x}} e^{i\vec{p}\cdot\vec{x}} h(\vec{x}, n_t),$$

where

$$p_i = \frac{2\pi n_i}{N}, \quad n_i = 0, 1, \dots, N-1,$$

and where  $i = 1, 2, 3$  and  $N^3$  is the spatial part of the lattice. Typically, for numerical calculation of the masses, the operator is projected to zero momentum, i.e.  $\vec{p} = \vec{0}$ , then as derived in [2], the corresponding *Euclidean correlator* or *Euclidean correlation function* is

$$C(n_t) = \langle \tilde{h}(\vec{0}, n_t) \bar{h}(\vec{0}, 0) \rangle = \sum_k \langle 0 | \hat{h} | k \rangle \langle k | \hat{h}^\dagger | 0 \rangle e^{-n_t E_k}. \quad (2)$$

This shows that the correlation function is a sum of exponential functions

$$C(n_t) = A_0 e^{-n_t E_0} + A_1 e^{-n_t E_1} + A_2 e^{-n_t E_2} + \dots.$$

We are using  $n_t$  as the time variable to emphasize that the variable takes discrete values when we are on the lattice, using lattice units  $a = 1$ . To revert to normal units, one of the energies or masses is used to set the scale.

In units where  $c = 1$ , the energy of a particle with rest mass  $m_0$  as a function of momentum is

$$E(p) = \sqrt{m_0^2 + p^2}.$$

For momentum  $p = 0$ , we have  $E = m_0$ , so that the leading contribution to the correlation function Eq. (2) comes from the lowest mass. For sufficiently large  $n_t$  the first exponential term dominates, and we have

$$C(z) \simeq A_0 e^{-m_0 z}, \quad (3)$$

where  $m_0$  is the rest mass of the particle being estimated. Given the computed values<sup>4</sup>  $C(z)$ , we can apply a least squares fit of this form to approximate the values of  $A_0$  and the mass  $m_0$ . In short, our simulation will produce a series of correlation function values  $C(1), C(2), C(3), \dots$  forming a discrete curve. We then apply a nonlinear fit of the form (3) with the free parameters  $A_0$  and  $m_0$ . Our fitting algorithm adjusts these two parameters until a best fit is found. Then the value of the parameter  $m_0$  is our mass estimate. This is illustrated in the example fits shown in Fig. (16), where

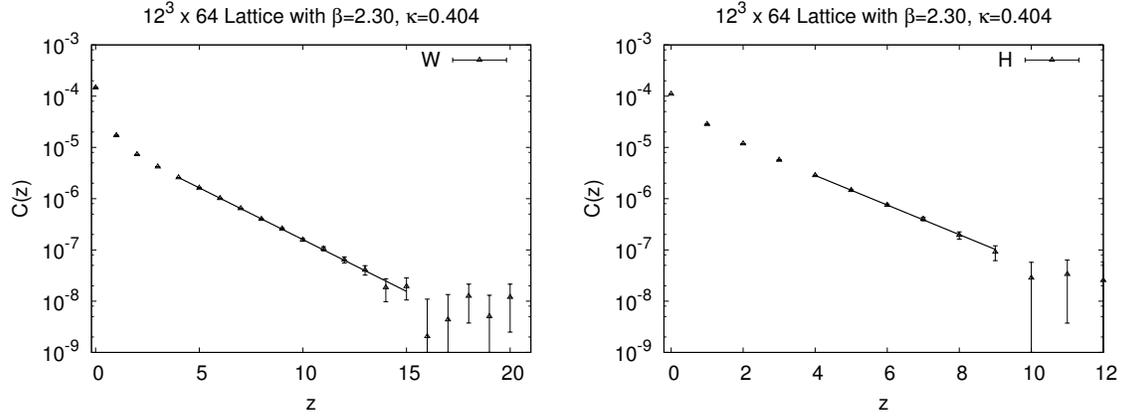


Figure 16: Here we have typical semi-log plots of  $C(z)$  for the W correlation function (on the left) and the Higgs correlation function (on the right). The solid lines display the 1-mass, (2-parameter) fits of the form Eq. (3) for the W mass and Higgs mass. Initial points are excluded because of unwanted higher mass contributions to the correlation function. I.e. initial points are contaminated by the exponentials that we have neglected in our fit. Later points are excluded because statistical errors become large at larger time values.

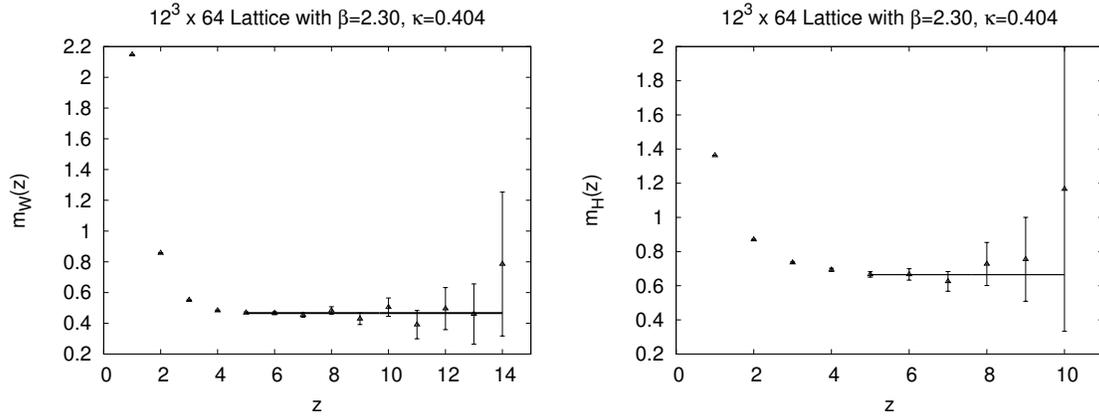


Figure 17: An example of the ratio fit. Here we plot  $m(z)$  from Eq. (4) for the W mass (on the left) and the Higgs mass (on the right). The asymptotic value  $m(z) \rightarrow m_0$  is estimated by applying a horizontal fit after excluding some of the initial points.

this “1-mass fit” is performed for the W mass (left panel) and the Higgs mass (right panel). In this example, a  $12^3 \times 64$  lattice was used, with  $\beta = 2.30$  and  $\kappa = 0.404$ .

To check for consistency or obtain a better fit we can include the second exponential term

$$C(z) \simeq A_0 e^{-m_0 z} + A_1 e^{-m_1 z},$$

and perform a four parameter fit with the parameters  $A_0$ ,  $m_0$ ,  $A_1$ , and  $m_1$ . In that case, the parameter of primary interest is still  $m_0$ —the mass associated with the ground state energy.

We can also estimate  $m_0$  by looking at the ratio of consecutive values of the correlation function

$$\frac{C(z)}{C(z+1)} = \frac{A_1 e^{-m_0 z} + A_2 e^{-m_1 z} + A_3 e^{-m_2 z} + \dots}{A_1 e^{-m_0(z+1)} + A_2 e^{-m_1(z+1)} + A_3 e^{-m_2(z+1)} + \dots}.$$

For large  $z$ ,

$$\frac{C(z)}{C(z+1)} \simeq \frac{A e^{-m_0 z}}{A e^{-m_0(z+1)}} = e^{m_0}.$$

Taking the natural logarithm of both sides gives us

$$m(z) = \ln \left( \frac{C(z)}{C(z+1)} \right), \quad (4)$$

where  $m(z) \rightarrow m_0$  as  $z \rightarrow \infty$ . Asymptotically, this logarithm approaches our mass  $m_0$ . Since  $m(z) > m_0$ , our mass estimate here is an upper bound. We call this approach the “ratio fit.”

Fig. (17) illustrates a typical ratio fit of the W mass in the left panel and the Higgs mass in the right panel. In this example, the simulation was on a  $12^3 \times 64$  lattice with  $\beta = 2.30$  and  $\kappa = 0.404$ . To obtain the asymptotic value, a horizontal line is fitted to Eq. (4). Since the mass is estimated from the asymptotic value of Eq. (4), some initial points must be discarded.

In our model,  $\beta$  and  $\kappa$  are free parameters, so the mass we calculate on the lattice is not immediately the actual mass of the physical particle. The lattice calculated mass is a function of  $\beta$  and  $\kappa$ , and we don’t necessarily know the correct  $\beta$  and  $\kappa$  that we should use in order to get an estimate of the actual mass of the particle. However, these measurements of the Higgs and W boson mass spectra are still useful. For example, exploring the dependence of particle masses on the parameters  $\beta$  and  $\kappa$  may give us insights into the continuum limit of this theory, and could even point to new physics.

---

<sup>4</sup>We use  $z$  rather than  $n_t$  because the correlation function is really of time *differences*. E.g.  $C(z=2)$  is the average value the correlation function gives for all pairs of points separated by a time distance of 2.

## References

- [1] B. Berg, *Markov Chain Monte Carlo Simulations and Their Statistical Analysis* (World Scientific, 2004).
- [2] C. Gattringer and C. B. Lang, *Quantum Chromodynamics on the Lattice* (Springer, 2010).
- [3] D. J. Griffiths, *Introduction to Electrodynamics, Ed. 3* (Prentice-Hall, 1999).
- [4] W. Langguth, I. Montvay, and P. Weisz, “Monte Carlo Study of the Standard SU(2) Higgs Model.” Nucl. Phys. **B27** (1986) 11-49.
- [5] I. Montvay and G. Munster, *Quantum Fields on a Lattice* (Cambridge, 1994).
- [6] G. Munster and M. Walzl, “Lattice Gauge Theory: A Short Primer” (arXiv:hep-lat/0012005). ([link](#))
- [7] D. V. Schroeder, *An Introduction to Thermal Physics* (Addison-Wesley, 2000).
- [8] J. Smit, *Introduction to Quantum Fields on a Lattice* (Cambridge, 2002).
- [9] M. Wagner et al, “An introduction to lattice hadron spectroscopy for students without quantum field theoretical background” (arXiv:1310.1760, 2013). ([link](#))
- [10] K.G. Wilson, “The Origins of Lattice Gauge Theory” Nucl. Phys. B **140** (2005) 3-19.

# A Appendix

## A.1 Program 1

```
#!/usr/bin/env python
"""
Ising Model

Uses a small lattice (e.g. 20x20) and displays
only the final lattice state.
"""

from __future__ import division, print_function
import numpy as np
import random as rand
import pylab as plt

def deltaU(i, j):
    """
    For a hypothetical dipole flip, compute the potential energy
    change. Each dipole has four nearest neighbors--left, top, right
    and bottom.

    Notice the periodic boundary conditions.
    """
    top = s[size-1,j] if i==0 else s[i-1,j]
    bottom = s[0,j] if i==size-1 else s[i+1,j]
    left = s[i, size-1] if j==0 else s[i,j-1]
    right = s[i,0] if j==size-1 else s[i,j+1]
    Ediff = 2*s[i,j]*(top+bottom+left+right)
    return Ediff

def initialize():
    """
    Initialize the lattice by randomly selecting 1 or -1
    for each dipole.
    """
    for i in range(size):
        for j in range(size):
            s[i,j] = 1 if rand.random()<0.5 else -1

#####
# MAIN PART OF PROGRAM
#####

size = 20                                # Size of the square lattice
T = 10                                    # Temperature in units of epsilon/k
upperLimit = 1000*size*size              # Each dipole is flipped an
```

```

# average 1000 times
s = np.zeros((size, size), dtype=int) # Create the lattice as a 2D array
initialize() # Initialize the array
plt.figure() # Initialize the plot

for iteration in range(1, upperLimit): # The main iteration loop
    i = rand.randint(0, size-1) # Choose a random dipole
    j = rand.randint(0, size-1)
    Ediff = deltaU(i, j) # Compute the energy change of a
                        # hypothetical flip
    if Ediff <= 0: # Flip the dipole if energy is reduced
        s[i,j] = -s[i,j]
    else: # Else, the Boltzmann factor gives
        if rand.random() < np.exp(-Ediff/T): # the probability of flipping
            s[i,j] = -s[i,j]

    if iteration % 100000 == 0:
        print((iteration/upperLimit)*100, "% done")

plt.imshow(s, interpolation='nearest') # Plot the final configuration
plt.show()

```

## A.2 Program 2

```

#!/usr/bin/env python
"""
Ising Model

This program displays the final lattice state
as well as the correlation function for the final
lattice state.
"""

from __future__ import division, print_function
import numpy as np
import random as rand
import pylab as plt

def deltaU(i, j):
    """
    For a hypothetical dipole flip, compute the potential energy
    change. Each dipole has four nearest neighbors--left, top, right
    and bottom.

    Notice the periodic boundary conditions.
    """

```

```

"""
top = s[size-1,j] if i==0 else s[i-1,j]
bottom = s[0,j] if i==size-1 else s[i+1,j]
left = s[i, size-1] if j==0 else s[i,j-1]
right = s[i,0] if j==size-1 else s[i,j+1]
Ediff = 2*s[i,j]*(top+bottom+left+right)
return Ediff

def initialize():
    """
    Initialize the lattice by randomly selecting 1 or -1
    for each dipole.
    """
    for i in range(size):
        for j in range(size):
            s[i,j] = 1 if rand.random()<0.5 else -1

def corr(r):
    """
    The correlation function for distance r
    """
    avg = 0
    for i in range(size):
        for j in range(size):
            me = s[i,j] # self
            ab = s[(i-r)%size, j] # above
            be = s[(i+r)%size, j] # below
            le = s[i, (j-r)%size] # left
            ri = s[i, (j+r)%size] # right

            avg += me*ab + me*be + me*le + me*ri

    avg = avg/(4*size*size)
    M = np.sum(s)/(size*size) # Overall magnetization
    return avg - M*M

#####
# MAIN PART OF PROGRAM
#####

size = 100 # Size of the square lattice
T = 2 # Temperature in units of epsilon/k
upperLimit = 1000*size*size # Each dipole is flipped an
# average 1000 times
s = np.zeros((size, size), dtype=int) # Create the lattice as a 2D array
initialize() # Initialize the array
plt.figure() # Initialize the plot
Mlist = [] # List to store magnetization values

```

```

for iteration in range(1, upperLimit):      # The main iteration loop
    i = rand.randint(0, size-1)            # Choose a random dipole
    j = rand.randint(0, size-1)
    Ediff = deltaU(i, j)                   # Compute the energy change of a
                                           # hypothetical flip
    if Ediff <= 0:                          # Flip the dipole if energy is reduced
        s[i,j] = -s[i,j]
    else:                                    # Else, the Boltzmann factor gives
        if rand.random() < np.exp(-Ediff/T): # the probability of flipping
            s[i,j] = -s[i,j]

    Mlist += [np.abs(np.sum(s))]

    if iteration % 100000 == 0:
        print((iteration/upperLimit)*100, "% done")

corrfun = []
xlist = []
for r in range(1, int(size/2)+1):          # Compute the correlation function
    xlist += [r]
    corrfun += [corr(r)]

plt.subplot(1,2,1)                          # Plot the final configuration
plt.imshow(s, interpolation='nearest')
plt.axis('off')
plt.subplot(1,2,2)                          # and the correlation function
plt.plot(xlist, corrfun)
plt.show()

```