

# Project 2: Monte Carlo methods for physical computations

Gabriel Smadi  
College of Engineering,  
College of Arts  
Sciences,  
Physics Department,  
Syracuse University,  
gsmadi@syr.edu

April 3, 2017

## Abstract

Monte Carlo methods are investigated for physical computations. Their usage is inspected by approximating the volume of a hypersphere. Also, using a Markov Chain Monte Carlo (MCMC) algorithm, specifically the Metropolis algorithm, a ferromagnetic material is modeled by means of the Ising model.

## 1 Introduction

Monte Carlo methods can arrive at the same result of any deterministic solution through the means of randomness. Through the generation of random numbers, Monte Carlo methods can achieve accurate results for several types of computations such as integrations and optimizations. However, Monte Carlo methods also prove useful for simulating complex systems such as the Ising model, which is explored in the final section of this writing.

There are several aspects that make Monte Carlo methods attractive in a wide myriad of problems. First, using such methods it is relatively easy to incorporate complex boundary conditions, such as the integration over a intricate surface. Also, these random methods can achieve better performance over higher dimensions than other algorithms. For example, integration over  $d$  dimensions, Simpson's Rule converges to the result with an error  $\propto N^{-4/d}$ , whereas Monte Carlo integration converges to the result with an error  $\propto 1/\sqrt{N}$ .

## 2 Computation of a hypersphere volume

A enlightening Monte Carlo method problem is the computation of a hypersphere's volume. The simplest hypersphere to start with is one with 2 dimensions, most commonly known as a circle.

Since a circle does not have sufficient dimensions to contain a volume, the analogue of volume for two dimension is area. Thus, for a circle, the area is given by,

$$A_{circle} = \pi r^2 \tag{1}$$

To simplify matters, we compute the area of a circle of radius  $r = 1$ . The methodology employed to compute the area of a circle is described below,

1. Generate a random coordinate  $(x_i, y_i)$  inside a square of size  $1 \times 1$ .
2. For a circle, the randomly generated point is inside the circle if  $x_i^2 + y_i^2 \leq r^2$ .
3. If randomly generated coordinate satisfies the condition in step 2, increment  $n$  by 1.
4. Perform step 1, 2, 3 for  $N$  times.

After completing the  $N$  iterations, the total number of generated coordinates is contained in  $n$ . Thus, to approximate the area of the circle we use the following expression,

$$A_{circle} \approx 4\left(\frac{n}{N}\right)r^2 \quad (2)$$

Equation 2 contains a factor of 4 because the generated coordinates lie inside a quarter of a circle. The ratio  $n/N$  signifies the number of points that fell inside the quarter circle versus the total number of points generated. Since we are computing the area of a circle of radius  $r = 1$ , then by equation 1 we should expect the area to be  $\approx \pi$ . In Figure 1, the simulation to approximate the area of a circle is illustrated for different iterations.

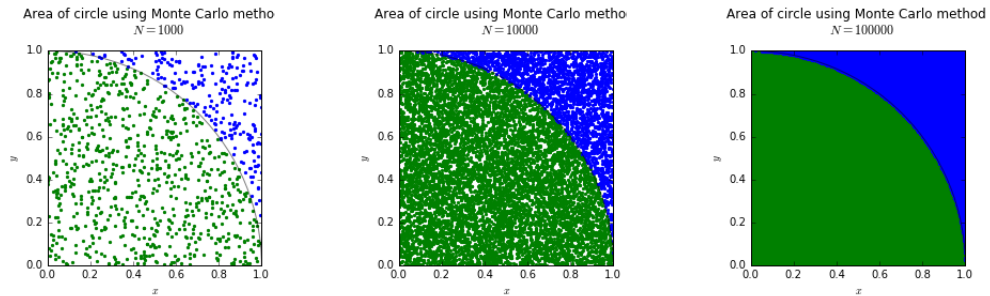


Figure 1: Approximation of of circle area via Monter Carlo method

The simulation was carried out for  $N = 10$  until  $N = 100000000$ . In the table below are the results, for several iterations.

Iterations	Computed	Error
10	2.4	23.6%
100	2.88	8.3%
1000	3.132	0.3%
10000	3.1316	0.3%
100000	3.13648	0.2%
1000000	3.139876	0.05%
10000000	3.141691	0.003%
100000000	3.141530	0.001%

Table 1: Circle area approximation for several  $N$  iteration.

The results from table 3 can be appreciated visually in the figure below.

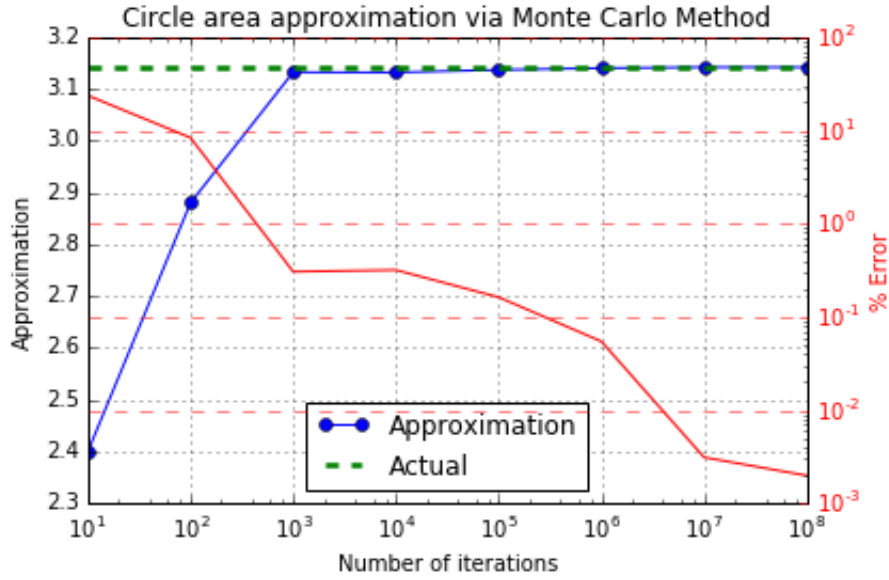


Figure 2: Convergence of circle area approximation to actual value as  $N \rightarrow \infty$

After  $N = 1000$  the approximation contains a % Error of about 0.3% or  $A_{approx} \approx 3.132$ , which is quite close to the value of  $\pi$ .

Now, since we have seen satisfactory results for the approximation of the area of a circle we can do the same with the volume of a sphere, where dimension  $d = 3$ . For a sphere, the procedure is the same as a circle the only different is that for step 2, the condition is restated with an extra dimension as  $x_i^2 + y_i^2 + z_i^2 \leq r^2$ .

The volume of a sphere is given by,

$$V_{sphere} = \frac{4}{3}\pi r^3 \quad (3)$$

thus, for a sphere of radius  $r = 1$ , we should expect  $V_{sphere} \approx 4.189$ . The convinience of using Monte Carlo methods for such computations is that no matter the dimensionality of the computation, the form of equation 2 still holds even for a sphere of dimension  $d = 3$ . In other words, the form of equation 2 holds for any  $d$ . Hence, the task at hand, is to simply generate random coordinates inside a hypercube (e.g.  $d = 2$ : square,  $d = 3$ : *cube*, etc.), count the ocurrences inside the hypersphere and compute the volume using equation 2.

Therefore, to generally define the approximation of a hypersphere for any dimension  $d = 3$ , in layman terms a sphere.

$$V_{hypersphere,d} = \left(\frac{n}{N}\right)(2r)^d \quad (4)$$

The factor of 2 is inside the exponent term becuase, for a circle we computed a quarter of the area, for a sphere we shall computed an eighth of the volume, and so forth.

Below the simulation for computing the volume of a hypersphere of dimension  $d = 3$  or in layman terms a sphere.

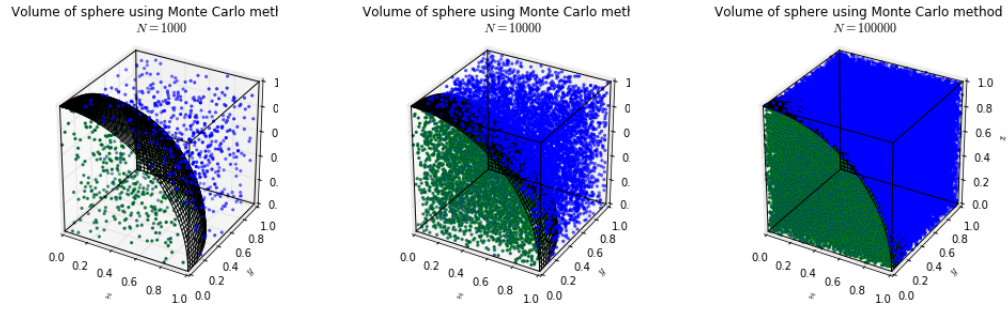


Figure 3: Approximation of sphere volume via Monter Carlo method

Iterations	Computed	Error
10	4.8	14.6%
100	4.64	10.8%
1000	4.088	2.4%
10000	4.1616	0.6%
100000	4.1704	0.4%
1000000	4.1858	0.07%
10000000	4.188134	0.02%
100000000	4.189403	0.01%

Table 2: Sphere volume approximation for several  $N$  iteration.

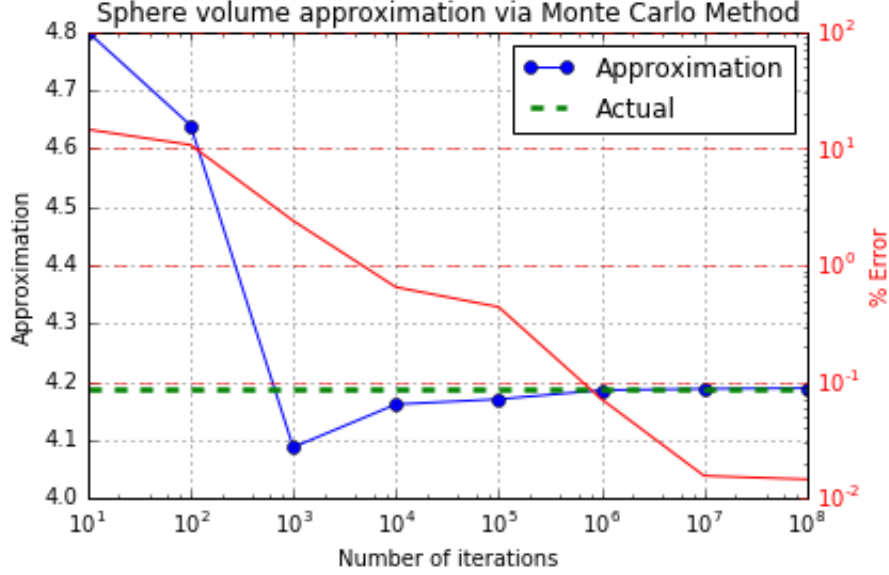


Figure 4: Convergence of sphere volume approximation to actual value as  $N \rightarrow \infty$

After  $N = 10000$  the volume approximation is very close to the actual value, with a % Error of 0.6%.

Now that we have tested our Monte Carlo method for managable hypersphere configurations with satisfactory results, let us investigate a higher dimensional hypersphere. From theory, the volume of a 10 dimensional hypersphere is given by,

$$V_{10-dim} = \frac{\pi^5}{120} r^{10} \quad (5)$$

Hence, for a hypersphere of radius  $r = 1$ ,  $V_{10-dim} \approx 2.55$ . For 10 dimensions, the condition for checking if a random coordinate is inside the hypersphere is given by,

$$\sum_{n=1}^{10} x_i \leq r^2 \quad (6)$$

where  $x_1 = x$ ,  $x_2 = y$ ,  $x_3 = z$ , would be the typical x, y, z coordinate space we live in. Given the "curse of dimensionality", visualizing the simulation would be cumbersome and unintuitive due to the difficulty of representing 10 dimensions. Thus, we view the results directly in this case.

Iterations	Computed	Error
10	0.0	100%
100	0.0	100%
1000	1.024	59.8%
10000	2.8672	12.4%
100000	2.580480	1.2%
1000000	2.56	0.4%
10000000	2.560205	0.4%
100000000	2.542408	0.3%

Table 3: Volume approximation of 10 dimensional hypersphere for several  $N$  iteration.

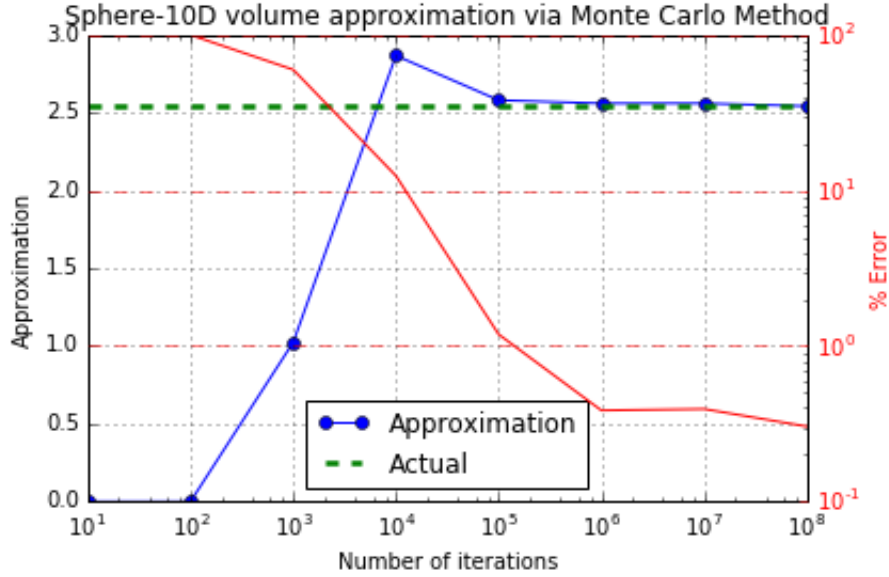


Figure 5: Convergence of 10 dimensional hypersphere volume approximation to actual value as  $N \rightarrow \infty$

For a 10 dimensional hypersphere, the approximate value converges closely to the actual value around  $N = 10^5$ . Contrasting the results with the circle ( $d = 2$ ) and sphere ( $d = 3$ ), the general trend appears to be the more dimensions, the larger the required  $N$  to converge to the actual value. Intuitively this would makes sense since as the number of dimension  $d$  is growing the less probable it is for a randomly generated coordinate to be inside the hypersphere.

### 3 Ising Model

The Ising model is a model of ferromagnetism where phase transitions can be identified. The simplest and most approachable configuration is a the Square-lattice Ising model and the one we shall be investigating. The model is laid out in a square lattice of size  $N = L \times L$ . From this model we may study a set up observables, particularly, we shall be being attention to the magnetization of the configuration.

The Hamiltonian of a particular location on the lattice is given by,

$$H_i = -J \sum_{j_{nn}} s_i s_j \quad (7)$$

where,  $J$  is a coupling constant,  $s_i$  is the location choosen to measure the energy and  $s_j$  are the neighboring spins to  $s_i$ . In simulation we shall take  $J = +1$ , which implies the material in question would be ferromagnetic. Also, in the implementation presented  $k_B$  is taken to be 1. Also, the ferromagnetic is absent of a magnetic field in our simulation.

The magnetization of the system is given by,

$$\langle M \rangle = \frac{1}{N N_{mc}} \sum_{s_i, s_j}^N s_{i,j} \quad (8)$$

where  $N$  is the size of the lattice,  $N_{mc}$  is the number of monte carlo loops. The equation above will simply sum over all the spins in the lattice and that will be the average magnetization. Note in theory, the average magnetization must be multiplied with a corresponding probability, however, in this case the system is being updated by a corresponding probability thus such multiplication is not neccesary.

The important computational detail that concerns the Ising model implementation is deciding when and how to update the system. This is where the Metropolis algorithm plays its part. The Metropolis algorithm is a Markov Chain Monte Carlo (MCMC), that permits random sampling of a probability distribution. In our context, the probability distibution in concern is the probability density of the systems energy, where the transition probability is given by  $e^{\delta H/k_B T}$ . The Markov chain portion of the algorithm is concerned with the random walk that will permit us to visit each state of the model that minimizes the change in energy at a particular temperature. While, the Monte Carlo portion of the algorithm permits us to sample the system many times to obtain an average value for an observable (e.g. magnetization, energy, etc.) at a particular state of the system. The diagram below shows the Metropolis algorithm with is corresponding parts dissected.

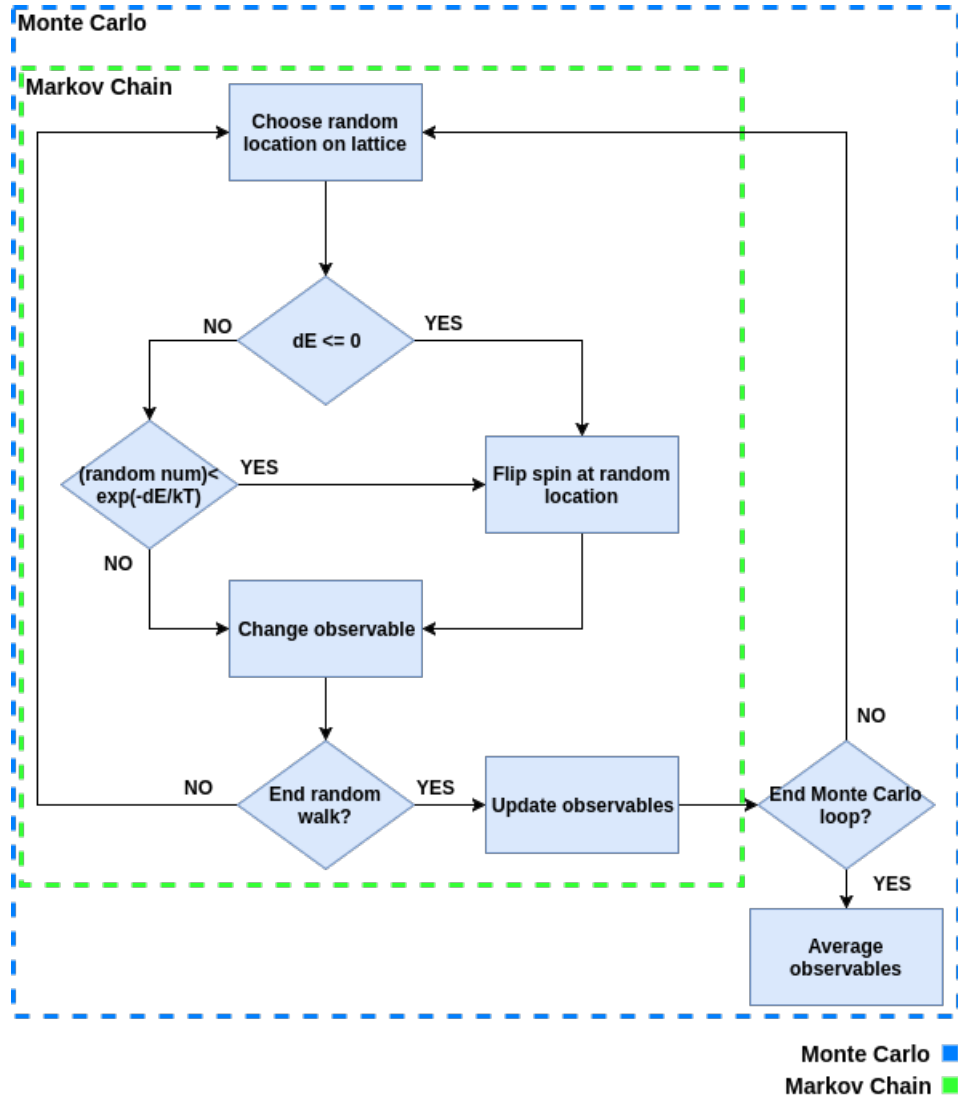


Figure 6: Metropolis algorithm adapted from Kotze[4]

For the simulation of the Ising model, the observables studied were  $\langle M^2 \rangle$  and  $\langle |M| \rangle$ . We also In the implementation the Metropolis algorithm is used to update the Ising model for temperatures from 0.1 to 5.0 with 0.1 increments. Below we visualize the simulation of 128 x 128 lattice Ising model for different temperatures.



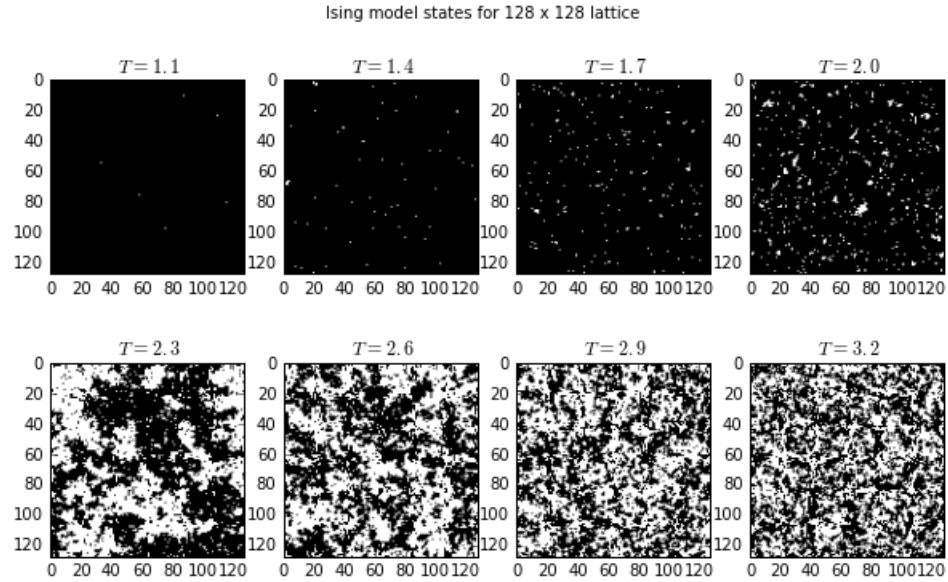


Figure 7: Evolution of Ising model at different temperatures

From the figure above we see how the spins are organized in parallel for low temperatures. As the temperature increases the spins start desorganizing themselves randomly. Specifically, after the Curie temperature  $T_c$  is where the spins start to get disorganized. Later, we shall find the Curie temperature by visualizing magnetic susceptibility.

The average magnetization for different lattice sizes can be appreciated in the figure below. Around the Curie temperature the spins are aligned completely either spin up or spin down. After the Curie temperature, the spins average out the magnetization to zero.

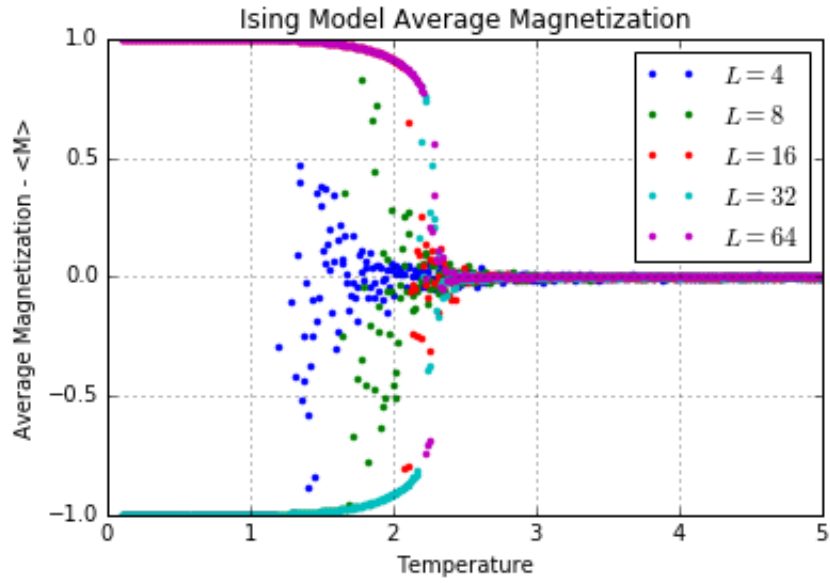


Figure 8: Average magnetization for different lattice sizes

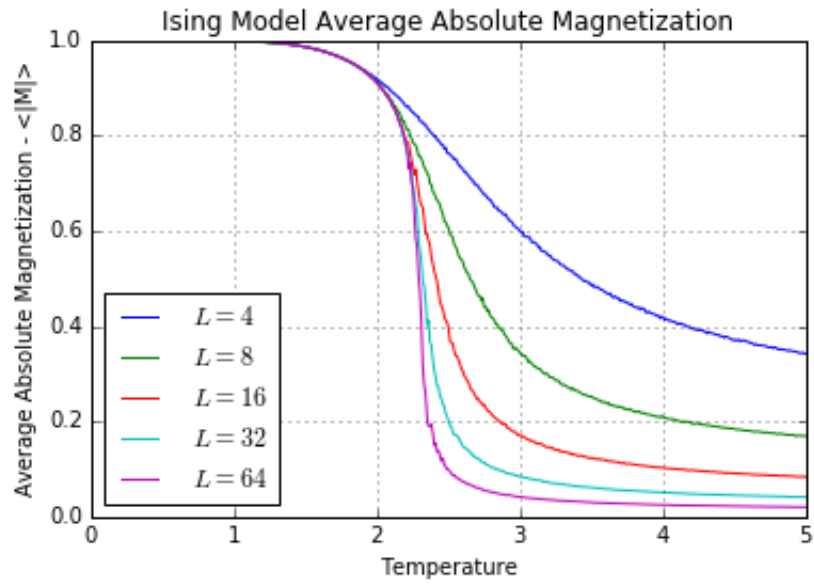


Figure 9: Absolute average magnetization for different lattice sizes

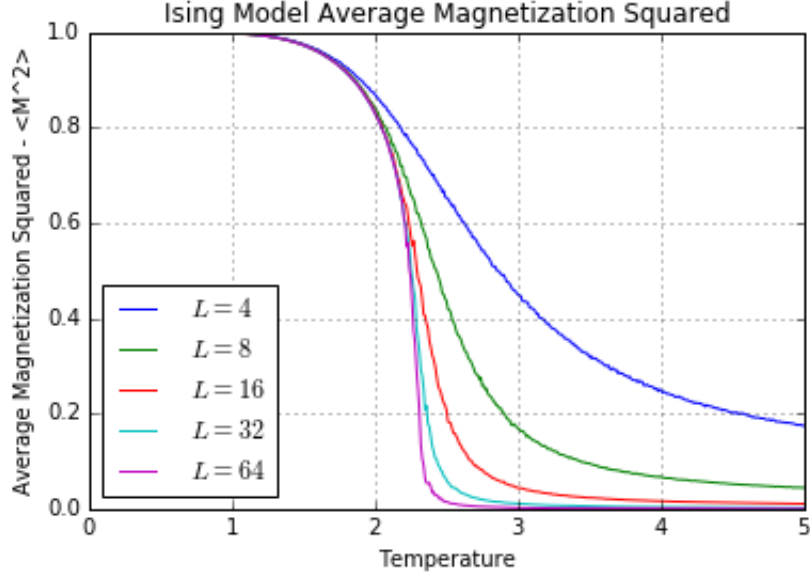


Figure 10: Normalized average magnetization squared for different lattices sizes

In the infinite volume limit, the magnetic susceptibility converges at the Curie temperature  $T_c$ . In theory, the value of the Curie temperature is given by,

$$T_c = \frac{2}{\ln(1 + \sqrt{2})} = 2.269185 \quad (9)$$

To find our Curie temperature we must find the temperature where the magnetic susceptibility is maximum. The largest lattice size for which data was obtained is  $N = 64 \times 64$ . Thus we prefer this lattice for later computations. The magnetic susceptibility was calculated as,

$$\chi'(T) = \frac{\langle M^2 \rangle - \langle M_{abs,avg}^2 \rangle}{k_B T} \quad (10)$$

with uncertainty,

$$\delta(\chi') = \delta(\langle M^2 \rangle) + \delta(\langle M_{abs,avg}^2 \rangle) \quad (11)$$

The average absolute magnetization squared was used as suggested by Kotze given that using the average magnetization provided skewness to the left for small lattice sizes. The figure below shows the magnetic susceptibility obtained for several lattice sizes.

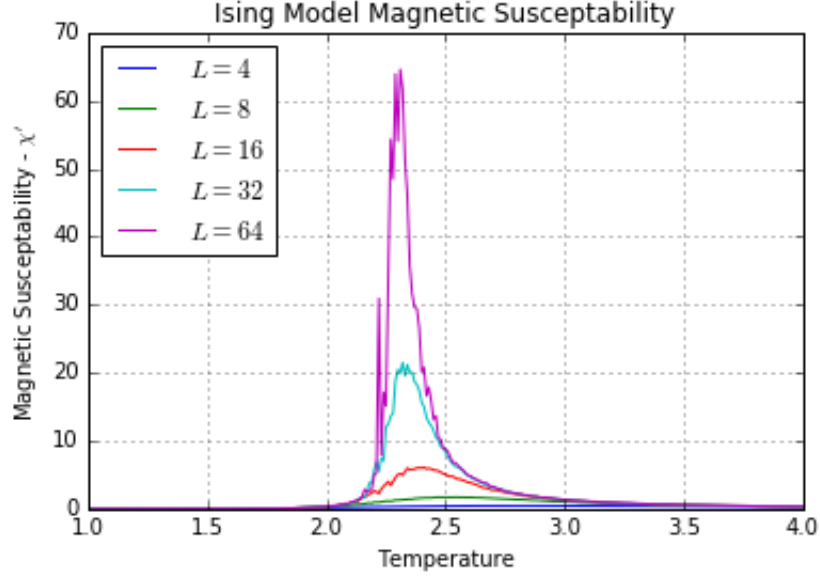


Figure 11: Convergence of errors as  $h \rightarrow 0$

From the data for a  $64 \times 64$  we find the Curie temperature  $T_c$ , where,

$$\chi'_{max} = 64.59 \quad (12)$$

The temperature in terms of the magnetic susceptibility is given by,

$$T(\chi') = \frac{\langle M^2 \rangle - \langle M_{abs,avg}^2 \rangle}{k_B \chi'} \quad (13)$$

which implies that,

$$T_{c,approx} = T(\chi'_{max}) = \frac{\langle M^2 \rangle - \langle M_{abs,avg}^2 \rangle}{k_B \chi'_{max}} \quad (14)$$

with uncertainty,

$$\delta T = \left| \frac{dT(\chi')}{d\chi'} \right| \delta \chi' = \left| \frac{\langle M^2 \rangle - \langle M_{abs,avg}^2 \rangle}{\chi'^2} \right| (\delta \langle M^2 \rangle + \delta \langle M_{abs,avg}^2 \rangle) \quad (15)$$

Thus, from equation 14, we find that,

$$T_{c,approx} = T(\chi'_{max} = 64.59) = 2.31 \quad (16)$$

and with uncertainty,

$$\delta T = \frac{2.31}{64.59} (0.47 + 0.38) = 0.03 \quad (17)$$

Therefore, our final reportable value for  $T_c$  is,

$$T_c = 2.31 \pm 0.03 \tag{18}$$

## 4 Conclusion

We have seen the power of Monte Carlo methods for physical computations. From the calculations of the hypersphere volume we saw how easy it is to incorporate boundary condition and to generalize these to greater dimensions. Also, we saw how quick these algorithms can converge to actual values even for high dimensions as the 10 dimensional hypersphere.

Beyond the power for computations, Monte Carlo methods are ideal for simulations as well. From the Metropolis algorithm we saw how Monte Carlo methods together with Markov chains can form a robust combination for sampling probability distributions. Using the Metropolis algorithm, we were able to update and visualize the Ising model and to find the Curie temperature obtaining consistent results from theory.

## References

- [1] Wikipedia contributors. *Monte Carlo method*. Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 28 Mar. 2017.
- [2] Wikipedia contributors. "Markov chain Monte Carlo." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 21 Feb. 2017.
- [3] Wikipedia contributors. "Square-lattice Ising model." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 26 Mar. 2016.
- [4] Jacques Kotze, *Introduction to Monte Carlo methods for an Ising model of ferromagnetism*. `arXiv:cond-mat.stat-mech/0803.0217`