Luke R. Johnson

CBE 641 – Nano Transport Processes

Computational Project

Due: April 26th, 2017

Project – The Ising Model

Presented to: Dr. John Crocker

Introduction

The Ising Model is a concept developed by Dr. Ising as a graduate student for his dissertation in 1925, under his advisor, Dr. Wilhelm Lenz, who invented the concept in 1920, to model the alignment of atoms and their spins in a lattice of other atoms, accounting for their interactions with other neighboring spins. What's interesting about this model is the way that temperature plays a role in the overall average alignment of spins in a system. Below a ferromagnet's critical temperature, most spins align in either the upward or downward direction, indicating that the material is acting as a ferromagnetic with a positive coupling constant. However, there is a sharp change, or steep convergence, in overall alignment near the critical temperature of the system, where it then takes on an overall neutral alignment to the right of the critical temperature. An important concept applied to the Ising model is that of the mean-field theory, in which we can use the averages of specific parameters to describe behavior near the critical temperature.

Specifically, for a 2-D Ising Ferromagnet, it has been determined that the critical temperature is approximately 2.2 joules per Boltzmann constant. As stated above, temperatures near the critical temperature allow us to determine the critical exponents, an intensive feature of systems that allow us to determine important thermodynamic properties. For the ferromagnet, these system properties are heat capacity, magnetization, correlation length, and magnetic susceptibility. The critical exponents are expressed as a power law for the difference between the critical temperature and the varied temperature.

$$M(T) = (T_c\text{-}T)^{\beta}$$

$$\chi(T) = (|T_c\text{-}T|)^{-\gamma}$$

$$\zeta(T) = (|T_c\text{-}T\})^{-\upsilon}$$

$$C(T) = (|T_c\text{-}T\})^{-\alpha}$$

The purpose of this computational practice is not just to obtain these critical exponents through a fitting analysis, but to compare them to the ones that were obtained by Onsager, who derived an expression for the free energy of the Ising model for an absent external field. These values were then obtained from his exact analytical model:

$$T_c = 2.2\frac{J}{k_B}$$

$$\alpha = 0, \beta = \frac{1}{8}, \gamma = \frac{7}{4}, \nu = 1$$

From this experiment, it is desirable to gain a better understanding of the Ising model and the physics/statistical thermodynamics that play a role in describing and causing this phenomenon. I had initially planned to examine superconductivity and critical temperatures of 2D materials related to my research by using the Ising model, but time permitted did not allow for that. There is a lab in SEAS at MSE that examines the magnetization of MXenes, peaking my intrigue in Ising model and wanting to learn more about it, in case future collaboration ever proved to be formidable. My computational experience with the language of choice, MATLAB, is moderate at best. I was a teaching assistant for a first-year engineering sequence for three years in MATLAB and EXCEL, both of which I have obtained working decency in. As a side note, I am not taking Numerical Methods and this is my first encounter with a MATLAB – like project since starting graduate school. I've produced closer to 100 pages of MATLAB code throughout my academic career.

Methodology/Script Development

Keeping the goal of determining the critical exponents in mind, it is desirable to first develop an algorithm for giving trends of magnetization, energy, and correlation length as a function of temperature for each lattice size of interest. Initially, a lattice size of a 20x20 was analyzed, which was a good selection based on the handout provided. In developing the code, it was decided to use several nested loops since multiple variables would be changed throughout the implementation of the MATLAB code. For the final script, the overall variable changing is lattice size, since all other loop variables are governed by it (besides temperature).

A temperature array spacing of 0.01 was used to correctly portray the desired trends. Unfortunately, for the desired lattice sizes of this study, the temperature and number of sweep variables assigned to smaller lattices proved to result in jobs that would end up taking weeks to converge. For every time the script looped to a new lattice size, the lattice was given a new initial state by randomizing the elements in the array to be either 1 or -1. Once the new lattice was constructed, a random element was selected and the spin was flipped by making the element the negative of itself. From there, the difference in energy between the initial and final state was calculated. The energy difference formula is shown below.

$$\Delta E = -2S_i\left(J\sum_{j\in N(i)} Sj\right)$$

One of the more elaborate parts of the script is the ability to enact periodic boundary conditions to examine the spin pair interaction with an existing neighbor or a neighbor across the boundary by specifying conditions within two for loops. Once the lattice was constructed and boundary changes were taken into consideration, a Metropolis Monte Carlo condition was established to either accept or reject a move based on how the neighboring pair interactions changed. Using the formula above, we can initially accept the move if this change in energy is positive, i.e., lower energy than the initial state. However, if not, we can sometimes accept the move by using a Boltzmann distribution, which is defined below.

$$\text{if } \Delta E \geq 0, \text{accept flip}$$

$$\text{if } \Delta E < 0, \qquad W = e^{\frac{\Delta E}{k_B T}}$$

$$\text{if } W > r, \quad 0 \leq r \leq 1, \qquad \text{accept flip}$$

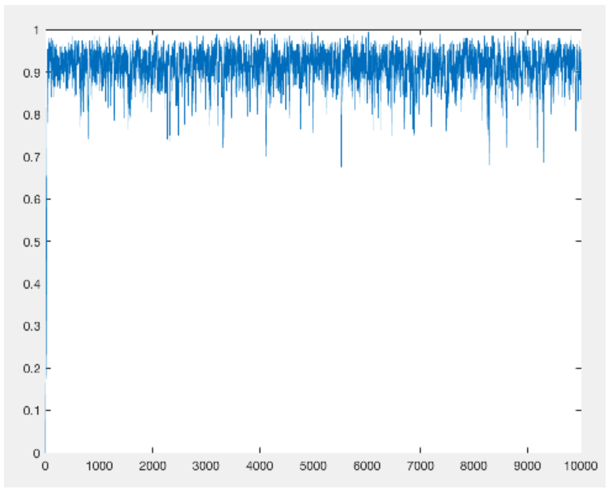
These changes either called for the spin matrix to allow the change, or to remain the same as it was prior to starting the loop. From this point, the average magnetization of the spins was recorded and collected in a matrix for further analysis, where average magnetization is a function of the number of sweeps iterated. Using 1000 sweeps, the very last sweep's information was recorded and placed into an array where the average magnetization, after running through an appropriate number of moves, is a function of the system temperature.

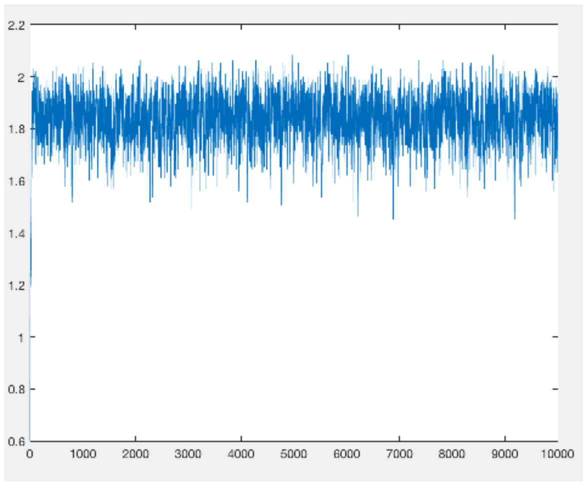
This process was repeated, iterating through temperature at 0.01 J/kB from 1 to 4 with 1000 sweeps. Once completed, the system size changed from 10 to 20. Afterwards, the script parameters were changed to iterating at 0.2 J/kB from 1 to 4 from 50 to 100 with 800 sweeps until the loop was completed and the data appropriately collected. Not only was average magnetization as a function of sweeps and temperature recorded, but so were the energy of the system specific heat capacity after 1000 iterations of making moves at random spins on the lattice. A more detailed discussion of the data, as well as a description on how the correlation functions were obtained is given in the results.

Results

Two important trends to initially notice are the average magnetization as a function of the number of sweeps and average magnetization as a function of temperature. These plots end up being very messy due to the randomness resulting from selecting a point, creating the matrix, and accepting/rejecting a move. However, these expressions were cleaned up nicely by using plotting software for presentation purposes. Raw data plots are included at the end of this paper in Appendix B.

Figures 1 and 2 below show these trends of average magnetization for a 20x20 spin lattice. As expected, the magnetization calibrates to its expected value after 10,000 sweeps through the Metropolis Monte Carlo iterations, showing that our code is correctly describing the long-term behavior of the system given a set number of moves, which relate to the time allowed under a specific temperature. At a temperature of 2 J/kB, the simulation converges to the expected averaged magnetization value of 0.9115, and an expected average energy value of 1.8374 J.





Figures 1 and 2: Average magnetization (left) and Average energy (right) of spins

It is of interest to determine the critical exponents of the Ising system. The objective is to determine these critical exponents by using data of thermodynamic properties near T=Tc, and fitting it to a model that is the inverse of the difference between a varying temperature and the critical temperature. We can do this by first developing appropriate methods of finding two thermodynamic properties of our choosing. It is crucial to note that if we can determine only two of the four exponents of interest, then we have a full description of them all because of the following expressions that are used to relate of the exponents.

$$\upsilon = \frac{2}{d + \frac{\alpha}{\upsilon}}; \qquad \gamma = \left(\frac{\gamma}{\upsilon}\right)\frac{2}{d + \frac{\alpha}{\upsilon}}; \qquad \beta = \frac{\left(d - \frac{\gamma}{\upsilon}\right)}{\left(d + \frac{\alpha}{\upsilon}\right)}$$

Since it is known how the magnetization changes with temperature, it is wise to obtain β. We can plot magnetization against the difference between Tc and T, and selectively determine a good fit by plotting against lattices of two different sizes. For the simulation, magnetizations from lattice sizes of 10x10 and 20x20 were plotted on the same axes and compared. Figure 3 below shows the trends obtained from both plots. The difference between critical temperature and temperature raised to the beta power was plotted, and a fit for the critical exponent was estimated by editing the critical temperature and the power until the two plots overlapped.

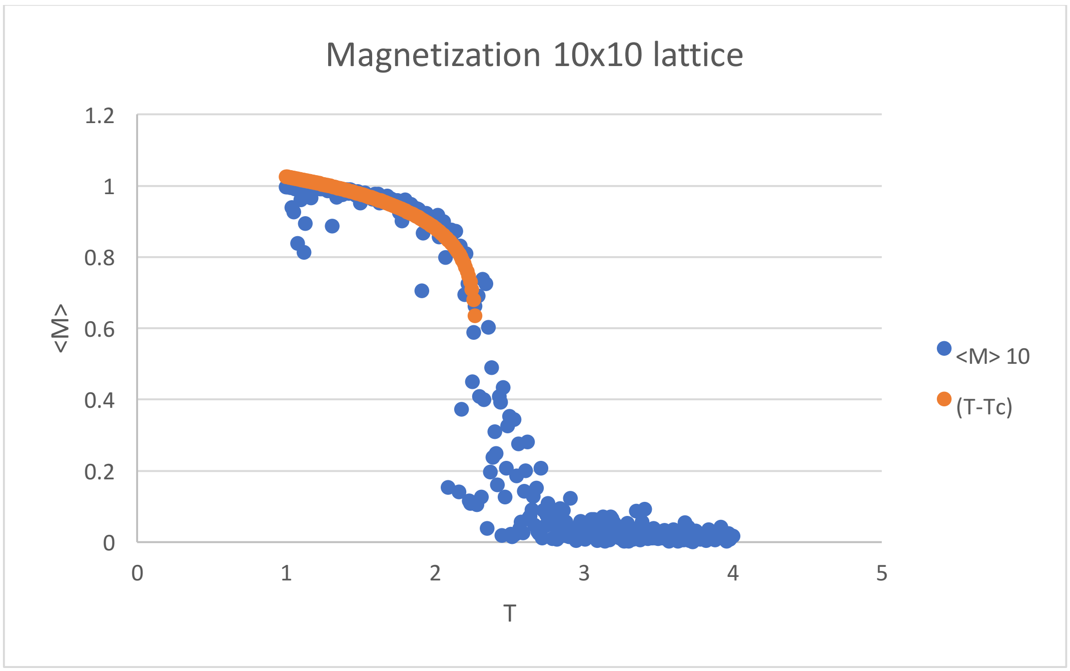


Figure 3: M(T) for a 10x10 lattice, with critical exponent fit

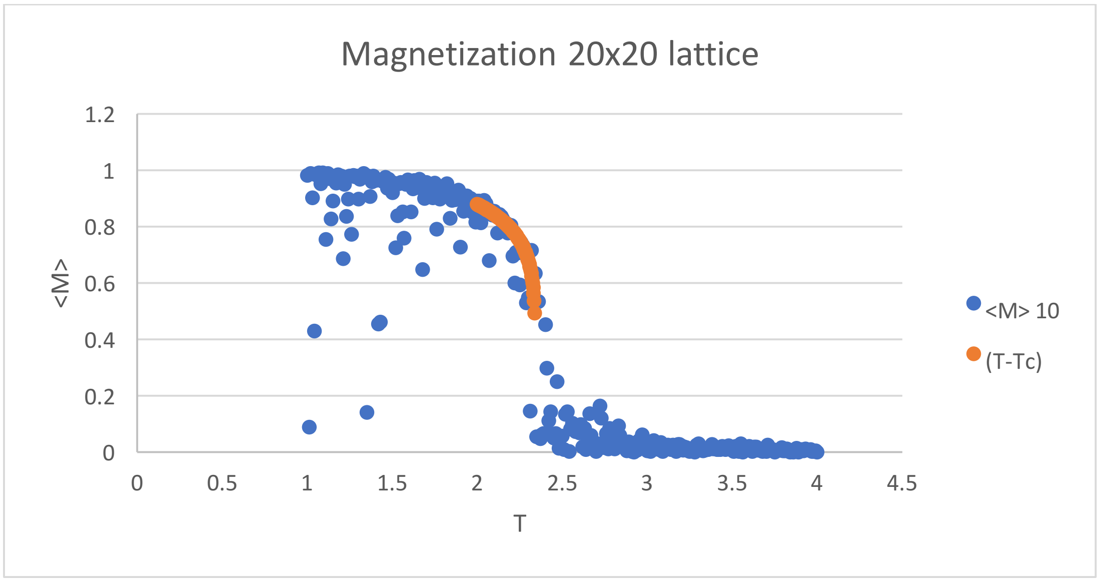


Figure 4: M(T) for a 20x20 lattice, with critical exponent fit

The appropriate critical exponent was then determined. For a 10x10 Ising lattice, the critical exponent was 0.1, which is near the Onsager solution of 0.125, but it crucial to note that an increase of this lattice size gave a critical exponent of 0.12. This shows that as we increase the lattice size, we can get better approximations of the critical parameters. Additionally, the critical temperatures obtained from these plots are an important factor in how the critical temperatures are determined. For a 10x10 lattice, the critical temperature was found to be 2.28 J/kB, and for a 20x20 lattice, was found to be 2.34. After looking at the 10 L and 20 L lattices, we can apply the same analysis to a 50 L and 100 L lattice. All critical parameters are provided in Table 1 in the Conclusions section.

We can then apply this analysis to the energy and its derivative as a function of temperature, the heat capacity. The heat capacity as a function of temperature was calculated by taking the difference between the energy points and their respective temperatures. Following the analysis, Figures 4 and 5 show the average energy and heat capacity for a 20x20 lattice, plotted on a fitted power law expression as a function of the difference between the critical temperature and varying temperature. For the heat capacity, it is appropriate to show that the log dependence of the temperature diverges at the critical temperature. This is demonstrated by plotting a mirror exponential temperature function over the specific heat capacity plots. There appears to be some divergence at the critical temperature of 2.3 J/kB, again, it is difficult to visualize due to the randomness of the system and the energies as a functions of temperature.

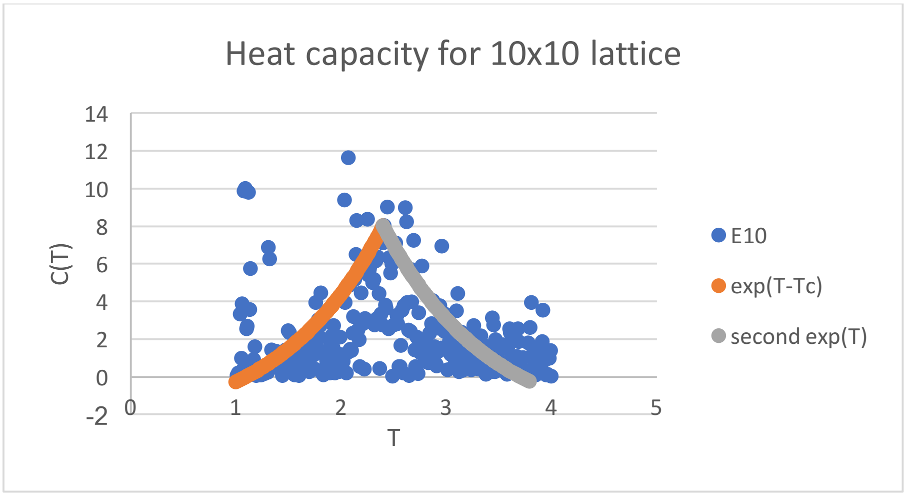


Figure 5: C(T) for a 20x20 lattice, with critical exponent fit

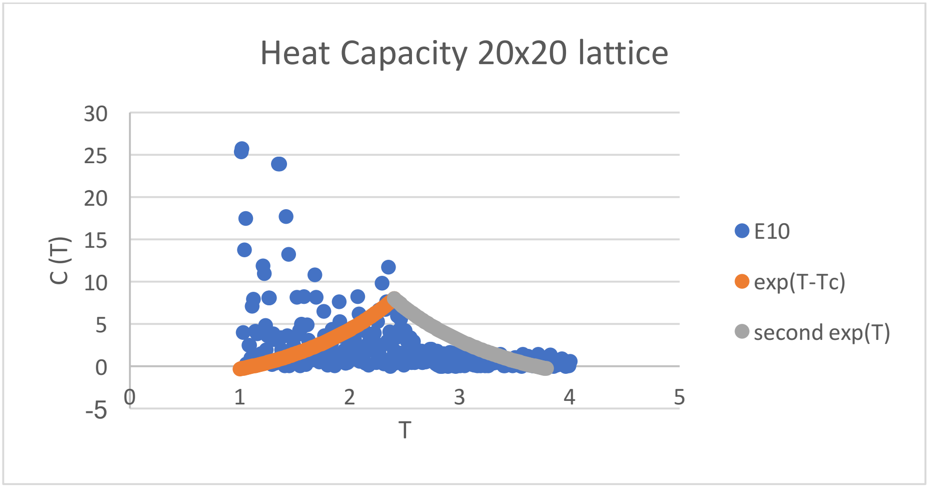


Figure 6: M(T) for a 20x20 lattice, with critical exponent fit

To give a better view of how the heat capacity is changing, A polynomial fit was obfrom the energy of systems as a function of time, which gave remarkably nice trends. The energy function obtained as a cubic polynomial of a 10x10 lattice, along with its derivative is:

$$E(T) = -0.0762T^4 + 0.9836T^3 - 4.3379T^2 + 7.0407T - 1.53; \quad R^2 = 0.99413$$

$$dE/dT = -0.3048T^3 + 2.9508T^2 - 8.6758T + 7.0407$$

Figures 7 and 8 show the energy plot and capacities as a function of temperature. Although the trend isn't what it desired, it is better at showing that there is a point in the fitted function where the derivative switches signs, which is an indication of the energy function diverging.

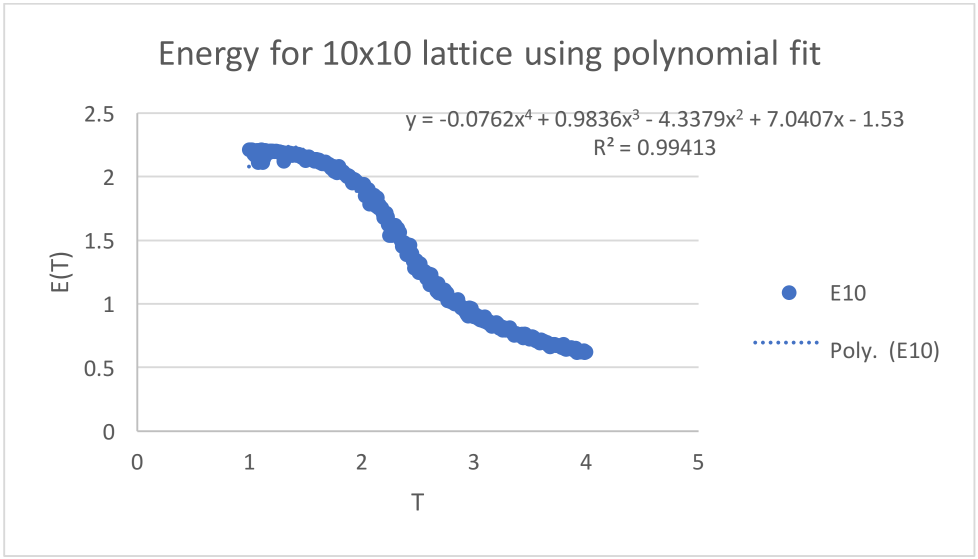


Figure 7: E(T) polynomial fit for a 10x10 lattice

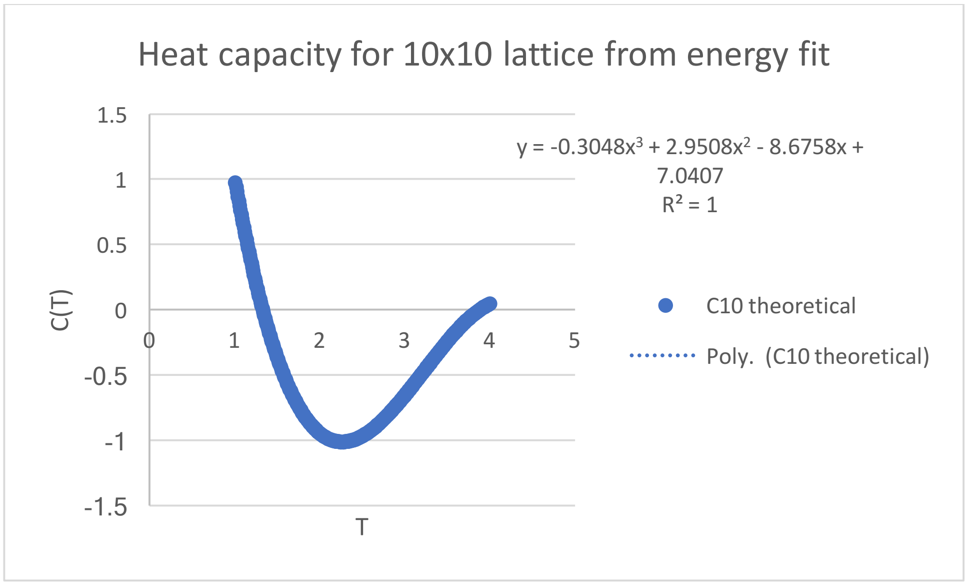


Figure 8: C(T) from the polynomial fit for a10x10 lattice

Finally, the correlation length was calculated as a function of temperature by first obtaining exponential fits of the correlation functions obtained after every temperature run. This portion of the project was by far the most challenging, as it required almost a week of thought, relative to finding the other 2 exponents. After coming to a Eureka moment (and realizing that the Excel autocorrelation function wasn't providing exactly what was needed), I coded a correlation function by simply sampling all the rows that changed as the result of a Monte Carlo step, and multiplying a fixed spin with a spin as a function of position and taking the ensemble mean. I was then able to collect the correlation functions as a function of position, and then fit the exponential trend to obtain the correlation length for each temperature by using MATLAB's FIT command.

The autocorrelation function is a function of the distance between spins on the row extracted and the correlation length, which is a constant value. Using a general form of A*e^(B*x), the value of B was computed, which is the negative reciprocal of the correlation length. Figure 9 below shows the plotted correlation length as a function of temperature for a 50x50 lattice. Finally, the length was fit near the critical temperature, where the third critical exponent was determined. This plot gave a clear Onsager critical exponent of 1 was from a 50x50 lattice running through 1000 sweeps per temperature. Figure 10 shows the power law fit on the correlation length as a function of temperature.

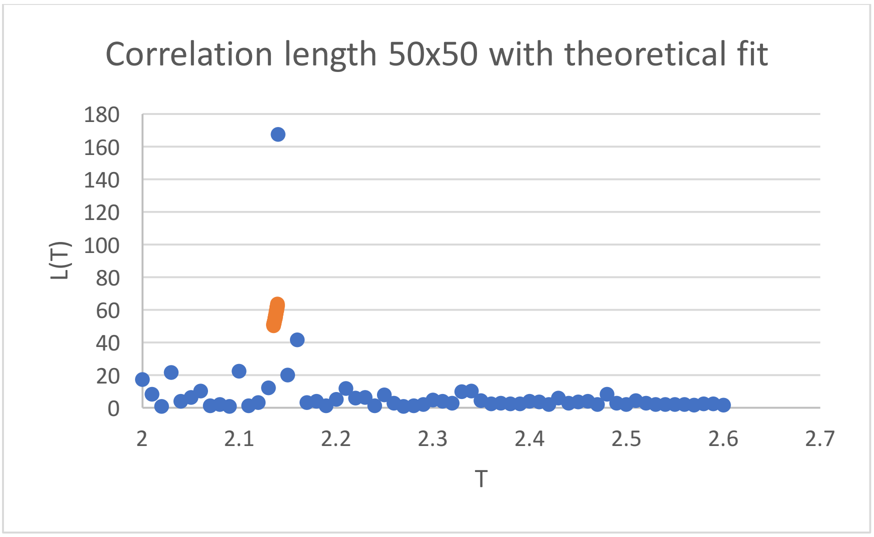


Figure 10: Correlation length with power law fit

Conclusion

The table below summaries the results obtained from the computations, the Onsager 2-D solutions, and the percent error obtained from estimating the beta coefficient. The results seem reasonable and within small margins of error, concluding that the Onsager solutions can be obtained from analyzing the Ising model, with enough computational time, an appropriate lattice size, and analytic tools.

Table 1 - Summarized Results from 1000 sweep Metropolis Monte Carlo Ising Model

L	System property	Value/Exponent	Value	Onsager Soln	% err	Determined
10	Critical Temperature	Tc	2.28	2.2	4%	Fit ((<M>))
20	Critical Temperature	Tc	2.34	2.2	6%	Fit ((<M>))
40	Critical Temperature	Tc	2.31	2.2	5%	Fit ((<M>))
50	Critical Temperature	Tc	2.27	2.2	3%	Fit ((<M>))
10	Magnetization	β	0.1	0.125	20%	Fit
20	Magnetization	β	0.12	0.125	4%	Fit
40	Magnetization	β	0.135	0.125	8%	Fit
50	Magnetization	β	0.127	0.125	2%	Fit
10	Heat Capacity	α	0	0	-	Fit
20	Heat Capacity	α	0	0	-	Fit
40	Heat Capacity	α	0	0	-	Fit
50	Heat Capacity	α	0	0	-	Fit
10	Correlation Length	υ	1	1	-	Fit
20	Correlation Length	υ	1	1	-	Fit
30	Correlation Length	υ	1	1	-	Fit
50	Correlation Length	υ	1	1	-	Fit
10	Magnetic susceptibility	γ	1.75	1.75	-	Relations
20	Magnetic susceptibility	γ	1.75	1.75	-	Relations
50	Magnetic susceptibility	γ	1.75	1.75	-	Relations

Some difficulties encountered include incorrectly implementing the Metropolis Monte Carlo, which was then resolved by simply reading through the packet provided. Another issue involved correctly comparing the thermodynamic properties as a function of time, as well as fitting the average system properties as a function of temperature.

Not realizing it until the critical exponents were of interest, one huge error involved double counting the spin pair interactions, which result in incorrect energy trends as a function of temperature. Fortunately, all the above troubles were dealt with quickly over the past 5-6 weeks. Although writing the code only took a few hours, there were weeks involved in debugging the script, and multiple scripts edited to finally obtain a select few that was ideal for collected data needed to find the critical exponents.

One point of importance is that using a lattice size large enough, with appropriate sweeps, is most appropriate for running simulations and getting back the appropriate Onsager solutions for critical temperature and the exponents, to avoid issues seen from the data presented. Had more time been available, better fitting analysis and a look at the Ising model in 3 dimensions would have been sought after.

Appendix A: MATLAB Script 1: Determine the magnetization, energy, and heat capacity as a function of the temperature

```
%CBE641 Computational Project Final code to find the mag, energy
%start code
clc; clear all; format compact
T=[1:0.01:2,2:0.005:2.4];
aa=1000;
tic
u=1;
for k=[10 20 40 50]
    x=k;
    y=k;
    s=zeros(length(T),aa*x^2);
    xv=zeros(length(T),aa*x^2);
for j=1:1:length(T) %randomize the matrix every time the temperature is changed

S=zeros(x,y); %randomly synthesize the spins of the structure
J=1;
h=1;

for m=1:1:x
    for n=1:1:y
bset=[-1,1];
pos=randi(length(bset));
S(m,n)=bset(pos);
end
end
%spins have been randomized
hm=zeros(x,y);

for i=1:1:aa*x^2 %Start the secondary loop; ends after a successful sweep
i;
    Sn=S;
v=[1:1:x];
s1=datasample(v,1); %randomly select a spin to change
s2=datasample(v,1);
Sn(s1,s2)=-1*Sn(s1,s2);
```

```
H=sum(sum(hm));

m=s1;

n=s2;

c=m-1;

d=m+1;

e=n-1;

f=n+1;

        if m-1==0

            c=x;

        end

        if m+1==x+1

            d=1;

        end

        if n-1==0

            e=y;

        end

        if n+1==y+1

            f=1;

        end

dE=-2*J*(S(m,n)*S(c,n)+S(m,n)*S(d,n)+S(m,n)*S(m,e)+S(m,n)*S(m,f));

if dE>=0

    W=1;

    S=Sn; %disp('Move was initially accepted')

elseif dE<0

    W=exp((dE)/T(j));

    r=rand;

    if r<W

        S=Sn;% disp('Move was accepted randomly from Boltzmann distribution')

    end

end


if i==u*x^2
ss=mean(mean(S));

E1=zeros(x,x-1);

E2=zeros(x-1,x);

for n=1:1:x-1

    if n==1

        E1(:,n)=(S(:,x)+S(:,2)).*S(:,1);

    elseif n>=2 & n<=x-1

    E1(:,n)=S(:,n).*S(:,n+1);
    end
end
for m=1:1:x-1
    if m==1
        E2(m,:)=(S(x,:)+S(2,:)).*S(1,:);
        elseif m>=2 & m<=x-1
    E2(m,:)=S(m, :).*S(m+1,:);
    end
end
EE=mean(mean(E1))+mean(mean(E2));
sss(u)=ss;
EEE(u)=EE;
u=u+1
end
end %end the loop to find new S
ssss(j)=mean(sss);
EEEE(j)=mean(EEE);
u=1

end %Complete temperature loop
if k==40; ssss50=ssss; EE50=EEEE;
elseif k==50; ssss100=ssss; EE100=EEEE;
end
end %complete lattice size loop
clc
disp('Congratulations, the Ising Ferromagnetic model has collected the data!!')
toc
%export data to Excel
xlswrite('magnetization2.xlsx', [transpose(ssss50), transpose(ssss100)], 'A1');
xlswrite('energy2.xlsx',[transpose(EE50), transpose(EE100)],'A1');
```

Appendix B: Script 2: Determine the autocorrelation function, and the respective correlation length as a function of temperature

```
%CBE641 Computational Project Final code

%start code

clc; clear all; format compact

T=[2:0.01:2.6];

aa=1000;

tic

u=1

for k=[10 20 30 50]

    x=k;

    y=k;

    s=zeros(length(T),aa*x^2);

    xv=zeros(length(T),aa*x^2);
 U=zeros(aa,x)
GRR=ones(1,x)
for j=1:1:length(T) %randomize the matrix every time the temperature is changed


S=zeros(x,y); %randomly synthesize the spins of the structure

J=1;

h=1;


for m=1:1:x

    for n=1:1:y

bset=[-1,1];

pos=randi(length(bset));

S(m,n)=bset(pos);

end

end

%spins have been randomized

hm=zeros(x,y);



for i=1:1:aa*x^2 %Start the secondary loop; ends after a successful sweep

Sn=S;

v=[1:1:x];

s1=datasample(v,1); %randomly select a spin to change

s2=datasample(v,1);

Sn(s1,s2)=-1*Sn(s1,s2);

H=sum(sum(hm));

m=s1;
```

```
n=s2;
        if m-1==0
            c=x;
        else c=m-1;
        end
        if m+1==x+1
            d=1;
        else d=m+1;
        end
        if n-1==0
            e=y;
        else e=n-1;
        end
        if n+1==y+1
            f=1;
        else f=n+1;
        end
dE=-2*J*(S(m,n)*S(c,n)+S(m,n)*S(d,n)+S(m,n)*S(m,e)+S(m,n)*S(m,f));
if dE>=0
    W=1;
    S=Sn;
    %disp('Move was initially accepted')
elseif dE<0
    W=exp((dE)/T(j));
    r=rand;
    if r<W
        S=Sn;
       % disp('Move was accepted randomly from Boltzmann distribution')
    else
        S=S;
       % disp('Move was not accepted')
    end
end
if i==u*x^2;
U(u,:)=S(x/2,:);
u=u+1
end
```

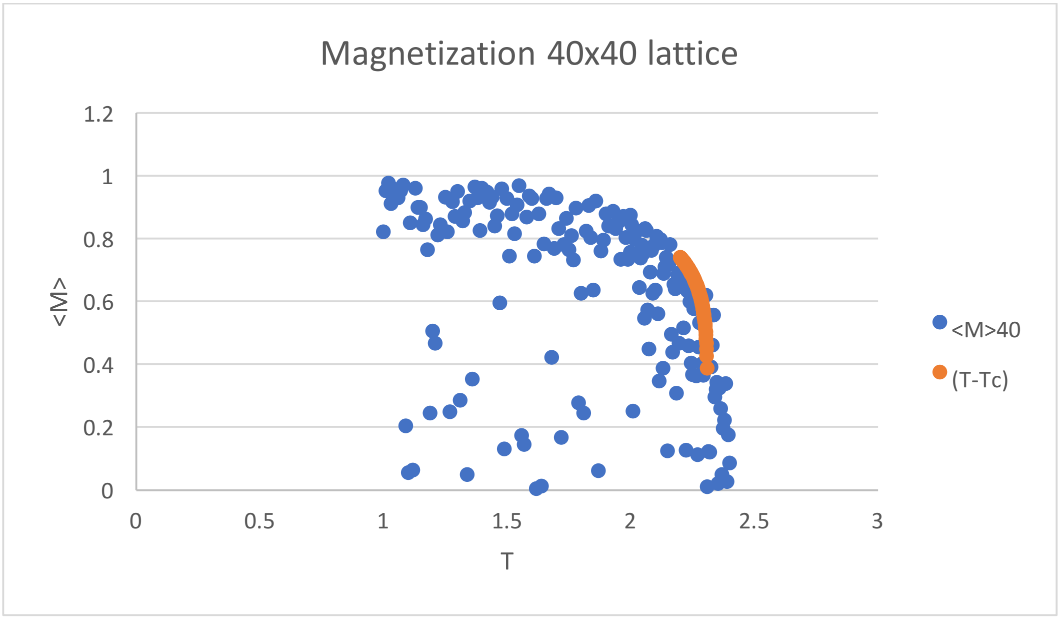
```
end %end the loop to find new S

for r=1:1:x

G(r)=mean(U(:,1).*U(:,r));

end




for r=1:1:x

GG(r)=G(r)-mean(mean([U(:,1),U(:,r)]))^2;

end



GRR(j,:)=GG;
u=1

end %Complete temperature loop
if k==30, GG30=GRR
else GG50=GRR
end


GRRR=transpose(abs(GRR(:,1:x/2)));

r=[1:1:x/2];

for u=1:1:length(T)

    fitt=fit(transpose(r),(GRRR(:,u)),'exp1');

    cory=coeffvalues(fitt);

    corlen(u)=1/-cory(2);

end

if k==10
    cormel1=corlen;
elseif k==20
    cormel2=corlen;
elseif k==30
    cormel3=corlen;
elseif k==50
    cormel4=corlen;
end



end%complete lattice size loop

clc

disp('Congratulations, the Ising Ferromagnetic model has collected the data!!')

toc

%export data to Excel
xlswrite('corlength.xlsx', [transpose(cormel1), transpose(cormel2), transpose(cormel3),
transpose(cormel4)], 'A1');
```

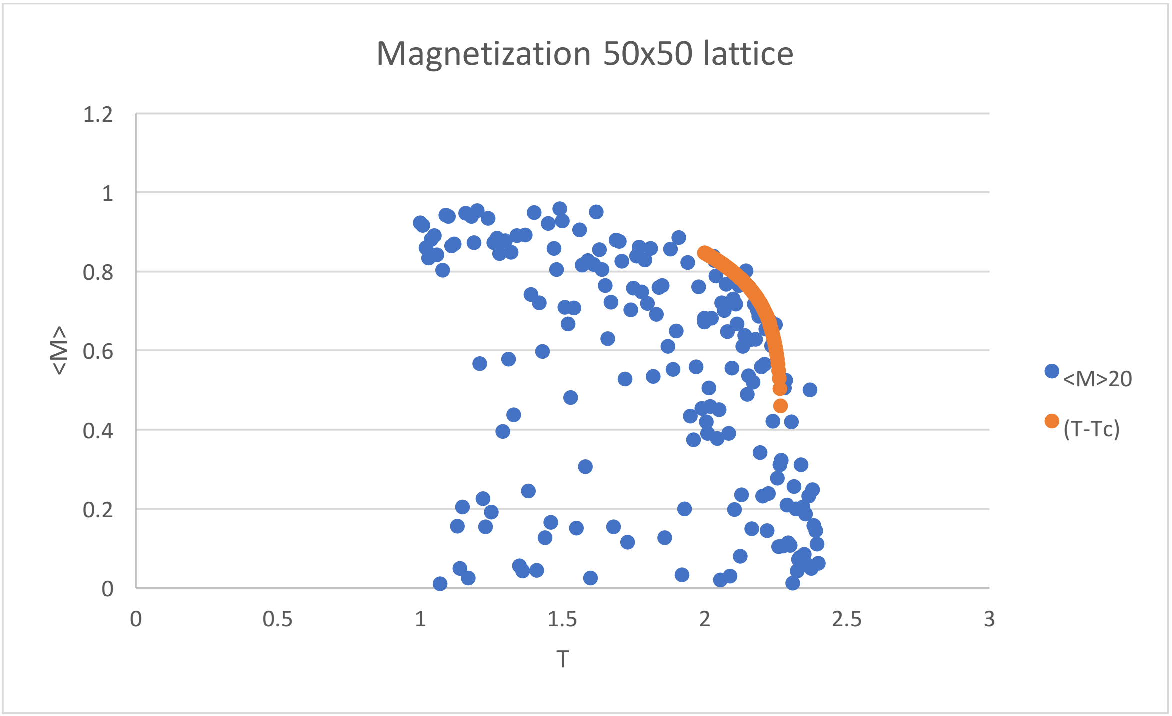
Appendix C: Script 3: Simple script to determine the magnetization and energy as a function of the number of sweeps

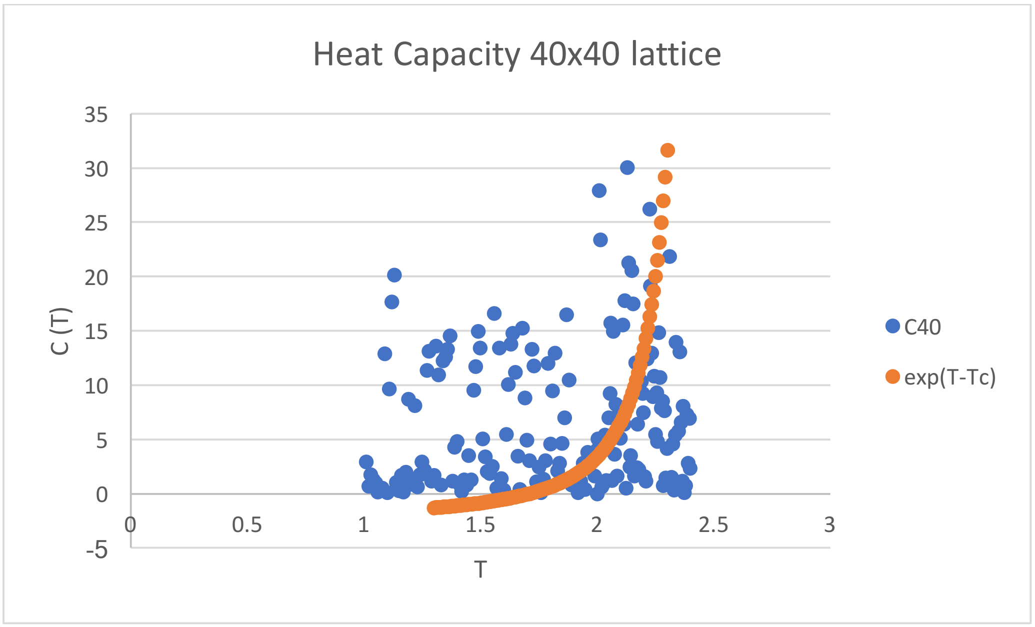
```
%CBE641 Computational Project Final code
%start code
clc; clear all; format compact
T=2
aa=10000;
tic

k=20
    x=k;
    y=k;

S=zeros(x,y); %randomly synthesize the spins of the structure
J=1;
h=1;

for m=1:1:x
    for n=1:1:y
bset=[-1,1];
pos=randi(length(bset));
S(m,n)=bset(pos);
end
end
%spins have been randomized
hm=zeros(x,y);

for i=1:1:aa*x^2 %Start the secondary loop; ends after a successful sweep
i;
    Sn=S;
v=[1:1:x];
s1=datasample(v,1); %randomly select a spin to change
s2=datasample(v,1);
Sn(s1,s2)=-1*Sn(s1,s2);
H=sum(sum(hm));
m=s1;
n=s2;
c=m-1;
d=m+1;
e=n-1;
f=n+1;
        if m-1==0
            c=x;
        end
        if m+1==x+1
            d=1;
        end
        if n-1==0
            e=y;
        end
        if n+1==y+1
            f=1;
        end
dE=-2*J*(S(m,n)*S(c,n)+S(m,n)*S(d,n)+S(m,n)*S(m,e)+S(m,n)*S(m,f));
if dE>=0
    W=1;
    S=Sn; %disp('Move was initially accepted')
elseif dE<0
    W=exp((dE)/T);
    r=rand;
    if r<W
        S=Sn;% disp('Move was accepted randomly from Boltzmann distribution')
    end
end

ss=mean(mean(S));
E1=zeros(x,x-1);
E2=zeros(x-1,x);
for n=1:1:x-1
    if n==1
        E1(:,n)=(S(:,x)+S(:,2)).*S(:,1);
    elseif n>=2 & n<=x-1
    E1(:,n)=S(:,n).*S(:,n+1);
    end
end
```

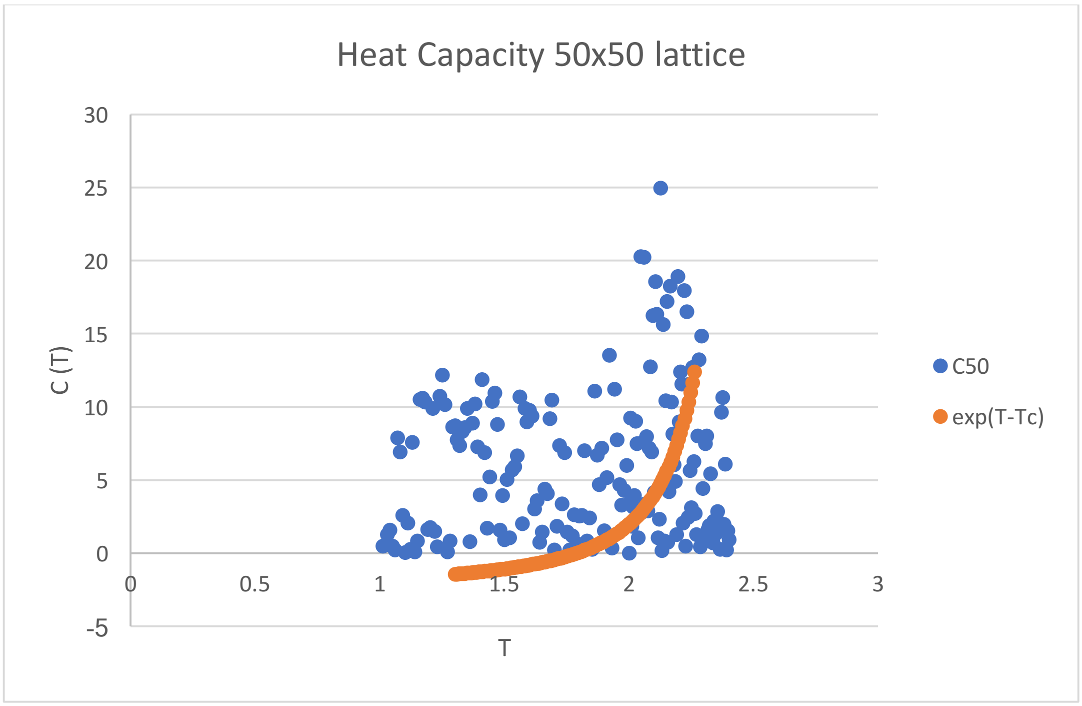
```
for m=1:1:x-1
    if m==1
        E2(m,:)=(S(x,:)+S(2,:)).*S(1,:);
        elseif m>=2 & m<=x-1
    E2(m,:)=S(m, :).*S(m+1,:);
    end
end
EE=mean(mean(E1))+mean(mean(E2));
sss(i)=ss;
eee(i)=ee;

end %end the loop to find new S

for k=1:1:aa
  ssss(k)=sss(k*x^2);
  eeee(k)=eee(k*x^2);
end


clc
disp('Congratulations, the Ising Ferromagnetic model has collected the data!!')
toc
k=[1:1:aa]
plot(k,ssss)
figure 2
plot(k,eeee)
```

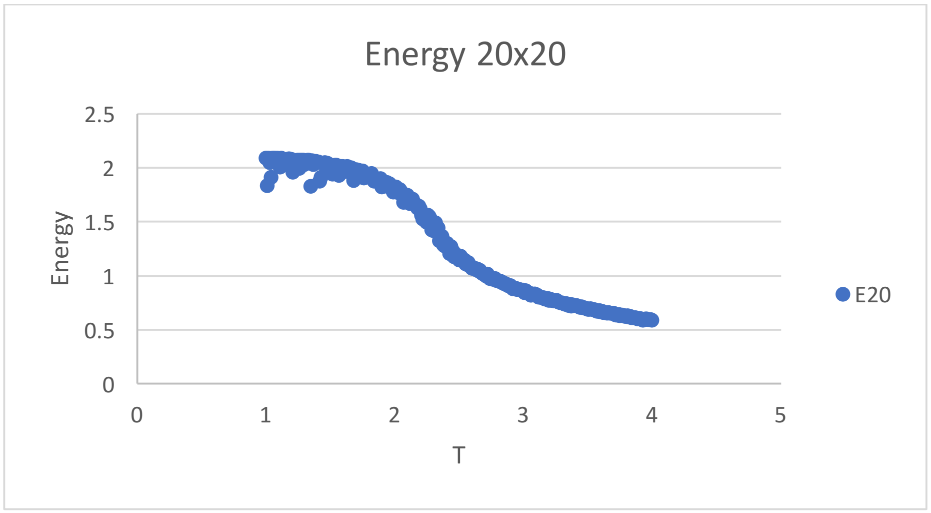
Appendix D: Figures: Additional figures not included in the report

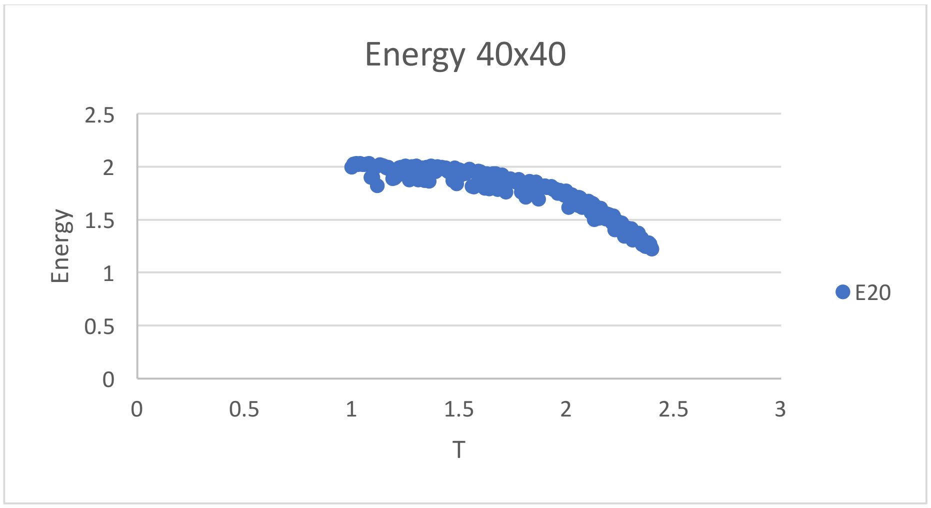


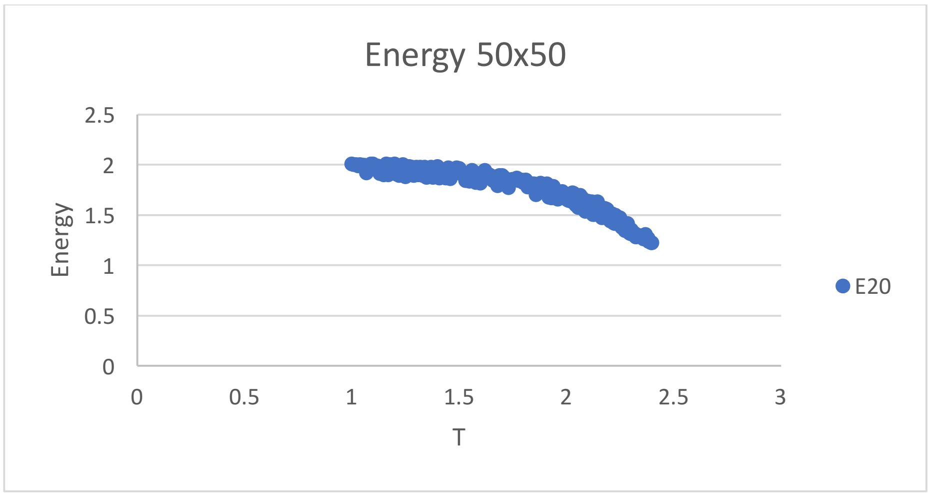


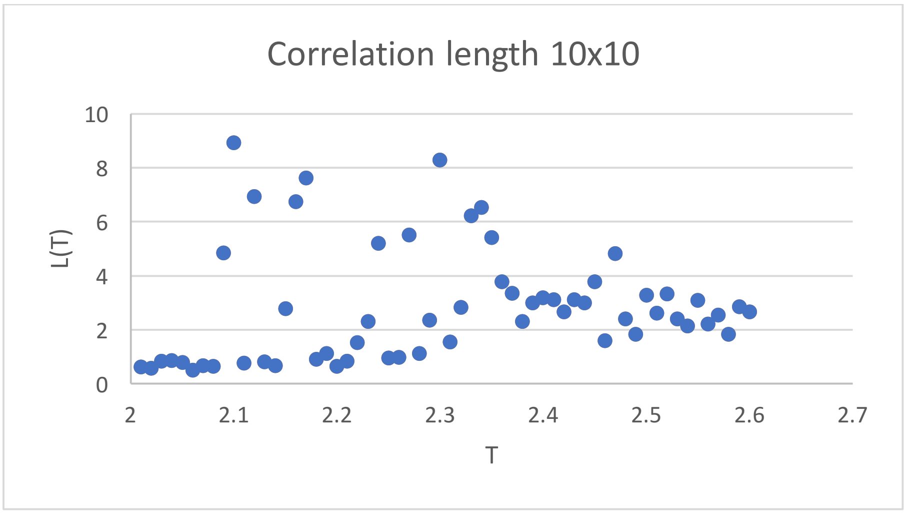

Heat Capacity 40x40 lattice
35
30
25
20
15
10
5
0
-5
0
0.5
1
1.5
2
2.5
3
C (T)
T
C40
exp(T-Tc)

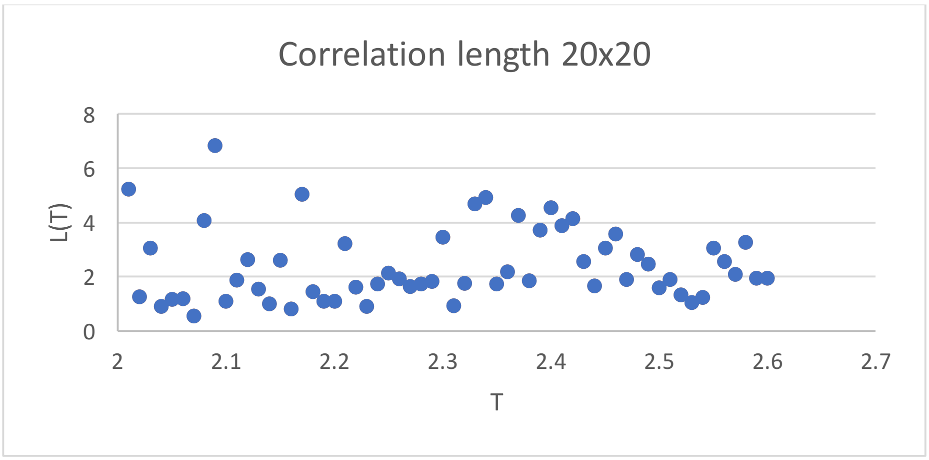

Heat Capacity 50x50 lattice
30
25
20
15
10
5
0
-5
0
0.5
1
1.5
2
2.5
3
C (T)
T
C50
exp(T-Tc)

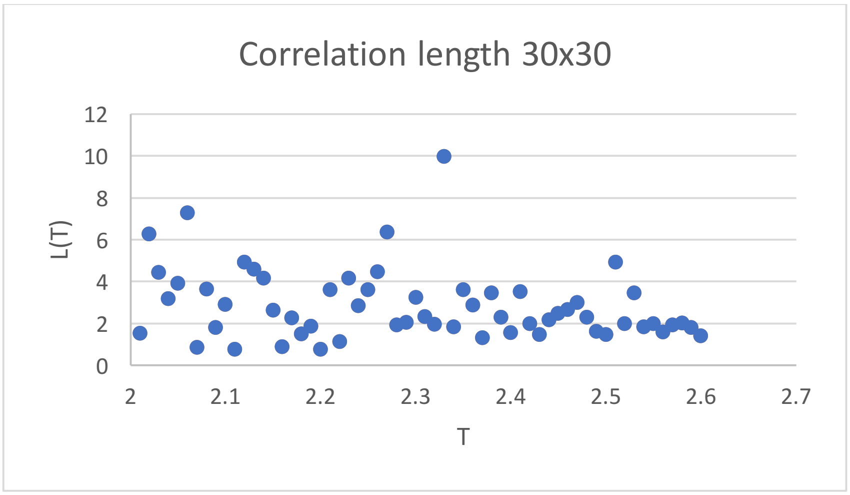

Energy 20x20
Energy
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
T
● E20


Energy 40x40
Energy
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
T
● E20


Energy 50x50
Energy
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
T
● E20


Correlation length 10x10
L(T)
T
0
2
4
6
8
10
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7


Correlation length 20x20
L(T)
T
0
2
4
6
8
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7


Correlation length 30x30
L(T)
T
0
2
4
6
8
10
12
2
2.1
2.2
2.3
2.4
2.5
2.6
2.7