

Simulación del modelo de Ising en 2D

Práctica computacional - Física Teórica 3 (1^{er} cuat. 2018) - Clase P. Minnini

Ignacio A. Poggi - ignaciop.3@gmail.com

2 de julio de 2018

1. Notas

Se tomaron las siguientes constantes a lo largo del trabajo:

- $k_B = 1$
- $J = 1$
- $B = 0$

En el Apéndice se transcribe el código fuente utilizado en lenguaje Matlab.

2. Ejercicio: Tiempo de equilibración

Para obtener el numero de iteraciones necesarias para que la energía y la magnetización alcancen un equilibrio, se eligieron tres tamaños de la red posibles: $L = 16$, $L = 32$ y $L = 64$. A su vez, para cada L se analizaron los siguientes valores de temperatura: $T = 1.8$ ($T < T_c$), $T = 2.27$ ($T = T_c$) y $T = 3$ ($T > T_c$).

Se fijo una cota superior razonable de $n = 10000$ iteraciones totales; de esta manera se puede cubrir una rango lo suficiente grande para permitir que cada sistema termalice y luego alcance el equilibrio; sin esperar tiempos muy largos de ejecución del código.

2.1. Resultados

Se encontró que para $T = 1.8$, el numero de iteraciones para que la energía y la magnetización de cada sistema llegue al equilibrio varía con el tamaño de la red; es decir a mayor L , más iteraciones se necesitan para que el sistema alcance el equilibrio en los parámetros observados.

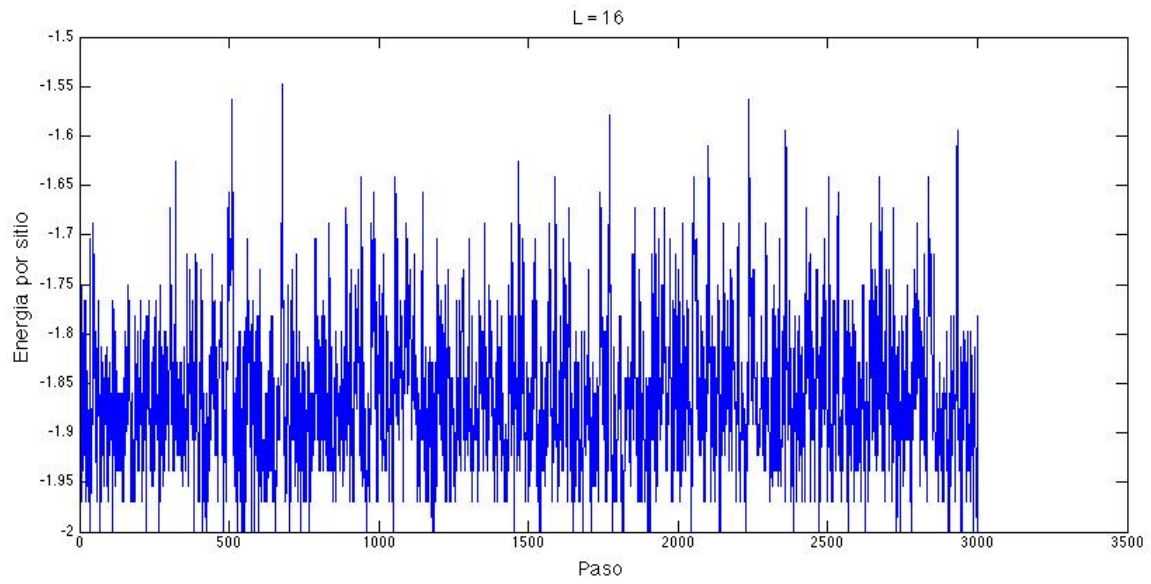


Figura 1: Convergencia de la energía por sitio para $L = 16$ y $T = 1.8$.

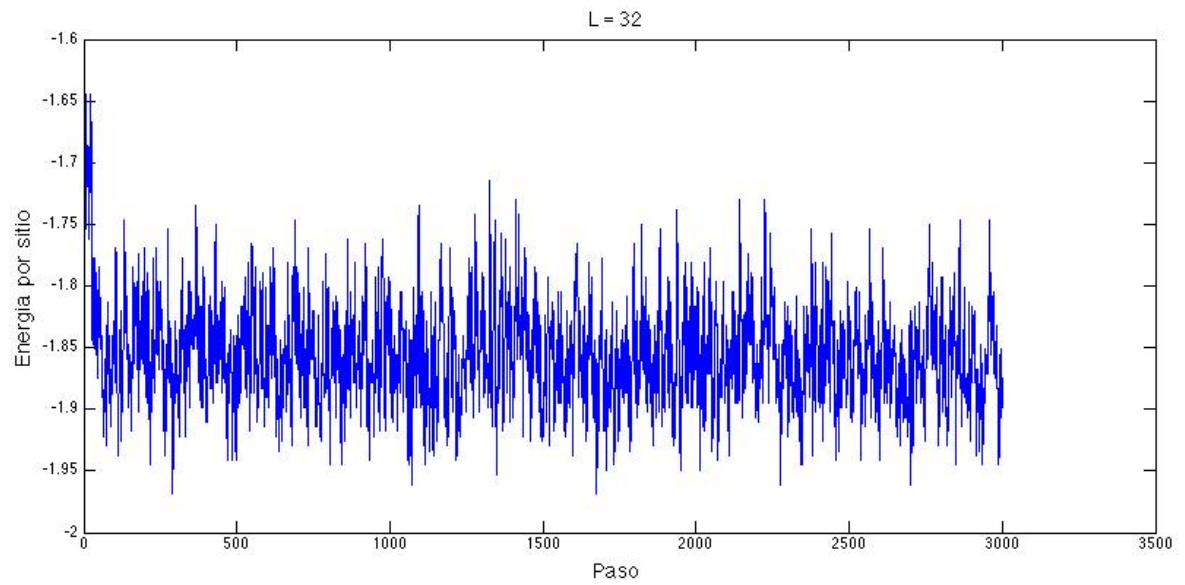


Figura 2: Convergencia de la energía por sitio para $L = 32$ y $T = 1.8$.

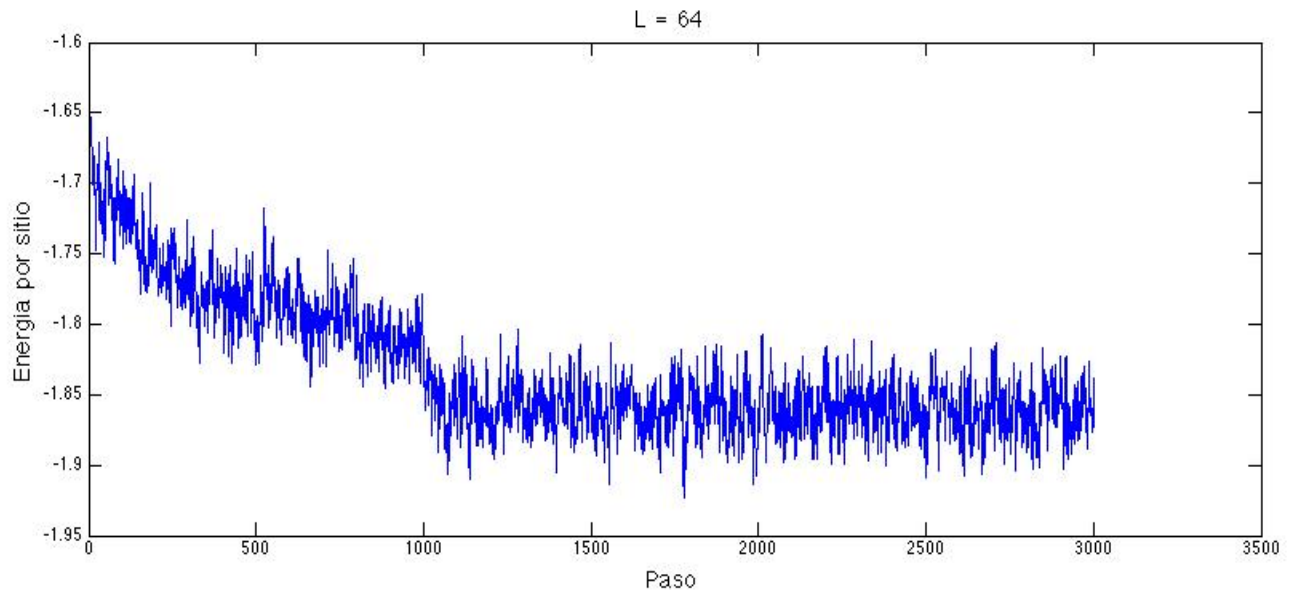


Figura 3: Convergencia de la energía por sitio para $L = 64$ y $T = 1.8$.

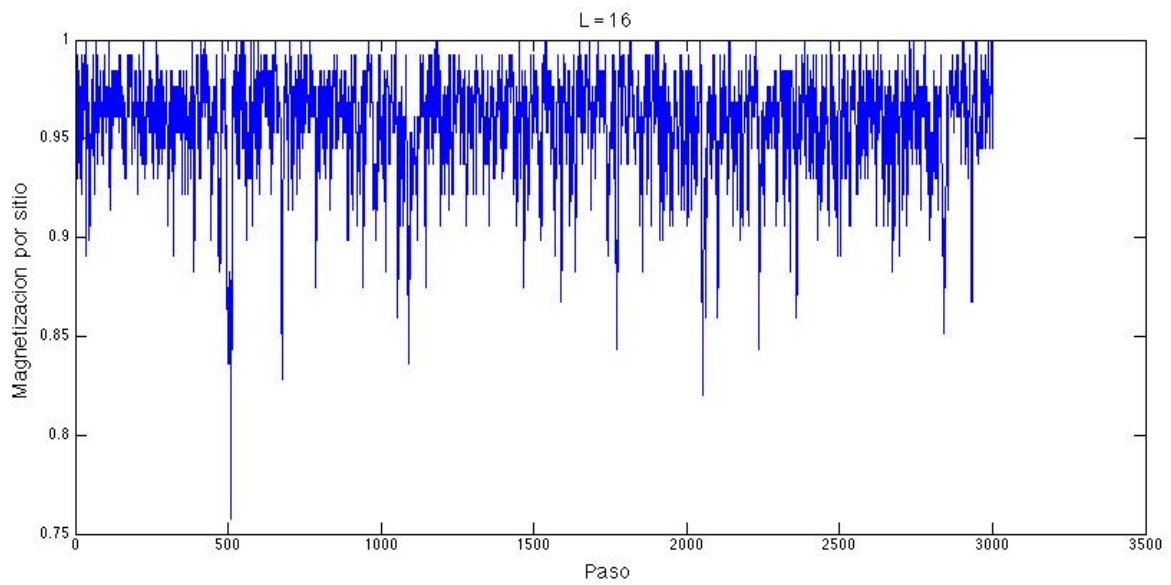


Figura 4: Convergencia de la magnetización por sitio para $L = 16$ y $T = 1.8$.

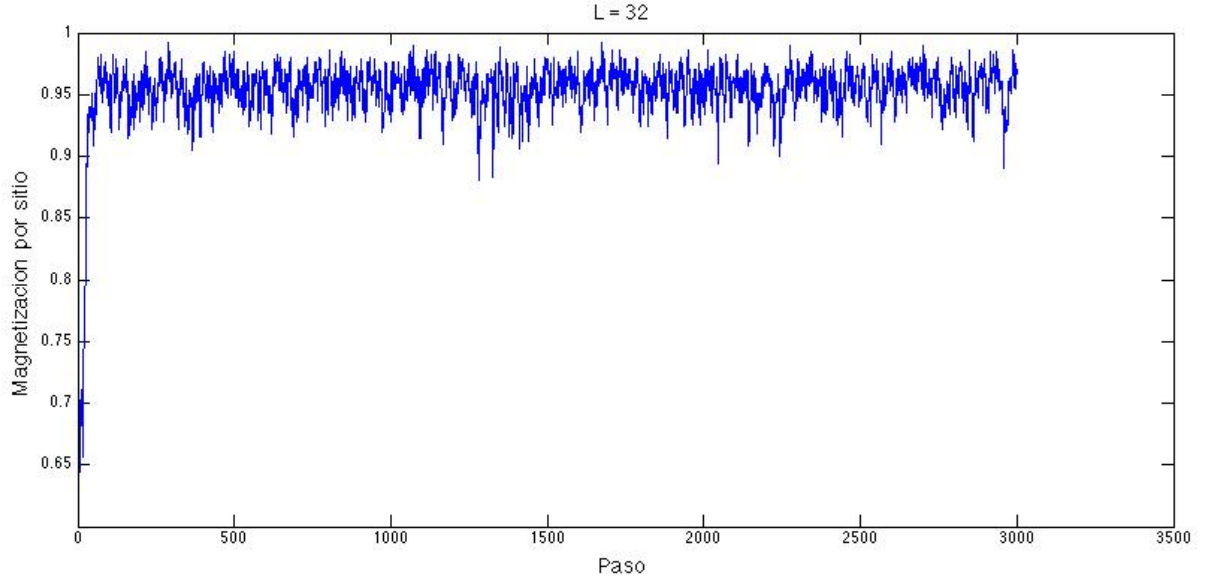


Figura 5: Convergencia de la magnetización por sitio para $L = 32$ y $T = 1.8$.

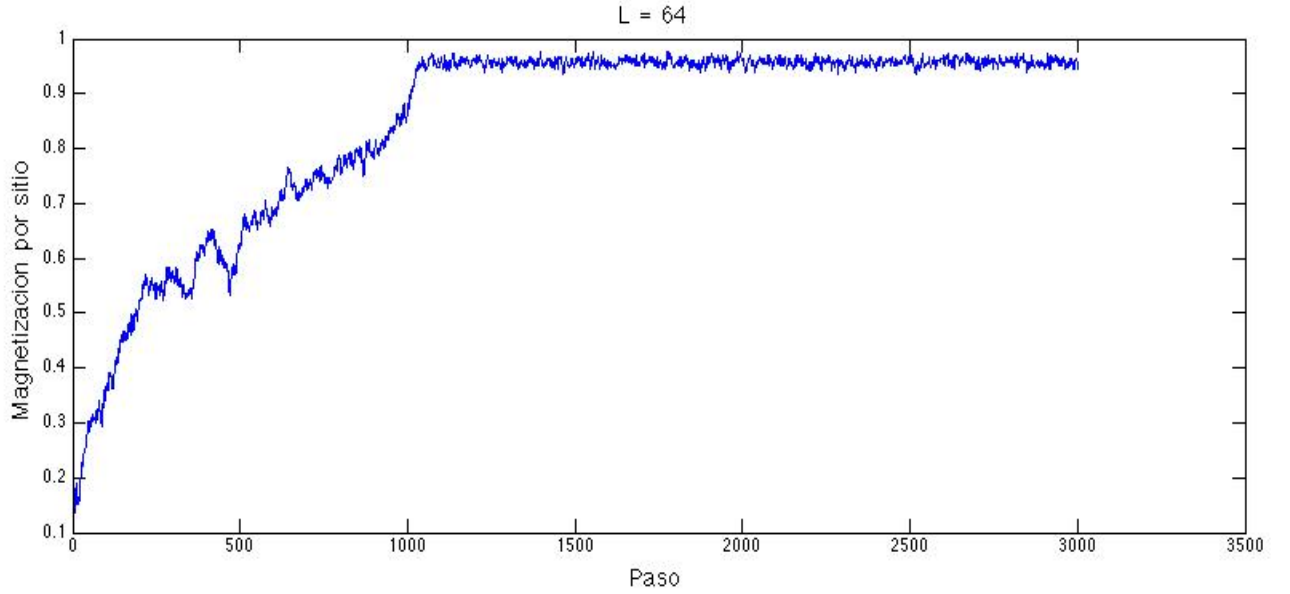


Figura 6: Convergencia de la magnetización por sitio para $L = 64$ y $T = 1.8$.

En las siguientes tablas, se muestran los resultados obtenidos de n para la energía y la magnetización por sitio en cada caso:

Cabe aclarar que, si bien en algunos casos (por ejemplo para $L = 64$), el n observado puede ser menor al informado en la tabla, se escogió un enfoque más conservativo; esto significa que no se tabuló un n justo en la zona de termalización-equilibrio; sino que se eligió un valor posterior.

$L = 16$	$L = 32$	$L = 64$
$n \approx 250$	$n \approx 500$	$n \approx 1250$

Cuadro 1: Cantidad de iteraciones n para alcanzar el equilibrio en e y m a $T = 2.27 = T_c$, para cada valor de L .

En el caso $T = 2.27 = T_c$, se pudo observar que la energía oscila en torno a un mismo valor medio ($e \approx -1,4$) casi inmediatamente al inicio de la ejecución del código; además de que dicho valor es más estable a medida que L es mayor, como se observa en las siguientes figuras:

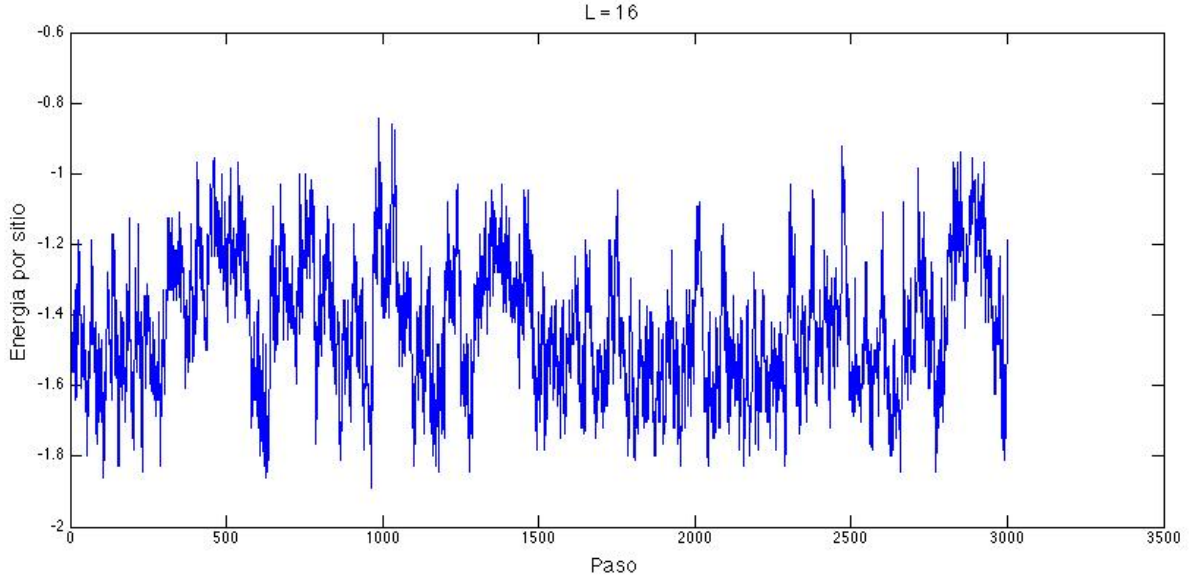


Figura 7: Convergencia de la energía por sitio para $L = 16$ y $T = 2.27$.

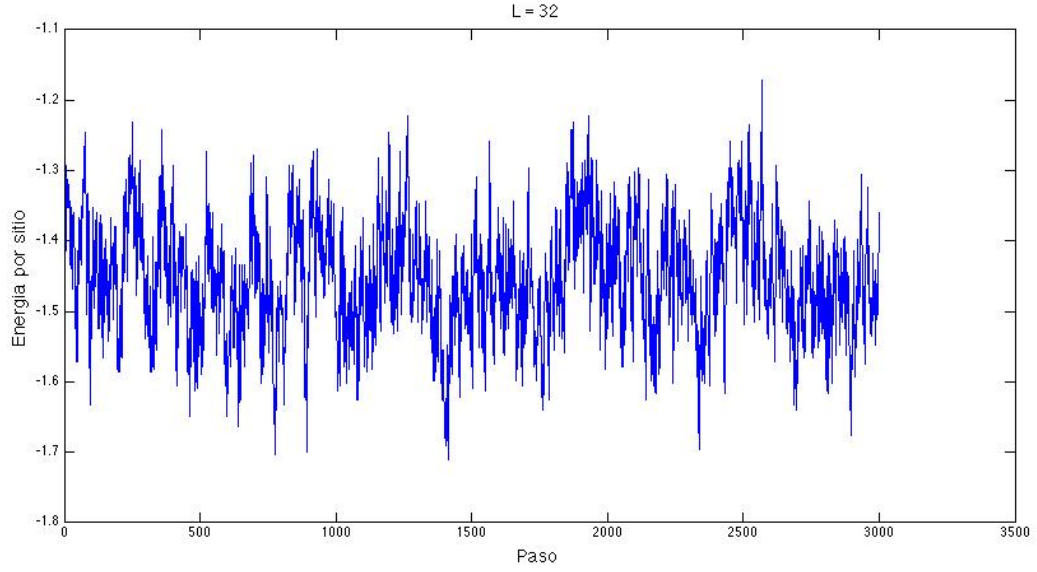


Figura 8: Convergencia de la energía por sitio para $L = 32$ y $T = 2.27$.

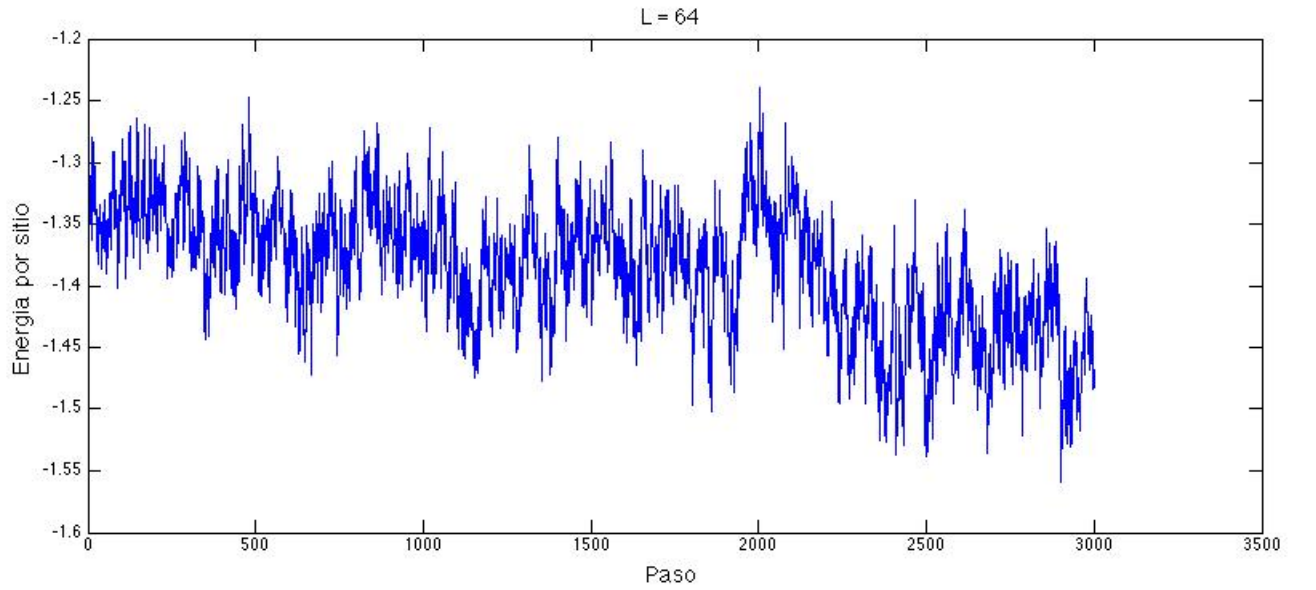


Figura 9: Convergencia de la energía por sitio para $L = 64$ y $T = 2.27$.

Como es de esperar en este caso, el valor absoluto de la magnetización oscila entre 0 y -1 de manera muy brusca para cada valor de L estudiado, por lo tanto no se puede decir que dicho observable converja para algún valor de n ; pero sí se puede asegurar que este comportamiento oscilatorio se mantiene durante toda la ejecución del programa con el valor de T considerado en este caso; lo que es consistente con la teoría.

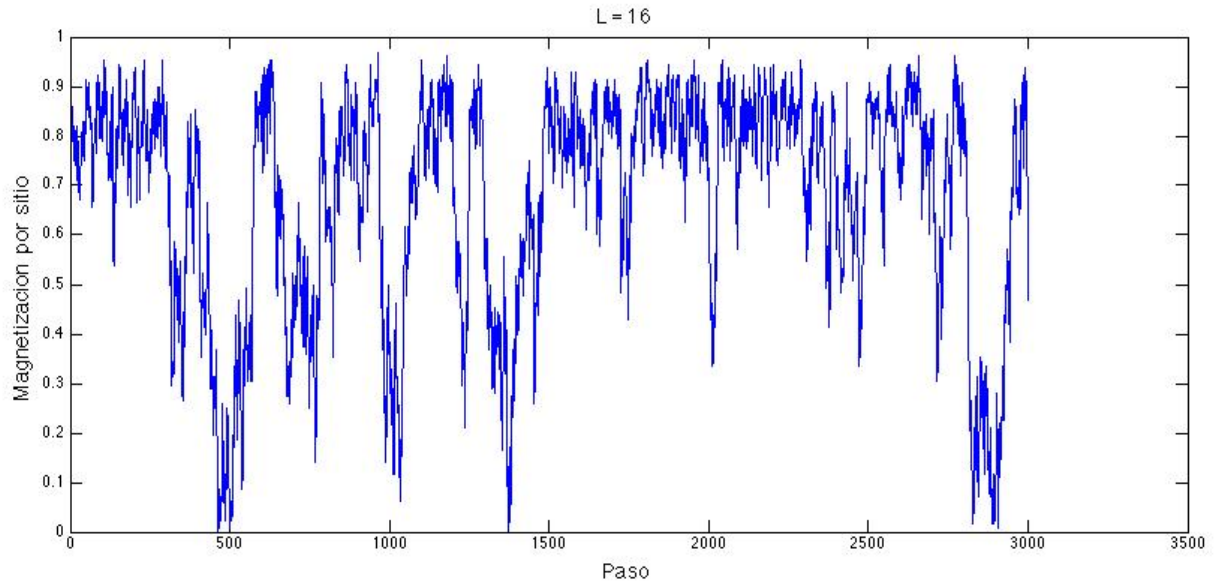


Figura 10: Convergencia de la magnetización por sitio para $L = 16$ y $T = 2.27$.

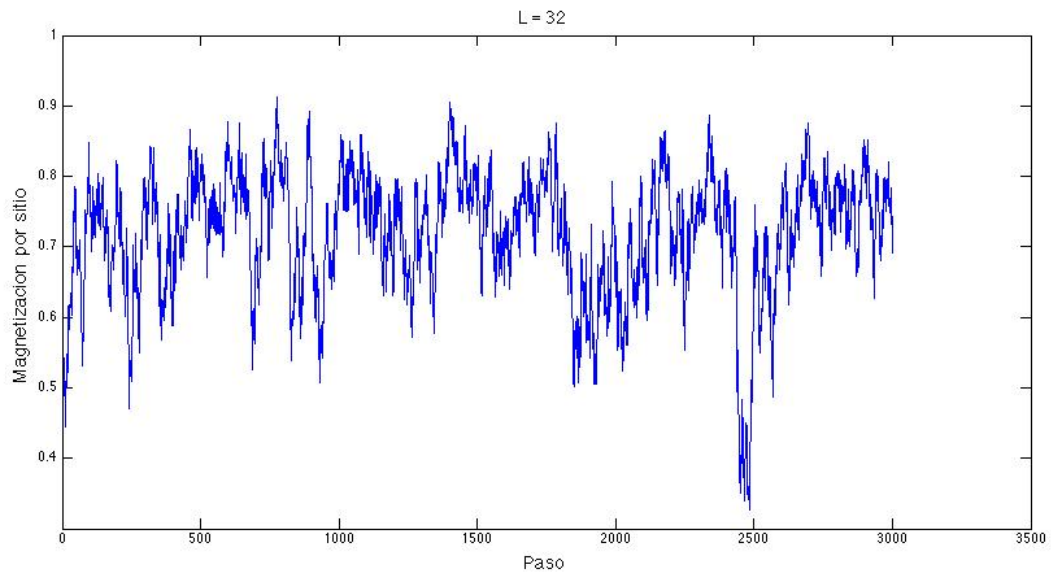


Figura 11: Convergencia de la magnetización por sitio para $L = 32$ y $T = 2.27$.

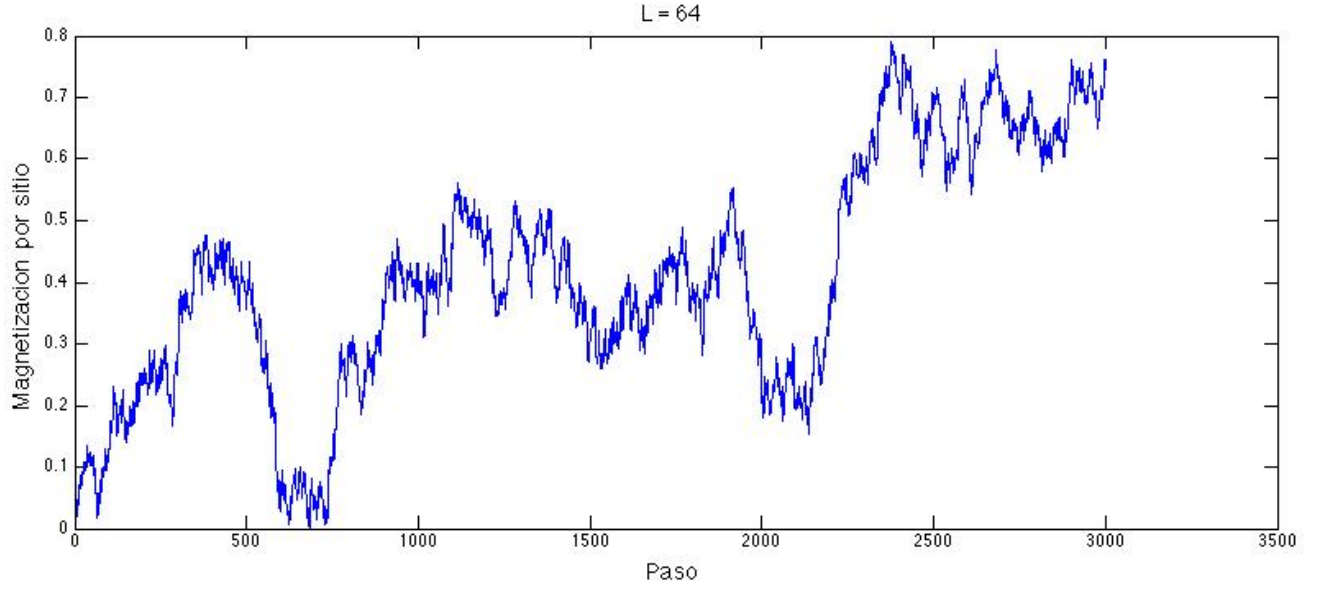


Figura 12: Convergencia de la magnetización por sitio para $L = 64$ y $T = 2.27$.

Por último, para $T = 3 > T_c$, tanto la energía como la magnetización de cada L converge aproximadamente a -0.8 y 0 respectivamente para un n muy pequeño: se podría tomar uno casi inmediato al comienzo de la ejecución del programa. En este caso, los gráficos para e y m son similares para los tres valores de L . Como ejemplo, se muestran los mismos para $L = 64$:

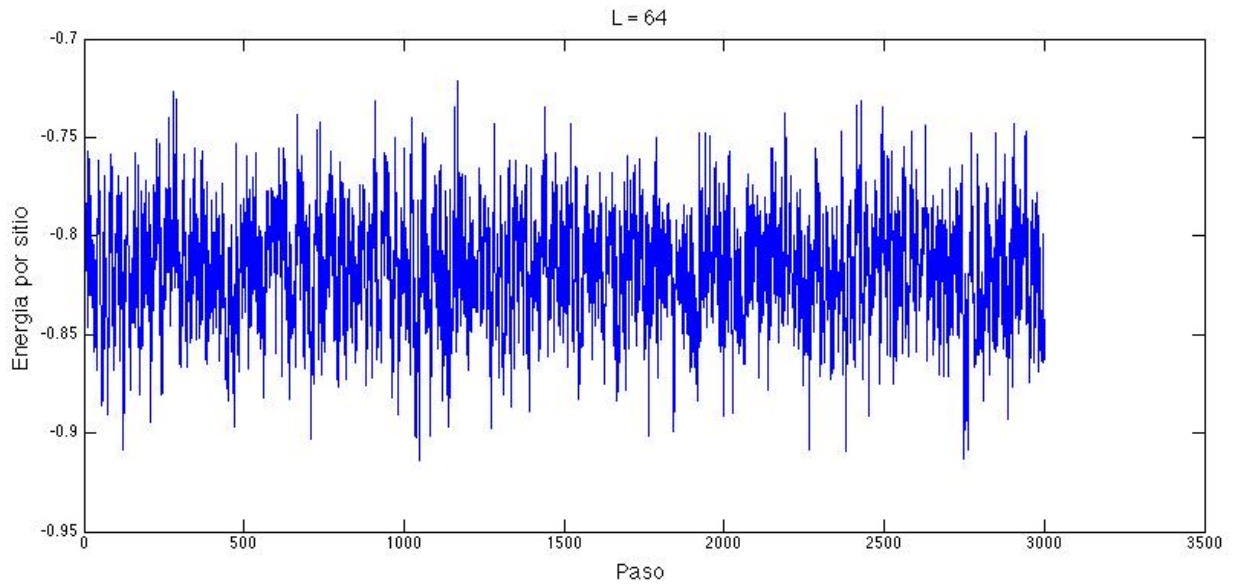


Figura 13: Convergencia de la energía por sitio para $L = 64$ y $T = 3$.

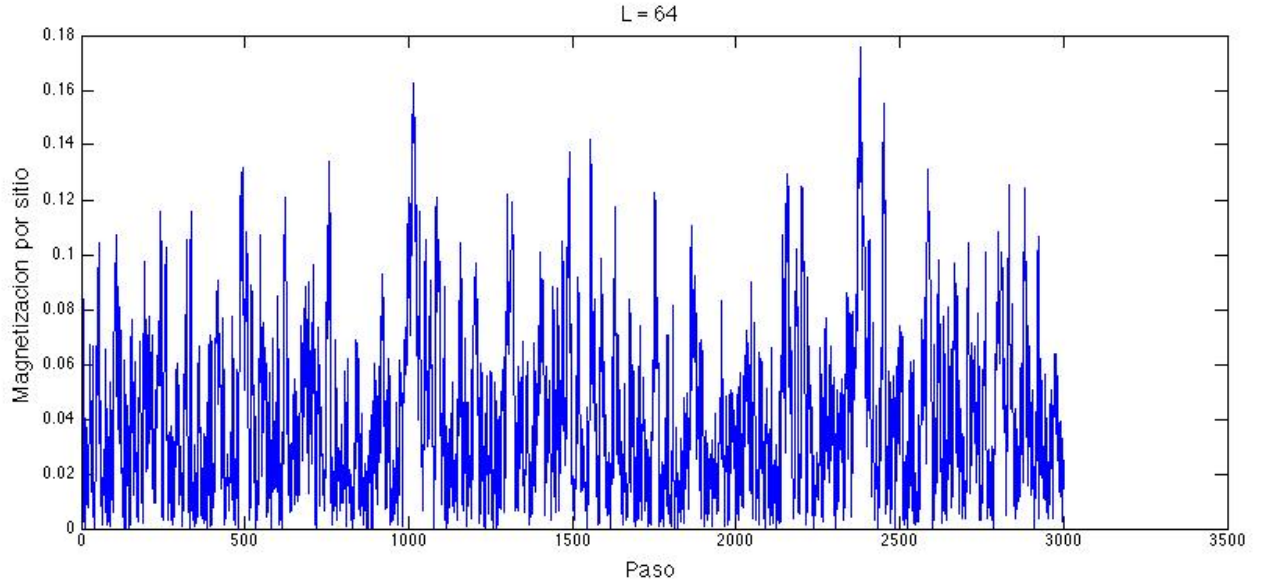


Figura 14: Convergencia de la magnetización por sitio para $L = 64$ y $T = 3$.

2.2. Conclusión

Para este ejercicio, el régimen mas interesante a estudiar fue el caso $T < T_c$, donde se obtuvo que a mayor L , son necesarias más cantidad de iteraciones para que el sistema alcance el equilibrio en los observables medidos. Para $T > T_c$, n fue despreciable para llegar a dichos equilibrios; en el caso $T = T_c$, no tiene sentido hablar de convergencia en la magnetización por la naturaleza oscilatoria del mismo en este régimen; pero para la energía el comportamiento es similar al caso $T > T_c$, por lo tanto se puede concluir que, en líneas generales, la convergencia de los valores medidos de los observables depende de la temperatura, pero a temperaturas bajas es necesario considerar también el tamaño de la red.

3. Ejercicio: Energía y magnetización en función de la temperatura

En este ejercicio se tuvieron presentes los resultados del anterior en lo referido a la cantidad de iteraciones necesarias para que el sistema llegue al equilibrio. Por esto, para calcular la energía y magnetización media ($\langle e \rangle$ y $\langle m \rangle$ respectivamente), se eligió un $n = 3000$ para valores de T entre 0 y 10; con pasos de 0.2; debido a que es un numero razonable de iteraciones para que el sistema termalice, llegue al equilibrio y un tiempo extra poder tomar valores medios significativos.

3.1. Resultados

A continuación, se muestran los resultados adquiridos para $L = 16$:

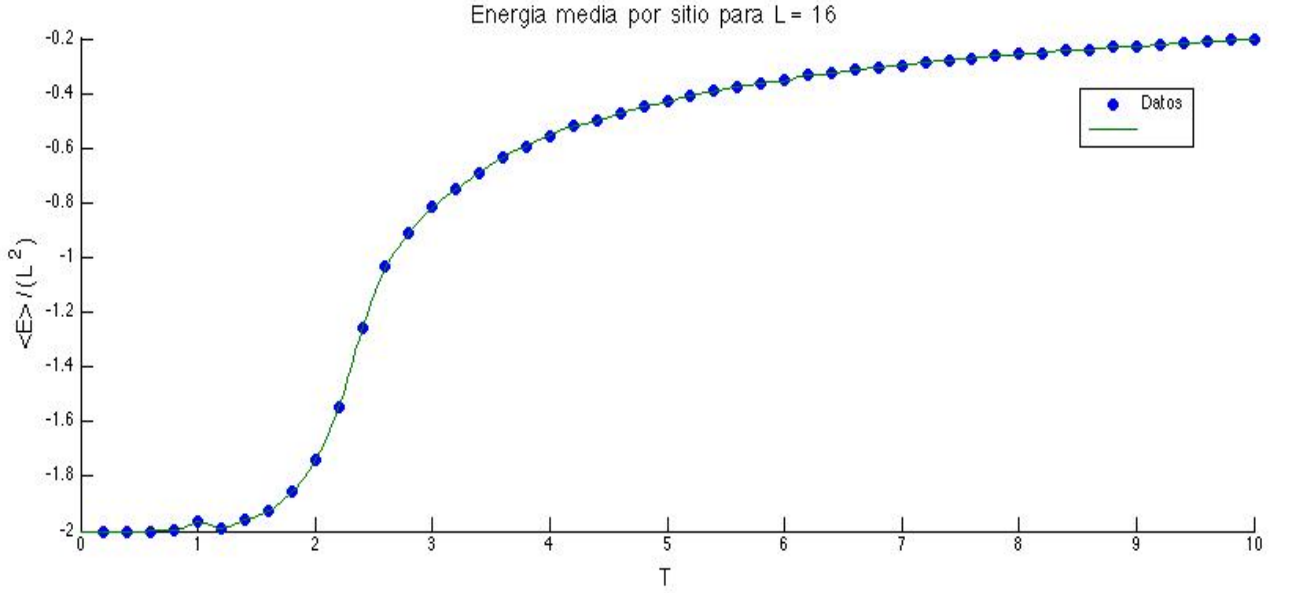


Figura 15: Energía media por sitio para $L = 16$.

Se puede observar que, para $T \rightarrow 0$, la energía media tiende a -2 ; de acuerdo con la teoría (tomando $J = 1$). En un entorno de T_c , se da un salto hacia un valor mayor; esto se corresponde a la transición de fase, que se observará más adelante como una discontinuidad en los gráficos correspondientes al C_v . La energía tiende a estabilizarse alrededor de -0.2 cuando $T \gg T_c$.

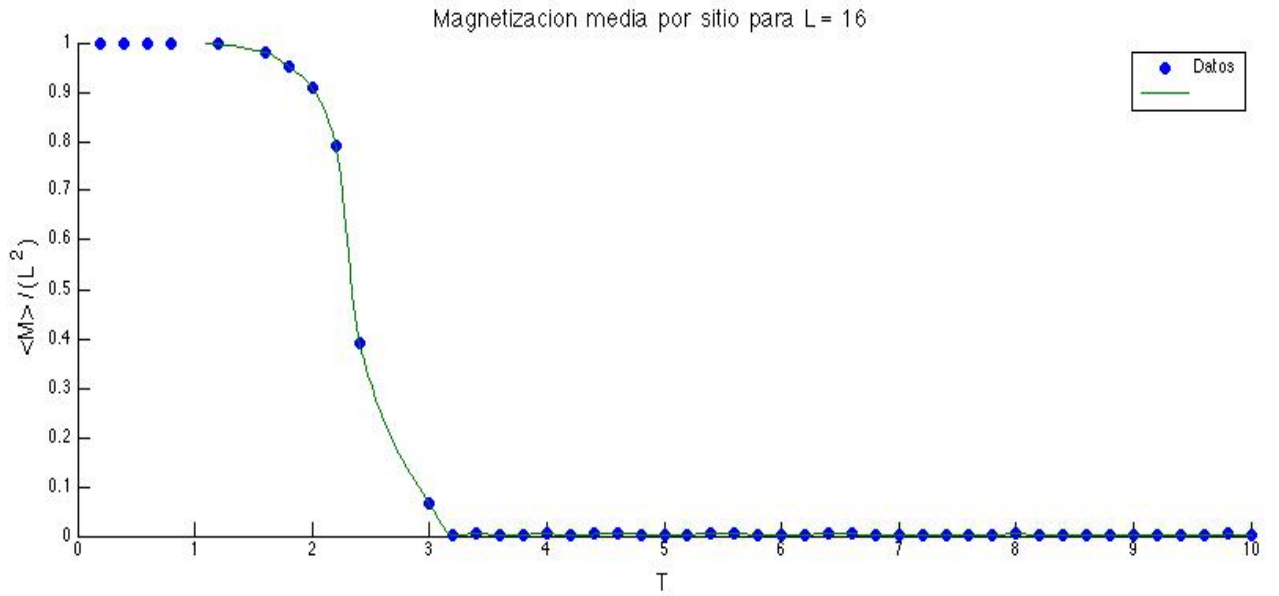


Figura 16: Magnetización media por sitio para $L = 16$.

Acá se puede ver que la magnetización mantiene un valor medio estable en 1 para valores de $T < T_c$; esto puede interpretarse como que todos los spines tienen el mismo valor. Al llegar a la temperatura crítica se observa un salto (discontinuidad) en el valor de $\langle m \rangle$ correspondiente al cambio de fase, convergiendo rápidamente a 0 para temperaturas mayores a T_c , lo que indica la pérdida de magnetización de la muestra.

Para calcular el calor específico y la susceptibilidad magnética, se utilizaron las fórmulas dadas por el enunciado, relacionándolas de la siguiente manera:

- $C_v = \frac{\sigma_E^2}{T^2}$
- $\chi_m = \frac{\sigma_M^2}{T}$

Los resultados para estos observables fueron los siguientes:

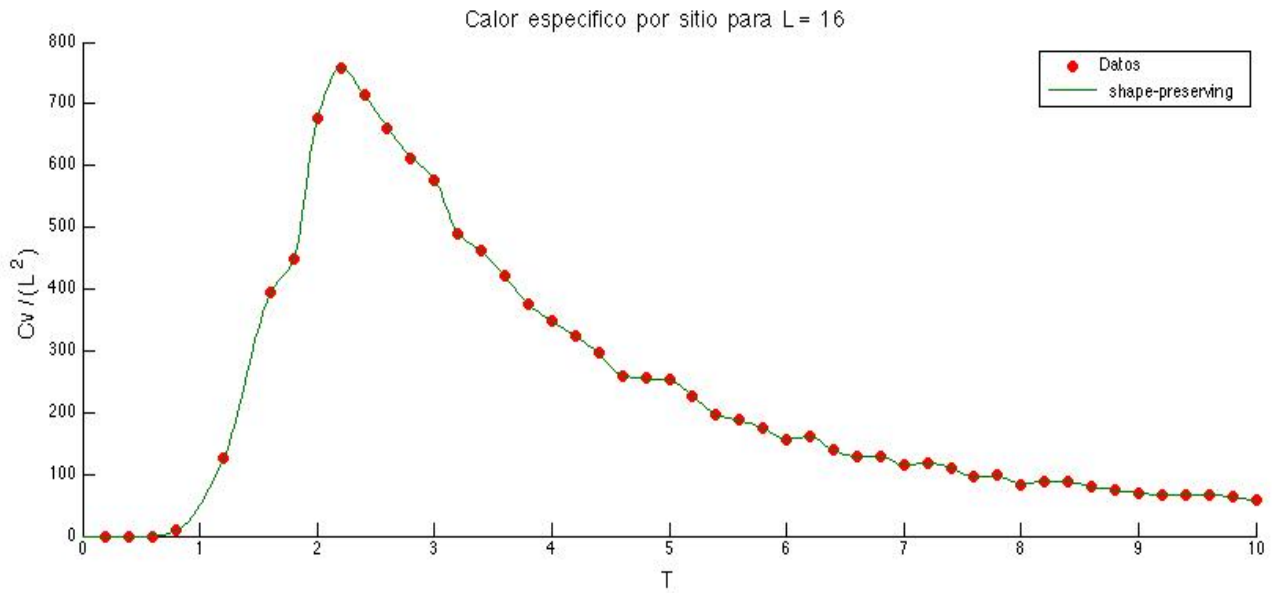


Figura 17: Calor específico por sitio para $L = 16$.

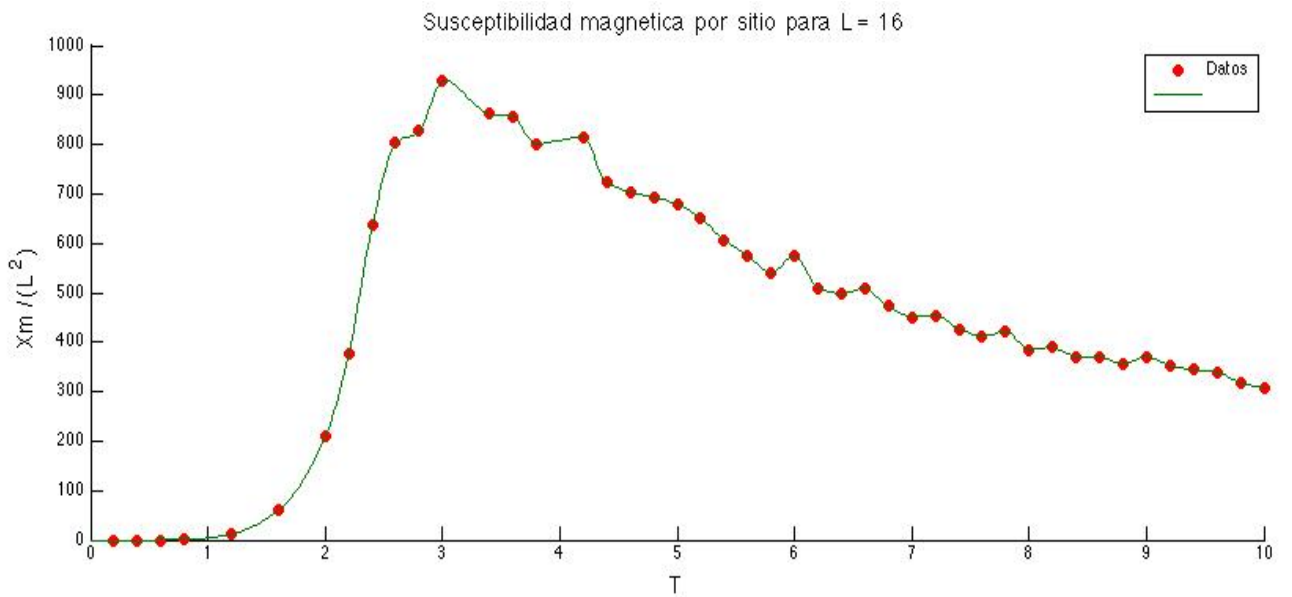


Figura 18: Susceptibilidad magnética por sitio para $L = 16$.

En los gráficos se puede observar el aumento del C_v y χ_m a medida que la temperatura se acerca al valor crítico, donde las mismas son máximas (cambio de fase). En este punto se puede ver la discontinuidad en $C_v = \frac{dE}{dT}$ y $\chi_m = \frac{dM}{d\beta}$, para luego decaer a valores cercanos a 0.

Para finalizar este ejercicio, se muestran los gráficos obtenidos de C_v y χ_m para $L = 32$:

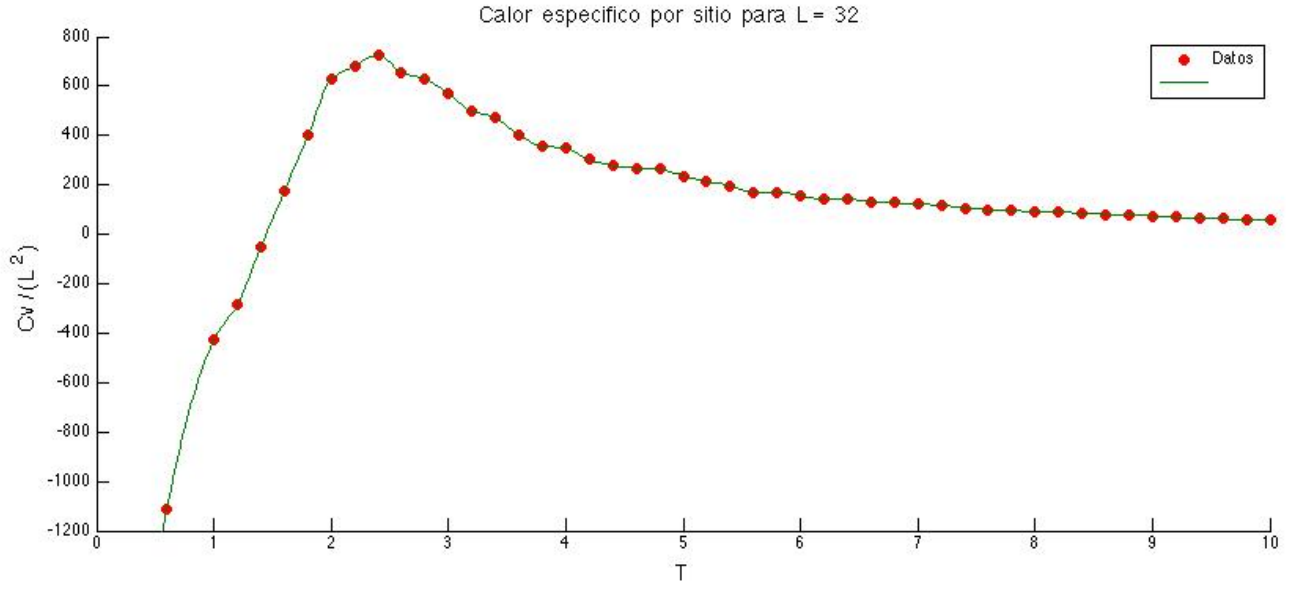


Figura 19: Calor específico por sitio para $L = 32$.

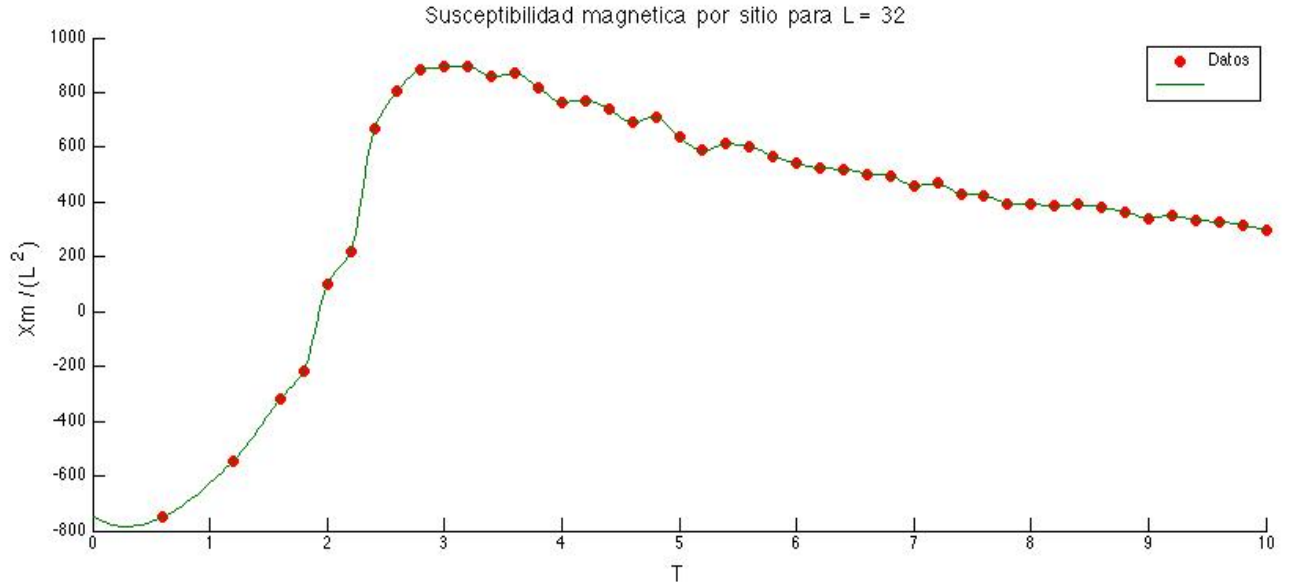


Figura 20: Susceptibilidad magnética por sitio para $L = 32$.

3.2. Conclusiones

Para $L = 32$ se puede que el decaimiento de ambos observables es aún más rápido con respecto a los correspondientes a $L = 16$, luego de la temperatura crítica T_c ; concordando con mayor exactitud

con el salto que produce la discontinuidad en las derivadas de E y M .

En ambos casos, el cambio de fase observado en C_v y χ_m se dio alrededor de un valor de la temperatura crítica en $T_c = 2.27$; de acuerdo con el mencionado en el enunciado para $T_c = \frac{2}{\ln(1+\sqrt{2})} \approx 2,2691...$

4. Ejercicio: Longitud de correlación en función de la temperatura

Para calcular la longitud de correlación, se eligieron los vecinos a un spin cualquiera en cada iteración del código; calculando primero la función de correlación entre ellos.

A continuación se presentan los resultados para una red de tamaño $L = 16$ y $L = 32$. La cantidad de iteraciones se mantuvo fija en $n = 3000$, como en el ejercicio anterior:

4.1. Resultados

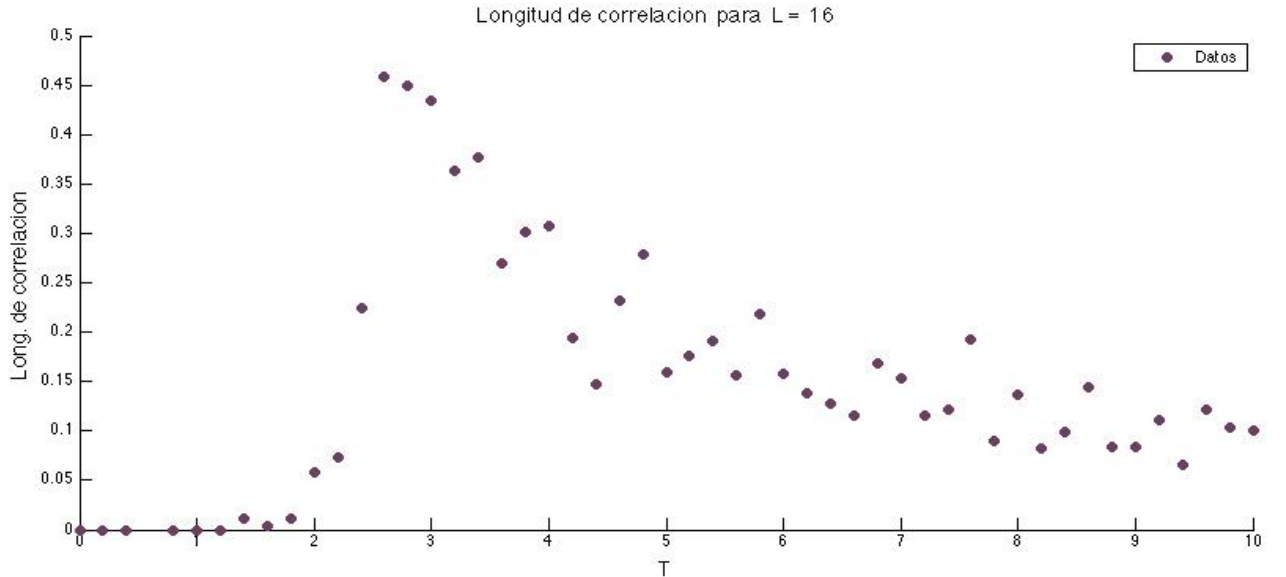


Figura 21: Longitud de correlación para $L = 36$.

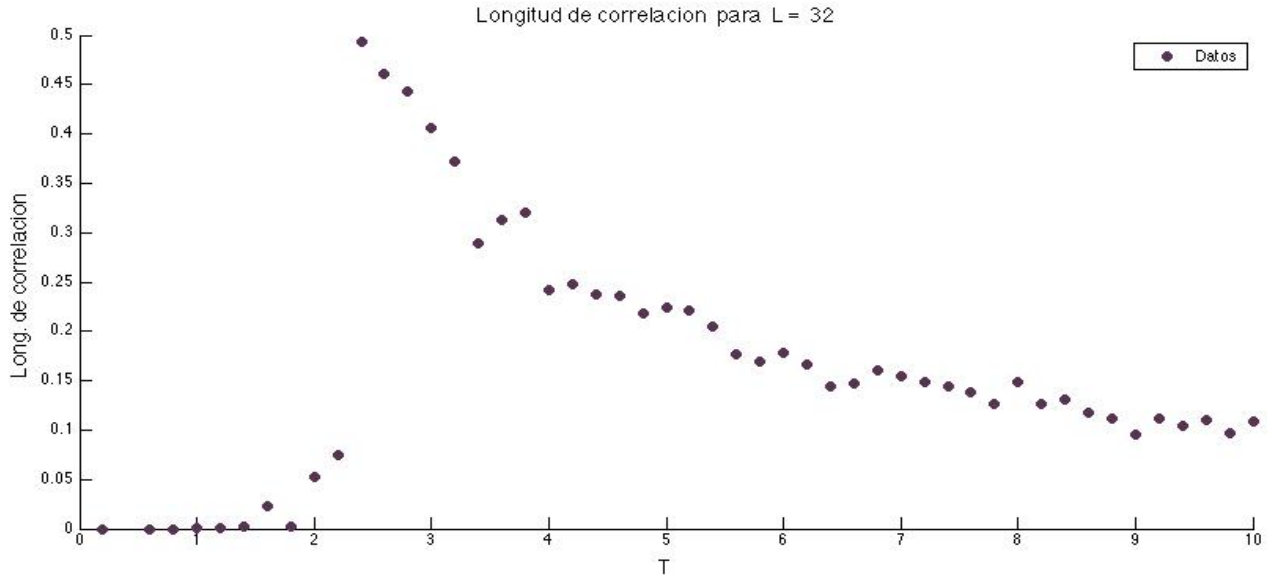


Figura 22: Longitud de correlación para $L = 32$.

4.2. Conclusiones

En primer lugar, noté que los resultados tienen menor dispersión para $L = 32$, sobre todo para $T > T_c$; lo que permite inferir en primera instancia que a medida que L aumenta, los valores se corresponden en mayor medida a la teoría, la cual indica que en un entorno de T_c , la longitud de correlación sigue una ley de potencias y no una exponencial como en el resto de las zonas.

Esto quiere decir que, a medida que la temperatura se acerca a T_c , la longitud de correlación entre espines diverge, lo que implica que espines lejanos están correlacionados, es decir, el estado de uno afecta a otro con una distancia mayor a la de sus primeros o segundos vecinos, por ejemplo. Por tanto, el sistema cerca de una transición de fase pierde memoria de su estructura microscópica y comienza a presentar nuevas correlaciones macroscópicas de largo alcance.

5. Apéndice

5.1. Código fuente de la función generadora de números enteros aleatorios (/codigo/randint.m)

```

1  %Funcion auxiliar que genera una matriz de nxm con los numeros enteros
2  %generados uniformemente en el intervalo (a,b)
3  function rn = randint(n,m,a,b)
4      rn = 1 + floor((a - 1) + (b - (a - 1))*rand(n,m));
5  end

```


5.2. Código fuente de la función de energía total del sistema (/codigo/En.m)

```

1 function e = En(S)
2     e = 0;
3
4     for i=1:length(S)
5         for j=1:length(S)
6
7             %Para cada spin, considero el spin vecino de arriba (fila i-1)
8             %el de su derecha (columna j+1)
9             iup = mod(i - 1 - 1, length(S)) + 1;
10            jright = mod(j + 1 - 1, length(S)) + 1;
11
12            e = e - S(i,j)*(S(iup,j)+S(i,jright));
13        end
14    end
15 end

```

5.3. Código fuente de la función de Ising 2D en un paso (/codigo/ising2Dpaso.m)

```

1 function [S,dE,dM,Sis,SiSjs] = ising2Dpaso(S,beta)
2     dM = 0;
3     dE = 0;
4
5     Sis = [];
6     SiSjs = [];
7
8     %Realiza el paso para cada spin de S
9     for it=1:length(S)*length(S)
10
11         %Elijo los indices i,j al azar con la funcion auxiliar definida
12         indices = randint(1,2,1,length(S));
13         i = indices(1); j = indices(2);
14
15         %Considero los indices de los spines vecinos al S(i,j)
16         iup = mod(i - 1 - 1, length(S)) + 1;
17         idown = mod(i + 1 - 1, length(S)) + 1;
18         jleft = mod(j - 1 - 1, length(S)) + 1;
19         jright = mod(j + 1 - 1, length(S)) + 1;
20
21         %Calculo la diferencia de energia al cambiar el spin S(i,j)
22         E_old = -S(i,j)*(S(iup,j)+S(idown,j)+S(i,jleft)+S(i,jright));
23         E_new = S(i,j)*(S(iup,j)+S(idown,j)+S(i,jleft)+S(i,jright));
24
25
26         dEt = E_new - E_old;
27
28         %Aplico el criterio de eleccion y guardo en las variables
29         %correspondientes dM y dE; actualizo S(i,j)
30         if ((dEt <= 0) || (rand() < exp(-beta*dEt)))

```

```

31         S(i,j) = -S(i,j);
32         dM = dM + 2*S(i,j);
33         dE = dE + dEt;
34     end
35
36     S_corr = (S(iup,j)+S(idown,j)+S(i,jleft)+S(i,jright));
37     Si = S(i,j);
38     SiSj = Si * S_corr / 4;
39     Sis = [Sis; Si];
40     SiSjs = [SiSjs; SiSj];
41 end
42 end

```

5.4. Código fuente de la función de Ising 2D para un valor de T (/codigo/ising2D0.m)

```

1  function [E, M, Cv, Xm, corr_func] = ising2D0(L,T,npasos,a)
2      %Script general para hacer una corrida a un set de parametros
3      %beta, lado de la red)
4
5      beta = 1/T;
6
7
8      %propongo un estado inicial al azar
9      %S es una matriz de 1 y -1 indicando las dos proyecciones de
10     %spin
11     S = 2*(rand(L,L) > 0.5) - 1;
12
13     %Condiciones iniciales y vectores para guardar los valores de energia,
14     %magnetizacion, energia^2 y magnetizacion^2 para un valor de T
15     npre = 100;
16     energia = zeros(npasos + 1,1);
17     magnet = zeros(npasos + 1,1);
18     energia2 = zeros(npasos + 1,1);
19     magnet2 = zeros(npasos + 1,1);
20
21
22     %Pretermalizo
23     %La funcion ising2Dpaso hace un nuevo elemento de la cadena de Markov. La tienen
24     %que escribir uds!
25     for n=1:npre
26         [S, dE, dM, Sis, SiSjs] = ising2Dpaso(S,beta);
27     end
28
29     %La funcion En calcula la energia de la red. Tambien la tienen que escribir uds!
30     energia(1) = En(S);
31     magnet(1) = sum(sum(S));
32     energia2(1) = energia(1)*energia(1);
33     magnet2(1) = magnet(1)*magnet(1);
34
35     %Ciclo para calcular el estado de S luego de termalizar
36     for n=1:npasos
37         [S, dE, dM, Sis, SiSjs] = ising2Dpaso(S,beta);

```

```

37
38     energia(n+1) = energia(n) + dE;
39     magnet(n+1) = magnet(n) + dM;
40     energia2(n+1) = energia2(n) + dE*dE;
41     magnet2(n+1) = magnet2(n) + dM*dM;
42
43     %El tercer parametro de entrada es para mostrar/ocultar los graficos.
44     %Util para el ejercicio siguiente
45     if a == 'y'
46         if(mod(n,10) == 0)
47             imagesc(S); shading flat;
48             title(['n = ' num2str(n) ' T = ' num2str(1/beta) ' e = ' num2str(
                     energia(n)/(L*L)) ' m = ' num2str(magnet(n)/(L*L))]);
49             drawnow;
50         end
51     end
52 end
53
54 %Valores medios de los vectores previamente calculados:
55 %E>, <M>, Cv = (1/T^2)*(<E^2> - <E>^2), Xm = (1/T)*(<M^2> - <M>^2)
56 E = mean(energia);
57 M = mean(magnet);
58 E2 = mean(energia2);
59 M2 = mean(magnet2);
60 Cv = (E2 - E*E)/(T*T);
61 Xm = (M2 - M*M)/T;
62
63 %Variables para calcular la longitud de correlacion: <Si> y <SiSj>
64 Si_avg = mean(Sis);
65 SiSj_avg = mean(SiSjs);
66 corr_func = abs(SiSj_avg - Si_avg*Si_avg);
67
68
69 %Otros plots que pueden/deben hacer, sobreescribe los anteriores.
70 if a == 'y'
71     figure()
72     plot(energia/(L*L))
73     title(['L = ' num2str(L)])
74     xlabel('Paso')
75     ylabel('Energia por sitio')
76
77     figure()
78     plot(abs(magnet)/(L*L))
79     title(['L = ' num2str(L)])
80     xlabel('Paso')
81     ylabel('Magnetizacion por sitio')
82 end
83 end

```

5.5. Código fuente de la funcion de Ising 2D para varios valores de T (/codigo/ising2D1.m)

1 *%Script general para hacer una corrida a un set de parametros*

```

2  %beta, lado de la red
3
4  %Lado de la red
5  L = 16;
6
7  %Valores de T y vectores iniciales para <E>, <M>, Cv y Xm
8  T = 0:0.2:10;
9  Energia_t = [];
10 Magnet_t = [];
11 Cv_t = [];
12 Xm_t = [];
13 corr_funcs = [];
14
15
16 %Calculo, para cada temperatura, el estado del sistema y sus variables
17 %termodinamicas y las guardo en los vectores definidos anteriormente
18 for t=T
19
20     [E, M, Cv, Xm, corr_func] = Ising2D0(L,t,3000,'n');
21
22
23     Energia_t = [Energia_t; E];
24     Magnet_t = [Magnet_t; M];
25
26     Cv_t = [Cv_t; Cv];
27     Xm_t = [Xm_t; Xm];
28     corr_funcs = [corr_funcs; corr_func];
29
30
31
32 end
33
34
35
36 %Otros plots que pueden/deben hacer, sobreescibe los anteriores.
37 figure()
38 scatter(T, Energia_t/(L*L), 'b.')
39 title(['Energia media por sitio para L = ', num2str(L)])
40 legend('Datos')
41 xlabel('T')
42 ylabel('<E> / (L^2)')
43
44 figure()
45 scatter(T, abs(Magnet_t)/(L*L), 'b.')
46 title(['Magnetizacion media por sitio para L = ', num2str(L)])
47 legend('Datos')
48 xlabel('T')
49 ylabel('<M> / (L^2)')
50
51 figure()
52 scatter(T, Cv_t/(L*L), 'r.')
53 title(['Calor especifico por sitio para L = ', num2str(L)])
54 legend('Datos')
55 xlabel('T')

```

```

56 ylabel('Cv / (L^2)')
57
58 figure()
59 scatter(T, Xm_t/(L*L), 'r. ')
60 title(['Susceptibilidad magnetica por sitio para L = ', num2str(L)])
61 legend('Datos')
62 xlabel('T')
63 ylabel('Xm / (L^2)')
64
65 figure()
66 scatter(T, corr_funcs, 'g. ')
67 title(['Longitud de correlacion para L = ', num2str(L)])
68 legend('Datos')
69 xlabel('T')
70 ylabel('Long. de correlacion')

```