

Mecánica Estadística: Tarea 4

Modelo de Ising - Transiciones de Fase

Iván Mauricio Burbano Aldana

27 de marzo de 2018

1. Dualidad

1.1. Desarrollo de baja temperatura

1. Como las condiciones de frontera son periódicas, la red cuadrada se puede considerar como embebida en un toro. En tal caso, cada espín tiene 4 vecinos. Por lo tanto la suma de los vecinos en cada vértice de la red es $4N$. En este cálculo cada par de vecinos se cuenta dos veces, una por cada vértice en el par. Luego el número total de pares de vecinos es

$$s = \frac{4N}{2} = 2N. \quad (1)$$

2. Vemos que el término

$$\sigma_i \sigma_j = \begin{cases} 1 & \sigma_i = \sigma_j \\ -1 & \sigma_i \neq \sigma_j \end{cases} \quad (2)$$

pues $\sigma_i, \sigma_j \in \{-1, 1\}$. Ahora bien, podemos caracterizar r como

$$r = |\{\langle i, j \rangle | \text{el espín } i \text{ es antiparalelo al espín } j\}| = |\{\langle i, j \rangle | \sigma_i \neq \sigma_j\}|. \quad (3)$$

De manera análoga

$$s - r = |\{\langle i, j \rangle | \text{el espín } i \text{ es paralelo al espín } j\}| = |\{\langle i, j \rangle | \sigma_i = \sigma_j\}|. \quad (4)$$

Luego

$$\sum_{\langle i, j \rangle} \sigma_i \sigma_j = 1|\{\langle i, j \rangle | \sigma_i = \sigma_j\}| - 1|\{\langle i, j \rangle | \sigma_i \neq \sigma_j\}| = s - r - r = s - 2r. \quad (5)$$

3. Se tiene una función de partición es

$$\begin{aligned} Z(T) &= \sum_{(\sigma_1, \dots, \sigma_N) \in \{-1, 1\}^N} e^{\beta J \sum_{\langle i, j \rangle} \sigma_i \sigma_j} = \sum_{(\sigma_1, \dots, \sigma_N) \in \{-1, 1\}^N} e^{L(s-2r)} \\ &= e^{Ls} \sum_{(\sigma_1, \dots, \sigma_N) \in \{-1, 1\}^N} e^{-2Lr} = e^{Ls} \sum_{r=0}^s A_r e^{-2Lr} \end{aligned} \quad (6)$$

donde A_r es el número de configuraciones $(\sigma_1, \dots, \sigma_N) \in \{0, 1\}^N$ con r pares de espines antiparalelos. Ahora bien, note que la configuración $(\sigma_1, \dots, \sigma_N) \in \{0, 1\}^N$ tiene el mismo número de espines antiparalelos que $(-\sigma_1, \dots, -\sigma_N)$. Luego $A_r = 2\omega_r$ donde ω_r es el numero de configuraciones $(\sigma_1, \dots, \sigma_N) \in \{0, 1\}^N$ con r pares de espines antiparalelos salvo multiplicación por -1 . Luego

$$Z(T) = 2e^{Ls} \sum_{r=0}^s \omega_r e^{-2Lr}. \quad (7)$$

Mas aún, las únicas configuraciones que tienen 0 pares de espines antiparalelos son $(1, \dots, 1)$ y $(-1, \dots, -1)$. Luego $\omega_0 = 1$ y

$$Z(T) = 2e^{Ls} \left(e^0 + \sum_{r=1}^s \omega_r e^{-2Lr} \right) = 2e^{Ls} \left(1 + \sum_{r=1}^s \omega_r e^{-2Lr} \right). \quad (8)$$

4. Note que si $T \ll J/k_B$ entonces $L = J/k_B T \gg 1$. Entonces $e^{2L} \gg 1$ y

$$e^{-2L} = \frac{1}{e^{2L}} \ll 1. \quad (9)$$

Luego el factor

$$1 + \sum_{r=1}^s \omega_r e^{-2Lr} = \sum_{r=0}^s \omega_r (e^{-2L})^r \quad (10)$$

es una expansión en serie de potencias de e^{-2L} que puede ser aproximada ignorando los términos de orden superior. Es por esto que la expresión (8) es una desarrollo de baja temperatura de la función de partición.

1.2. Auto-dualidad y determinación de la temperatura crítica

1. Ver figura 1.

2. Ver figura 2.

3. Note que el número de pares de vecinos antiparalelos en la figura 2 es $r = 20$ (salvo multiplicación por -1). Esto se puede contar fácilmente notando que cada linea azul en la red original separa a un par de espines antiparalelos en la red dual. Esto por supuesto es una característica general: *cualquier diagrama de r barras determina una configuración con r pares de vecinos antiparalelos en la red dual y viceversa*. En particular, el número de diagramas con r barras es igual al de configuraciones con r pares de espines vecinos antiparalelos. Por lo tanto, bajo esta dualidad la función de partición se transforma en

$$Z^*(T) = 2e^{sL} \left(1 + \sum_{r=1}^s \Omega_r e^{-2Lr} \right) \quad (11)$$

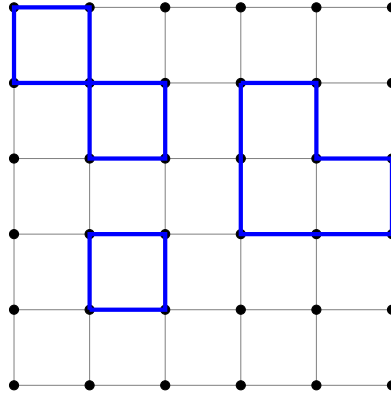


Figura 1: Un gráfico de 20 barras que contribuye sobre un retículo cuadrado de 36 vértices.

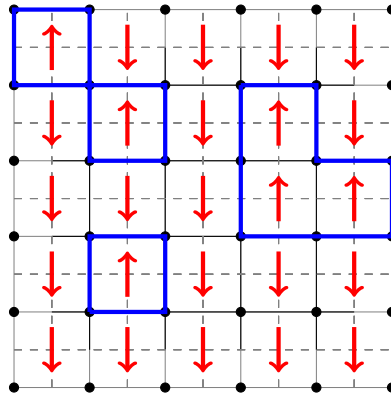


Figura 2: En la red dual sobre la figura 1 se dibujan espines hacia arriba dentro de los polígonos y espines hacia abajo fuera.

4. En efecto,

$$\begin{aligned}
& 2^{N-1} \cosh(L)^s e^{-sL^*} Z^*(T^*) \\
&= 2^{N-1} \cosh(L)^s e^{-sL^*} 2e^{sL^*} \left(1 + \sum_{r=1}^s \Omega_r e^{-2L^* r} \right) \\
&= 2^N \cosh(L)^s \left(1 + \sum_{r=1}^s \Omega_r \left(e^{-2L^*} \right)^r \right) \\
&= 2^N \cosh(L)^s \left(1 + \sum_{r=1}^s \Omega_r \tanh(L)^r \right) = Z(T).
\end{aligned} \tag{12}$$

5. Se tiene que

$$e^{-sL^*} = e^{-2L^* s/2} = \left(e^{-2L^*} \right)^{s/2} = \tanh(L)^{s/2} = \frac{\sinh(L)^{s/2}}{\cosh(L)^{s/2}}. \tag{13}$$

Por lo tanto

$$\begin{aligned}
\cosh(2L) &= \cosh(L)^2 + \sinh(L)^2 = \cosh(L)^2 \left(1 + \tanh(L)^2 \right) \\
&= \cosh(L)^2 \frac{1 + \tanh(L)^2}{\cosh(L)^2 - \sinh(L)^2} = \frac{1 + \tanh(L)^2}{1 - \tanh(L)^2} \\
&= \frac{1 + (e^{-2L^*})^2}{1 - (e^{-2L^*})^2} = \frac{1 + e^{-4L^*}}{1 - e^{-4L^*}} = \frac{e^{2L^*} + e^{-2L^*}}{e^{2L^*} - e^{-2L^*}} = \coth(2L^*)
\end{aligned} \tag{14}$$

y

$$\begin{aligned}
\sinh(2L) &= 2 \sinh(L) \cosh(L) = 2 \frac{\sinh(L)}{\cosh(L)} \cosh(L)^2 = 2 \tanh(L) \cosh(L)^2 \\
&= \frac{2 \tanh(L) \cosh(L)^2}{\cosh(L)^2 - \sinh(L)^2} = \frac{2 \tanh(L)}{1 - \tanh(L)^2} = \frac{2e^{-2L^*}}{1 - (e^{-2L^*})^2} \\
&= \frac{2e^{-2L^*}}{1 - e^{-4L^*}} = \frac{2}{e^{2L^*} - e^{-2L^*}} = \operatorname{csch}(2L^*).
\end{aligned} \tag{15}$$

Se tiene entonces que

$$\begin{aligned}
\cosh(L)^s e^{-sL^*} &= \cosh(L)^s \frac{\sinh(L)^{s/2}}{\cosh(L)^{s/2}} \\
&= \cosh(L)^{s/2} \sinh(L)^{s/2} \\
&= (\cosh(L) \sinh(L))^{s/2} \\
&= \sinh(2L)^{s/2} / 2^{s/2} \\
&= \frac{\cosh(2L^*)^{s/2}}{\cosh(2L^*)^{s/2}} \sinh(2L)^{s/2} / 2^{s/2} \\
&= \frac{\cosh(2L^*)^{s/2}}{\cosh(2L^*)^{s/2}} \frac{1}{\sinh(2L^*)^{s/2}} / 2^{s/2} \\
&= \frac{\coth(2L^*)^{s/2}}{\cosh(2L^*)^{s/2}} / 2^{s/2} \\
&= \frac{\cosh(2L)^{s/2}}{\cosh(2L^*)^{s/2}} / 2^N
\end{aligned} \tag{16}$$

de lo cual concluimos

$$\begin{aligned}
\frac{Z(T)}{2^{(N-1)/2} \cosh(2L)^{s/2}} &= \frac{2^{N-1} \cosh(L)^s e^{-sL^*} Z^*(T^*)}{2^{(N-1)/2} \cosh(2L)^{s/2}} \\
&= \frac{2^{N-1} \frac{\cosh(2L)^{s/2}}{\cosh(2L^*)^{s/2}} / 2^N Z^*(T^*)}{2^{(N-1)/2} \cosh(2L)^{s/2}} \\
&= \frac{2^{-1} Z^*(T^*)}{2^{(N-1)/2} \cosh(2L^*)^{s/2}} = \frac{Z^*(T^*)}{2^{(N+1)/2} \cosh(2L^*)^{s/2}}
\end{aligned} \tag{17}$$

6. Note que

$$\begin{aligned}
\frac{Z^*(T^*)}{\cosh(2L^*)^{s/2}} &= \frac{2^{(N+1)/2}}{2^{(N-1)/2}} \frac{Z(T)}{\cosh(2L)^{s/2}} \\
\overrightarrow{\text{límite termodinámico}} & \frac{Z(T)}{\cosh(2L)^{s/2}}
\end{aligned} \tag{18}$$

7. Se tiene que la energía libre de Gibbs en el límite termodinámico es

$$\begin{aligned}
\beta f(T) &= -\frac{1}{N} \ln(Z(T)) = -\frac{1}{N} \ln\left(\frac{\cosh(2L)^{s/2}}{\cosh(2L^*)^{s/2}} Z(T^*)\right) \\
&= -\frac{s}{2N} \ln\left(\frac{\cosh(2L)}{\cosh(2L^*)}\right) - \frac{1}{N} \ln(Z(T^*)).
\end{aligned} \tag{19}$$

Dado que el término

$$-\frac{sk_B T}{2N} \ln\left(\frac{\cosh(2L)}{\cosh(2L^*)}\right) \tag{20}$$

es analítico, se concluye que f presenta discontinuidades indicadoras de una transición de fase a temperatura T_c si y solo si el termino $\ln(Z(T_c))$ o equivalentemente $\ln(Z(T_c^*))$ las presenta. Pero en tal caso se concluye que f también presentaría esta discontinuidad en T_c^* .

8. Si solo hay una transición de fase se tiene que $T_c = T_c^*$. Luego por la ecuación (15) se tiene $\sinh(2L_c) = \text{csch}(2L_c)$. Como \sinh es positivo en los positivos, esto implica

$$\sinh\left(\frac{2J}{k_B T_c}\right) = 1, \quad (21)$$

es decir,

$$T_c = \frac{2J}{k_B \text{arsinh}(1)} \approx 2,27J/k_B. \quad (22)$$

2. Simulación Monte Carlo del modelo de Ising con el algoritmo de Metropolis

2.1. Elementos de teoría

1. La probabilidad $W(C_1 \rightarrow C_2)$ pasar de la configuración C_1 a la configuración C_2 donde C_2 y C_1 difieren por solo un espín depende de la diferencia de energía $\Delta E = H(C_2) - H(C_1)$ entre las configuraciones. Según el algoritmo

$$W(C_1 \rightarrow C_2) = \begin{cases} 1 & \Delta E \leq 0 \\ e^{-\beta \Delta E} & \Delta E > 0 \end{cases} \quad (23)$$

2. Se tiene la condición de balance detallado

$$\begin{aligned} P_{eq}(C_1)W(C_1 \rightarrow C_2) &= \begin{cases} \frac{e^{-\beta H(C_1)}}{Z} & H(C_2) - H(C_1) \leq 0 \\ \frac{e^{-\beta H(C_1)}}{e^{-\beta H(C_1)} e^{-\beta(H(C_2) - H(C_1))}} \frac{1}{Z} & H(C_2) - H(C_1) > 0 \end{cases} \\ &= \begin{cases} \frac{e^{-\beta H(C_1)}}{Z} & H(C_2) - H(C_1) < 0 \\ \frac{e^{-\beta H(C_1)}}{Z} & H(C_2) - H(C_1) = 0 \\ \frac{e^{-\beta H(C_1)}}{e^{-\beta H(C_1)} e^{-\beta(H(C_2) - H(C_1))}} \frac{1}{Z} & H(C_2) - H(C_1) > 0 \end{cases} \\ &= \begin{cases} \frac{e^{-\beta H(C_2)} e^{-\beta(H(C_1) - H(C_2))}}{Z} & H(C_2) - H(C_1) < 0 \\ \frac{e^{-\beta H(C_2)}}{Z} & H(C_2) - H(C_1) = 0 \\ \frac{e^{-\beta H(C_2)}}{Z} & H(C_2) - H(C_1) > 0 \end{cases} \quad (24) \\ &= \begin{cases} \frac{e^{-\beta H(C_2)} e^{-\beta(H(C_1) - H(C_2))}}{Z} & H(C_1) - H(C_2) > 0 \\ \frac{e^{-\beta H(C_2)}}{Z} & H(C_1) - H(C_2) \leq 0 \end{cases} \\ &= P_{eq}(C_2)W(C_2 \rightarrow C_1) \end{aligned}$$

2.2. Implementación

Según (22) en nuestra simulación deberíamos ver una transición de fase. Para poder observarla necesitamos representar el límite termodinámico y por lo

tanto considerar un número grande de sitios. Sin embargo, dada la naturaleza del algoritmo de Metropolis, un número grande de sitios requiere un número considerable de iteraciones para la estabilización. Esto se debe a que para barrer un número representativo de configuraciones debemos permitir que cada sitio tenga la oportunidad de cambiar. Intentando hallar un balance entre el número de sitios y el número de iteraciones se decidió tomar un retículo cuadrado de 100 espines junto con 1000000 de iteraciones. De estas, las primeras 10000 fueron consideradas como parte del régimen estabilizador y por lo tanto ignoradas para el cálculo de la magnetización.

En primer lugar consideraremos las curvas de energía obtenidas. Como se puede observar en la figura 3, A medida que la temperatura aumenta la energía promedio y sus fluctuaciones también. Esto concuerda con la percepción intuitiva de la temperatura. Este fenómeno se puede rastrear a (24). En el algoritmo implementado cada configuración C se obtiene con una probabilidad $e^{-\beta H(C)}$. Por lo tanto a medida que la temperatura aumenta, es decir, β disminuye, la probabilidad de obtener configuraciones de altas energías aumenta.

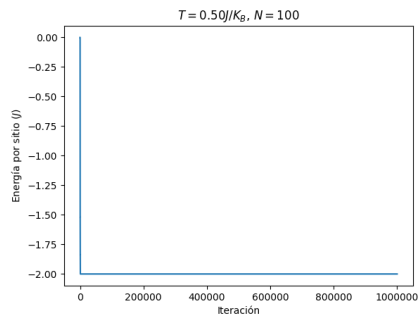
Para hallar la curva de magnetización en función de temperatura en la figura 4, se barrieron 50 temperaturas entre $0.5J/k_B$ y $3J/k_B$. Para cada temperatura la magnetización se calculó tomando la magnetización total en cada iteración y luego calculando el promedio descartando las primeras 10000 iteraciones. En la figura se observa que la magnetización decae para temperaturas mayores a $2J/k_B$. Esto concuerda con la transición de fase hallada teóricamente en (22). Con esta figura podemos distinguir que a temperaturas menores que la crítica el sistema es magnético. En efecto, las fluctuaciones térmicas son bajas y la interacción en el Hamiltoniano hace que los espines apunten en la misma dirección (asumiendo $J > 0$). Debido a esto, los momentos magnéticos en cada sitio contribuyen a la magnetización del material obteniendo así propiedades magnéticas. La transición de fase corresponde a que a altas temperaturas las fluctuaciones térmicas se vuelven más relevantes. En tal caso, la interacción entre espines se vuelve despreciable y estos toman posiciones aleatorias. De esta manera la contribución del momento magnético de cada sitio se vuelve destructiva y se anula la magnetización del material.

El código debidamente documentado fue

```
import numpy as np
import random
import matplotlib.pyplot as plt
from matplotlib import rc

#Se definen los parametros de la simulacion

L_c = 0.5*np.arcsinh(1)
temperaturas = np.linspace(0.5, 3.2, 1000)
N = 10
nsteps = 1000000
nsteps_equilibrio = 10000
```



energia_1_265306.png

energia_1_520408.png

energia_2_642857.png

Figura 3: Se muestra la energía de la configuración obtenida en cada iteración para tres temperaturas distintas. En el título se indica la temperatura de la simulación T y el número de espines considerados N .

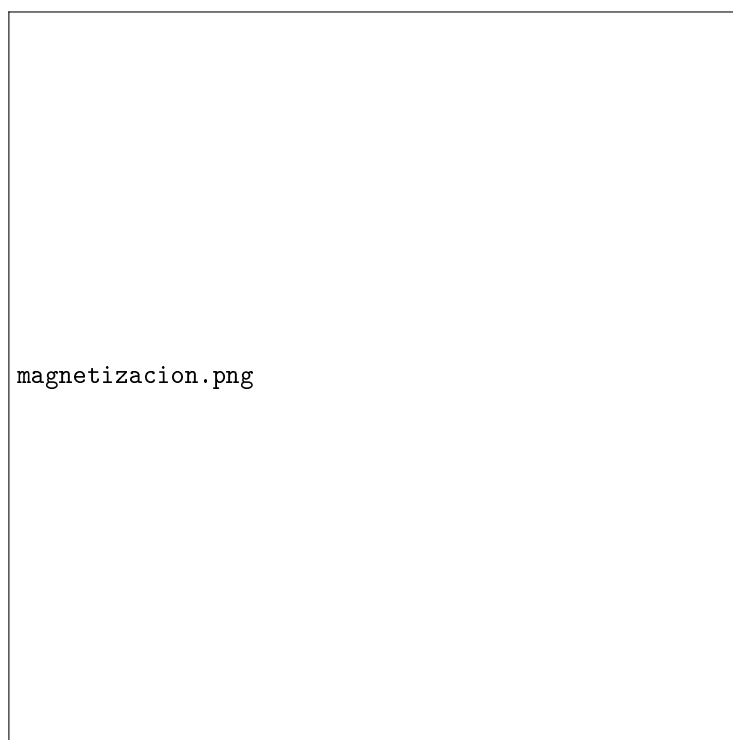


Figura 4: Se muestra la magnetización total en función de la temperatura.

```

#Se crea una funcion que calcula la energia de un
reticulo en unidades de J

def H(sigma):
    E = 0
    for i in range(0, len(sigma)):
        for j in range(0, len(sigma[i])):
            E += sigma[i,j]*(sigma[i, (j+1) %
                len(sigma[i])] + sigma[i, (j
                -1) % len(sigma[i])] + sigma[(
                i+1) % len(sigma), j] + sigma
                [(i-1) % len(sigma), j])

    E = -0.5*E
    return E

#Se crean las listas de energia y magnetizacion promedio

E_promedio = np.zeros(len(temperaturas))
M_promedio = np.zeros(len(temperaturas))

#Se realiza el método para cada temperatura

for t in range(0, len(temperaturas)):

    L = 1.0/temperaturas[t]

    #Se crea el reticulo de espines

    reticulo = np.zeros((N, N))

    #Se crea una lista de energias y magnetizaciones

    E = np.zeros(nsteps)
    M = np.zeros(nsteps)

    #Se le asigna un valor aleatorio en {-1,1} a cada
    sitio en el reticulo

    for i in range(0, N):
        for j in range(0, N):
            reticulo[i,j] = random.randrange
                (-1,2,2)

    #Se inicializa la lista de energias y
    magnetizaciones

```

```

E[0] = H(reticulo)
M[0] = np.sum(reticulo)

for n in range(1, nsteps):

    #Se selecciona una entrada aleatoria en
    el reticulo

    i = random.randrange(0, len(reticulo))
    j = random.randrange(0, len(reticulo[i]))

    #Se calcula el cambio en la energia

    Delta_E = 2*reticulo[i,j]*(reticulo[i, (j
    +1) % len(reticulo[i])] + reticulo[i,
    (j-1) % len(reticulo[i])] + reticulo[(i
    +1) % len(reticulo), j] + reticulo[(i
    -1) % len(reticulo), j])

    #Se decide si se acepta el cambio

    if (Delta_E <= 0) or (random.uniform(0,
    1) < np.exp(-L*Delta_E)):
        reticulo[i,j] = -reticulo[i,j]

    #Se calcula la nueva energia
    E[n] = H(reticulo)
    M[n] = np.sum(reticulo)

#Se hacen graficas de la evolucion de la energia

fig, ax = plt.subplots()
ax.plot(range(0, nsteps), E/(N*N))
ax.set_xlabel("Iteración")
ax.set_ylabel("Energía por sitio ($J$)")
ax.set_title("$T= %.2f$ J/K_B$, $N= %d$" % (
    temperaturas[t], N*N))
fig.savefig("energia_%.png" % temperaturas[t])
plt.close(fig)

#Se guarda el valor de la energia promedio y la
magnetizacion promedio

E_promedio[t] = np.mean(E[nsteps_equilibrio:])

```

```

M_promedio[t] = np.mean(M[nsteps_equilibrio:])

#Se grafica la magnetizacion

fig, ax = plt.subplots()
ax.scatter(temperaturas, M_promedio/(N*N), alpha = 0.3)
ax.set_xlabel("Temperatura ($J/K_B$)")
ax.set_ylabel("Magnetización por sitio")
ax.set_title("$N=%d$, $%d$ pasos con $%d$ para
             estabilizar" % (N*N, nsteps, nsteps_equilibrio))
fig.savefig("magnetizacion.png")
plt.close(fig)

```