

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/322255721>

Proyecto final (FS-5221): Modelo de Ising

Working Paper · March 2016

DOI: 10.13140/RG.2.2.27592.83208

CITATIONS

0

READS

286

1 author:



[Manuel Agustín Morgado](#)

Simon Bolivar University

7 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Undergraduate research [View project](#)



Advance Laboratory Experience [View project](#)

Proyecto final: Modelo de Ising. Física Computacional FS-5221

Manuel Morgado

Carnet: 08-10779

Universidad Simón Bolívar. Marzo 2016

Abstract

Se estudiará una simulación computacional del *Modelo de Ising*, por medio de variables físicas como: la *magnetización* (M), la *energía* (E), el *calor específico* (C_v) y la *susceptibilidad* (χ_o). Demostrando así, la eficiencia del *Algoritmo de Metropolis* para este tipo de simulación.

La física moderna ha encontrado como aprovechar la tecnología actual (i.e las computadoras), para que tanto físicos teóricos como experimentales puedan explotar y facilitar su trabajo. Por un lado, los físicos teóricos pueden llevar a cabo simulaciones de sus modelos teóricos para tener al menos, una primera aproximación y evidencia de la real descripción que lleva a cabo su modelo. Por otra parte, los físicos experimentales, pueden realizar simulaciones a computadora para comparar con sus resultados en el laboratorio, de esta forma ellos tienen una segunda referencia de los resultados esperados.

De este modo, ha surgido el campo de la *Física Computacional* como una delgada interface entre la *Física Teórica* y *Experimental*.

Modelos como el de *E. Ising* propuesto en 1925 (por sugerencia de *W. Lenz*), para describir el comportamiento de materiales ferromagnéticos, han sido incansablemente estudiados por la física computacional empleando métodos estocásticos para el cálculo de variables físicas como: la magnetización, la energía, el calor específico y la susceptibilidad del sistema. Con lo cual se logra obtener una descripción física satisfactoria para este tipo de materiales.

El método estocástico por excelencia es el *Método de Monte Carlo (MMC)*; sin embargo, este presenta un gran número de variaciones entre las cuales se destaca el *Algoritmo de Metropolis*, el cual será empleado en este trabajo.

MARCO TEÓRICO

El modelo que presenta una primera aproximación para describir el comportamiento de los materiales ante el magnetismo, considera que el material esta compuesto por una red

FS-5221. Marzo, 2016. M. Morgado.

de partículas que poseen *espín* (momento angular intrínseco). Estos espines contribuyen a la magnetización del material, lo cual define el tipo de material magnético.

Los materiales magnéticos más relevantes son: los paramagnetos, diamagnetos, ferromagnetos y los anti-ferromagnetos. Los cuales sin presencia de campo electro magnetico presentan una distribución de espines siguientes:



Si se considera que existe un campo magnético no nulo, entonces:



Modelo de Ising.

Ising desarrolló un modelo el cual permite estudiar los materiales ferromagnéticos, con la finalidad de demostrar que no existían transiciones de fase (cambios de estado) en sistemas de espines de 1 dimensión. El modelo consiste en considerar N^2 partículas con espín S , en una red $N \times N$. Donde los valores de los espines pueden ser $\{-1, 1\}$ (*spin down*, *spin up*), de modo que la energía puede estar representada por el valor del Hamiltoniano:

$$\mathcal{H} = -\mathcal{J} \sum_{\langle i,j \rangle} S_i \cdot S_j - H \sum_{i=1}^N S_i, \quad (1)$$

donde \mathcal{J} corresponde a la constante de acoplamiento entre espines ($\mathcal{J} > 0$ para ferromagnetos, $\mathcal{J} < 0$ para anti-ferromagnetos y $\mathcal{J} = 0$ si los espines no interactúan), la sumatoria $\sum_{\langle i,j \rangle}$ corresponde a una suma en la vecindad del i -ésimo espín (visto en la red, es entre los primeros vecinos

del (i, j) -ésimo espín), H corresponde al valor del campo magnético externo [1].

De este modo, la función de partición del sistema puede ser escrita como:

$$\mathcal{Z} = \sum_{\{S_i\}} e^{-\beta \mathcal{H}\{S_i\}} = \sum_{\{S_i\}} \prod_{\langle i, j \rangle} e^{-\mathcal{J} S_i \cdot S_j / k_B T} . \quad (2)$$

Es posible escribir la función de partición en términos de la energía del μ -ésimo microestado (E_μ), así como escribir la probabilidad de la partícula de estar en esta energía:

$$P_\mu = \frac{1}{\mathcal{Z}} e^{-\beta E_\mu} , \quad (3)$$

donde $\beta = 1/k_B T$.

Variables Físicas.

Con la expresión de la función de partición, es posible obtener las expresiones de distintas variables física interesantes al momento de hacer el estudio de estos materiales, como hemos mencionado antes, la magnetización por espín (m), la magnetización total (\mathcal{M}), el calor específico (C_v) y la susceptibilidad magnética (χ_o) son variables sencillas de obtener a partir de este modelo empleando el formalismo de la *Mecánica Estadística* [2].

La *Magnetización Total* viene dada por:

$$\begin{aligned} \mathcal{M} &= K_B T \frac{\partial \ln \mathcal{Z}}{\partial H} = \left\langle \sum_{i=1}^N S_i \right\rangle \\ &= \sum_{\mu} P_{\mu} \sum_{i=1}^N S_i(\mu) = \frac{1}{N} \sum_{i=1}^N S_i , \end{aligned} \quad (4)$$

y donde la solución exacta del modelo permite tener la siguiente expresión para la *magnetización total*:

$$\mathcal{M} = [1 - \sinh^{-4}(2\beta \mathcal{J})]^{1/8} ,$$

la *Magnetización por Espín* viene dada por:

$$m = \frac{\mathcal{M}}{N} , \quad (5)$$

la *Susceptibilidad Magnética*, que nos dice cuanto cambia un parámetro extensivo cuando un parámetro intensivo aumenta. Esto es, cuanto cambia la susceptibilidad, en función de la temperatura y viene dada por:

$$\chi_o = \frac{1}{N} \frac{\partial \langle \mathcal{M} \rangle}{\partial H} = \frac{\beta}{N} \left(\langle \mathcal{M}^2 \rangle - \langle \mathcal{M} \rangle^2 \right) , \quad (6)$$

y el *Calor Específico*, que nos dice cuanto cambia la energía con el aumento de temperatura, esta determinado por:

$$C_v = \frac{1}{N} \frac{\partial \langle E \rangle}{\partial T} = \frac{K_B \beta^2}{N} \left(\langle E^2 \rangle - \langle E \rangle^2 \right) . \quad (7)$$

Es de esperar que exista una divergencia de esta variable en la *Etapa de Transición*. En esta etapa de transición queda establecida una *Temperatura Crítica*¹ (T_c) a partir de la cual, el valor de la magnetización puede ser distinta de 0, debido a la variación aleatoria de la magnetización entre los valores -1, 0 y 1. Y su expresión es:

$$T_c = \frac{\mathcal{Z} \mathcal{J}}{2 K_B} , \quad (8)$$

que es obtenida a partir de la relación de dualidad de Kramers–Wannier [3]. El valor aproximado en la literatura es $T_c \approx 2,2693K$.

Números Aleatorios.

Un número aleatorio es un número escogido como si se obtuviera de una distribución específica, tal que una larga selección de estos números reproduce dicha distribución. Generalmente se requiere que estos números presenten una independencia entre sí, de forma que no pueda establecerse una correlación entre dos de estos números sucesivos [4].

Los números generados aleatoriamente por computadora son llamados algunas veces como números “*pseudoaleatorios*”, mientras que el término “*aleatorio*” se emplea para aquellos que surgen de procesos físicamente impredecibles.

En este trabajo se emplea la palabra “*aleatorio*” para referirnos de números que provienen de una “*distribución aleatoria uniforme*”, la cual consiste en que todos los números de un intervalo tiene una misma probabilidad de aparecer.

Método de MC. (Algoritmo de Metrópolis).

El método de **MC** es ampliamente utilizado para resolver numéricamente distintos modelos de sistemas físicos (tanto estocásticos como deterministas) de gran complejidad matemática de manera satisfactoria. Empleando para ello, grupos de números *pseudoaleatorios* generados por computadora.

Existe un gran abanico de variaciones al método de **MC** original; sin embargo, el *Algoritmo de Metrópolis* resulta ser bastante sencillo, ya que a partir de una variable estocástica se puede realizar la simulación [5].

¹Las unidades de temperatura serán expuestas a lo largo del trabajo en grados Kelvin (K), sin embargo, sus unidades correctas son de $[J/K_B]$.

Este algoritmo puede ser resumido en los siguientes pasos:

1. Se escoge aleatoriamente un estado del sistema (ε).
2. Se calcula la diferencia de energía (ΔE) de dicho estado, por medio de una perturbación.
3. Se compara ($\Delta E \leq 0$). Si cumple, se asigna nuevo estado (ε^*), sino, acepta dicho estado con probabilidad $p = e^{\beta \Delta E}$.

Teniendo en cuenta esto, presentaremos a continuación como este algoritmo se aplica al *Modelo de Ising*.

Primero, imaginemos que tenemos una red de espines con valores aleatorios de *up* y *down*. Luego, escogemos un espín de ubicación aleatoria en la red ($S_{(i,j)}$), al cual calculamos la diferencia de energía (ΔE) por medio de (1), esto es la energía debida a la interacción con los vecinos cercanos y el campo magnético externo (el cual se considerará cero en parte del trabajo):

$$\Delta E = -J \cdot S_{(i,j)} \cdot \{S_{(i,j-1)} + S_{(i,j+1)} + S_{(i-1,j)} + S_{(i+1,j)}\} - H \cdot S_{(i,j)} \quad (9)$$

Esto se puede apreciar en la siguiente figura:

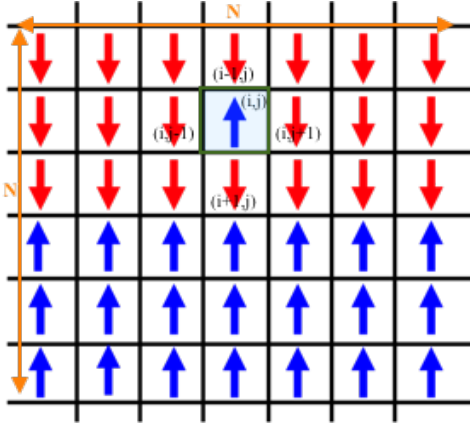


Figura 1: Esquema de red de espines, con 1ros vecinos. Las flechas rojas indican espines *Down* y las azules espines *Up*.

En donde se tienen condiciones de borde, los cuales pueden ser periódicas o con un pared. Las primeras consiste en que la red se encuentra en un toroide (topológicamente hablando), mientras que la condición de pared considera que los elementos de los borde tienen 3 vecinos, a excepción de las esquinas, quienes solo tienen 2.

Por último, se realiza la etapa fundamental del algoritmo. Debe revisarse la condición sobre $\Delta E \leq 0$. Si la cumple: $S_{(i,j)} = -S_{(i,j)}$, sino entonces, $S_{(i,j)} = -S_{(i,j)}$ cuando un número *pseudoaleatorio* '*r*' cumpla que: $r \leq p = e^{\beta \Delta E}$.

Entonces es aquí donde se puede reducir el tiempo de simulación del **MC**, sacrificando la precisión, tomando en cuenta un número pequeño de decimales de ΔE , *r*, y *p*.

PROCEDIMIENTO EXPERIMENTAL

Para llevar a cabo este trabajo se realizó una simulación del sistema por medio de una computadora empleando el intérprete de *Python*. En dicha simulación se obtuvieron las distintas variables físicas, para distintos valores del parámetro de temperatura. A continuación veremos un *pseudocódigo* del ejecutado en *Python* para simular este modelo. El *pseudocódigo* se puede observar en el esquema del **Algoritmo 1**

Input: Red, Tamaño, Temperatura, # Iteraciones
Output: Red, $M(T)$, $E(T)$, $C_v(T)$, $\chi_o(T)$, $Corr(d)$

```

1 begin
2   -Calcula matriz de Red
3   -Inicia constantes y variables [J, B, Kb, dT, ite, T, E, M, Cv, Xo]
4   for  $T \in [0, 4]K$  do
      // Loop que itera sobre distintos valores de la Temperatura.
5     for #iteraciones  $\in [0, 2000]$  do
      // Realiza 2000 iteraciones.
6       for  $(i,j) \in [(0,0), (N,N)]$  do
          // Itera sobre los  $N^2$  elementos de la matriz, de manera aleatoria.
7         -Pide una posición aleatoria dentro de red  $(i,j)$ .
8         -Calcula el valor del  $\Delta E$  para el espín  $S_{(i,j)}$ .
          // Aplica Algoritmo de Metrópolis
9         if  $\Delta E \leq 0$  then
10          -Cambia estado:  $S_{(i,j)} = -S_{(i,j)}$ 
11          else if  $e^{\beta \Delta E} > \#Aleatorio$  then
12            -Cambia estado:  $S_{(i,j)} = -S_{(i,j)}$ 
13          end
14        end
15      end
16    end
17  end
18  -Calcula todas las variables físicas.
19 end

```

Algoritmo 1: *Pseudocódigo* del Modelo de Ising

En el código de *Python* desarrollado (**Anexo 1**), se emplearon varias librerías:

- *numpy* y *math*

Estas librerías es usada para acceder a operaciones matemáticas más generales que las de *Python* (i.e generación de matrices y vectores, generación de números aleatorios) y algunas prestaciones para graficar [*numpy.meshgrid()*].

- *matplotlib.pyplot*

Esta librería se emplea para graficar, tanto la simulación a tiempo real, por uso de [*imshow()*, *cla()*, *etc*] y comandos comunes de gráficas [*plot()*]. Un par de librerías extras usadas fueron *mpl_toolkits.mplot3d* para obtener un gráfico en 3D y *matplotlib.ticker* para emplear una notación científica en los ejes.

- *numpngw*

Es un librería que permite generar archivos tipo imagen con secuencia.

- *random*

Esta librería es realmente útil para la generación de números aleatorios, los cuales son necesarios al momento de obtener una matriz aleatoria de espines y de seleccionar una posición aleatoria en ella.

- *time* y *sys*

Este par de librerías si bien no son necesarias para la simulación del modelo, es útil al momento de crear una barra de progreso que permita tener una idea del estatus de la corrida de la simulación.

- *datetime*

Esta es otra librería que podría considerarse sin utilidad para la simulación; sin embargo, es útil para cuantificar el tiempo de computo empleado para llevar a cabo la simulación completa.

Y se crearon distintas funciones que serán brevemente explicadas a continuación:

- *iAle()* :

Esta función fue construida con la finalidad de hacer un estudio de la uniformidad de la distribución de los números generados “aleatoriamente” por *Python*.

- *iRed(L, tipo)*

Esta función define la red de espines de tamaño L . Según una condición de $\mathcal{J}/K_B T \rightarrow 0$ entonces, los valores de espines iniciales es aleatorio. Mientras que si $\mathcal{J}/K_B T \rightarrow \infty$ los valores de espines iniciales son todos 1.

- *iCb(k, L, tipo = 1)*

Esta función define el tipo de condición de borde sobre la coordenada “ k ”, según el tipo de condición que se introduzca (tipo = 1 para condición de borde periódico y tipo = 2 para condición de borde de pared).

- *idE(red, L, i, j)*

Esta función define la variación de energía del espín en la posición (i, j) de la red. Esta función aplica la ecuación (8).

- *iMet(red, T, L, m, n)*

Esta función define el *Algoritmo de Metrópolis*, como fue descrito en la sección anterior para el *Modelo de Ising*. Para ello, requiere de entradas a la matriz que define la red, la temperatura y la posición (i, j) del espín.

- *iSing(red, L, T, ite)*

Esta función define la iteración del *Algoritmo de Metrópolis* para los N^2 elementos de la matriz, con un número determinado de iteraciones. Requiere de entradas a: la red, el tamaño de la red (L), la temperatura y el número de iteraciones.

En el código del anexo también se encuentran unas variaciones, que prácticamente realizan la misma tarea, sin embargo, esta construida para cumplir con ciertas tareas: *iSingVIVO(red, L, T, ite)*, se encarga de mostrar la simulación en vivo y *iSingSimple(red, L, T, ite)*, para realizar el estudio de la magnetización por espín.

- *iMag(red)*

Función que calcula el valor total de los aportes de los espines (Magnetización).

- *cor(red, L)*

Función que calcula la correlación de los espines en una red tamaño L , en función de las distancias relativas entre ellos.

- *iERed(red, L)*

Esta función calcula el valor de la energía en la red completa de tamaño L .

- *iCv(red)*

Esta función calcula el calor específico del sistema.

- $plotmS(mS)$, $ploteS(eS)$, $plotMT(MT, T)$, $plotXT(XT, T)$ y $plotCor(Corr, d)$

Estas son funciones para llevar a cabo la construcción de gráficas de cada una de las distintas variables físicas del sistema.

Con la simulación a computadora, se llevaron a cabo tareas:

1. Verificar la uniformidad de la distribución de los números aleatorios.
2. Obtener gráficamente la simulación del sistema, usando el *Modelo de Ising*, considerando ciertas condiciones de borde (Periódicas y con pared).
3. De la simulación, graficar la variable de la *Magnetización por Espín* del sistema ($m(t)$), para 4 valores interesantes de la temperatura.
4. Para 5 valores del tamaño de la red ($L \in \{5, 10, 20, 30, 40, 50\}$), obtener las gráficas de la *Magnetización* como función de la temperatura ($M(T)$) y la *Susceptibilidad Magnética* en función de la temperatura ($X_o(T)$), cuando el sistema haya alcanzado el equilibrio. Obtener la temperatura crítica.
5. Encontrar la *función de correlación* entre espines:

$$\langle S_i S_j \rangle - \langle S_i \rangle \langle S_j \rangle \quad (10)$$

en función de la distancia, de modo que se obtenga una idea del nivel de interacción o el efecto de un cambio de espín (*spin-flip*), sobre el resto (que tan correlacionados está un espín con otro que se encuentre a una distancia d). De aquí surgen un par de conceptos adicionales, que serán explicados en la sección de discusión de resultados.

RESULTADOS

Primero buscamos observar la uniformidad de la distribución de números aleatorios en *Python*. Para ello realizamos un histograma en dos dimensiones donde se muestran las probabilidades cruzadas de dos conjuntos de datos aleatorios (*pseudoaleatorios*) para un total de 1 millón de números.

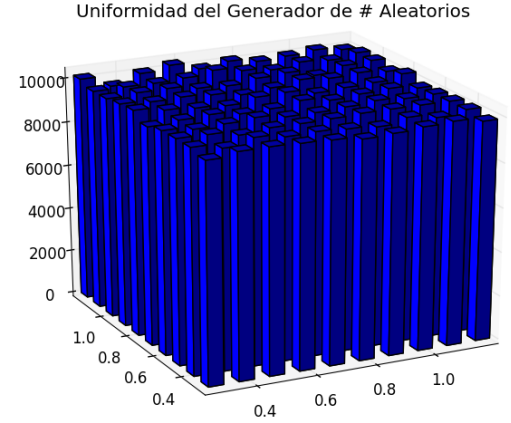


Figura 2: Distribución Uniforme de 1 millón de números *pseudoaleatorios* en *Python*.

Como segunda tarea, se realizó por simulación el *Modelo de Ising*, creando una animación que se podrá encontrar en el **Anexo 1** (adjunto a este trabajo), donde se observa la evolución del sistema en el tiempo. Para una red de 30×30 espines iniciados con valor *aleatorio*, a **temperatura crítica** ($2,26K$), sin campo magnético, con $\mathcal{J} = 1$ y *condiciones de borde periódicas*.

Evolucionando del estado:

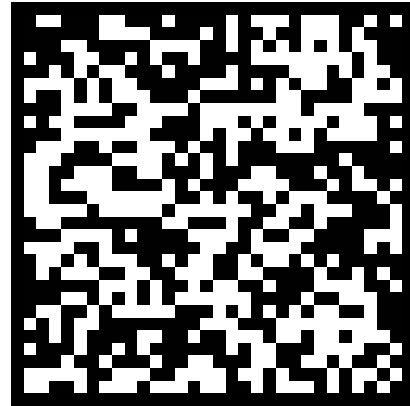


Figura 3: Estado inicial de la red de espines en simulación.

al estado:

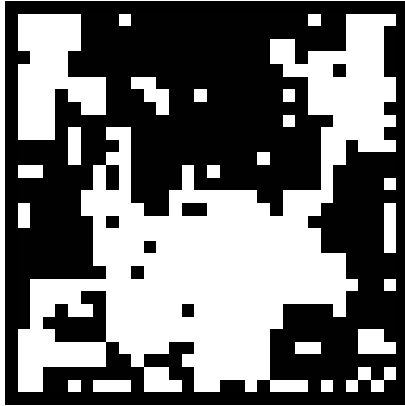


Figura 4: Estado final de la red de espines en simulación, después de 1.8 millones de iteraciones.

y con condiciones de borde de pared para una red con 150×150 , del estado:

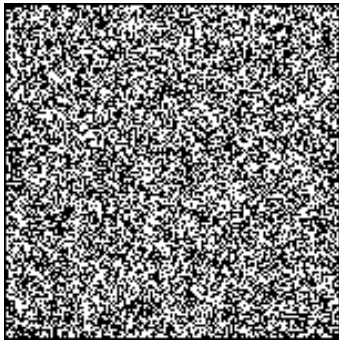


Figura 5: Estado inicial de la red de espines en simulación.

al estado:

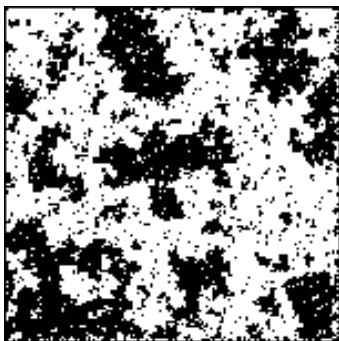


Figura 6: Estado final de la red de espines en simulación, después de 45 millones de iteraciones.

Como tercera tarea, se obtuvo la gráfica de la *Magnetización por Espín* en función del tiempo ($m(t)$) para 4 valores

de la temperatura ($T \in \{1,45, 2,26, 3,45, 5,00\}K$). Para una red de tamaño 20×20 , con condiciones periódicas, sin campo magnético, inicializada con espines 1 y $J = 1$.

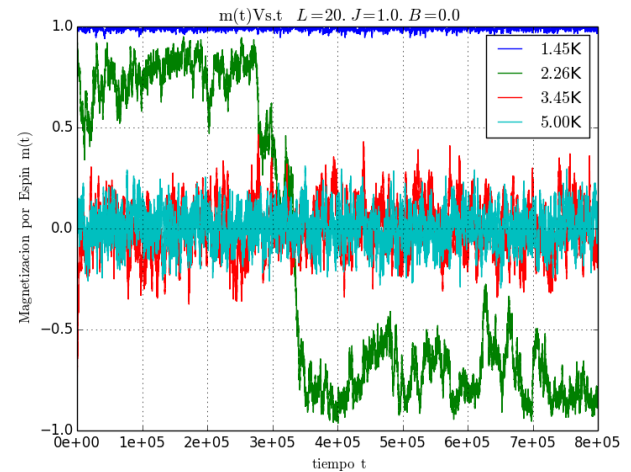


Figura 7: *Magnetización por Espín* en función del tiempo ($m(t)$), para diferentes temperaturas (T).

Adicionalmente, en esta parte se simuló para todos los valores de L , con la finalidad de obtener una relación entre el tamaño de la red y el tiempo promedio de cómputo.

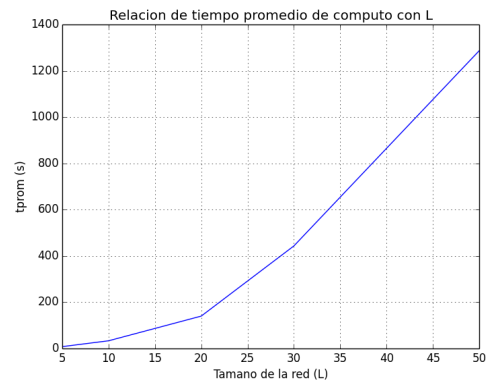


Figura 8: *Tiempo promedio de cómputo Vs Tamaño de Red.*

La cuarta parte consistió en obtener los valores de distintas variables físicas en función de la temperatura, para distintos valores del tamaño de la red, luego que el sistema alcanzase el equilibrio.

Donde se obtuvo el siguiente gráfico de la magnetización para 40 temperaturas con condiciones de borde:

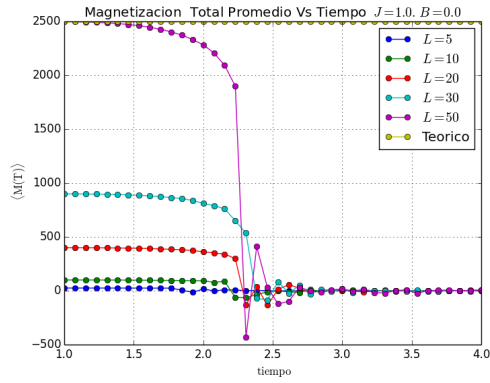


Figura 9: Magnetización Total Promedio en función de la temperatura, para distintos valores de L . (Aprox. 4 horas de tiempo de computo)

Del mismo modo se procedió para obtener la gráfica de la Susceptibilidad Magnética:

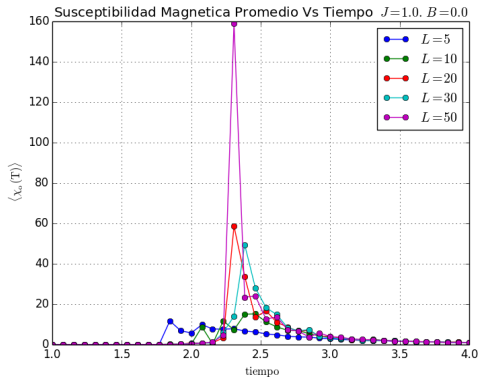


Figura 10: Susceptibilidad Magnética Promedio en función de la temperatura, para distintos valores de L . (Aprox. 4 horas de tiempo de computo)

En la quinta parte graficamos la correlación entre espines de la red usando (9), en función de la distancia que los separa (ubicación en la red). Como se observa en la siguiente figura.

Nota: el programa al calcular las correlaciones daban malos resultados, desconocemos su razón al revisar varias veces el código no fue detectado, se presume una mala toma de datos. Pero hemos colocas un gráfico de referencia.

DISCUSIÓN Y CONCLUSIONES

En la [figura 2](#) podemos darnos cuenta que la distribución de números *pseudoaleatorios* muestra un comportamiento uniforme, ya que si bien existen diferencias entre las barras de uno u otro número, no representan una diferencia

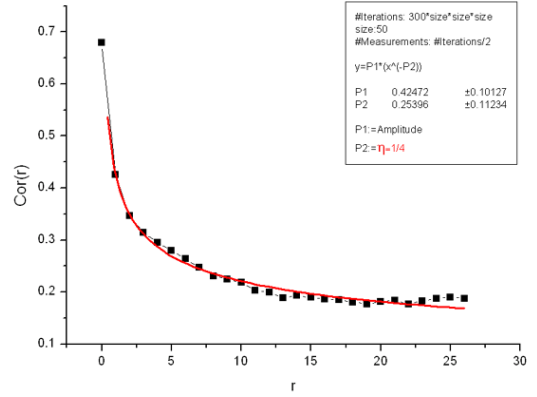


Figura 11: Función de Correlación de espines para un tamaño de red y temperatura dados.

considerable en comparación con el número total de *bins* de las barras.

En la segunda parte, cuando realizamos la simulación ([figura 3 y 4](#)) para todos los valores de " L " (tamaño de la red), observamos que entre mayor sea L , más representativo a la realidad aparenta ser. Adicionalmente, no existió diferencia (ni en lo que refiere a la física de la simulación, ni en el aspecto computacional) entre las condiciones de borde periódicas y con la pared, a pesar de ser llevado a cabo con 2 tipos procesadores (2 núcleos y 4 núcleos).

Se realizó un par de animaciones adicionales que pueden ser encontradas en el **Anexo 1**, con tamaño de red 150×150 que muestran lo real de este modelo (Por ejemplo: [figura 5 y 6](#)).

En tercer lugar, podemos observar en la [figura 7](#) que para bajas temperaturas se obtiene un resultado esperado, que consiste en que todo los espines se encuentren alineados (consideramos un material Ferromagnético, $\mathcal{J} > 0$). Por otro lado para valores

Puede observarse claramente en la gráfica ([figura 9](#)) que para un valor mayor a T_c la magnetización espontánea es nula, ya que las fluctuaciones térmicas eliminan cualquier posibilidad de que una vecindad consiga alinearse, mostrando de ese modo un comportamiento *paramagnético*. Por el contrario, para temperaturas menores de T_c , si existe tal magnetización espontánea. Claramente, el efecto total es dependiente del tamaño de la red.

Se puede observar en la ([figura 11](#)), que por debajo de T_c la susceptibilidad tiende a infinito, a medida que se acerca a la temperatura de transición (en este punto no existe un cambio en el valor de la magnetización). Luego de pasar dicha etapa, la susceptibilidad decae a cero. Esto es una demostración de la existencia de un estado de transición existente en el sistema, independientemente de su tamaño.

En la (figura ??), se puede observar que la correlación entre espines consiste en una función que decae en función de la distancia. Esto es porque a temperaturas entre 0 y T_c el material se comporta con magnetización 1 o -1; sin embargo, luego de pasar por T_c , puede continuar magnetizado de manera local, esto es gracias a la aparición de grupos de espines (*Clusters*), los cuales se hacen más pequeños a medida que aumenta la temperatura. El tamaño de estos *clusters* es lo que se conoce por el nombre de *Longitud de Correlación*, la cual tiene un punto de divergencia en T_c .

Finalmente, considerar un error en estas medidas, puede tratarse de manera global considerando el número de iteraciones que se llevan a cabo, en el proceso, es decir $\#ite = 2000 * L * L$ donde L es el tamaño de la red. Y si consideramos el error (Δ):

$$\Delta = \pm \frac{\sigma}{\sqrt{\#ite}}, \quad (11)$$

entonces, $\Delta \approx \pm 10^{-4}$, lo cual es totalmente aceptable.

RECOMENDACIONES

Entre las recomendaciones más relevantes observadas:

- Aumentar a un número mayor de iteraciones (2000), con la finalidad de disminuir el error de los valores de las variables, ya que el error del método de **MC** va como $(1/\sqrt{\#iteraciones})$.
- Aumentar el tamaño de la red permite tener una simulación más real de un sistema de espines.
- Aumentar cantidad de valores considerados para la Temperatura, de manera de obtener mayor cantidad de datos a graficar y tener resultados más precisos.
- Intentar otros algoritmos como el de *Wolff*.
- Hacer un estudio que falte en este trabajo sobre la *Energía* y el *Calor Específico*.

REFERENCIAS

Los libros y publicaciones consultadas por orden de aparición son las siguientes:

[1] H. Gould, J. Tobochnik, and W. Christian. *An Introduction to Computer Simulation Methods*. Third Edition.

[2] D. P. Landau and K. Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*.

[3] Wikipedia. [https://en.wikipedia.org/wiki/Kramers\OT1\textendashWannier_duality](https://en.wikipedia.org/wiki/Kramers%20T%20textendashWannier_duality)

[4] Weisstein and Eric W. *Random Number*. From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/RandomNumber.html>

[5] M. Wallin. *Monte Carlo simulation in statistical physics*. Theoretical Physics, KTH, SE-100 44 Stockholm,

Sweden. (September, 2005)

[6] M. Medo and Y. Zhang. *Simulations for the Ising model (and more) Numerical Methods for Physicists, Lecture 6*. Physics Department, Fribourg University, Switzerland. (March, 2013).

[7] L. Witthauer and M. Dieterle <http://quantumtheory.physik.unibas.ch/people/bruder/Semesterprojekte2007/p1/index.html>

ANEXOS

A continuación presentamos los anexos al informe:

- Anexo 1: Carpeta organizada de gráficas relevantes.
- Anexo 2: Código en python de la simulación.