

Modelo de Ising 2D: Explorando la paralelización

Santiago Ruiz Piedrahita*

*Instituto de Física, Facultad de Ciencias Exactas y Naturales,
Universidad de Antioquia UdeA, Calle 70 No. 52-21, Medellín, Colombia.*

(Dated: May 2, 2025)

El modelo de Ising es una representación fundamental en física estadística para describir sistemas magnéticos y transiciones de fase. En este trabajo, se implementaron simulaciones de Monte Carlo para estudiar su comportamiento en redes bidimensionales, en ausencia de campo magnético externo. Se analizaron la evolución de la energía, la magnetización, la capacidad calorífica y la susceptibilidad magnética, evidenciando la presencia de una transición de fase en la temperatura crítica predicha teóricamente. Además, se evaluó el rendimiento computacional del algoritmo mediante implementaciones en serie y paralelas utilizando la biblioteca `multiprocessing` de Python. Los tiempos de ejecución fueron comparados en dos entornos: Google Colab y un servidor local. El estudio demostró que la paralelización mejora significativamente la eficiencia del cómputo, siendo el servidor local el entorno más adecuado para simulaciones de gran escala.

Palabras clave: Modelo de Ising, Método de Montecarlo, Paralelización.

I. INTRODUCCIÓN

Los modelos de sistemas con variables interactivas han sido fundamentales para comprender fenómenos físicos complejos. Uno de los más estudiados en física estadística es el modelo de Ising, propuesto originalmente por Wilhelm Lenz y desarrollado por su estudiante Ernst Ising [1]. Este modelo fue concebido para describir la transición de fase entre un estado paramagnético y uno ferromagnético.

Aunque su versión unidimensional no presenta transición de fase, investigaciones posteriores demostraron que el modelo de Ising en dos o más dimensiones exhibe un comportamiento crítico bien definido [2, 3]. Gracias a estas propiedades, el modelo se ha convertido en una herramienta central en el estudio de la termodinámica de sistemas magnéticos, y también se ha extendido a campos como la mecánica estadística y la teoría de redes [4].

Dado que no existe una solución analítica para el modelo en dimensiones mayores a uno, su análisis requiere de métodos numéricos. Los algoritmos de Monte Carlo, especialmente el algoritmo de Metrópolis, han demostrado ser efectivos para explorar las propiedades del sistema a distintas temperaturas [4, 5].

El modelo de Ising en dos dimensiones se estudia mediante simulaciones de Monte Carlo para analizar propiedades como energía, magnetización y susceptibilidad. Dado el alto costo computacional de estas simulaciones, se implementan versiones en serie y en paralelo, utilizando la biblioteca `multiprocessing`. Se comparan los tiempos de ejecución en Google Colab y en un servidor local, permitiendo evaluar la eficiencia y las condiciones

óptimas de uso de cada entorno para el estudio de sistemas físicos complejos.

II. MODELO DE ISING

El modelo de Ising es uno de los sistemas fundamentales en física estadística para el estudio de transiciones de fase. Consiste en una red de espines clásicos $\sigma_i = \pm 1$ dispuestos sobre una retícula de dimensión finita. Cada espín interactúa únicamente con sus vecinos más cercanos y puede estar influenciado por un campo magnético externo. La energía total del sistema está descrita por el Hamiltoniano:

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j - h \sum_i \sigma_i, \quad (1)$$

donde J es la constante de acoplamiento (positiva para interacciones ferromagnéticas), h representa el campo magnético externo, y la suma $\langle i,j \rangle$ se realiza sobre todos los pares de espines vecinos en la red.

El sistema evoluciona bajo las reglas de la mecánica estadística, y la probabilidad de que se encuentre en una configuración de espines $S = \{\sigma_1, \sigma_2, \dots\}$ está dada por la distribución de Boltzmann:

$$P(S) = \frac{1}{Z} e^{-\mathcal{H}(S)/k_B T}, \quad (2)$$

donde T es la temperatura, k_B la constante de Boltzmann, y Z la función de partición, definida como

* santiago.ruizp@udea.edu.co

$$Z = \sum_S e^{-\mathcal{H}(S)/k_B T}, \quad (3)$$

con la suma extendida a todas las posibles configuraciones del sistema. En general, esta suma no puede calcularse de forma exacta para redes bidimensionales grandes, por lo que es necesario recurrir a métodos de simulación numérica.

III. ALGORITMO DE METROPOLIS

Para simular el modelo de Ising y estudiar su comportamiento térmico, se emplea el método de Monte Carlo con el algoritmo de Metropolis [6]. Este método permite generar configuraciones de espines con una probabilidad que respeta la distribución de Boltzmann, asegurando así el muestreo correcto del sistema en equilibrio.

El algoritmo parte de una configuración inicial de espines (aleatoria o alineada) y realiza iteraciones en las que propone el cambio de un espín seleccionado al azar. La decisión de aceptar o rechazar dicho cambio se basa en la variación de energía ΔE que implica la inversión del espín:

$$\Delta E = 2J\sigma_i \left(\sum_{\langle j \rangle} \sigma_j + h \right), \quad (4)$$

donde la suma se realiza sobre los espines vecinos del sitio i . Si $\Delta E < 0$, el cambio se acepta automáticamente. En caso contrario, se acepta con probabilidad:

$$P = e^{-\Delta E/k_B T}. \quad (5)$$

Este proceso se repite de forma iterativa sobre todos los sitios de la red y durante múltiples ciclos, hasta que el sistema alcanza el equilibrio térmico. A partir de allí, se recolectan datos de observables físicos mediante promedios temporales sobre configuraciones sucesivas.

IV. ENTORNO DE COMPUTACIÓN Y ESTRATEGIAS DE PARALELIZACIÓN

El desarrollo del proyecto se estructuró en tres fases principales, combinando simulaciones físicas con análisis de rendimiento computacional:

Fase 1: Implementación del modelo de Ising 2D

Se implementó una clase `IsingModel2D` que representa una red cuadrada de espines con interacciones de primer vecino. La simulación considera condiciones de frontera periódicas, sistemas ferromagnéticos ($J = 1$) y ausencia de campo magnético externo ($h = 0$). La configuración inicial es ordenada (todos los espines alineados).

Para validar el modelo, se realizaron pruebas a temperatura fija con $L = 20$, verificando la evolución de la energía y la magnetización. Posteriormente, se exploró la dependencia del sistema con el tamaño de la red y se estudió el cambio de fase térmica variando la temperatura en redes de distinto tamaño.

Fase 2: Computación en paralelo

Se incorporó paralelización utilizando la biblioteca `multiprocessing` de Python. Para paralelizar, se asignan diferentes temperaturas a los distintos núcleos, permitiendo que cada núcleo realice simulaciones independientes para cada temperatura de manera simultánea.

La paralelización fue validada asegurando que los resultados físicos coincidieran con los obtenidos en la versión secuencial. Luego, se repitió el estudio del cambio de fase con esta implementación paralela, manteniendo la estructura de análisis por temperatura y tamaño del sistema.

Fase 3: Análisis de tiempos de cómputo

Finalmente, se realizó un estudio cuantitativo de los tiempos de ejecución. Se evaluaron dos escenarios computacionales:

- **Google Colab:** Entorno en la nube con acceso limitado a recursos computacionales compartidos.
- **Servidor local:** Sistema con múltiples núcleos y mayor capacidad de procesamiento.

En cada entorno, se compararon los tiempos de ejecución entre la versión secuencial y la paralela del código. Las simulaciones se realizaron para diferentes tamaños de red ($L \in [20, 40, \dots, 100]$), permitiendo observar el impacto del paralelismo en función de la escala del sistema y el entorno de cómputo. Este análisis permite evaluar la eficiencia computacional y la escalabilidad de las estrategias implementadas.

V. RESULTADOS Y ANÁLISIS

Como primer paso, se probó la implementación del modelo para una red pequeña de tamaño $L = 20$, manteniendo la temperatura constante y observando la evolución temporal de la energía y la magnetización por espín. El objetivo fue verificar que el sistema evoluciona correctamente hacia un equilibrio térmico, y que los resultados son consistentes con lo esperado físicamente. Estos resultados sirvieron como validación preliminar de la implementación antes de extender el análisis a sistemas más grandes.

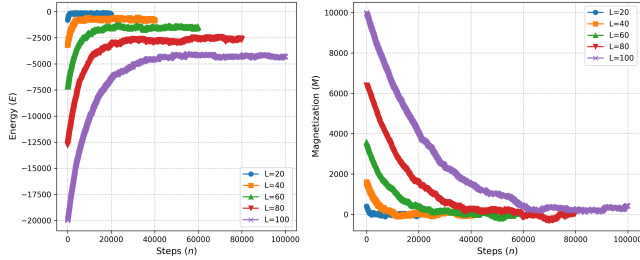


FIG. 1. Convergencia de energía y magnetización promedio por espín para distintos tamaños de red L a temperatura fija.

Una vez validado el modelo, se estudió la evolución de la energía y la magnetización promedio por espín para diferentes tamaños de red: $L = 20, 40, 60, 80, 100$, a una temperatura fija. El objetivo fue estimar cuántos pasos Monte Carlo son necesarios para alcanzar el equilibrio térmico según el tamaño del sistema.

Como se observa en la Fig. 1, a medida que L aumenta, el sistema requiere un mayor número de pasos para estabilizarse. Esta observación permite ajustar el número de pasos utilizados en las simulaciones posteriores, asegurando que los promedios se calculen en régimen estacionario.

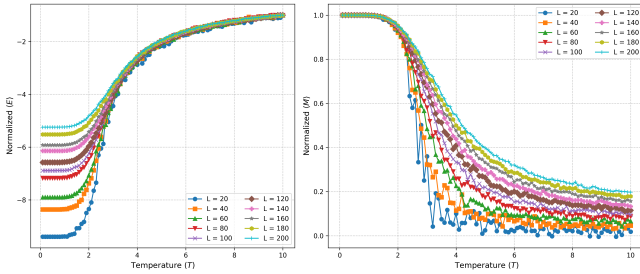


FIG. 2. Evolución de la energía y la magnetización en función de la temperatura para diferentes tamaños de red. Los resultados se muestran para una simulación paralelizada en el servidor utilizando 100 valores de temperatura y L de 20 hasta 200 en incrementos de 20.

La Fig. 2 muestra la evolución de la energía y la magnetización en función de la temperatura. En la gráfica se

presentan los resultados paralelizados para 100 valores de temperatura y tamaños de red L que van desde 20 hasta 200, con incrementos de 20. Esta gráfica es la más general, ya que proporciona la mayor cantidad de información, y se muestra tanto para la computación en serie como en paralelo, en Google Colab y en el servidor local, ya que los resultados obtenidos son idénticos en todos los casos.

El análisis de los resultados revela el cambio de fase del sistema al observar el comportamiento de la energía y la magnetización a medida que se varía la temperatura. Este procedimiento se repitió para diferentes valores de L , observándose que el sistema presenta un cambio de fase a una temperatura crítica $T_c = 2.279$, tal como lo predice el modelo de Onsager.

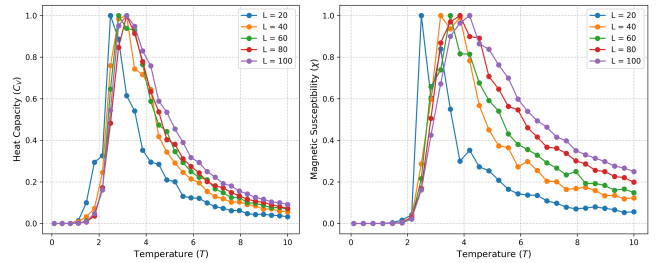


FIG. 3. Capacidad calorífica y susceptibilidad magnética en función de la temperatura para el sistema de computación en serie en Google Colab.

De manera similar, en la Fig. 3 se muestran las gráficas de la capacidad calorífica y la susceptibilidad magnética para el sistema de computación en serie en Google Colab. En estas gráficas, se observa cómo ambas magnitudes responden al cambio de temperatura. Al igual que en el análisis previo, se identifica el valor de la temperatura crítica $T_c = 2.279$, que corresponde al punto de transición de fase, como lo predice el modelo de Onsager.

Estas magnitudes son cruciales para el estudio de las transiciones de fase, ya que la capacidad calorífica muestra un pico en T_c , indicando un cambio en la estructura termodinámica del sistema, mientras que la susceptibilidad magnética también muestra un aumento en esta temperatura, reflejando una mayor respuesta del sistema a pequeños cambios en el campo magnético cerca de T_c .

Cabe destacar que se obtienen los mismos resultados tanto utilizando Google Colab como el servidor, y tanto en el caso de computación en serie como en paralelización. Esta consistencia en los resultados es importante y se analizará más a fondo en términos de los tiempos de cómputo en las siguientes secciones.

Para la Fig. 4 se muestra el tiempo de cómputo en función de L para diferentes entornos y estrategias de cómputo. En la gráfica se incluyen cuatro líneas, representando los tiempos de ejecución para computación

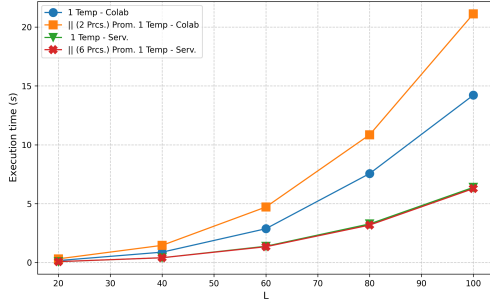


FIG. 4. Tiempo de cómputo en función de L para una sola temperatura en los entornos de Google Colab y el servidor. Se comparan los tiempos de cómputo en serie y en paralelo.

en serie y paralelizada en Google Colab y en el servidor. Para la computación en serie, se calcula el tiempo requerido para la evolución de una sola temperatura, con Google Colab mostrando un tiempo significativamente mayor a medida que L aumenta, especialmente para $L = 100$, comparado con el servidor. En el caso de la paralelización, para el servidor se obtiene un tiempo de cálculo igual que para la computación en serie, mientras que para Google Colab, se observa que el tiempo de ejecución en paralelo aumenta de manera pronunciada, desde aproximadamente 14 segundos hasta 22 segundos para los tamaños de red más grandes. Es importante destacar que este análisis se realiza considerando una sola temperatura.

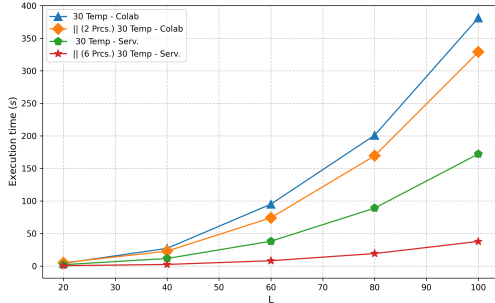


FIG. 5. Tiempo de cómputo total en función de L para diferentes entornos y estrategias de cómputo. El análisis se realiza para 30 valores de temperatura, mostrando los tiempos de ejecución tanto en serie como en paralelo, con 2 núcleos en Google Colab y 6 núcleos en el servidor.

La Fig. 5 muestra el tiempo de cómputo total en función de L para diferentes entornos y estrategias de cómputo, considerando 30 valores de temperatura. Para la computación en serie, se calcula el tiempo requerido para la evolución de todos los valores de temperatura, sumando los tiempos de ejecución para cada uno de los 30 valores. En el caso de Google Colab, se observa que el tiempo de ejecución aumenta significativamente a me-

da que L crece, especialmente para $L = 100$, debido a las limitaciones gráficas y de procesamiento que presenta Colab. En comparación, el servidor muestra tiempos más bajos tanto en serie como en paralelo, lo que indica que tiene una mayor capacidad de procesamiento y menos restricciones de recursos.

Para la computación paralelizada, se emplean 2 núcleos en Google Colab y 6 núcleos en el servidor. En Google Colab, la mejora al paralelizar es limitada, ya que el tiempo de ejecución solo disminuye en un minuto, y aún así, los tiempos de ejecución pueden llegar a ser de hasta 7 minutos en serie. En contraste, en el servidor, la paralelización con 6 núcleos ofrece una mejora mucho más significativa, reduciendo el tiempo de cómputo de 172 segundos a 37 segundos, lo que representa una reducción de aproximadamente 80% en comparación con la ejecución en serie. Esto da como resultado que el mejor método, sin lugar a dudas, es paralelizar y hacerlo en el servidor, debido principalmente a la ausencia de interfaz gráfica en este entorno, lo que mejora considerablemente la eficiencia computacional.

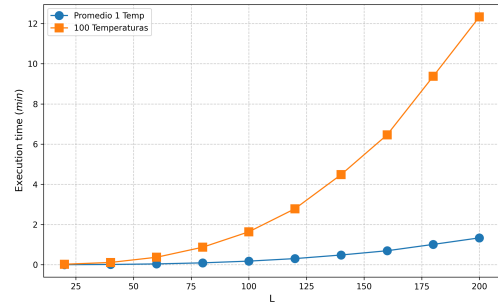


FIG. 6. Tiempos de cómputo total y promedio por temperatura en función de L , utilizando paralelización con 12 núcleos en el servidor para 100 temperaturas, ampliando el sistema hasta $L = 200$.

Finalmente, se realizó un análisis extendido aumentando el tamaño del sistema hasta $L = 200$ y considerando 100 valores de temperatura. Este estudio se llevó a cabo únicamente en el entorno que ofreció el mejor rendimiento: el servidor, utilizando paralelización. Para maximizar la eficiencia, se emplearon los 12 núcleos disponibles.

En la Fig. 6 se presentan los tiempos de cómputo en función de L , tanto para el promedio de tiempo requerido por una sola temperatura como para el tiempo total de las 100 temperaturas paralelizadas. Para $L = 200$, el tiempo por temperatura alcanza aproximadamente 80 segundos, mientras que el tiempo total de la simulación es de alrededor de 12 minutos y medio.

Cabe destacar que el aumento observado en los tiempos de cómputo se debe no solo al incremento en el

TABLE I. Comparación de características de los entornos de computación en Google Colab y el servidor local.

Característica	Google Colab	Servidor Local
Arquitectura	x86_64 (64-bit)	x86_64 (64-bit)
Procesador	Intel Xeon @ 2.20GHz	Intel Core i7-8700 @ 3.20GHz
Núcleos	2 (1 núcleo físico \times 2 hilos)	12 (6 núcleos \times 2 hilos)
RAM Disponible	12 GB (aprox. 11 GB libres)	15 GB (aprox. 14 GB libres)

tamaño del sistema, sino también a que el número de pasos de Monte Carlo necesarios para alcanzar el equilibrio térmico crece con L , tal como se discutió previamente en la Fig. 1. Por otro lado, los resultados físicos correspondientes a este análisis (energía y magnetización en función de la temperatura para todos los valores de L) se mostraron en la Fig. 2.

VI. CONCLUSIONES

El modelo de Ising en dos dimensiones permitió observar con claridad el fenómeno de transición de fase en sistemas magnéticos, evidenciado por el comportamiento de la energía, la magnetización, la capacidad calorífica y la susceptibilidad magnética al variar la temperatura. La coincidencia entre los resultados obtenidos numéricamente y el valor teórico de la temperatura crítica ($T \approx 2.269$) valida la correcta implementación del algoritmo de Monte Carlo y la pertinencia del enfoque estadístico empleado para describir sistemas físicos complejos.

El análisis de rendimiento computacional demostró que la paralelización es una estrategia fundamental para reducir significativamente los tiempos de simulación, especialmente en sistemas de gran tamaño. Si bien Google Colab ofrece una alternativa accesible, el servidor local optimizado presentó una mejora notable, especialmente al utilizar múltiples núcleos. Esto confirma que, para estudios computacionales exigentes, la combinación de paralelización y hardware dedicado representa la opción más eficiente y escalable.

VII. RECURSOS

El código fuente del proyecto está disponible en el siguiente repositorio de GitHub:

<https://github.com/santiagoruizp/Computacion-Avanzada/tree/main/Proyecto%20Paralelizaci%C3%B3n>

Las características de los entornos computacionales se muestran en la Tabla I:

-
- [1] E. Ising, Beitrag zur theorie des ferromagnetismus, Z. Physik **31**, 253 (1925).
 - [2] L. Onsager, Crystal statistics. i. a two-dimensional model with an order-disorder transition, Physical review **65**, 117 (1944).
 - [3] H. Gould, J. Tobochnik, and W. Christian, *An introduction to computer simulation methods*, Vol. 1 (Addison-Wesley Reading, 1988).
 - [4] D. Landau and K. Binder, *A guide to Monte Carlo simulations in statistical physics* (Cambridge university press, 2021).
 - [5] A. L. Gelover-Santiago, *Simulación del modelo de Ising con el método de Monte Carlo* (Universidad Nacional Autónoma de México, 2005).
 - [6] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, Equation of state calculations by fast computing machines, The journal of chemical physics **21**, 1087 (1953).