

Kawasaki

Maria Esther Angulo Ruiz

July 2024

Resumen

En este informe se presenta la implementación y análisis de una simulación basada en el modelo de Kawasaki. Se examinan las propiedades y comportamiento del sistema utilizando un algoritmo de Monte Carlo que conserva el número total de espines durante la simulación, intercambiando dos espines vecinos y evaluando la probabilidad de que ocurra dicho intercambio en base al valor de la diferencia de energía entre el estado inicial del sistema y el estado final.

1. Introducción

El modelo de Kawasaki, una variante del modelo de Ising, es una herramienta fundamental en física estadística para estudiar sistemas magnéticos y materiales con conservación de partículas. A diferencia del modelo de Ising tradicional, donde los espines pueden cambiar su estado de $+1$ a -1 y viceversa, la dinámica de Kawasaki conserva el número total de espines de cada tipo mediante la permutación de pares de espines vecinos. Esta característica lo hace especialmente adecuado para simular sistemas donde se conserva una cantidad global, como la concentración de partículas en aleaciones binarias o la magnetización en ciertos materiales.

El objetivo principal de este informe es describir la implementación de un código que simula el modelo de Kawasaki utilizando el método de Monte Carlo. Se analizarán las propiedades del sistema en función de parámetros como la temperatura y se evaluará la magnetización y otros observables como la densidad de partículas en un eje o la energía media. La simulación se lleva a cabo en una red bidimensional, lo que permite estudiar la formación de dominios y la evolución temporal del sistema.

1.1. Fundamentos del Modelo de Kawasaki

En el modelo de Kawasaki, los espines están dispuestos en una red bidimensional y pueden tomar valores de $+1$ o -1 . La energía del sistema está determinada por la interacción entre espines vecinos, descrita por el Hamiltoniano:

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

donde J es la constante de interacción, S_i y S_j son los espines en los sitios i y j , y la suma se realiza sobre todos los pares de vecinos.

El proceso de Monte Carlo en la dinámica de Kawasaki implica seleccionar aleatoriamente un par de espines vecinos y evaluar el cambio de energía (ΔE) asociado con su permutación. La permutación se acepta con una probabilidad basada en el criterio de Metropolis:

$$P(\Delta E) = \begin{cases} 1 & \text{si } \Delta E \leq 0, \\ \exp(-\Delta E/k_B T) & \text{si } \Delta E > 0, \end{cases} \quad (2)$$

donde k_B es la constante de Boltzmann y T la temperatura del sistema.

1.2. Objetivos del Estudio

Este estudio se enfoca en:

- Implementar un algoritmo de Monte Carlo para el modelo de Kawasaki.
- Analizar la magnetización por partícula y su dependencia con la temperatura.
- Observar la formación de dominios y la evolución temporal del sistema.
- Analizar otros observables relevantes en función de la temperatura y la dimensión de la red bidimensional de partículas.

2. Código: metodología y algoritmo.

En esta sección se describe la metodología y el algoritmo utilizados para la simulación del modelo de Kawasaki. Este modelo se implementa en una red bidimensional de $N+2 \times N+2$ espines utilizando un algoritmo de Monte Carlo, que se detalla a continuación. Una anotación relevante es que la matriz central de $N \times N$ espines es con la que trabajamos, las filas y columnas exteriores a esta matriz solo se utilizan para asegurar las condiciones de contorno periódicas.

2.1. Generación de la matriz inicial

La simulación comienza con la generación de una matriz de espines aleatorios. Cada espín puede tener un valor de $+1$ o -1 , distribuidos de manera uniforme en la matriz. Para implementar condiciones de contorno periódicas, se agregan filas y columnas adicionales alrededor de la matriz original, siendo la matriz central $N \times N$ la que nos interesa.

2.2. Condiciones de contorno periódicas

Las condiciones de contorno periódicas se manejan replicando los bordes de la matriz en las filas y columnas adicionales. Esto asegura que los espines en los bordes de la matriz tengan vecinos correctos en el lado opuesto de la misma.

2.3. Cálculo de energía, magnetización y densidad.

La energía del sistema se calcula considerando las interacciones entre cada espín y sus vecinos más cercanos (arriba, abajo, izquierda y derecha). Primero se calcula la energía del sistema antes del cambio. Luego se hace el cambio, y por último se calcula la energía después del cambio. Así obtenemos la diferencia de energía ΔE que evaluamos en el algoritmo de Metrópolis. La magnetización del sistema se obtiene sumando todos los valores de los espines en la matriz, proporcionando una medida de la alineación global de los espines. En nuestro caso la magnetización se va a mantener constante a lo largo de toda la simulación. A lo largo de la simulación, se calculan y registran la energía media y la magnetización del sistema. Estos valores se promedian sobre el número total de pasos de Monte Carlo para obtener estimaciones precisas de las propiedades del sistema en equilibrio. También se calcula la densidad media de partículas en la dirección y para diferentes temperaturas y tamaños.

2.4. Simulación de Monte Carlo

El algoritmo de Monte Carlo utilizado en esta simulación sigue estos pasos:

1. Selección aleatoria de espines: se selecciona aleatoriamente un par de espines vecinos en la matriz. 2. Evaluación de intercambio: si los espines seleccionados son diferentes, se evalúa un posible intercambio. 3. Cálculo de energía: se calcula la energía del sistema antes y después del intercambio de los espines. 4. Criterio de aceptación: si el intercambio reduce la energía del sistema, se acepta automáticamente. Si incrementa la energía, se acepta con una probabilidad proporcional a $e^{-\Delta E/T}$, donde ΔE es el cambio en la energía y T es la temperatura. 5. Actualización de la matriz: si se acepta el intercambio, se actualiza la matriz de espines y se ajustan las condiciones de contorno periódicas en consecuencia.

Estos pasos se repiten para un gran número de iteraciones, típicamente $PMC \times N^2$ pasos de Monte Carlo, donde PMC es un parámetro que controla la duración de la simulación.

2.5. Medición de Energía y Magnetización

3. Resultados

La simulación produce datos que representan la configuración de espines en varios estados y las correspondientes energías, magnetizaciones y densidades. Estos datos se analizan para entender el comportamiento del sistema bajo diferentes condiciones, como la variación de la temperatura o el número de espines de la red.

3.1. Simulación para diferentes espines y temperaturas.

En las animaciones representadas se observa la convergencia para temperaturas bajas y la no convergencia para las altas. Esto se observa en las animaciones ,que se encuentran en el repositorio asociado, para el parámetro asociado con la temperatura; $T \in 1.0, 2.0, 3.0$ y el número de espines, con parámetro asociado $N \in 16, 32, 48$. Para N mayores de 48, el código tarda mucho en ser ejecutado.

3.2. Magnetización.

Se muestra a continuación las imágenes, donde se observa la representación de la magnetización por dominios para cada instante de tiempo y para diferentes tamaños y temperaturas:

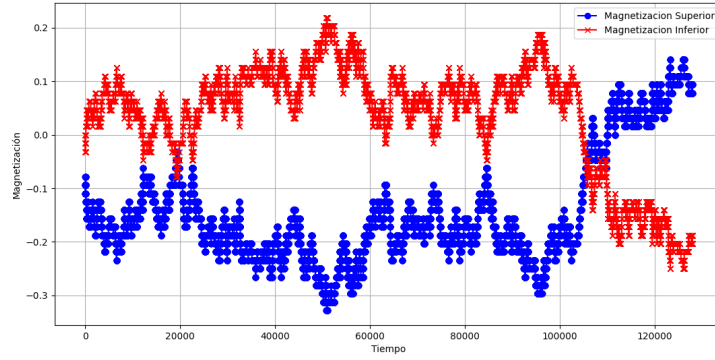


Figura 1: Representación de la magnetización por dominios para $T=1$ y $N=16$.

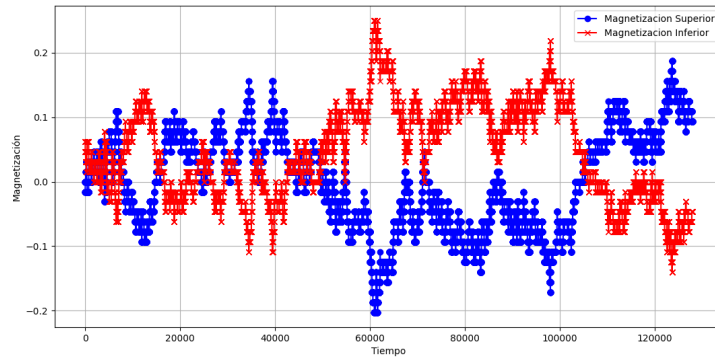


Figura 2: Representación de la magnetización por dominios para $T=2$ y $N=16$.

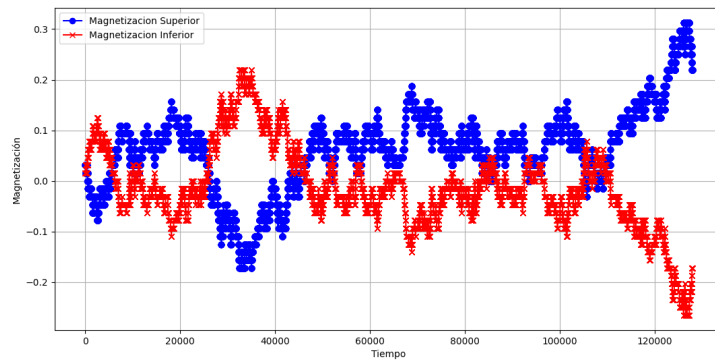


Figura 3: Representación de la magnetización por dominios para $T=3$ y $N=16$.

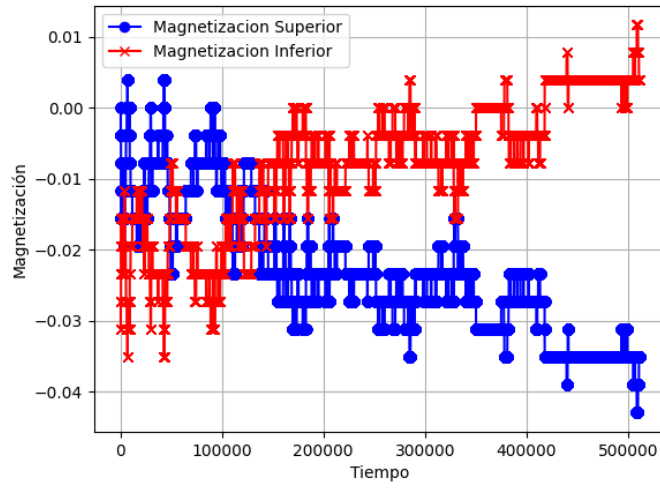


Figura 4: Representación de la magnetización por dominios para $T=1$ y $N=32$.

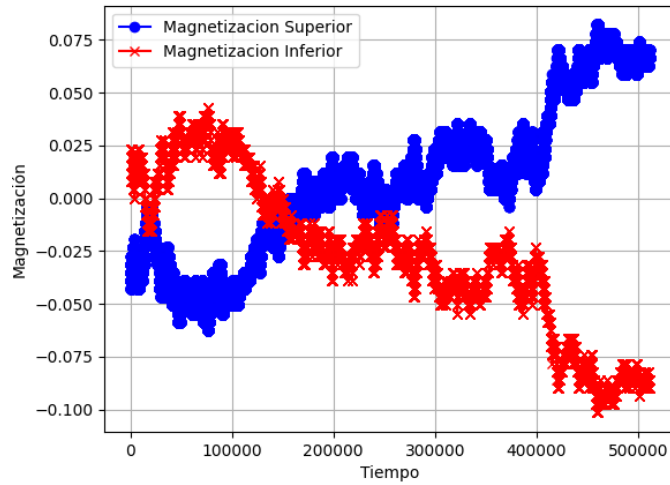


Figura 5: Representación de la magnetización por dominios para $T=2$ y $N=32$.

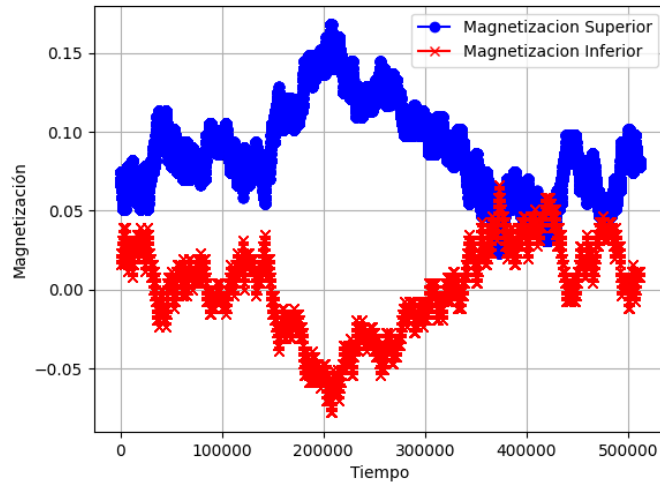


Figura 6: Representación de la magnetización por dominios para $T=3$ y $N=32$.

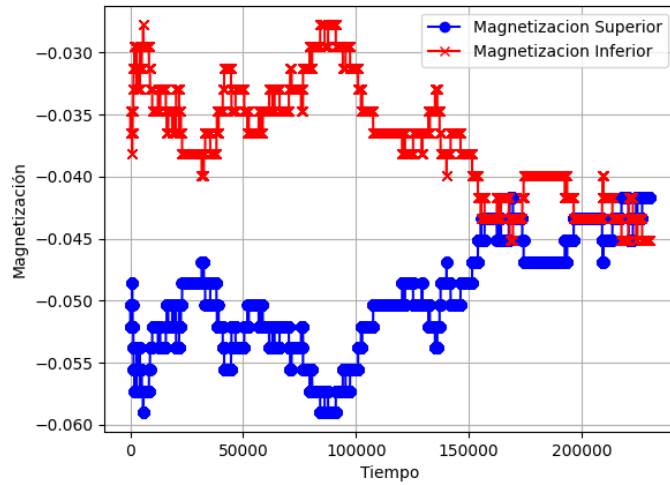


Figura 7: Representación de la magnetización por dominios para $T=1$ y $N=48$.

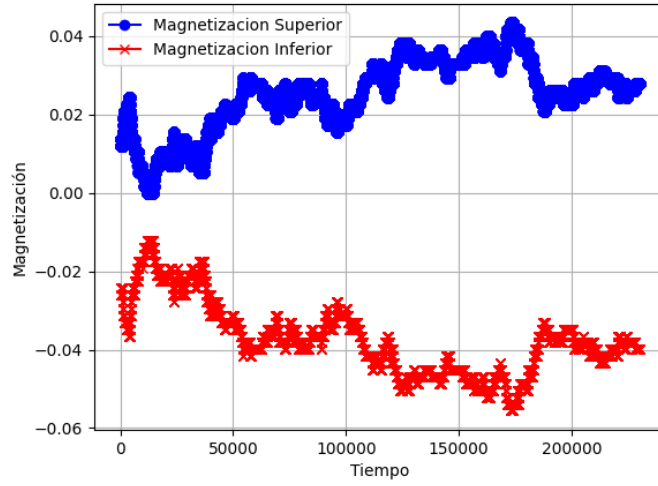


Figura 8: Representación de la magnetización por dominios para $T=2$ y $N=48$.

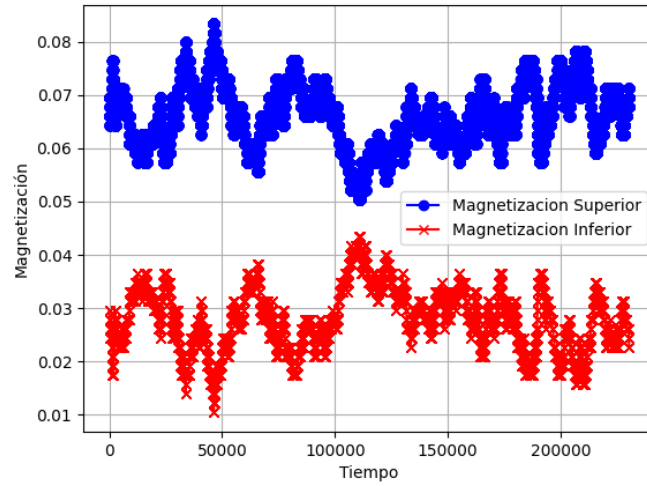


Figura 9: Representación de la magnetización por dominios para $T=3$ y $N=48$.

Se observa una clara simetría entre los valores de la magnetización de la parte superior de la matriz y la parte inferior para todas las temperaturas y tamaños. Esto es lógico pues la magnetización se conserva, luego ambas partes deben contrarrestarse, siendo la magnetización total en cada instante la misma que la inicial.

3.3. Densidad de partículas.

Se muestra a continuación tres imágenes, donde se observa la representación de la densidad de partículas media en la dirección y en función de la temperatura y para

diferentes tamaños:

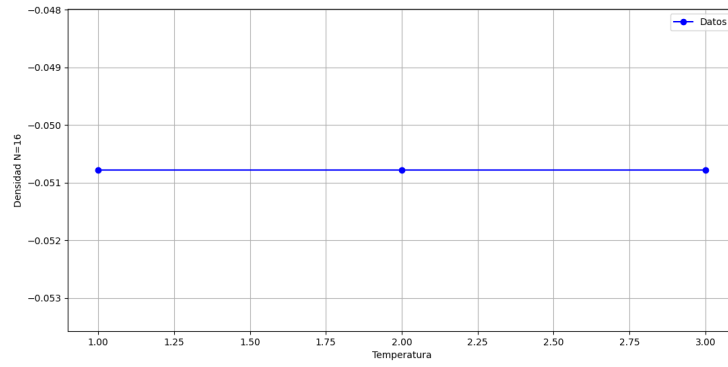


Figura 10: Representación de la densidad media por partícula para $T = 1, 2, 3$ y $N=16$.

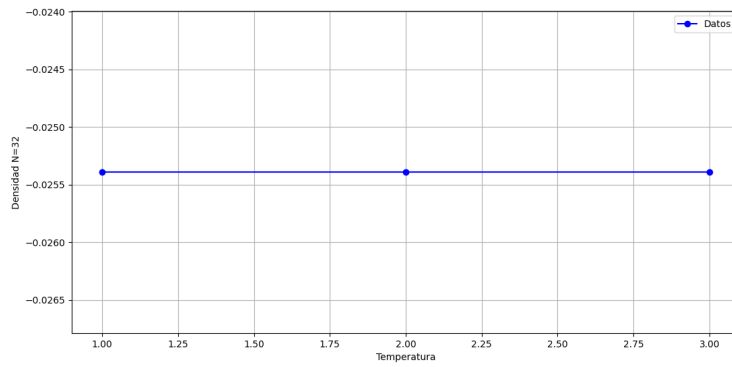


Figura 11: Representación de la densidad media por partícula para $T = 1, 2, 3$ y $N=32$.

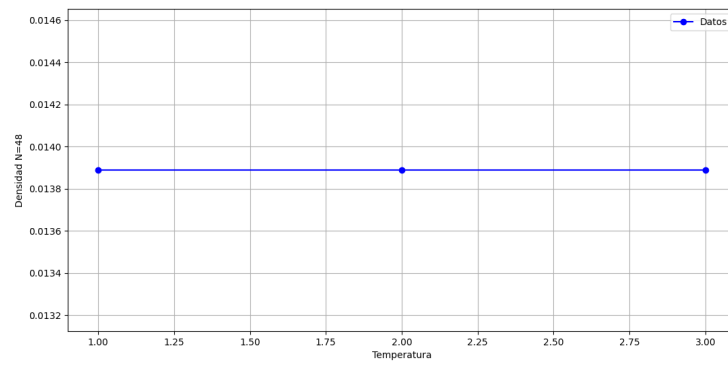


Figura 12: Representación de la densidad media por partícula para $T = 1, 2, 3$ y $N=48$.

Se observa que la densidad de partículas media en a dirección del eje y se mantiene constante como era de esperar.

3.4. Energía media.

Se muestra a continuación tres imágenes, donde se observa la representación de la energía media en función de la temperatura y para diferentes tamaños:

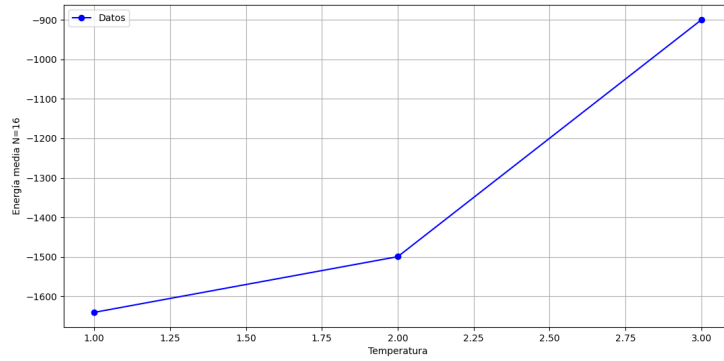


Figura 13: Representación de la energía media por partícula para $T = 1, 2, 3$ y $N=16$.

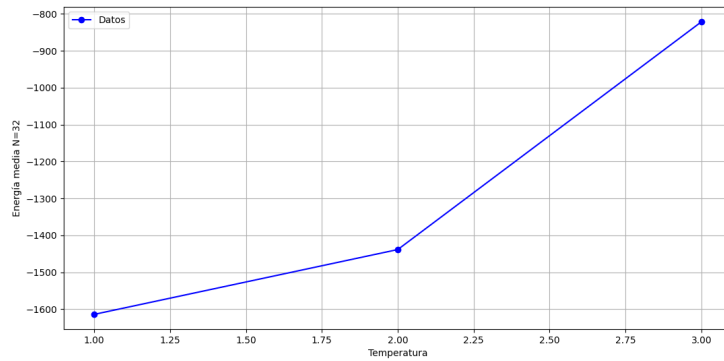


Figura 14: Representación de la energía media por partícula para $T = 1, 2, 3$ y $N=32$.

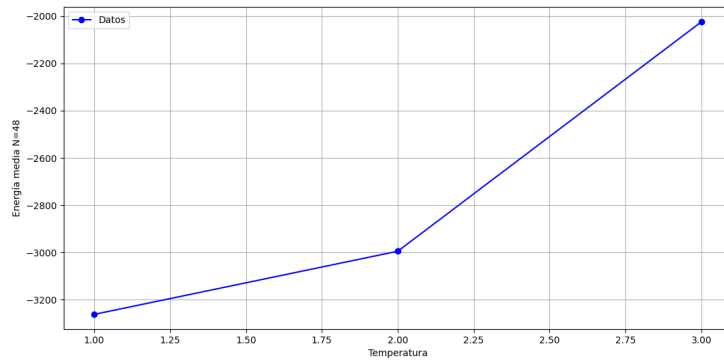


Figura 15: Representación de la energía media por partícula para $T = 1, 2, 3$ y $N=48$.

Se observa que la energía media aumenta con la temperatura para los tres casos de N . Es cierto que a medida que aumenta N , el aumento de la energía media de una temperatura a otra es más pronunciado.

3.5. Temperatura crítica.

Observando los datos de magnetización total vemos que la magnetización aumenta abruptamente al aumentar la temperatura obteniendo el cambio de fase buscado entre un material ordenado y uno desordenado.

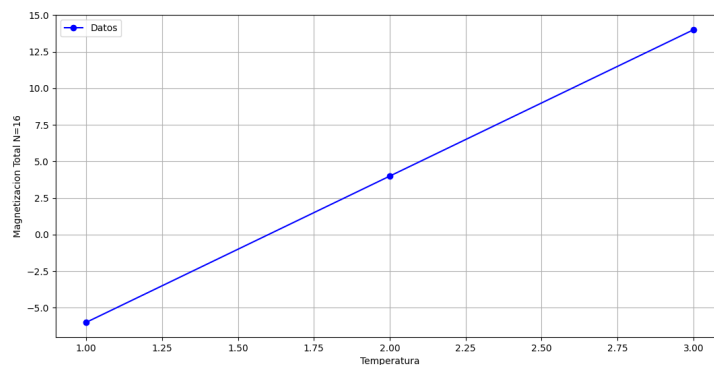


Figura 16: Representación de la magnetización total para $T = 1, 2, 3$ y $N=16$.

3.6. Optimización

El programa al ser ejecutado en mi ordenador no me permitía hacer cálculos con valores mayores de $N=48$.

4. Conclusión

El programa realizado es muy lento para valores altos de N , quizás por el ordenador en el que ha sido ejecutado o porque el código no está bien optimizado. Se observan resultados esperados en la mayoría de apartados y las animaciones se corresponden con las esperadas según el modelo de Kawasaki.