

Modelo de Ising cuántico unidimensional en una grilla de N espines: Análisis espectral

Johnny Buzo, Natalia Calderón, Joseph Castillo, Gerald Schmitz, Dilan Vega

1. Contexto

El modelo de Ising fue originalmente introducido para estudiar el ferromagnetismo y probablemente es uno de los modelos más estudiados en la física estadística. Los espines están dispuestos en una estructura de grafo dada y cada uno puede estar en uno de dos estados: $+1$ (“arriba”) o -1 (“abajo”) [1]. Estos espines interactúan entre sí de manera estocástica, pero cada espín tiene la tendencia a alinearse con sus vecinos, ya que esto resulta en configuraciones de baja energía para el sistema. En la literatura de física estadística, los investigadores han considerado principalmente el modelo de Ising en estructuras de grillas o grafos completos.

El modelo de Ising es fundamental para comprender conceptos como la transición de fase, donde un sistema experimenta un cambio abrupto en sus propiedades físicas a medida que se varía un parámetro, como la temperatura. Este modelo ha sido estudiado en diversas geometrías de red y condiciones de contorno para comprender mejor los fenómenos críticos en la física de la materia condensada[2].

2. Hamiltoniano con productos tensoriales

El Hamiltoniano del modelo de Ising unidimensional para N espines $1/2$ en una grilla periódica, se puede escribir de la siguiente forma:

$$\hat{H} = -J \sum_{i=1}^N \sigma_i^z \sigma_{i+1}^z - g \sum_{i=1}^N \sigma_i^x$$

Donde se definen las matrices de Pauli actuando sobre el espín i de la siguiente forma:

$$\sigma_i^\alpha = I_2 \otimes I_2 \otimes \cdots \otimes \sigma^\alpha \otimes \cdots \otimes I_2$$

Sin embargo, el producto tensorial de matrices identidad es otra matriz identidad del doble de dimensión, de forma que se puede simplificar a dos productos tensoriales:

$$\sigma_i^\alpha = I_{2i} \otimes \sigma^\alpha \otimes I_{2(N-i-1)}$$

Partiendo de esto, se crea una función $S(N, \sigma^\alpha)$ que devuelve una matriz $S\alpha$ que contiene todas las N matrices σ_i^α :

$$S\alpha = [\sigma_1^\alpha, \sigma_2^\alpha, \dots, \sigma_N^\alpha]$$

De forma que $S\alpha[0] = \sigma_1^\alpha$ y $S\alpha[N-1] = \sigma_N^\alpha$

Al ser una grilla periódica, se consideró la condición de frontera periódica:

$$\sigma_{(N+k)}^z = \sigma_k^z$$

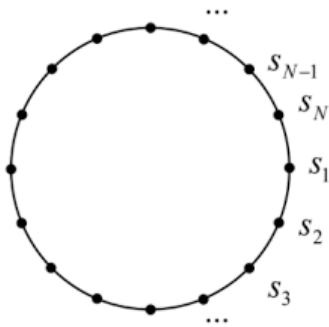


Figura 1: Ejemplificación de las condiciones de frontera

3. Escalar el Hamiltoniano a la mayor cantidad de espines posibles

Para esto se pasa a trabajar con sistemas de matrices dispersas, cuya cualidad es que almacenan y trabajan solo con los elementos no nulos, lo cual ahorra memoria y tiempo de ejecución, además se puede usar kron de scipy en lugar de np.kron, lo cual hace eficiente el trabajo con matrices dispersas. El hecho de trabajar con matrices dispersas optimiza el proceso ya que no se gasta tiempo ni memoria en la ejecución de operaciones sobre valores nulos, de esta forma se puede trabajar eficientemente con matrices muy grandes, tomando en cuenta que se debe de hacer el sistema para que reciba cualquier N y sabiendo que las dimensiones de las matrices son de 2^*N entonces para valores grandes de N las matrices son de grandes dimensiones y este métodos permite realizar las operaciones necesarias con más agilidad.

4. Diagonalizar el sistema y visualizar los autovalores

Para realizar la diagonalización, se utilizó la biblioteca Numpy, en la que se obtuvo una matriz diagonalizada junto a sus respectivos autovalores y autovectores de modo que para valores dados de N=2, J=2 y g=1:

$$\begin{bmatrix} -2,82842712 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ 0 & 0 & 2,82842712 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

Autovalores
-2,82842712
-2,00000000
2,82842712
2,00000000

5. ¿Qué se puede implementar para acelerar la operación de la diagonalización?

En primer lugar se debe de optimizar el código lo mejor posible, por ejemplo mediante métodos iterativos para trabajar matrices grandes y dispersas, para ese caso se puede destacar el método de Lanczos que se especializa especialmente en el cálculo de los valores y vectores propios de un sistema lineal de forma iterativa. Realizar paralelizaciones del código para optimizar el uso de la memoria y así acelerar el proceso de diagonalización. De igual forma se puede directamente hacer uso de bibliotecas de alto rendimiento tal como lo es Eigen para el caso de c++ pues es optimizada para cálculos de álgebra lineal y fue precisamente la utilizada en c++. El utilizar matrices dispersas para los cálculos simplifica mucho el tiempo de ejecución al no almacenar los datos no nulos. Hay varias formas de acelerar este proceso, en la presente aplicación se implementaron con éxito algunas de las mismas.

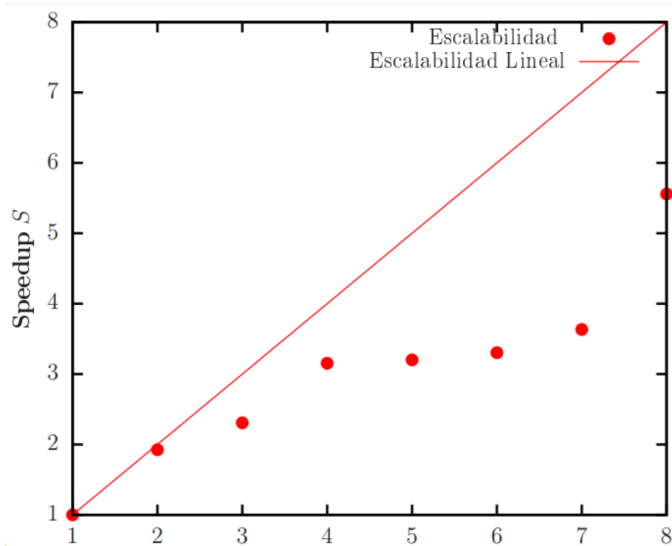


Figura 2: Speedup.dat

6. Dificultades

1. La interpretación teórica a la hora de tomar en cuenta las condiciones de frontera.
2. El mecanismo matemático necesario para convertir los productos tensoriales a un lenguaje de programación.
3. Pasar de un lenguaje interpretado, como Python, a otro compilado como puede ser C++, esto debido a la complejidad en la sintaxis de este último.

7. Utilidad

1. **Estudio de Propiedades Termodinámicas:** El análisis espectral de can-

tidades termodinámicas como la energía libre, la energía interna y la capacidad calorífica en el modelo de Ising proporciona información detallada sobre el comportamiento del sistema en diferentes condiciones, incluidas las transiciones de fase.

2. **Simulaciones Numéricas:** El análisis espectral se utiliza en simulaciones numéricas del modelo de Ising para estudiar propiedades críticas, comportamiento a diferentes temperaturas y tamaños de red, y para comparar con resultados teóricos y experimentales.
3. **Caso práctico:** En el artículo "Phase Transition of the Ising Model on Fractal Lattice" se emplea el análisis espectral de la matriz de densidad para investigar la estructura de entrelazamiento en el modelo de Ising en una red fractal, lo que proporciona información valiosa sobre la naturaleza de las transiciones de fase y las propiedades críticas del sistema en esta configuración particular.

Referencias

- [1] Simone Baldassarri. Ising model on clustered networks: A model for opinion dynamics. *Physica A*, 623:128811, 2023.
- [2] Jozef Genzor, Andrej Gendiar, and Tomotoshi Nishino. Phase transition of the Ising model on fractal lattice. *arXiv:1509.05596v2 [cond-mat.stat-mech]*, December 2015.