Miniproyecto #2
Paramagnetismo y Ferromagnetismo. Simulación Monte Carlo.

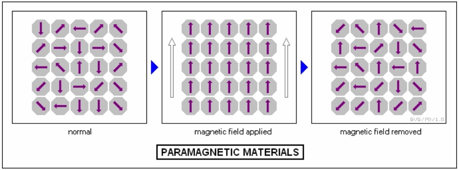
Johans Restrepo Cárdenas

Instituto de Física. Universidad de Antioquia.

18 de abril de 2023

Planteamiento del problema.

Representación esquemática (2D) de un paramagneto.



Escriba un programa para simular (en el espacio 2D) al menos tres isotermas $\{T_1, T_2, T_3\}$ de curvas de magnetización en función de un campo magnético externo aplicado $(M$ vs. $H)$ usando un método de Montecarlo con dinámica de Metropolis para cada uno de los siguientes sistemas:

1. Un paramagneto tipo Ising con $S = 1/2$
2. Un paramagneto Heisenberg cuántico con $S = 3/2$
3. Un paramagneto Heisenberg clásico con $S = \infty$
4. Un ferromagneto tipo Ising con $S = 1/2$

Planteamiento del problema.

El informe debe incluir el código fuente con comentarios. Ejemplo:

ising.py

```python
import random
#--------------------------------------------------------
# Implementación de condiciones de frontera periódicas
#--------------------------------------------------------
def pbc(i):
    if i+1 > SIZE-1:
        return 0
    elif i-1 < 0:
        return SIZE-1
    else:
        return i

#--------------------------------------------------------
# Cálculo de la energía interna (J_ex = 1)
#--------------------------------------------------------
def energy(spin, i, j):
    return -1 * spin[i,j] * (spin[pbc(i-1),j] + spin[pbc(i+1),j] +
                spin[i,pbc(j-1)] + spin[i,pbc(j+1)]) - spin[i,j] * H

#--------------------------------------------------------
# Microestado inicial aleatorio (T=inf, H=0)
#--------------------------------------------------------
def build_system():
    spin = np.random.random_integers(0,1,(SIZE,SIZE))
    spin[spin == 0] = -1
    return spin

#--------------------------------------------------------
# Loop Monte Carlo principal - Dinámica de Metropolis
#--------------------------------------------------------
def main(T):
    spin = build_system()

    for step, x in enumerate(range(STEPS)):
        j = np.random.randint(0,SIZE)
        i = np.random.randint(0,SIZE)

        Delta_E = -2. * energy(spin, i, j)

        if Delta_E <= 0.:
            spin[i,j] *= -1
        elif np.exp(-1./T*Delta_E) >= np.random.rand():
            spin[i,j] *= -1

#--------------------------------------------------------
# Run
#--------------------------------------------------------
def run():
    print 'Entre el valor de temperatura (0.1-100)'
    T = float(raw_input())
    main(T)
```

Planteamiento del problema.

El ejemplo anterior muestra un código en Python en el que se definen los siguientes **métodos** y que usted debe tener en cuenta:

- Definición de condiciones de frontera periódicas (trabajar con SIZE = 100).
- Método de cálculo de la energía basado en el siguiente hamiltoniano por espín:

$$\mathcal{H}_i = -J\sigma_i \sum_{<j>} \sigma_j - \sigma_i H$$

  - El primer término da cuenta de la interacción del espín σ_i con sus primeros vecinos (en un paramagneto $J = 0$).
  - El segundo, la interacción de dicho espín con un campo externo H (término Zeeman).

Planteamiento del problema.

- Método para definir un estado inicial (en el ejemplo: espines aleatorios correspondiente a $T \to \infty$).
- Método que define la dinámica de Metropolis, visitando una red **cuadrada** e intentando inversiones de espín, calculando el cambio en la energía y teniendo en cuenta la siguiente razón de probabilidades de transición entre estados μ y ν:

$$\frac{W(\mu \to \nu)}{W(\nu \to \mu)} = \frac{e^{-\beta E_2}}{e^{-\beta E_1}} = e^{-\beta \Delta E}$$

Objetivos.

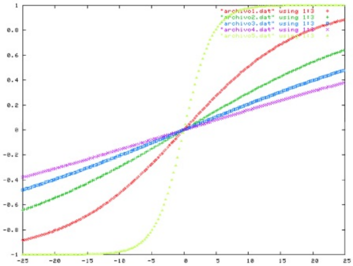
En el caso de un **paramagneto (PM)** ($J = 0$) y para los diferentes valores de S debe:

1. Obtener y graficar curvas M vs. H para tres temperaturas diferentes $\{T_1, T_2, T_3\}$.
2. Comprobar la ley de estados correspondientes graficando M vs. H/T.
3. Ajustar estos resultados con la curva teórica según corresponda (tanh(x) para Ising o $S = 1/2$, función de Brillouin $B_S(\eta)$ para $S = 3/2$ y función de Langevin $L(x)$ para $S = \infty$.

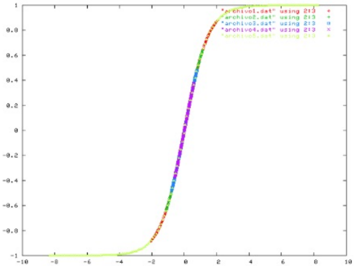
Objetivos.

Paramagnetismo: Ejemplos del tipo de curvas que se deberían obtener:

- *Curvas* M *vs.* H:



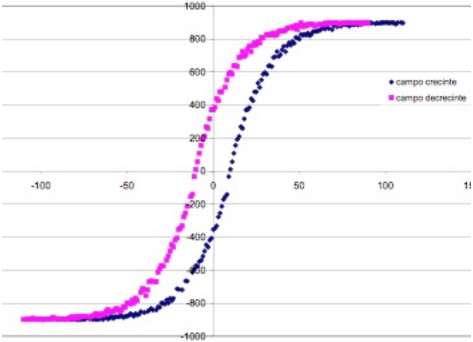
- *Curvas* M *vs.* H/T:



Objetivos.

En el caso de un **ferromagneto (FM)** ($J = 1$) tipo Ising $S = 1/2$ debe:

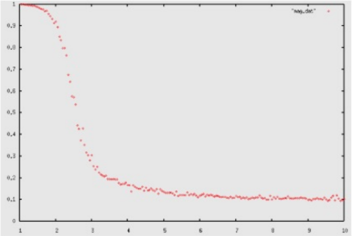
- Obtener y graficar al menos **una** isoterma M vs. H que revele la existencia del fenómeno de **histéresis**. Un ejemplo se ilustra a continuación:



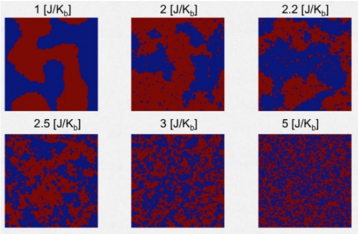
Objetivos.

- Obtener una curva de magnetización en función de la temperatura M vs. T que revele la existencia de una **transición de fase** y figuras de estados magnéticos (*snapshots*) para diferentes temperaturas como se ilustra a continuación ($\uparrow =$ rojo, $\downarrow =$ azul):

- *Curva M vs. T*:



- *Snapshots*:



Objetivos.

- Obtener una curva de la capacidad calorífica, calculada a partir de $C_V = (\langle E^2 \rangle - \langle E \rangle^2)/(k_B T^2)$, en función de la temperatura y a partir de su máximo determine la temperatura pseudocrítica de Curie $T_C(L)$. Comparela con el valor teórico $T_C(\infty)$ (investigar).
- Hacer un análisis, basado en los snapshots, de lo que ocurre antes y después de dicha temperatura de transición.
- Finalmente, modifique su programa para reproducir una **red triangular**, y analice el efecto de la frustración magnética para el caso de un **aniferromagneto (AFM)** tipo Ising con $J = -1$.

Sobre el informe.

El informe en forma de artículo debe contener:

- Título. Nombre autor, afiliación.
- Resumen y palabras claves.
- Introducción (estado del arte, motivación)
- Marco teórico (Paramagnetismo cuántico y clásico, Ferromagnetismo, método Monte Carlo y dinámica de Metropolis)
- Resultados y discusión (incluya las gráficas solicitadas).
- Conclusiones
- Bibliografía
- Agradecimientos
- Anexos (códigos)