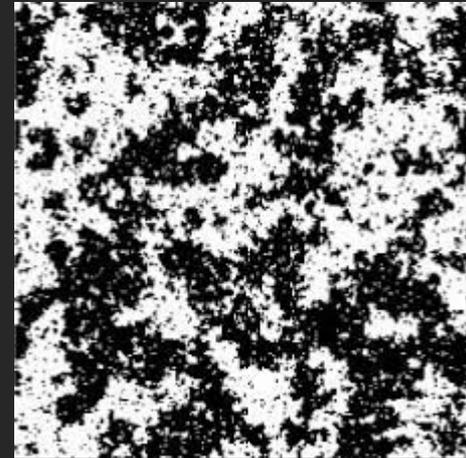


Algoritmo de Metropolis en C



6 de Mayo de 2021

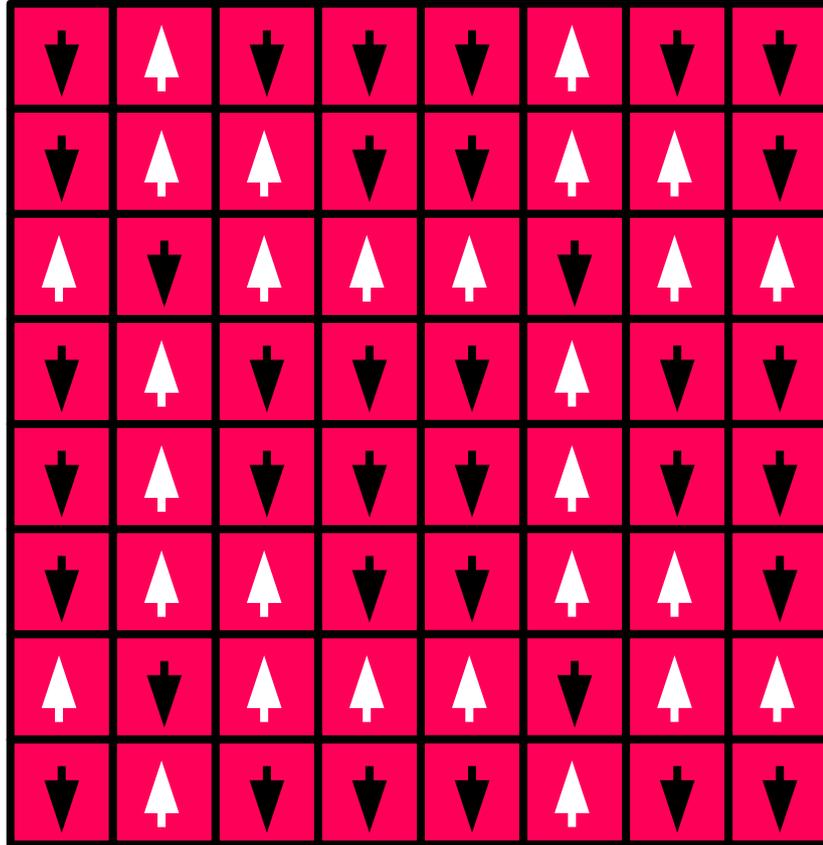
Esquema de la clase

Dividiremos la clase en dos partes

- Algoritmo de Metropolis
- Magnitudes termodinámicas: energía y magnetización

Poblamos la red

Podemos hacerlo como antes con una cierta probabilidad de ocupación
¿Como simulamos T alta?

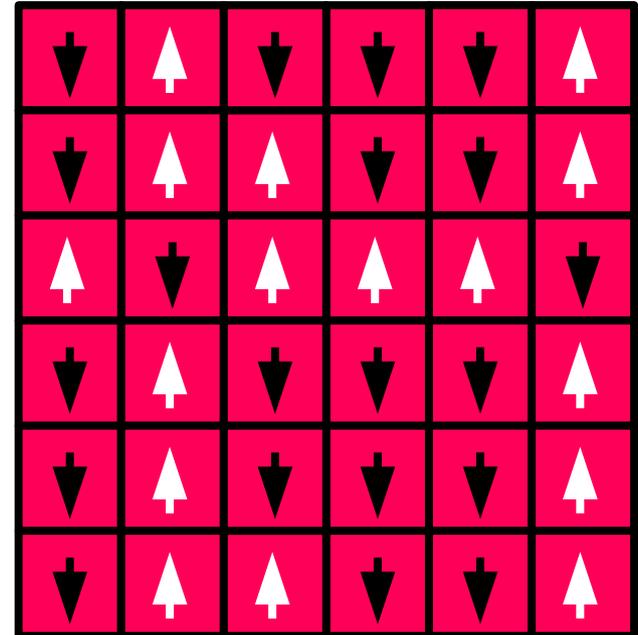


Algoritmo de Metropolis

Una vez poblada la red, la hacemos evolucionar (no en el tiempo!)

Les propongo usar sub-funciones:

```
Metropolis()  
{  
    pick_site();  
    flip();  
}
```



Algoritmo de Metropolis

`pick_site()`

Elegimos arbitrariamente un spin de la red

Veamos dos posibles formas:

a) Índice de 0 a N-1

b) Usando filas y columnas (tipo matriz). Usamos dos números random desde 0 hasta L-1

Ej. cada uno los podemos elegir usando `rand()%L`

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(a)

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

(b)

Algoritmo de Metropolis

flip()

Elegido un spin (azul) nos fijamos cuales son sus vecinos inmediatos (primeros vecinos).

Elegido = $\text{fila} * L + \text{columna}$

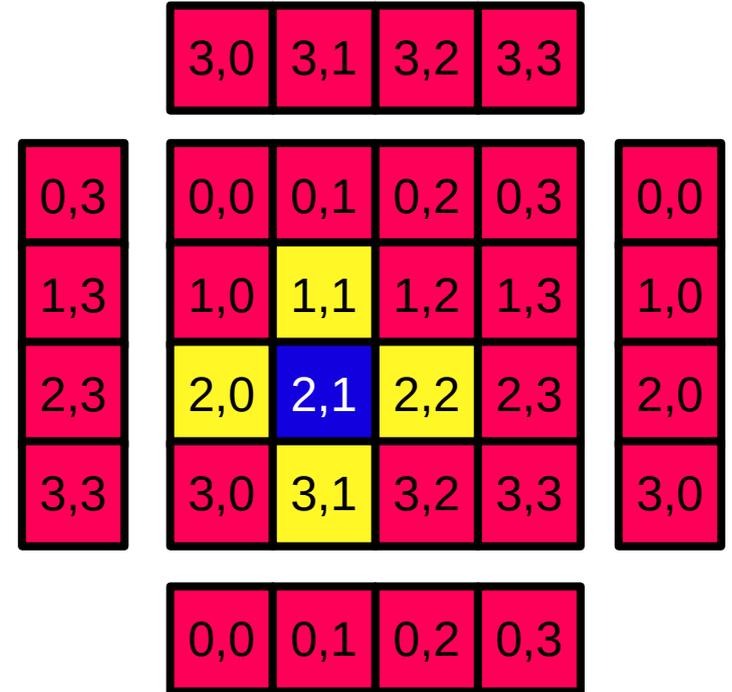
Debemos actualizar “fila” y “columna”:

$\text{fila_abajo} = (\text{fila} + 1 + L) \% L;$ $1 \% 4 = 1, 2 \% 4 = 2, 7 \% 4 = 3$

$\text{fila_arriba} = (\text{fila} - 1 + L) \% L;$

$\text{columna_izquierda} = (\text{columna} - 1 + L) \% L;$

$\text{columna_derecha} = (\text{columna} + 1 + L) \% L;$



Algoritmo de Metropolis

flip()

Elegido = fila*L + columna

Celda_abajo = fila_abajo*L + columna;

Celda_arriba = fila_arriba*L + columna;

Celda_izquierda = fila*L + columna_izquierda;

Celda_derecha = fila*L + columna_derecha;

$$E_k = -J s_e (s_1 + s_2 + s_3 + s_4) - B s_e$$

$$E_{k,flip} = -E_k$$

$$\Delta E = -2E_k$$

red[Celda_derecha]

	3,0	3,1	3,2	3,3	
0,3	0,0	0,1	0,2	0,3	0,0
1,3	1,0	1,1	1,2	1,3	1,0
2,3	2,0	2,1	2,2	2,3	2,0
3,3	3,0	3,1	3,2	3,3	3,0
	0,0	0,1	0,2	0,3	

Tomemos un desvío



creo_tabla()

Es la energía de interacción con los vecinos y con el campo (no es la energía total del sistema!)

$$E_k = -J s_e (s_1 + s_2 + s_3 + s_4) - B s_e$$

$$E_{k,flip} = -E_k$$

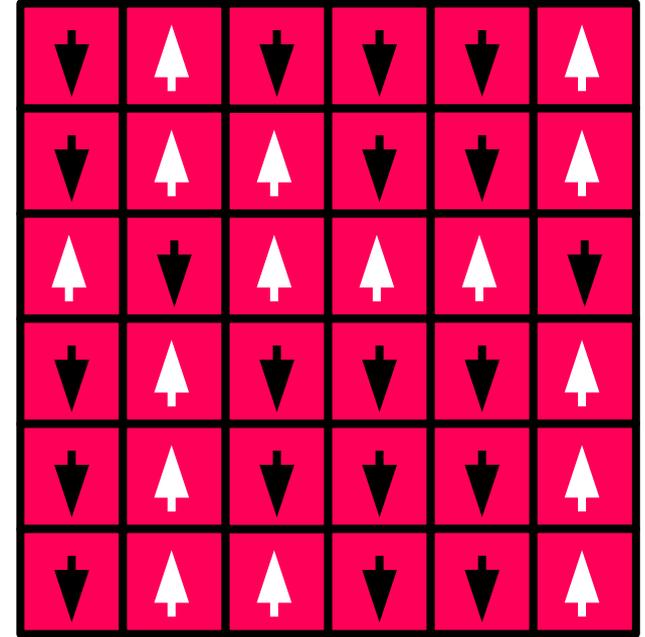
$$\Delta E = -2E_k$$

```
for (int i = 0; i < 5; i++)
{
    dE = -j*(4*i-8); //j=1
    exponente = (float)dE/T;
    vec_exp[i]=exp(exponente);
}
```

← Sin campo B

```
for (int i = 0; i < 2; i++)
{
    dE = 2*b*(2*i-1) ; //b=1
    exponente = (float)dE/T;
    vec_exp[i]=exp(exponente);
}
```

← Sin interacción entre espines



$$p_{ij} = e^{-\beta \Delta E}$$

Volvamos...



Algoritmo de Metropolis

flip()

Calculamos ΔE ;

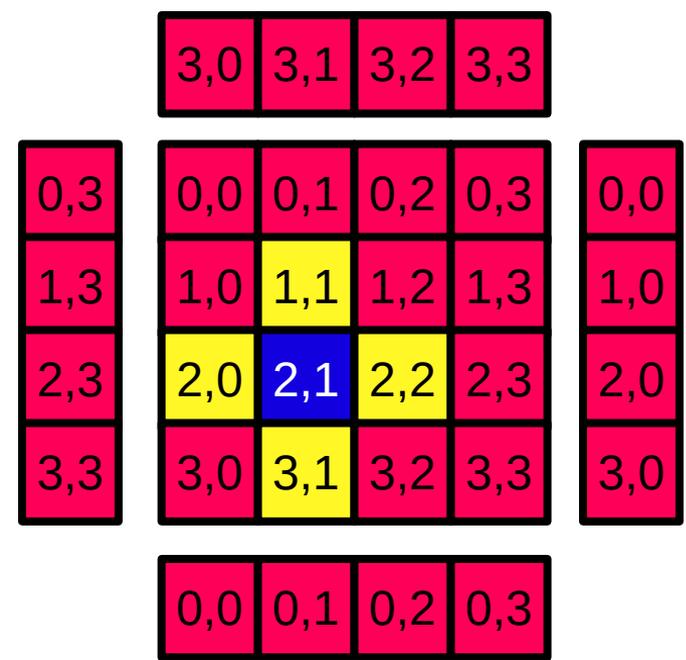
Si $\Delta E \leq 0 \rightarrow \text{spin_flip} \rightarrow \text{red}[\text{elegido}] = -\text{red}[\text{elegido}]$

Si $\Delta E > 0 \rightarrow$ aceptamos el spin_flip con probabilidad

$$p_{ij} = e^{-\beta \Delta E}$$

Usamos la tabla, ej. $p = \text{vec_exp}[-\Delta E/4 + 2]$

Listo el algoritmo de Metropolis!



```
for (int i = 0; i < 5; i++)
{
    dE = -j*(4*i-8); //j=1
    exponente = (float)dE/T;
    vec_exp[i]=exp(exponente);
}

for (int i = 0; i < 2; i++)
{
    dE = 2*b*(2*i-1) ; //b=1
    exponente = (float)dE/T;
    vec_exp[i]=exp(exponente);
}
```

Todavía nos falta estudiar como se comporta el sistema:

Energía
Magnetización



Funciones respuesta (?)

Energía inicial

Usamos bloques tipo “P” (loop desde 0 hasta L-1 tanto en filas como en columnas)

$$E_{inicial} = E_{inicial} - J * s_{actual} * s_{abajo} - J * s_{actual} * s_{derecha}$$

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Condiciones periódicas de contorno

Energía inicial (si hay campo B)

Nos movemos “libremente” por toda la red

$$E_{inicial} = E_{inicial} - B * s_{actual}$$

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Energía inicial total (por spin)

$$e_{inicial} = (E_J + E_B)/N^2$$

Salvo que cambie la temperatura, no la volvemos a calcular nunca mas!

En cada paso de Metropolis calculamos $\Delta E/N^2$ y se lo sumamos a la energía inicial.

	3,0	3,1	3,2	3,3	
0,3	0,0	0,1	0,2	0,3	0,0
1,3	1,0	1,1	1,2	1,3	1,0
2,3	2,0	2,1	2,2	2,3	2,0
3,3	3,0	3,1	3,2	3,3	3,0
	0,0	0,1	0,2	0,3	

Magnetización inicial

Simplemente es sumar todos los espines de la red

$$M_{inicial} = M_{inicial} + S_{actual}$$

$$m_{inicial} = M_{inicial}/N^2$$

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Salvo que cambie la temperatura, no la volvemos a calcular nunca mas!

En cada paso de Metropolis calculamos $\Delta M/N^2$ y se lo sumamos a la magnetización inicial.

Actualizamos E y M luego de cada paso de Metropolis

Sólo si hay spin flip cambia E y M



$$dM = 2$$



$$dM = -2$$

```
*energ =*energ+(double)dE/(double)(N*N);  
*m=*m+(double)dM/(double)(N*N);
```

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Pseudo-código para Ising

```
poblamos_red();

loop sobre temperaturas
{
    creo_tabla();
    calculo_E_inicial();
    calculo_M_inicial();

    termalizacion();

    loop sobre mediciones
    {
        loop sobre pasos de metropolis
        {
            metropolis();
        }
        vec_e[s] = energ;
        vec_m[s] = m;
    }
}
```

Me “aseguro” que estoy a esa temperatura (hay que averiguar cuantos pasos hacer)

Para descorrelacionar las mediciones (hay que averiguar cuantos pasos hacer)

Pasos a seguir

- Verificar que anda bien la tabla
- Verificar que calcula bien la magnetización y la energía inicial (probar configuraciones fáciles, ej. todos los espines para arriba).
- hasta ahora no usamos Metropolis-----
- Estudiar la termalización del sistema (obtenemos cuantos pasos hay que hacer para poner al sistema a esa temperatura). ¿depende de la config. inicial? Fijar $T=5.0$ y probar diferentes configuraciones iniciales (random, todos los espines up ó down).

```
loop sobre temperaturas
{
    creo_tabla();
    calculo_E_inicial();
    calculo_M_inicial();

    termalizacion();
}
}
```

Pasos a seguir

- Termalización:

Levantar las curvas de E y M en función de los pasos de Metropolis (k/N^2) ¿convergen?

```
loop sobre temperaturas
{
    creo_tabla();
    calculo_E_inicial();
    calculo_M_inicial();

    termalizacion();
}
}
```

Problema 2: Ising 2D por medio de Metropolis

Considere un arreglo cuadrangular de spines $s_i = \pm 1$ con condiciones periódicas de contorno. El hamiltoniano del sistema está dado por

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} s_i s_j - B \sum_i s_i \quad (2)$$

- Si $J = 0$ el hamiltoniano se reduce a la interacción de espines independientes en un campo magnético externo. Obtenga las soluciones analíticas para las variables termodinámicas en este caso y compare con los resultados de las simulaciones.
- Estudie el comportamiento del sistema en una red de 32×32 , para $B = 0$ y $J \in [0.1, 0.6]$. Estime en cada caso la frecuencia de sampleo adecuada para cada constante de acoplamiento empleada. Muestre que las correlaciones se hacen más importantes al acercarse al punto crítico.
- Estudie el comportamiento del sistema para diversos tamaños de red. Analice los efectos producidos por tamaño finito (use $B = 0$ y $J \in [0.1, 0.6]$).
- Explore la termodinámica del modelo para el caso antiferro ($J < 0$) con B finito.
- Haga lo propio para un modelo en el cual cada spin s_i interactúa ferromagnéticamente con sus primeros vecinos y antiferromagnéticamente con sus cuatro segundos vecinos (diagonales). Explique el fenómeno de *frustración*.

