

Fenòmens Col·lectius i Transicions de Fase

Pràctica 2

Simulació MC del model d'Ising
2D: evolució temporal

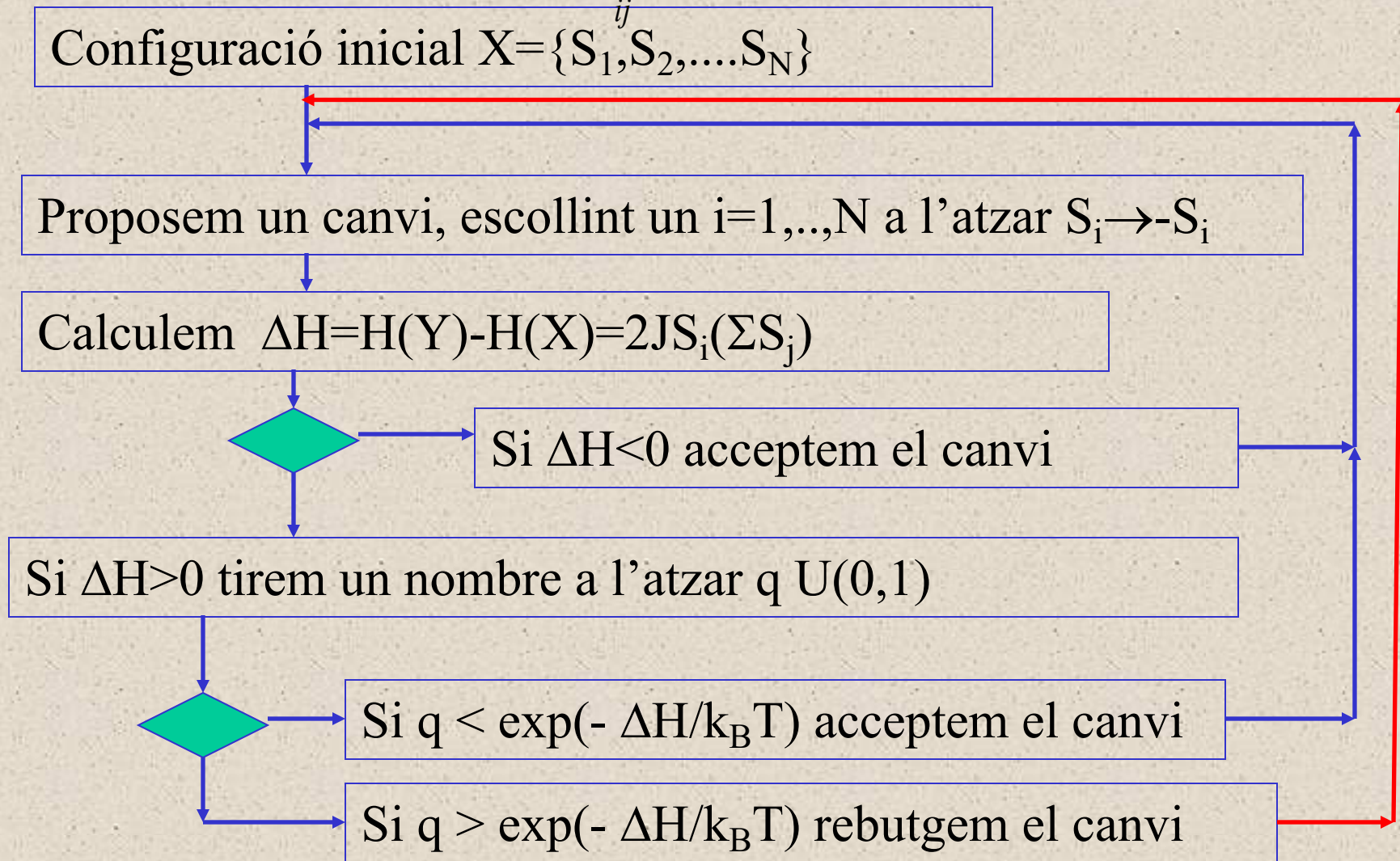
Objectius

Construirem un primer codi molt simple de simulació Monte Carlo `MC1.f` per al model d'Ising 2D en una xarxa quadrada, que ens permetrà generar una seqüència d'estats $\{S(I, J)\}$ corresponents a una cadena de Markov, que després d'un cert nombre inicial de passes, recorrerà l'espai de les fases amb els pesos corresponents a l'equilibri canònic .

(En aquesta primera versió simplement traurem informació de l'evolució de la cadena per pantalla)

Esquema algorisme de Metropolis

- **Model d'Ising:** $H = -J \sum_{ij}^{veins} s_i s_j$



Index: discussions prèvies

- "Temps" en una simulació MC
- Unitats reduïdes
- Efectes de contorn: condicions periòdiques
- Càlcul de l'energia

Exercici P2-exercici-1.f

- Hot spot de l'algorisme de metropolis
- Dades d'entrada

Exercici MC1.f

"Temps" en l'algorisme de Metropolis

En les simulacions d'equilibri, l'índex que marca les passes de la cadena de Markov, en principi, no té cap significat físic.

(Comentari: En el marc de la F.E. fora de l'equilibri, les cadenes de Markov es poden interpretar, de vegades, com evolucions temporals.)

De tota manera l'índex, s'utilitza per comparar una simulació amb una altra i mesurar-ne la "qualitat".


Mes passes signifiquen millors promitjos (sempre que les configuracions estiguin descorrelacionades)

Es defineix: **1 passa MC = N propostes de nova configuració**

Ajuda: estructura dels bucles principals


```
DO IMC=1,MCTOT
```

```
DO IPAS=1,N
```




Aqui farem la proposta de canvi
(gir d'un spin a l'atzar) i
decidirem si l'acceptem o no

```
ENDDO
```



Aqui ha acabat un passa de MC

```
ENDDO
```



Aqui han acabat totes les passes de MC

Unitats reduïdes

$$H(S_1, S_2, \dots, S_N) = -J \sum_{ij} S_i S_j - B \sum_i S_i$$

Definim: $H^* = H/J$, $B^* = B/J$ de forma que:

$$H^*(S_1, S_2, \dots, S_N) = - \sum_{ij} S_i S_j - B^* \sum_i S_i$$

La probabilitat d'acceptació que apareix en l'algorisme serà:

$$e^{-\frac{\Delta H}{k_B T}} = e^{-\frac{\Delta H J}{k_B T J}} = e^{-\frac{\Delta H^*}{T^*}}$$

- Temperatura reduïda $T^* = k_B T / J$

Efectes de contorn

- Convé minimitzar els efectes de contorn:
- Ex: (sistema 2d)

En un sistema real tenim de l'ordre de 10^{23} partícules.

En la simulació tindrem de l'ordre de 10^4 partícules.

Si els dos sistemes tenen la mateixa densitat:

$$S_{sim} = 10^{-19} S_{real}$$

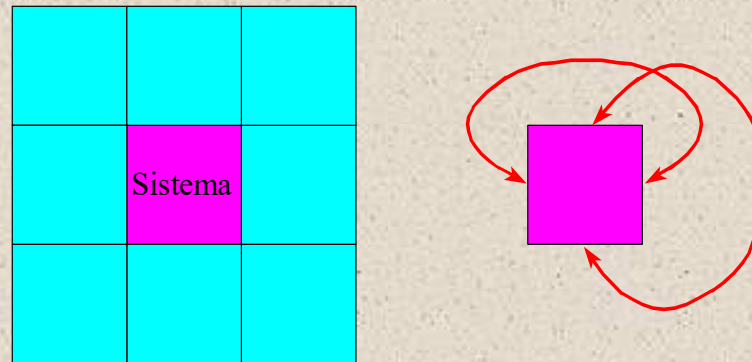
$$L_{sim} = 10^{-10} L_{real}$$

$$\frac{L_{sim}}{S_{sim}} = 10^9 \frac{L_{real}}{S_{real}}$$

- El sistema simulat esta molt més afectat pels contorns ($\times 10^9$) que el sistema real
- Per minimitzar aquest problema es fan servir les condicions periòdiques de contorn (Periodic Boundary Conditions)

Condicions Periòdiques de Contorn

Es simula un sistema infinit (però periòdic). Per tant es considera que el sistema que simulem esta envoltat de còpies idèntiques en totes les direccions. Això equival a considerar que la nostra xarxa d'espins té una estructura toroidal



Atenció, les CPC corregeixen els efectes del pes del contorn, perquè simulem un sistema infinit sense contorns. Però, en canvi, introdueixen fortes correlacions espacials (sistema periòdic) → Problemes per estudiar T.F.

Ajuda: implementació PBC

- Implementació de les condicions periòdiques de contorn en el model d'Ising 2d (si hi ha interacció a 1ers veïns)
 - Matriu d'espins `INTEGER*2 S(1:L,1:L)`
 - Vector `INTEGER*4 PBC(0:L+1)`
`PBC(0) = L, PBC(1)=1, ..., PBC(L)=L, PBC(L+1)=1`
 - Els primers veïns de `S(I,J)` seran
 - `S(I,PBC(J+1))`
 - `S(I,PBC(J-1))`
 - `S(PBC(I+1),J)`
 - `S(PBC(I-1),J)`
- Caldria fer modificacions si tinguéssim interaccions a segons veïns o bé si la xarxa és rectangular i no quadrada

Subrutina pel càlcul de l'energia

Molt similar al càlcul de la imantació, però cal passar també el vector PBC

```
REAL*8 FUNCTION ENERG(S,L,PBC)
INTEGER*2 S(1:L,1:L)
INTEGER*4 I,J,L
INTEGER*4 PBC(0:L+1)
REAL*8 ENE
ENE=0.0D0
DO I =1,L
    DO J=1,L
        ENE=ENE-S(i,j)*S(PBC(I+1),J)
+           -S(i,j)*S(I, PBC(J+1))
    ENDDO
ENDDO
ENERG=ENE
RETURN
END
```

Aquesta function la crideu al programa principal, per exemple, fent:

```
ENE=ENERG(S,L,PBC)
```

P2-exercici-1.f

Modifiqueu l'exercici `P1-exercici-3.f`, afegint la defició de les PBC i la funció `ENERG` i avalueu l'energia de la matriu que heu generat a l'atzar.

Feu-ho per $L=32$

Esperem un valor proper a zero

Si poseu tota la matriu a $S=+1$, tindreu una energia de $-2*L*L$

Hot spot de l'algorisme de Metropolis

- S'escull un spin a l'atzar $S(I, J)$ escollint dos indexs I i J a l'atzar
- Es calcula el canvi d'energia que es produiria si es girés aquest spin
- S'accepta o es rebutja

La programació en aquest punt ha de ser molt optimitzada perquè aquesta part s'executarà milions de vegades

Ajuda: Increment d'energia DE

Si es proposa el gir $S(I, J) \rightarrow -S(I, J)$ (sense fer-lo) l'increment d'energia ΔE associat al canvi seria

$$\begin{aligned}\Delta E &= E_{\text{final}} - E_{\text{inicial}} = \\ &= [\dots - (-S(I, J)) * \text{sumaveins} - \dots] - [\dots - S(I, J) * \text{sumaveins} - \dots] = \\ &= 2 * S(I, J) * \text{sumaveins}\end{aligned}$$

Es a dir, fem:

$$\begin{aligned}\text{suma} &= S(I, \text{PBC}(J + 1)) + S(I, \text{PBC}(J - 1)) + \\ &+ S(\text{PBC}(I + 1), J) + S(\text{PBC}(I - 1), J)\end{aligned}$$

$$\text{DE} = 2 * S(I, J) * \text{suma}$$

Ajuda: Probabilitat d'acceptació

- Probabilitat d'acceptació
 - Si $DE \leq 0$ s'accepta el canvi $S(I, J) = -S(I, J)$
 - Si $DE > 0$ es treu un nombre DELTA *uniforme* $U(0,1)$
 - Si $DELTA < \exp(-DE/TEMP)$ s'accepta el canvi
 $S(I, J) = -S(I, J)$
 - Si $DELTA > \exp(-DE/TEMP)$ no s'accepta

Ajuda: inici programa i dades d'entrada

C234567

IMPLICIT NONE

← Molt recomanable

C DECLARACIO DE VARIABLES

INTEGER*4 L

PARAMETER (L=32)

REAL*8 TEMP

INTEGER*4 SEED

INTEGER*4 MCTOT

REAL*8 genrand_real2

← Bloc de declaracio de variables

.....

← Afegirem totes les declaracions que calguin

C INSTRUCCIONS EXECUTABLES

TEMP = 2.4D0

SEED = 234567

MCTOT = 10000

← Primer, definir les dades d'entrada

Ajuda: dades calculades i inicialitzacions

C234567

.....

C CALCULATED VARIABLES

N=L*L



C Definim PBC

PBC(0)=L

PBC(L+1)=1

do i=1,L

 PBC(i)=i

enddo

C

CALL init_genrand(SEED)

Totes aquestes variables i vectors també els heu d'anar declarant a la part de dalt:

INTEGER*4 N

INTEGER*4 PBC(0:161)

Mesurar l'energia de 2 maneres diferents

A l'inici de la simulació, quan hem generat la matriu inicial, avaluem l'energia (ENE) i , si volem, la imantació (MAG) inicials.

```
ENE = ENERG(S,L,PBC)
```

Cada vegada que acceptem un canvi, actualitzem l'energia fent

```
ENE = ENE+DE
```

(noteu que això ho farem en 2 llocs del codi)

Cada vegada que acaba un pas de MC, avaluem l'energia de tota la matriu una altra vegada, en una variable diferent

```
ENEBIS = ENERG(S,L,PBC)
```

I escrivim el dos valors per pantalla: haurien de donar el mateix !!

```
WRITE(*,*) IMC, ENE, ENEBIS
```


MC-1.f

Copieu el codi P2-exercici-1.f, a un nou arxiu MC1.f i escriviu el codi per a simular el model d'Ising 2D amb l'algorisme de Metropolis

Feu que el codi escrigui per la pantalla, cada pas de MC, el index que indica el pas IMC, la imantació (MAG) i l'energia (ENE i ENEBIS)

Compileu i executeu el codi per $L=32$, $TEMP=1.3$ $MCTOT=3000$

Comprovacions:

Les columnes ENE i ENEBIS han de donar sempre igual

Per $TEMP=1.3$, al final hem de tenir aproximadament:

$$ENE \approx -32*32*2 = -2048$$

$$MAG \approx +/- L*L = +/- 1024$$

Per $TEMP=3.6$, al final hem de tenir ENE \approx negativa i petita

$$MAG \approx \text{petita}$$

MC-1.f

Sortida per pantalla L=32, TEMP=1.3 MCTOT=3000

MC=	2963	ENERGIA =	-2024.00000000000000	-2024.00000000000000	MAGNE =	-1018
MC=	2964	ENERGIA =	-2016.00000000000000	-2016.00000000000000	MAGNE =	-1016
MC=	2965	ENERGIA =	-2048.00000000000000	-2048.00000000000000	MAGNE =	-1024
MC=	2966	ENERGIA =	-2040.00000000000000	-2040.00000000000000	MAGNE =	-1022
MC=	2967	ENERGIA =	-2032.00000000000000	-2032.00000000000000	MAGNE =	-1020
MC=	2968	ENERGIA =	-2004.00000000000000	-2004.00000000000000	MAGNE =	-1012
MC=	2969	ENERGIA =	-1976.00000000000000	-1976.00000000000000	MAGNE =	-1002
MC=	2970	ENERGIA =	-2012.00000000000000	-2012.00000000000000	MAGNE =	-1010
MC=	2971	ENERGIA =	-2028.00000000000000	-2028.00000000000000	MAGNE =	-1016
MC=	2972	ENERGIA =	-2028.00000000000000	-2028.00000000000000	MAGNE =	-1016
MC=	2973	ENERGIA =	-2036.00000000000000	-2036.00000000000000	MAGNE =	-1020
MC=	2974	ENERGIA =	-2024.00000000000000	-2024.00000000000000	MAGNE =	-1016
MC=	2975	ENERGIA =	-2032.00000000000000	-2032.00000000000000	MAGNE =	-1020
MC=	2976	ENERGIA =	-2040.00000000000000	-2040.00000000000000	MAGNE =	-1022
MC=	2977	ENERGIA =	-2024.00000000000000	-2024.00000000000000	MAGNE =	-1018
MC=	2978	ENERGIA =	-2040.00000000000000	-2040.00000000000000	MAGNE =	-1022
MC=	2979	ENERGIA =	-2040.00000000000000	-2040.00000000000000	MAGNE =	-1022
MC=	2980	ENERGIA =	-2040.00000000000000	-2040.00000000000000	MAGNE =	-1022
MC=	2981	ENERGIA =	-2040.00000000000000	-2040.00000000000000	MAGNE =	-1022
MC=	2982	ENERGIA =	-2032.00000000000000	-2032.00000000000000	MAGNE =	-1020
MC=	2983	ENERGIA =	-2004.00000000000000	-2004.00000000000000	MAGNE =	-1012
MC=	2984	ENERGIA =	-1988.00000000000000	-1988.00000000000000	MAGNE =	-1004
MC=	2985	ENERGIA =	-2008.00000000000000	-2008.00000000000000	MAGNE =	-1014
MC=	2986	ENERGIA =	-2032.00000000000000	-2032.00000000000000	MAGNE =	-1020
MC=	2987	ENERGIA =	-2048.00000000000000	-2048.00000000000000	MAGNE =	-1024
MC=	2988	ENERGIA =	-2040.00000000000000	-2040.00000000000000	MAGNE =	-1022
MC=	2989	ENERGIA =	-2048.00000000000000	-2048.00000000000000	MAGNE =	-1024
MC=	2990	ENERGIA =	-2032.00000000000000	-2032.00000000000000	MAGNE =	-1020
MC=	2991	ENERGIA =	-2028.00000000000000	-2028.00000000000000	MAGNE =	-1018
MC=	2992	ENERGIA =	-2028.00000000000000	-2028.00000000000000	MAGNE =	-1018
MC=	2993	ENERGIA =	-2016.00000000000000	-2016.00000000000000	MAGNE =	-1016
MC=	2994	ENERGIA =	-2016.00000000000000	-2016.00000000000000	MAGNE =	-1016
MC=	2995	ENERGIA =	-2032.00000000000000	-2032.00000000000000	MAGNE =	-1020
MC=	2996	ENERGIA =	-2016.00000000000000	-2016.00000000000000	MAGNE =	-1016
MC=	2997	ENERGIA =	-2008.00000000000000	-2008.00000000000000	MAGNE =	-1014
MC=	2998	ENERGIA =	-2024.00000000000000	-2024.00000000000000	MAGNE =	-1018
MC=	2999	ENERGIA =	-2032.00000000000000	-2032.00000000000000	MAGNE =	-1020
MC=	3000	ENERGIA =	-2040.00000000000000	-2040.00000000000000	MAGNE =	-1022

Millora: MC1-millorat.f

El càlcul de l'exponencial és la part que consumeix més temps de CPU.

Pot tabular-se, ja que DE només pren uns pocs valors.

$$DE = -8, -4, 0, 4, 8$$

Definirem un vector $W(DE)$

```
REAL * 8      W(-8 : 8)
```

```
.....
```

```
DO DE = -8, 8
```

```
    W(DE) = d exp(-dfloat(DE) / TEMP)
```

```
ENDDO
```

Atenció perquè ara DE ha de ser integer