



Simulazione Monte Carlo del
modello di Ising

TECNICA MONTE CARLO

- Con tecnica Monte Carlo si intende una vasta categoria di tecniche e algoritmi che permettono di simulare numericamente un processo.
- l'idea è di trarre stime attraverso simulazioni.
- Nasce insieme ai primi calcolatori elettronici a metà degli anni 40 a Los Alamos nell'ambito del progetto Manhattan
- il nome Monte Carlo viene dato da Nicholas Metropolis facendo riferimento al famoso Casinò.



ALGORITMO DI METROPOLIS

- L'algoritmo che verrà sviluppato appartiene alla famiglia dei Monte Carlo dinamici
- L'algoritmo di Metropolis è basato su una strategia di modifiche nella configurazione di un sistema che dipendono da una decisione di accettazione-rigetto
- Da una configurazione arbitraria φ_0 si costruisce una successione di configurazioni $\varphi_1, \varphi_2, \dots, \varphi_n$ visitando tutti i siti n e si propone una mossa che viene accettata con probabilità: $P = \min(1, e^{-\delta E})$ e rifiutata con probabilità $1 - P$
- δE è la variazione di energia che si avrebbe accettando la mossa



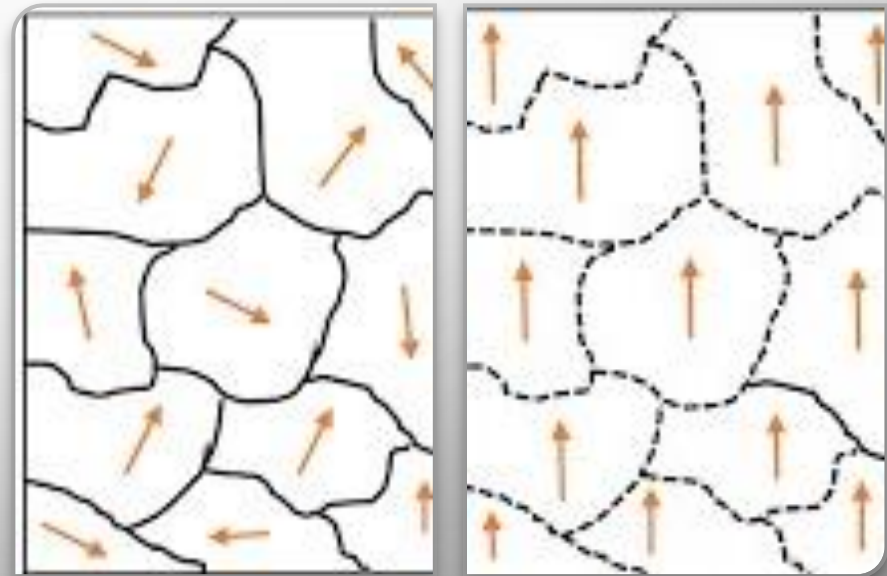
ALGORITMO DI METROPOLIS (2)

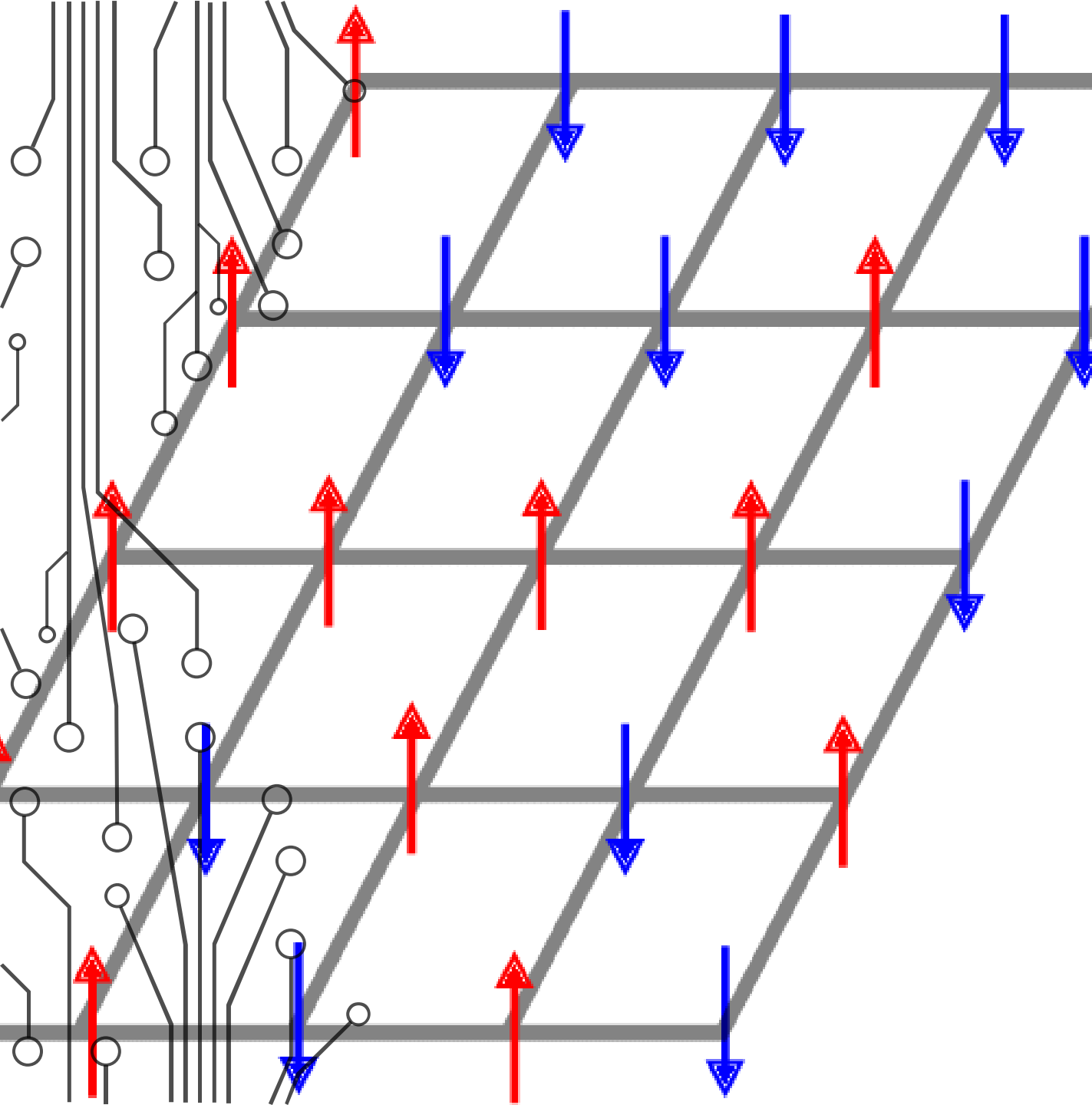
- La correttezza del metodo Metropolis è garantita dal principio di “bilancio dettagliato”
- si veda p come una matrice stocastica $p = P(j \rightarrow i)$
- p è detta reversibile rispetto alla distribuzione π se vale: $\forall(i, j): \pi_i P(i \rightarrow j) = \pi_j P(j \rightarrow i)$
- Se questo succede π è sicuramente distribuzione di equilibrio di p
- L’algoritmo di Metropolis soddisfa il bilancio dettagliato



MODELLO DI ISING

- Si tratta di un modello ideato per descrivere la transizione tra ferromagnetismo e paramagnetismo
- Il ferromagnetismo è la proprietà di alcuni materiali di magnetizzarsi molto intensamente sotto l'azione di un campo magnetico esterno e di restarlo a lungo quando il campo si annulla
- Questa proprietà è mantenuta solamente al di sotto di una determinata temperatura: la temperatura di Curie
- Se $T > T_c$ il materiale diventa paramagnetico
- La differenza tra ferromagneti e paramagneti è la permanenza di una magnetizzazione totale





MODELLO DI ISING (2)

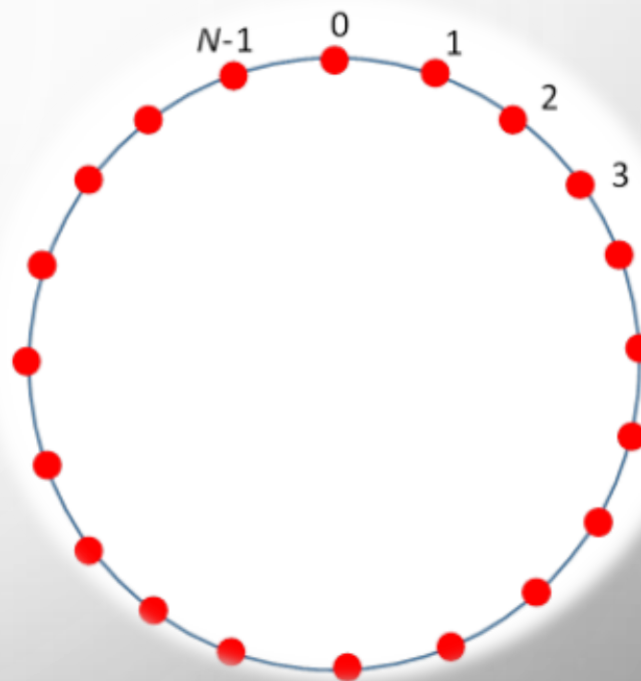
- Il modello è quello di un reticolo con N siti fissi con condizioni al contorno periodiche
- Ogni sito è definito da una variabile casuale di spin che ha due orientazioni possibili ± 1
- $E(s_1, \dots, s_N) = -J \sum s_i s_j - B \sum_{i=1}^N s_i$ è l'energia del sistema
- $Z_N(E, T) = \sum_{s_1=-1}^1 \dots \sum_{s_N=-1}^1 e^{-\beta E(s_1, \dots, s_N)}$ è la funzione di partizione

MODELLO IN UNA DIMENSIONE

- il reticolo di una dimensione è una linea formata da N siti con condizioni di periodicità al contorno
- Il sistema ha un soluzione analitica (Ernst Ising)
- La magnetizzazione media è:

$$\langle M \rangle = \frac{1}{N\beta} \frac{\partial}{\partial \beta} \ln(Z(\beta)) = \frac{\sinh(\beta B)}{\sqrt{e^{-4\beta J} + \sinh^2(\beta B)}}$$

- È funzione di β , J e B per tutti i loro valori quindi non presenta transizioni di fase
- Non si ha magnetizzazione spontanea ($\langle M \rangle = 0$ se $B = 0$)



MODELLO IN UNA DIMENSIONE - MATLAB

- si divide il reticolo in due vettori, uno per i siti dispari ed uno per i siti pari
- Questo per facilitare la ricerca dei primi vicini ed ottimizzare il programma
- Mossa di metropolis diventa:

```
r = rand(1,N/2);  
dE_pari = 2*J.*spin_pari.*(spin_dispari + vicini_disp +B);  
prob_pari = exp(-dE_pari*b);  
accept = r < prob_pari;  
spin_pari(accept) = -spin_pari(accept);
```

- Essendo una semplice inversione di spin $\delta E = 2E$

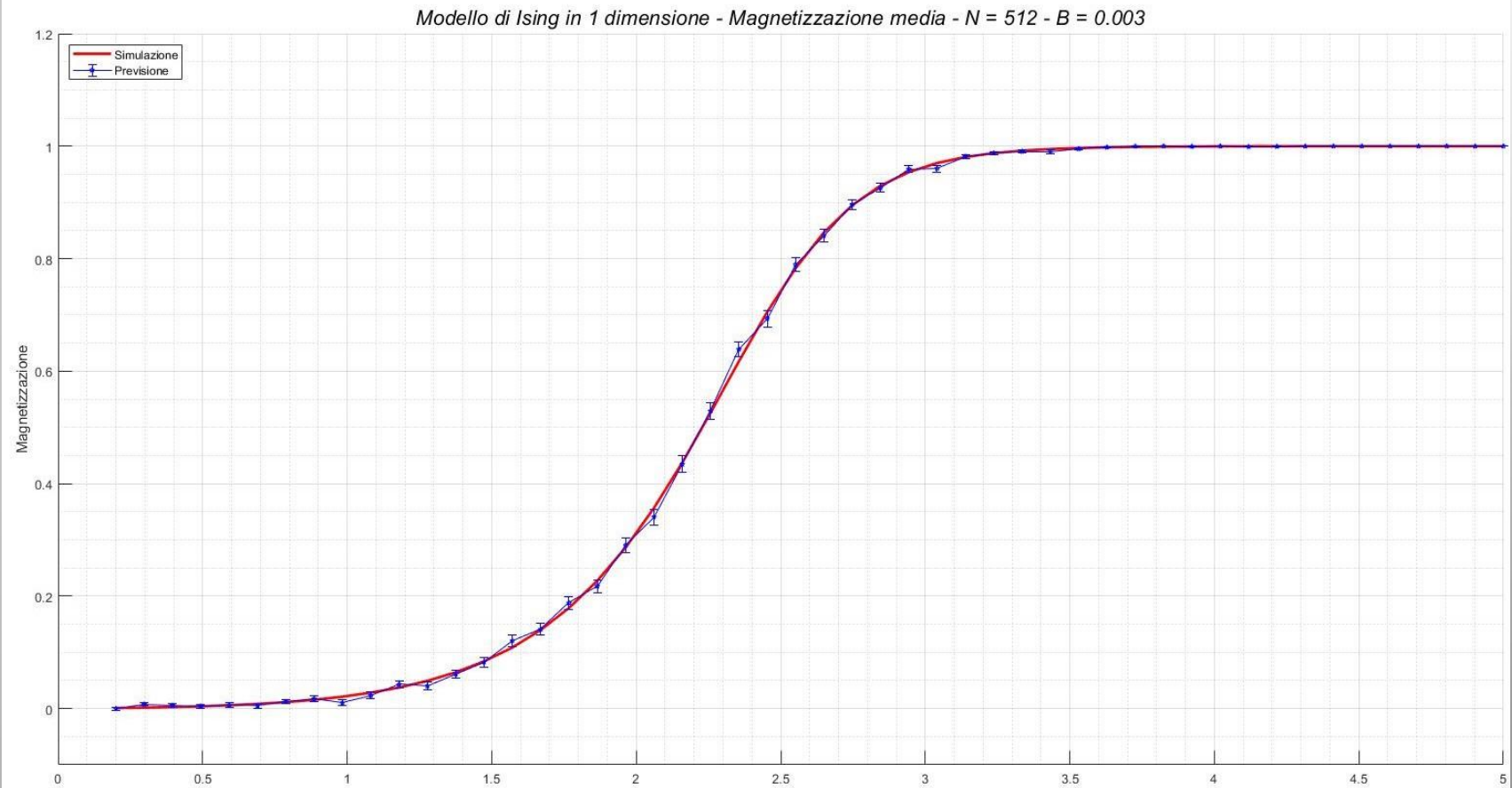
MODELLO IN UNA DIMENSIONE – MATLAB (2)

I primi vicini vengono
trovati mediante la
funzione circshift

```
A =  
    0    0    0    0    0    1    0    0    0    0    0  
  
>> circshift(A, 1)  
  
ans =  
    0    0    0    0    0    0    1    0    0    0    0  
  
>> circshift(A, -1)  
  
ans =  
    0    0    0    0    1    0    0    0    0    0    0
```

MODELLO IN UNA DIMENSIONE – MATLAB (3)

- In una dimensione la simulazione segue puntualmente la soluzione analitica
- Il grafico è tratto da una simulazione con $2e5$ sweeps e $1e3$ cicli di termalizzazione
- Al fine di ottimizzare il programma il corpo del codice viene suddiviso in diverse funzioni
- In una dimensione è stato pressoché superfluo ma tornerà molto utile in seguito



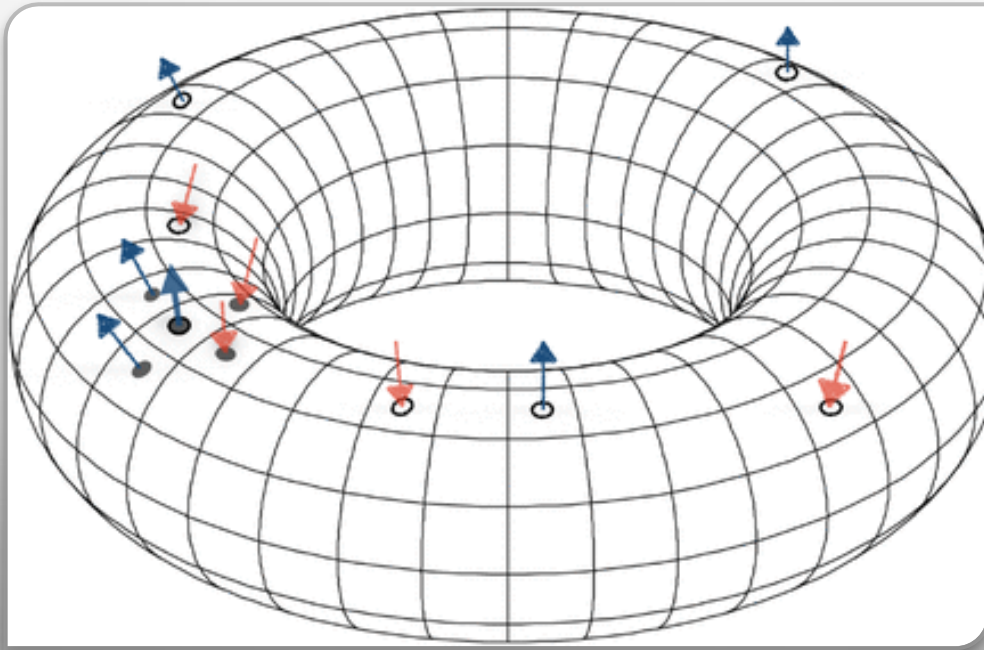
```
for cnt = 1:Nbeta
    b = beta(cnt);
    [spin_pari, spin_dispari] = thermalizzazione1D(therm, spin_pari, spin_dispari, J, B, b, N);
    %Inizio evoluzione sistema

    [spin_pari, spin_dispari, Magmedia] = evoluz1D(sweeps, spin_pari, spin_dispari, Magmedia, J, B, b, skip, N);

    Magn(cnt) = mean(Magmedia);           %magnetizzazione media per spin
    errMag(cnt) = std(Magmedia);         %errore sulla magnetizzazione media

    waitbar(cnt/Nbeta);
end
```

MODELLO IN DUE DIMENSIONI



- Reticolo piano con $N \times N$ siti sempre con condizioni periodiche al contorno
- Soluzione analitica per la magnetizzazione spontanea fornita da Lars Onsager:

$$M = (1 - \sin h(2\beta J)^{-4})^{\frac{1}{8}}$$

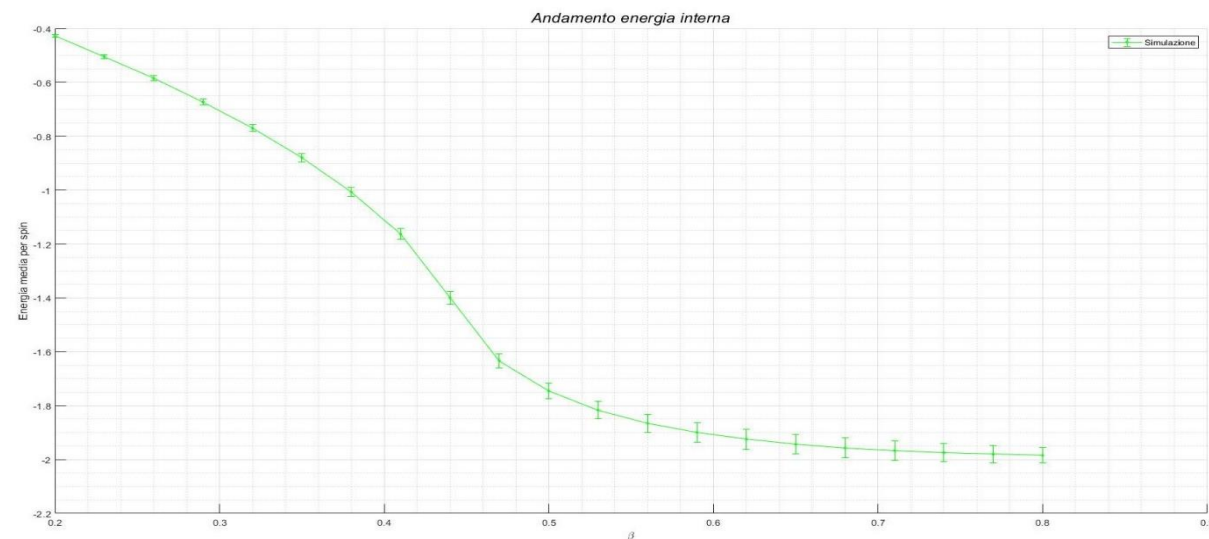
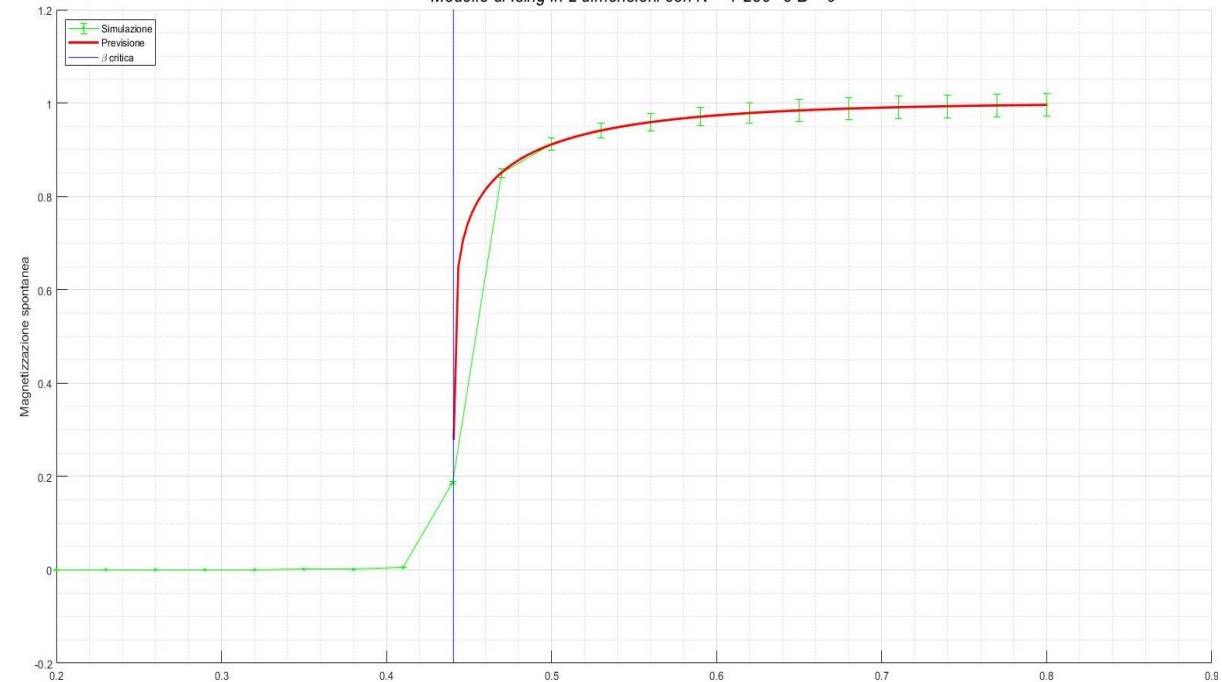
- Transizione di fase tra ferromagnetismo e paramagnetismo per $\beta \approx 0,44$

MODELLO IN DUE DIMENSIONI - MATLAB

- Considerando un reticolo di lato L composto da N siti pari e N siti dispari si ha un totale di $(2*N)^2$ siti risulta quindi comodo creare uno struct composto da quattro matrici $N \times N$ per descriverlo
- Viene studiata la magnetizzazione spontanea
- Simulazioni con $2e4$ sweeps e $1e3$ cicli di termalizzazione
- Sotto alla temperatura di Curie il materiale assume comportamento ferromagnetico
- Energia interna tende al valore di $-2J$ per basse temperature

```
spin.pp = sign(0.5 - rand(N,N)); %pari-pari  
spin.dp = sign(0.5 - rand(N,N)); %dispari-pari  
spin.dd = sign(0.5 - rand(N,N)); %dispari-dispari  
spin.pd = sign(0.5 - rand(N,N)); %pari-dispari
```

Modello di Ising in 2 dimensioni con $N = 4*256^2$ e $B = 0$



MODELLO IN DUE DIMENSIONI – MATLAB (2)

- Vengono create quattro funzioni che trovano i primi vicini di ognuna delle quattro matrici e restituiscono la variazione di energia
- Si utilizza ancora il comando circshift che oltre a permettere di mantenere le condizioni al contorno permette anche di ruotare una matrice nelle quattro direzioni

```
function dE = trovaviciniPariPari(pari_pari, dispari_pari, pari_dispari, J,N)
%trova i vicini degli elementi pari-pari e calcola la variazione di energia

pp_vicini = dispari_pari + circshift(dispari_pari, [0 1]) + pari_dispari + circshift(pari_dispari, [1 0]);

dE = J.*pari_pari.*(pp_vicini); %variazione energia se inverto spin
```

```
A =
     0     0     0
     0     1     0
     0     0     0

>> circshift(A , [1 0])

ans =
     0     0     0
     0     0     0
     0     1     0

>> circshift(A , [-1 0])

ans =
     0     1     0
     0     0     0
     0     0     0

>> circshift(A , [0 1])

ans =
     0     0     0
     0     0     1
     0     0     0

>> circshift(A , [0 -1])

ans =
     0     0     0
     1     0     0
     0     0     0
```

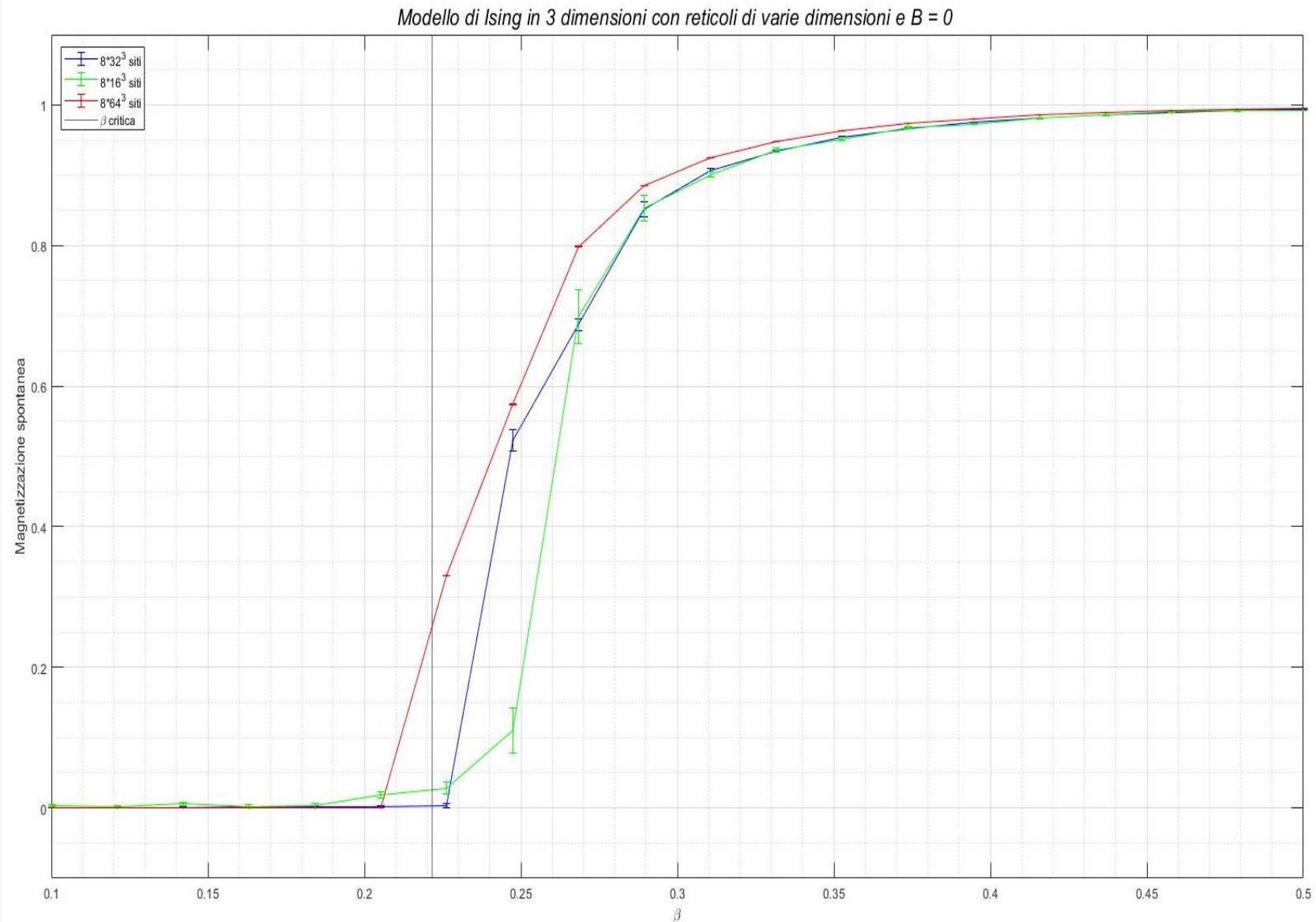
MODELLO IN TRE DIMENSIONI

```
spin.ppp = sign(0.5 - rand(N,N,N)); %pari-pari-pari
spin.ppd = sign(0.5 - rand(N,N,N)); %pari-pari-dispari
spin.pdp = sign(0.5 - rand(N,N,N)); %pari-dispari-pari
spin.dpp = sign(0.5 - rand(N,N,N)); %dispari-pari-pari
spin.ddp = sign(0.5 - rand(N,N,N)); %dispari-dispari-pari
spin.dpd = sign(0.5 - rand(N,N,N)); %dispari-pari-dispari
spin.pdd = sign(0.5 - rand(N,N,N)); %pari-dispari-dispari
spin.ddd = sign(0.5 - rand(N,N,N)); %dispari-dispari-dispari
```

- C'è da considerare un reticolo cubico con un lato di lunghezza L composto da N siti pari e N siti dispari quindi da un totale di $(2N)^3$ siti.
- Risulta quindi comodo creare il reticolo come uno struct di otto matrici differenti
- Dalla terza dimensione in poi non esiste una soluzione analitica
- studi numerici permettono di collocare la transizione di fase attorno a $\beta_c \approx 0,22165$

MODELLO IN TRE DIMENSIONI - MATLAB

- Lo script rimane grossomodo lo stesso con la differenza che alla variazione di energia contribuiscono sei primi vicini
- Con la crescita delle dimensioni del reticolo la curva mostra l'inizio della salita sempre più vicino al valore atteso di β critica
- Simulazione con $2e4$ sweeps e $1e3$ cicli di termalizzazione

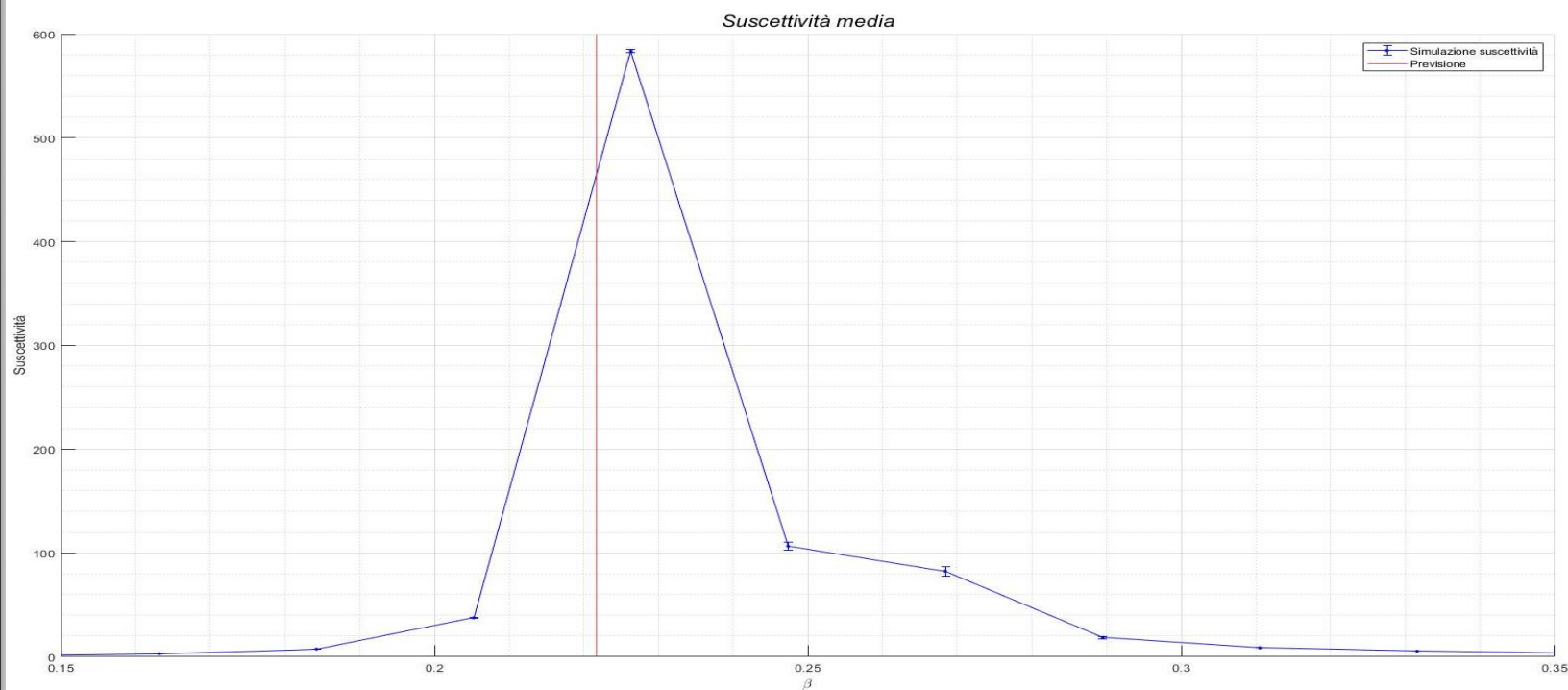
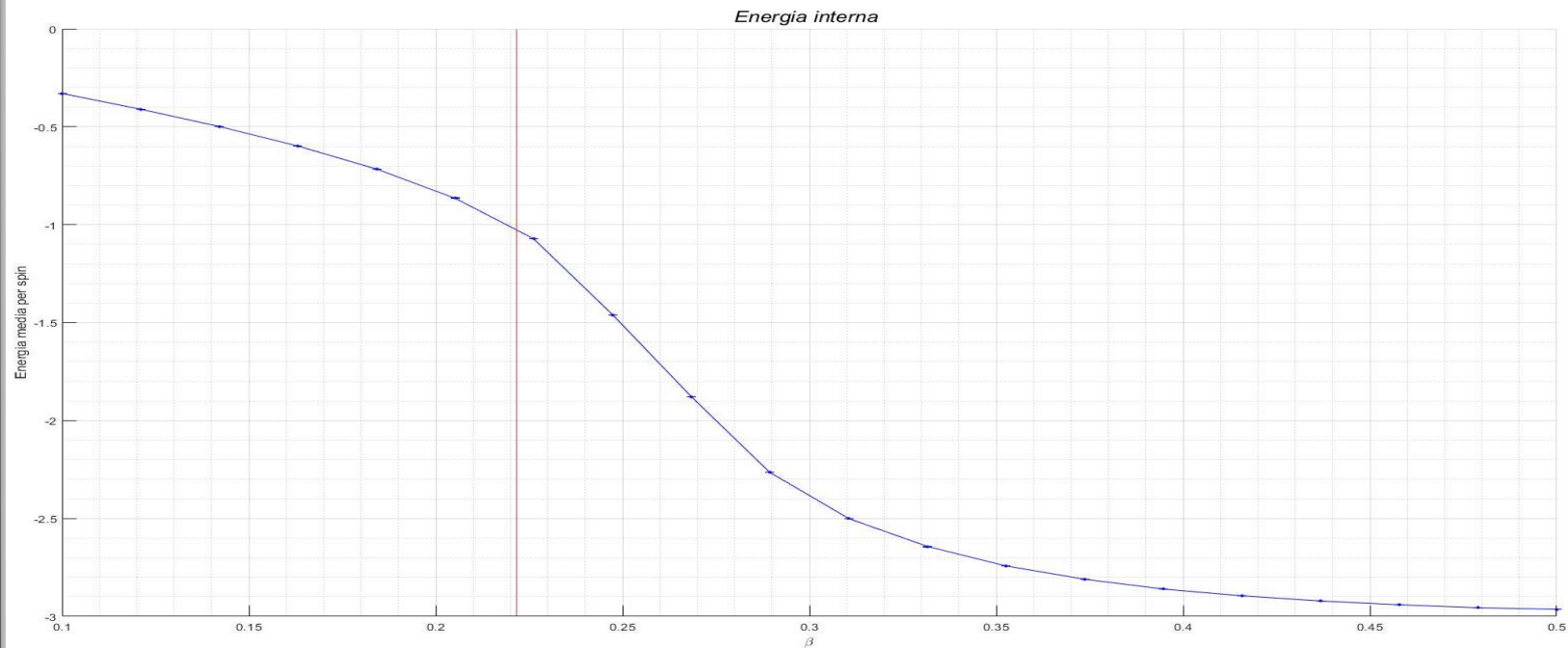


MODELLO IN TRE DIMENSIONI – MATLAB (2)

- Per quanto riguarda l'energia interna per spin ci si aspetta un andamento del tipo :

$$\begin{cases} \frac{\langle E \rangle}{N} \rightarrow 0 \text{ per } \beta \rightarrow 0 \\ \frac{\langle E \rangle}{N} \rightarrow -3J \text{ per } \beta \rightarrow \infty \end{cases}$$

- Viene riportata anche la suscettività media che ci si aspetta vada a zero sia per piccoli e grandi β con un asintoto verticale in corrispondenza della trasformazione di fase
- Simulazioni con $2e4$ sweeps e $1e3$ cicli di termalizzazione
- $N = 8 \cdot 64^3$



PROBLEMI DELL'ALGORITMO

- Qualunque programma basato sul metodo Monte Carlo risente di errori statistici che vanno come $N^{-\frac{1}{2}}$
- Ci sono inoltre due problemi fondamentali che portano errori sistematici nei Monte Carlo dinamici:
 - 1) bias di inizializzazione

Causato dalla distribuzione arbitraria di partenza della simulazione, c'è un passaggio iniziale in cui i dati non riflettono la distribuzione di equilibrio

Per ovviare questo problema si introduce la termalizzazione del sistema prima di iniziare a studiarne le proprietà

PROBLEMI DELL'ALGORITMO (2)

2) Autocorrelazione

A causa della natura stessa dell'algoritmo che ad ogni sweep propone piccole modifiche alla configurazione precedente gli stati vicini del sistema sono correlati tra loro

Questo risulta in una diminuzione dei campioni statisticamente indipendenti

Per risolvere questo problema va cercato un valore di skip in modo tale da annotare il valore un numero sweeps/skip di volte, ovvero annotare solamente i dati statisticamente indipendenti

PROBLEMI DELL'ALGORITMO (3)

- La funzione di autocorrelazione è definita come:

$$c(t) = \langle f(x_s)f(x_{s+t}) \rangle - \langle f \rangle^2$$

- È una funzione che decresce esponenzialmente nel tempo:

$$c(t) = c_0 e^{-\frac{t}{\tau}}$$

- Ricavando τ , il tempo medio in cui l'autocorrelazione cala di un fattore $1/e$, si avrà un'indicazione sul parametro skip

CALCOLO AUTOCORRELAZIONE

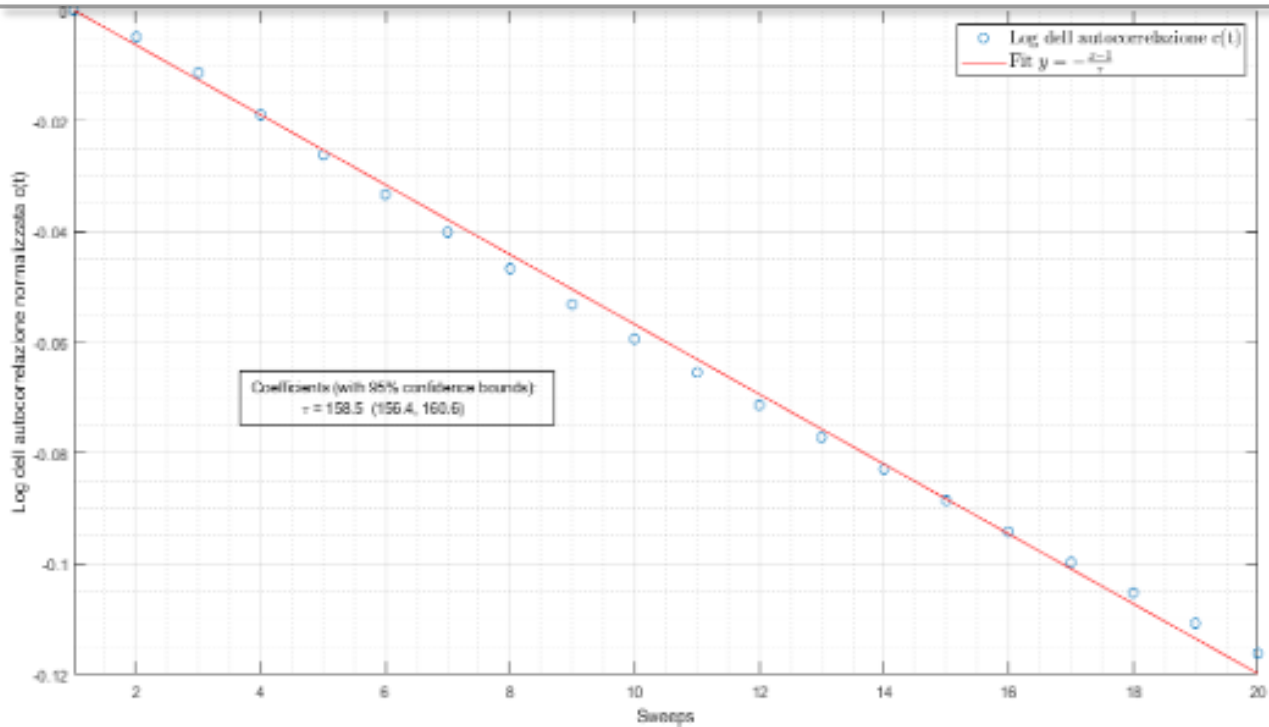
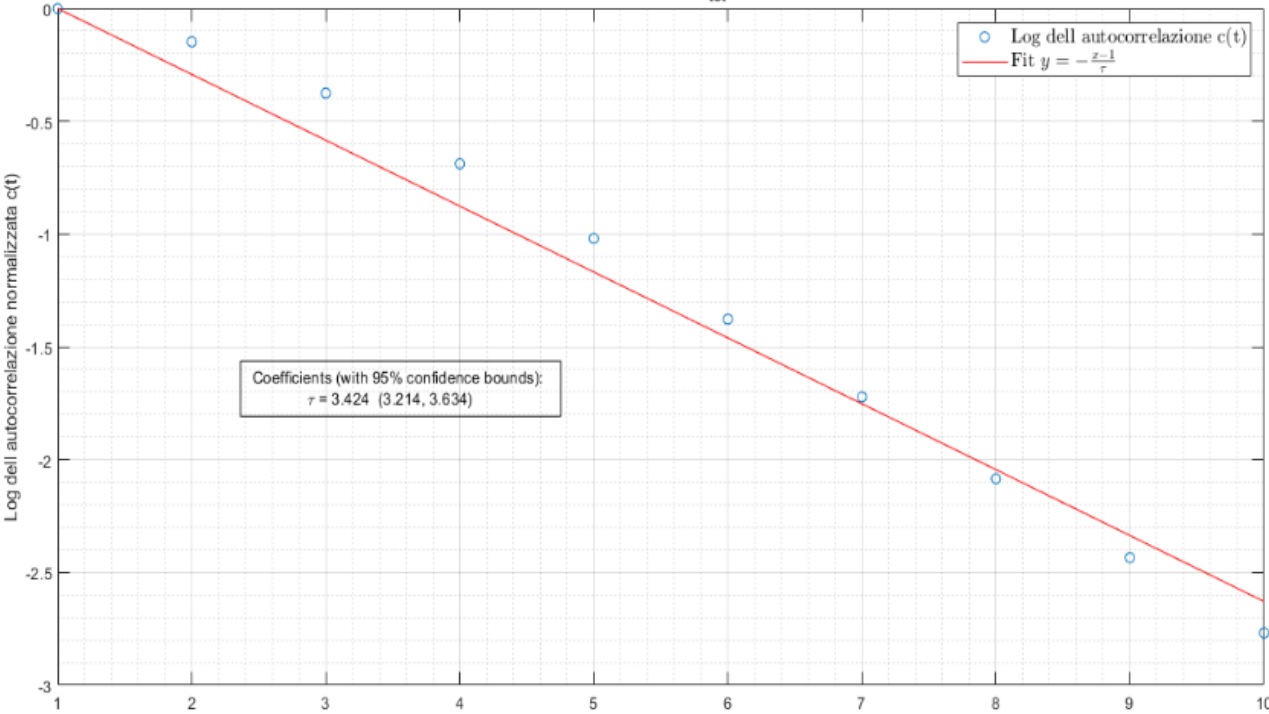
- Fissata la dimensione del reticolo $N_{tot} = 64^3$, ovvero il reticolo più grande analizzato, si calcola l'autocorrelazione per due valori di β : uno minore della temperatura di Curie ed uno in corrispondenza di essa
- L'autocorrelazione viene calcolata usando la trasformata di Fourier:

La prima parte di $c(t)$, $\langle f(x_s)f(x_{s+t}) \rangle$, può essere definita come una convoluzione e grazie al teorema di Wiener-Khintchine secondo il quale

$$F\left[\int_{-\infty}^{+\infty} f(x)f^*(x + \tau) dx\right] = |F[t(x)]|^2$$
 si può calcolare come una trasformata quindi $\langle f(x_s)f(x_{s+t}) \rangle$ è l'antitrasformata del modulo quadro della trasformata di f

```
c = fft(averageM(k,:)); %correlazione,  
%k definisce il valore di \beta a cui la stiamo calcolando  
ac = ifft(c.*conj(c)); %autocorrelazione  
  
ac = ac - (mean(averageM(k,:))2); %sottraggo la media al quadrato  
ac = (ac)./ac(1); %normalizzo l'autocorrelazione
```

CALCOLO AUTOCORRELAZIONE



- Graficando l'autocorrelazione in funzione del numero di sweeps e tramite un fit in scala logaritmica possiamo ricavare τ
- Per $\beta = 0.2$ $\tau = 3.4$ sweeps
- Per $\beta = 0.221 \approx \beta_C$ si ha $\tau = 159$ sweeps
- È evidente che più ci si avvicina alla beta critica più lenta è la decrescita dell'autocorrelazione, questo perché diventa sempre più difficile generare campioni tra loro indipendenti quando gli spin si allineano

STIMA ERRORI

- Una volta scelto lo skip di conseguenza a quanto detto (skip = 300) è possibile effettuare una stima degli errori
- L'errore è la deviazione standard della media ma va calcolata sul numero di elementi indipendenti su cui è stata calcolata la media ovvero sweeps/skip
- L'errore perciò per quanto riguarda la magnetizzazione è dM ed analogamente è stato calcolato per le altre quantità

```
%Magnetizzazione media per spin  
Mg(cnt) = abs(mean(M_media))/(8*N*N*N);  
errMagn(cnt) = std(M_media)/(8*N*N*N);  
dM = errMagn./sqrt(sweeps/skip);
```

REFERENCES

- E. Onofri, Lezioni di Metodi Probabilistici per la Fisica, seconda Edizione, settembre 2021
- A. Sokal, Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms, settembre 1996
- P. Butera e M. Comi, Extension to order b_{23} of the high-temperature expansions for the spin $-1/2$ Ising model on simple cubic and body-centered cubic lattices, 2000
- A. Pelissetto, Introduction to the Monte Carlo Method, settembre 1992
- Wikipedia, Teorema di Wiener-Khintchine
- Relazione di Armanini Elisabetta ed Irene Piazza per calcolo dell'autocorrelazione e stima del coefficiente τ , gennaio 2021