

Simulazione Monte Carlo del modello di Ising bidimensionale

Luglio 2023

Francesco Grieco, Mattia Orlandini

Indice

1	Introduzione al modello di Ising	3
2	Metodo Monte Carlo	5
2.1	Introduzione alle simulazioni Monte Carlo	5
2.2	Campionamento per importanza e con catene di Markov . . .	6
2.3	Algoritmo di Metropolis	7
3	Studio Monte Carlo del modello di Ising	8
3.1	Implementazione algoritmo di Metropolis	8
3.2	Parametri temporali simulazione	10
3.2.1	Tempi di termalizzazione	10
3.2.2	Tempi di correlazione	13
3.3	Osservabili termodinamiche	16
3.4	Temperatura critica e parametro di Binder	18
3.5	Esponente critico dinamico	20
3.6	Problemi dell'algoritmo di Metropolis	22
4	Algoritmo di Wolff	22
4.1	Implementazione	22
4.2	Tempi di termalizzazione	24
4.3	Tempi di correlazione ed esponente critico dinamico	25
4.4	Osservabili termodinamiche	29
5	Conclusioni	30

6 Appendice	31
Riferimenti bibliografici	32

1 Introduzione al modello di Ising

Il modello di Ising è un modello di ferromagnete definito su un generico reticolo con numero di coordinazione z ¹; ad ogni sito reticolare è associata una variabile classica σ che può assumere solo valore ± 1 chiamata spin. Per definizione l'Hamiltoniana di tale modello è:

$$H(\{\sigma\}) = -J \sum_{(i,j)} \sigma_i \sigma_j - h \sum_i \sigma_i \quad (1)$$

Il pedice di ciascuno spin indica la posizione reticolare; $J > 0$ e h sono rispettivamente l'accoppiamento ferromagnetico ed il campo magnetico esterno. Infine il simbolo (i, j) sotto la sommatoria sta ad indicare che essa è estesa solo alle coppie di siti reticolari primi vicini.

Quando $h = 0$ il sistema possiede una simmetria globale discreta Z_2 , ovvero l'Hamiltoniana è invariante sotto la trasformazione $\sigma_i \rightarrow -\sigma_i$ per tutti gli spin del reticolo. Dalla teoria (vedere [1], [2]) risulta che in dimensione $d > 1$ e $h = 0$ il modello presenta una transizione di fase.

Nel seguito ci proponiamo di studiare, tramite simulazione Monte Carlo, il comportamento del modello definito su un reticolo quadrato di lato L e quindi con un numero di spin $N = L^2$.

Consideriamo il sistema in contatto con un bagno termico alla temperatura T ($\beta = \frac{1}{kT}$); quindi usiamo l'insieme canonico e indichiamo la media canonica di una generica osservabile $A(\{\sigma\})$ con $\langle A \rangle_T$.

Una transizione di fase si manifesta nella non analiticità alla temperatura critica T_c di alcune grandezze termodinamiche associate al sistema in esame (vedere [1], [2]). Nel nostro caso le osservabili fisiche di riferimento sono la magnetizzazione e l'energia per spin definite come:

$$m \equiv \frac{1}{N} \sum_{i=1}^N \sigma_i \quad (2)$$

$$e \equiv -\frac{J}{N} \sum_{(i,j)} \sigma_i \sigma_j \quad (3)$$

Nel presente elaborato denoteremo le grandezze intensive con lettere minuscole mentre quelle estensive con lettere maiuscole; ad esempio per la magnetizzazione totale si avrà $M = Nm = \sum_{i=1}^N \sigma_i$. Si osservi che $m \in [-1, 1]$, $e \in [-2J, 2J]$.

¹Il numero di coordinazione di un reticolo è il numero primi vicini di un qualsiasi sito del reticolo

$\langle |m| \rangle_T$ è il parametro d'ordine della transizione: per $T > T_c$, cioè nella fase disordinata è nullo, mentre al di sotto della temperatura critica $\langle |m| \rangle_T = 1$, cioè tutti gli spin sono orientati allo stesso modo. A livello teorico una transizione di fase è legata ad una rottura spontanea della simmetria; nel caso del modello di Ising lo stato del sistema a $T < T_c$ non è più invariante sotto Z_2 e lo spazio delle configurazioni è separato in due regioni disgiunte che vengono mappate l'una nell'altra dalla simmetria rotta.

Altre due quantità fisiche rilevanti e misurabili sperimentalmente sono la suscettibilità magnetica e il calore specifico; esse sono due funzioni di risposta e sono collegate alle fluttuazioni (più precisamente alla varianza) della relativa grandezza termodinamica:

$$\chi \equiv \frac{\partial \langle |m| \rangle_T}{\partial h} = \frac{\beta}{N} (\langle M^2 \rangle - \langle M \rangle^2) = N\beta (\langle m^2 \rangle - \langle m \rangle^2) \quad (4)$$

$$c \equiv \frac{\partial \langle e \rangle_T}{\partial T} = \frac{\beta^2}{N} (\langle E^2 \rangle - \langle E \rangle^2) = N\beta^2 (\langle e^2 \rangle - \langle e \rangle^2) \quad (5)$$

Le non analiticità delle grandezze termodinamiche del sistema in vicinanza del punto critico sono descritte dagli esponenti critici che elenchiamo di seguito.

$$c \sim |t|^{-\alpha} \quad h = 0, t \rightarrow 0 \quad (6)$$

$$m \sim (-t)^\beta \quad t \rightarrow 0^- \quad (7)$$

$$\chi \sim |t|^{-\gamma} \quad h = 0, t \rightarrow 0 \quad (8)$$

$$h \sim |m|^\delta \quad t = 0 \quad (9)$$

$$\xi \sim |t|^{-\nu} \quad h = 0, t \rightarrow 0 \quad (10)$$

$$G_c^{(2)}(r) = \frac{1}{r^{d-2+\eta}} \quad h = 0, t \rightarrow 0 \quad (11)$$

Nelle precedenti relazioni $t = \frac{T-T_c}{T_c}$ è la temperatura ridotta, mentre ξ e $G_c^{(2)}$ sono rispettivamente la lunghezza di correlazione e la funzione di correlazione connessa a due punti. Definiremo meglio in seguito cosa sia la lunghezza di correlazione, mentre per una appropriata descrizione della funzione di correlazione rimandiamo a [1].

Il modello di Ising in $d=2$ (e $h = 0$) è esattamente risolubile e i relativi esponenti sono:

$$\alpha = 0 \quad \beta = \frac{1}{8} \quad \gamma = \frac{7}{4} \quad \delta = 15 \quad \nu = 1 \quad \eta = \frac{1}{4} \quad (12)$$

L'andamento del parametro d'ordine con la temperatura è:

$$\langle |m| \rangle_T = \left(1 - \left(\sinh \left(\frac{2}{T} \right) \right)^{-4} \right)^{\frac{1}{8}} \quad (13)$$

Infine la temperatura critica è pari a:

$$T_c = \frac{2}{\log(1 + \sqrt{2})} \simeq 2.269 \quad (14)$$

2 Metodo Monte Carlo

2.1 Introduzione alle simulazioni Monte Carlo

L'obiettivo tipico di una simulazione Monte Carlo di una sistema a contatto con un bagno termico è il calcolo del valore medio di alcune grandezze termodinamiche rispetto alla distribuzione canonica. Sia C una possibile configurazione del sistema (nel modello di Ising una configurazione è determinata dal valore degli N spin). In generale il prototipo di integrale da calcolare è:

$$I = \sum_C P(C) A(C) \quad (15)$$

dove $A(C)$ è una generica osservabile e $\sum_C P(C) = 1$; evidentemente $P(C)$ è una distribuzione di probabilità sullo spazio delle configurazioni del sistema (può essere una generica distribuzione).

Sia $W(C)$ un'altra distribuzione e consideriamo la grandezza aleatoria $s(C) \equiv \frac{P(C)A(C)}{W(C)}$; allora:

$$\langle s(C) \rangle_W \equiv \sum_C s(C) W(C) = \sum_C P(C) A(C) = I \quad (16)$$

Dalla statistica sappiamo che il miglior stimatore per il valore atteso di una variabile aleatoria è la media aritmetica e quindi se si estraggono N configurazioni del sistema con la distribuzione $W(C)$ si può valutare l'integrale come:

$$I \equiv \sum_C P(C) A(C) \simeq \frac{1}{N} \sum_{i=1}^N s(C_i) \quad (17)$$

Si ricordi che la media aritmetica è uno stimatore non polarizzato e consistente, ovvero tende al valore vero nel limite di un numero infinito di misure $N \rightarrow +\infty$.

2.2 Campionamento per importanza e con catene di Markov

Per ottenere una stima accurata di I dobbiamo decidere la distribuzione $W(C)$ con la quale estrarre le N configurazioni; la scelta più semplice è la distribuzione uniforme e in questo caso si parla di campionamento uniforme. Questa scelta non è ottimale; l'integrale I di interesse è una somma su un numero estremamente grande di configurazioni, ma nella maggioranza dei casi tale somma è dominata da un numero limitato di stati. Dal momento che nella simulazione viene selezionata solo una piccola frazione dello spazio delle configurazioni, se il campionamento è uniforme la probabilità di selezionare le poche configurazioni rilevanti è molto bassa e la stima di I sarà perciò poco attendibile.

Risulta opportuno scegliere $W(C)$ in modo da selezionare gli stati che contribuiscono significativamente ad I , cioè fare un campionamento per importanza. La strategia è dunque quella di usare come distribuzione di campionamento proprio $P(C)$, che nel caso di un sistema termostatoato è la distribuzione canonica; in questo modo l'equazione (17) diviene:

$$I \simeq \frac{1}{N} \sum_{i=1}^N A(C_i) \quad (18)$$

La difficoltà più grande in una simulazione Monte Carlo risiede proprio nella generazione di un insieme aleatorio di stati distribuiti secondo $P(C)$. Quasi tutti gli schemi Monte Carlo si basano su un processo di Markov come motore per generare il set di stati usati; in questo caso si parla di Markov Chain Monte Carlo.

Un processo di Markov è un meccanismo in tempo discreto che, dato uno stato i (abbreviazione per C_i) a un istante t genera un nuovo stato j al tempo successivo con una probabilità w_{ij} indipendente dal tempo e dagli stati precedentemente visitati; per questo un processo di Markov si dice senza memoria. In una simulazione Monte Carlo viene generata una catena di Markov di configurazioni (si può pensare di avere un camminatore che si muove sui diversi stati del sistema ad ogni passo temporale); il processo di Markov è scelto in modo tale che, indipendentemente dalla configurazione iniziale, dopo un tempo di termalizzazione necessario per raggiungere l'equilibrio produca stati distribuiti secondo $P(C)$. Una volta che è stato raggiunto

l'equilibrio, è sufficiente iterare il processo di Markov un numero di volte pari al tempo di misura t_{mis} scelto (in questo modo vengono selezionati $N = t_{mis}$ configurazioni) e quindi valutare I come:

$$I = \sum_{t=1}^{t_{mis}} A(C_t) = \sum_{t=1}^{t_{mis}} A_t \quad (19)$$

dove è stato posto $A(C_t) \equiv A_t$.

Affinché il processo di Markov abbia le proprietà richieste esso deve essere ergodico (ovvero partendo da un generico stato, deve poter raggiungere un qualsiasi altro se iterato per un tempo sufficiente) e soddisfare il bilancio dettagliato:

$$p_i w_{ij} = p_j w_{ji} \quad \forall i, j. \quad (20)$$

dove per semplicità di notazione è stato posto $P(C_i) = p_i$; nel seguito si avrà $p_i = \frac{e^{-\beta E_i}}{Z_C}$, con Z_C funzione di partizione canonica.

I vincoli sopra esposti lasciano comunque un buon grado di libertà nella scelta dei coefficienti di transizione w_{ij} che definiscono il processo.

2.3 Algoritmo di Metropolis

L'algoritmo di Metropolis è una possibile scelta delle probabilità di transizione w_{ij} che rispetta la condizione di bilancio dettagliato (20).

Assumendo che il sistema si trovi nella configurazione i ad un tempo t , l'algoritmo si compone di due parti principali:

1. viene proposto un nuovo stato j con probabilità simmetrica u_{ij} , cioè in modo che $u_{ij} = u_{ji}$, tra i possibili j connessi allo stato i .
2. il nuovo stato j viene accettato con probabilità $\min(1, \frac{p_j}{p_i})$; nel qual caso al tempo $t + 1$ la catena di Markov si trova nello stato j , sennò rimane nello stato i .

Quindi il camminatore si muoverà sicuramente su uno stato con probabilità maggiore, cioè $p_j > p_i$, e con certa probabilità se $p_j < p_i$. Le probabilità u_{ij} possono essere scelte in modo uniforme tra i possibili stati collegati, a condizione che in ogni stato ci sia lo stesso numero di nuove configurazioni raggiungibili, per mantenere la condizione di simmetria. È facile dimostrare che questa scelta dei coefficienti di transizione rispetta il bilancio dettagliato. Partendo da uno stato i la probabilità totale di andare in uno stato j è:

$$w_{ij} = u_{ij} \min\left(1, \frac{p_j}{p_i}\right), \quad (21)$$

ovvero la probabilità di proporre lo stato j per la probabilità che esso venga accettato. Sostituendo nella Eq. (20), risulta

$$p_j u_{ij} \min \left(1, \frac{p_j}{p_i} \right) = p_i u_{ji} \min \left(1, \frac{p_i}{p_j} \right) \quad (22)$$

dove $u_{ij} = u_{ji}$ per definizione, da cui è immediata l'uguaglianza.

Il vantaggio dell'algoritmo di Metropolis, oltre alla sua semplicità, è che è (generalmente) molto efficiente in quanto ha bisogno solo della generazione di due numeri casuali e del calcolo di due sole probabilità p_i e p_j .

3 Studio Monte Carlo del modello di Ising

3.1 Implementazione algoritmo di Metropolis

Si considerano ora le quantità in unità $J = k_b = 1$. Per il caso del modello di Ising l'algoritmo di Metropolis può essere semplificato. Questo perché si considera un reticolo quadrato, con 4 primi vicini, e le energie sono discrete. Descrivendo l'interazione di uno spin σ_i con i suoi vicini tramite un campo efficace h_{eff} , possiamo riscrivere quindi l'Hamiltoniana per lo spin σ_i :

$$H_i = \sigma_i h_{eff} = \sigma_i \sum_k \sigma_k, \quad (23)$$

dove la somma è sui primi vicini. Infatti l'energia di interazione di uno spin i con i suoi primi vicini può assumere solo 5 valori: $[-4, -2, 0, +2, +4]$, cioè vale -4 se i 4 vicini sono allineati come i , -2 se 3 vicini sono paralleli a i , ecc. Se si gira lo spin $\sigma_i \rightarrow -\sigma_i$ le interazioni cambiano segno, e diventano $[4, 2, 0, -2, -4]$. La distribuzione di probabilità degli spin è la distribuzione di Boltzman $p_j \propto \exp\left(-\frac{E_j}{T}\right)$. Consideriamo solo il contributo all'energia dello spin i -esimo, perché l'energia del resto del reticolo non cambia durante uno step Montecarlo. Per l'algoritmo di Metropolis dobbiamo calcolare i rapporti delle probabilità

$$\frac{p_j}{p_i} = \frac{\exp\left(-\frac{E_j}{T}\right)}{\exp\left(-\frac{E_i}{T}\right)} = \exp\left(-\frac{\Delta E}{T}\right), \quad (24)$$

che dipendono solo dalla differenza di energia ΔE , con p_j la probabilità dello stato con lo spin $-\sigma_i$. Essendo le energie possibili fissate, si ricava che anche ΔE può assumere solo il set finito di valori $[8, 4, 0, -4, -8]$. Per esempio il caso con $\Delta E = 8$ corrisponde a girare lo spin i che parte parallelo a tutti i suoi vicini, e quindi è energeticamente sfavorito. Osserviamo dalla Eq.

(24) che quando ho un $\Delta E \leq 0$, il rapporto $p_j/p_i \geq 1$. Ricordando che la probabilità di accettare un nuovo stato j nell'algoritmo di Metropolis è $\min(1, \frac{p_j}{p_i})$, quindi viene accettato sicuramente se $\Delta E \leq 0$.

```
@jit
def oneSweep2D(conf,T):# T temperatura
    # Realizzo un MCS selezionando gli spin in modo sequenziale
    # MCS sta per MonteCarlo sweep: in un MCS si propone una modifica di ogni spin
    N=conf.shape[0]

    for i in range(N):
        for j in range(N):
            heff=conf[(i-1)%N,j]+conf[(i+1)%N,j]+conf[i,(j+1)%N]+conf[i,(j-1)%N]
            #deltaE=2*heff*conf[i,j]
            sum=heff*conf[i,j]
            prob=LUT(T)
            if (sum<=0) or np.random.rand()<prob[sum]:
                conf[i,j]=-1.*conf[i,j]
    return conf
```

(a) Look up table, per il calcolo delle probabilità

```
@jit
def LUT(T):
    # look-up table
    prob=np.zeros(5)
    for i in range (2,5,2):
        prob[i]=np.exp(-2*i/T)
    return prob
```

(b) uno sweep Monte Carlo

Figura 1: Funzioni per la realizzazione in Python di uno sweep Montecarlo

Quindi ad ogni step Monte Carlo:

1. si sceglie lo spin i di cui voglio tentare il flip
2. si calcola l'energia di interazione con i primi vicini k tramite $h_{eff} = \sum_k \sigma_k$, dal quale si ricava la differenza di energia tra lo stato j e lo stato i come $\Delta E = 2h_{eff}$.
3. se $\Delta E \leq 0$ si accetta la nuova configurazione e si cambia $\sigma_i \rightarrow -\sigma_i$
4. senno si calcola il rapporto $\frac{p_j}{p_i}$, si genera un numero casuale $z \in [0, 1]$, e si inverte lo spin solo se $z < \exp(-\frac{\Delta E}{T})$

Per la scelta dello spin al passo 1 ci sono due alternative: o scegliere casualmente ad ogni ciclo il sito i , in modo equiprobabile tra tutti gli spin, o

scorrere sequenzialmente su tutta la griglia. Questi due metodi sono equivalenti computazionalmente, e rispettano entrambe la condizione $u_{ij} = u_{ji}$ per il bilancio dettagliato. Il secondo ha però il vantaggio di essere più efficiente, perché non si deve generare un numero casuale per la scelta dello spin a ogni step. Si definisce quindi un *Monte Carlo sweep*, che abbrevieremo come MCS, tentare di girare sequenzialmente tutti gli spin tramite la procedura sopra scritta. Nel caso di un reticolo quadrato, partiamo dallo spin in alto a sinistra, e proponiamo il flip di ogni spin in ordine, riga per riga. In Fig.1a possiamo vedere la realizzazione in Python. Usiamo il decoratore `@jit` della libreria *numba* per ottimizzare la funzione. La funzione prende come argomenti la configurazione da modificare e la temperatura. Scorriamo lungo tutti gli spin, tramite le coordinate (i, j) , e calcoliamo il campo efficace, sommando il valore dei 4 spin primi vicini con condizioni periodiche. Calcoliamo la quantità *sum* che può assumere i valori $[-4, -2, 0, 2, 4]$. Per i valori negativi accetto sicuramente la nuova configurazione con spin girato. Per $[2, 4]$, definisco un vettore *prob* di 5 elementi con solo indice 2 e 4 (partendo da indice 0) non nulli, pari alla probabilità p_j/p_i corrispondente al valore di *sum*. Genero un numero casuale $z \in [0, 1]$ tramite la funzione *random.random* della libreria *numpy*, e accetto il flip dello spin solo se è minore della probabilità sopra calcolata.

3.2 Parametri temporali simulazione

Per mostrare le transizioni di fase è necessario simulare il sistema a varie temperature, e fare la misura delle due quantità principali, la magnetizzazione m e l'energia e . Per una corretta misura di queste quantità, data una simulazione di un sistema di spin, è importante chiedersi:

1. dall'inizio della simulazione, dopo quanti Monte Carlo sweep è possibile fare la misura di m ed e .
2. quanto le misure effettuate siano correlate.

Quindi studiamo i tempi di termalizzazione t_{eq} e quelli di correlazione t_{mis} .

3.2.1 Tempi di termalizzazione

Per la misura durante una simulazione dobbiamo essere sicuri che il sistema sia all'equilibrio, sennò non è possibile misurare le quantità tramite la distribuzione canonica. Per il sistema in considerazione l'equilibrio dipende solo dalla temperatura T . Per trovare il tempo di termalizzazione basta verificare dopo quanti Monte Carlo sweep le quantità da misurare raggiungono

un valore costante.

Per fare questa verifica sono state definite tre configurazioni iniziali del sistema: inizialmente con tutti gli spin positivi (+), con tutti gli spin negativi (−) e tutti random (0).

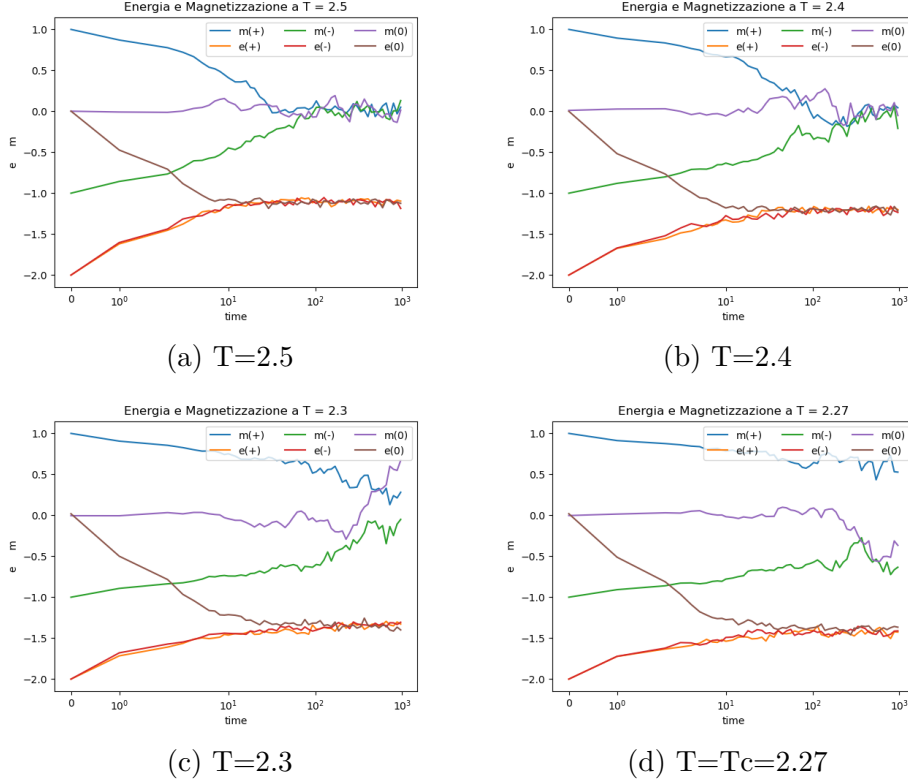


Figura 2: Termalizzazione energia e magnetizzazione per tre sistemi diversi a varie temperature, con $L = 100$.

Nella Fig.2 si osservano i tempi di termalizzazione di m ed e partendo da queste tre configurazioni a temperatura fissata $T > T_c$. In particolare dalle Fig. 2a e 2b, possiamo affermare che per $T > 2.3$ è sufficiente considerare un $t_{term} \simeq 10^3$, perché a questi tempi le tre configurazioni raggiungono lo stesso valore di m ed e . Alla temperatura critica, Fig. 2d, osserviamo che per le configurazioni (+), (−) rimangono ben separate, e (0) tende verso una di queste. Osserviamo quindi la rottura di simmetria del sistema, che caratterizza questa transizione di fase, e rottura di ergodicità, perché le due configurazioni con m positivo e negativo sono separate da una barriera di energia. Questo è visibile anche in Fig. 3a, infatti si vede che i tempi di termalizzazione dipendono dalla temperatura, e in particolare t_{term} aumenta

all'avvicinarsi della temperatura critica, rendendo più difficile una misura corretta.

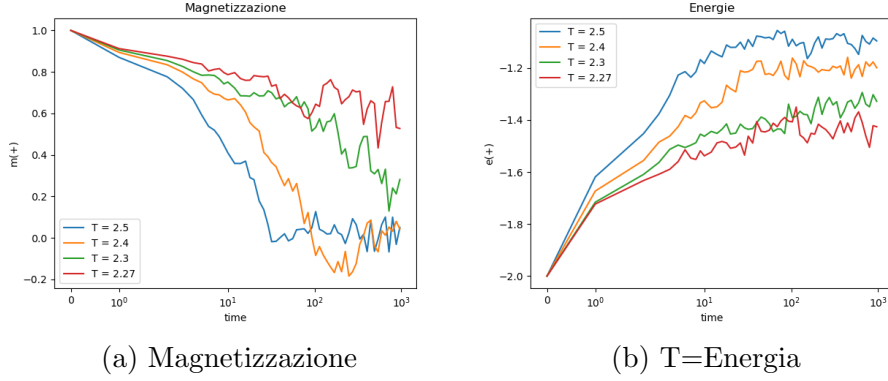


Figura 3: Termalizzazione del sistema (+) dell'energia e magnetizzazione al variare della temperatura, per $L = 100$.

E' interessante notare che i tempi t_{term} per l'energia non dipendono dalla temperatura, e i tre sistemi riescono a tendere allo stesso valore anche a temperature prossime alla temperatura critica. Questo si può intuire dal fatto che l'energia non è un parametro d'ordine della transizione di fase, e rimane simmetrica per lo scambio $\sigma_i \rightarrow -\sigma_i$ a tutte le temperature. Infatti in Fig2 l'energia dei sistemi (+) e (-) rimane circa uguale durante tutta la simulazione. Mostriamo in Fig 4 che le affermazioni fatte valgono anche per temperature $T < T_c$.

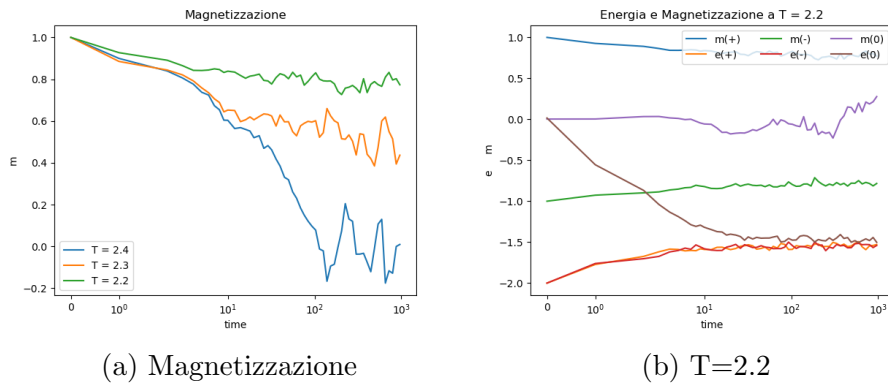


Figura 4: Risultati per temperature inferiori a quella critica, per $L = 100$.

3.2.2 Tempi di correlazione

Nella sezione precedente abbiamo studiato e stimato il tempo di termalizzazione; nel seguito ci concentriamo sulla simulazione all'equilibrio. Per stimare il valore di una generica osservabile è sufficiente usare la (19); si deve notare che nella nostra simulazione abbiamo misurato il tempo in Monte Carlo Sweep, quindi t_{mis} è il numero totale di MCS all'equilibrio. In una MCS l'algoritmo tenta di aggiornare tutte le variabili del sistema; nonostante ciò due configurazioni successive possono essere molto correlate, specialmente vicino al punto critico; questo fatto è rilevante per una corretta analisi statistica degli errori nella simulazione. Per valutare la rilevanza delle correlazioni temporali introduciamo, per una generica osservabile A , la relativa funzione di autocorrelazione connessa definita come:

$$\phi_A(\Delta t) \equiv \frac{\sum_{t=1}^{t_{mis}-\Delta t} A_t A_{t+\Delta t}}{t_{mis} - \Delta t} - \left(\frac{\sum_{t=1}^{t_{mis}-\Delta t} A_t}{t_{mis} - \Delta t} \right) \left(\frac{\sum_{t=1}^{t_{mis}-\Delta t} A_{t+\Delta t}}{t_{mis} - \Delta t} \right) \quad (25)$$

Nella simulazione risulta più conveniente usare la funzione:

$$C_A(\Delta t) \equiv \frac{\phi_A(\Delta t)}{\phi_A(0)} \quad (26)$$

Infatti, dalla definizione, $C_A(0) = 1$ e tende a 0 per $\Delta t \rightarrow +\infty$ in quanto, per grandi tempi, A_t e $A_{t+\Delta t}$ sono scorrelati e i due termini in (25) si elidono. Nel seguito, quando parleremo della funzione di autocorrelazione di un'osservabile, intenderemo sempre la (26).

Abbiamo simulato il modello a tre temperature diverse, incrementando progressivamente la taglia del sistema. Per ogni simulazione abbiamo impostato $t_{eq} = 1000$ e $t_{mis} = 10000$ e calcolato la funzione di autocorrelazione dell'energia usando la funzione `acf` della libreria `statsmodels.tsa.stattools` di python; i risultati sono riportati in Fig. 5.

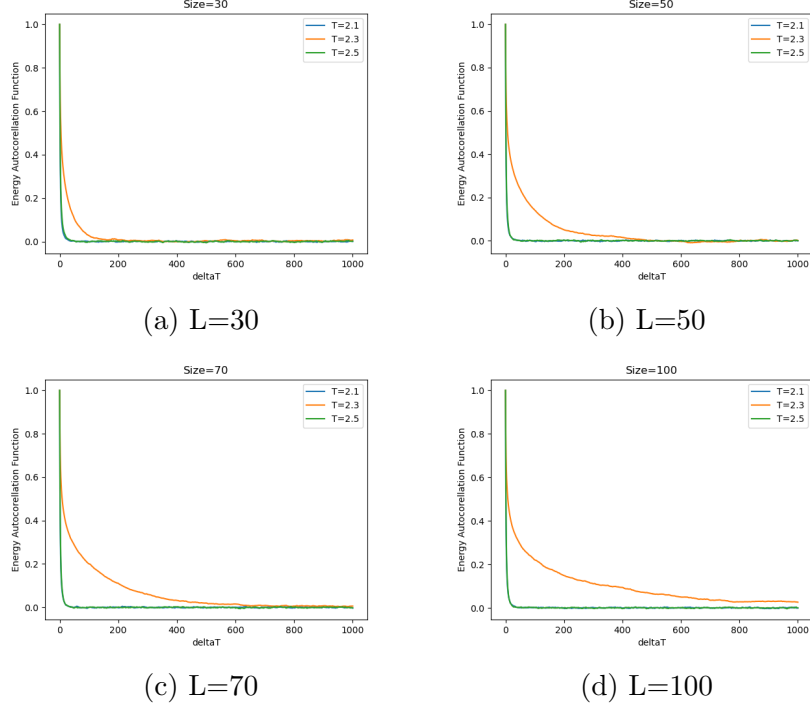


Figura 5: Funzioni di autocorrelazione dell'energia

Osserviamo dalla Fig. 5 che indipendentemente dalla taglia, la funzione di correlazione che decade a 0 più lentamente è quella corrispondente a $T = 2.3$, cioè quella più vicina alla temperatura critica (14); inoltre all'aumentare di L tale decadimento sembra diventare sempre più lento. A partire dalla funzione di autocorrelazione (26) possiamo definire il tempo di correlazione τ_A dell'osservabile A come:

$$C_A(\Delta t) \simeq \exp\left(-\frac{\Delta t}{\tau_A}\right) \quad (27)$$

Il tempo di correlazione è una misura del tempo (in MCS) che dobbiamo effettuare affinché due configurazioni generate siano indipendenti l'una dall'altra. Per verificare che il tempo di correlazione τ_E aumenti con la taglia del sistema simulato abbiamo eseguito un fit della funzione di autocorrelazione a $T = 2.3$ per le diverse taglie L usando la funzione *curvefit* della libreria *scipy.optimize*; la funzione di fit usata è:

$$y = a \exp\left(-\frac{t}{\tau}\right) + c \quad (28)$$

dove a , τ e c sono i tre parametri del fit. I risultati sono riportati in Tab.1.

L	$\tau_E(MCS)$
30	27.3 ± 0.3
50	69 ± 1
70	110 ± 2
100	143 ± 3

Tabella 1: Tempi di correlazione per l'energia a $T = 2.3$

Osserviamo che intorno a T_c il tempo di correlazione τ_E cresce con la taglia del sistema; quindi possiamo concludere che esso ha un massimo in vicinanza della temperatura critica e diverge nel limite termodinamico $L \rightarrow \infty$. A causa di questa divergenza si dice che la dinamica di inversione con singolo spin su cui si basa l'algoritmo di Metropolis presenta un rallentamento critico a T_c .

3.3 Osservabili termodinamiche

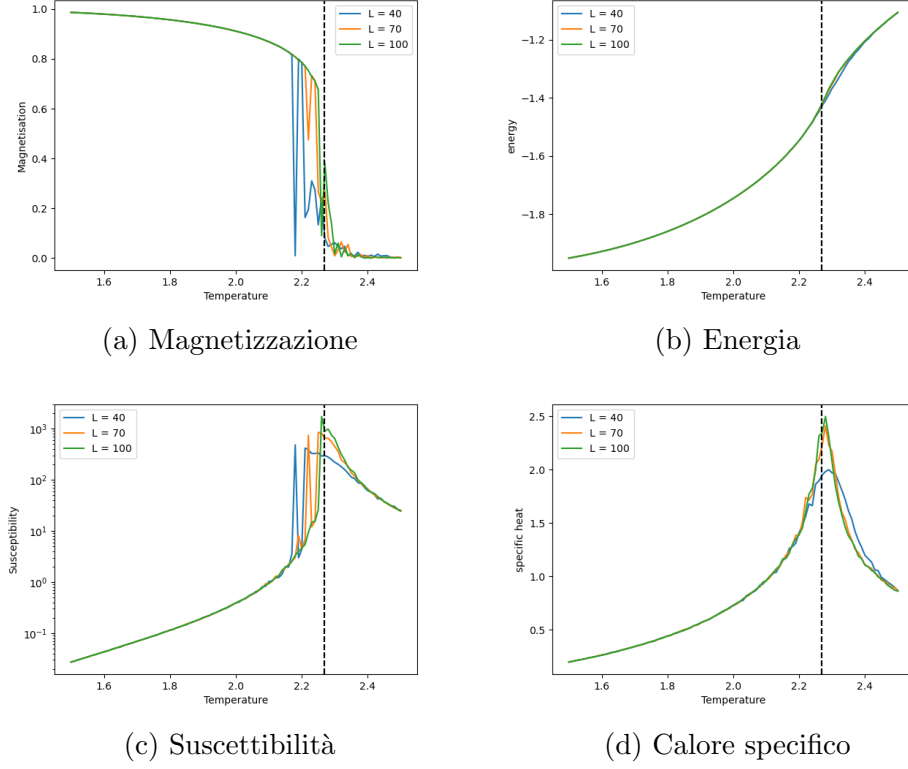


Figura 6: Risultati simulazione per diverse taglie.

Ora che abbiamo definito i parametri della simulazione possiamo verificare l'andamento di alcune quantità termodinamiche. Quelle che consideriamo sono magnetizzazione, energia suscettibilità magnetica e calore specifico, come definite in §1. Ricordiamo che misuriamo i tempi in Monte Carlo Sweep. Per ottenere i risultati in Fig. 6 sono stati usati i seguenti parametri:

- tempo di termalizzazione $t_{term} = 10^4$
- tempo di misura $t_{mis} = 10^5$. La misura viene effettuata facendo la media delle m ed e ottenute in questo intervallo.
- la temperatura viene diminuita ad ogni ciclo di $\Delta T = 0.01$ partendo da $T_{max} = 2.5$ fino a $T_{min} = 1.5$
- la simulazione viene fatta a tre taglie diverse $L = [40, 70, 100]$


```

@jit
def energy(conf): # Energia di una configurazione
    # Funzione che ha in input la configurazione iniziale
    # (una matrice) e calcola la corrispondente energia
    N = conf.shape[0] # la funzione shape restituisce le le dimensioni dell'oggetto;
                       # per una matrice restituisce numero di righe e colonne.
                       # Per specificare che vogliamo le righe si usa shape([0])
    E=0
    for i in range (N):
        for j in range (N):
            heff=conf[(i-1)%N,j]+conf[(i+1)%N,j]+conf[i,(j+1)%N]+conf[i,(j-1)%N] # PBC
            E=E-heff*conf[i,j]
    return(E/2)

```

Figura 7: Implementazione in Python della funzione *energy*

L'energia è calcolata ad ogni step tramite la funzione *energy* in Fig. 7. Le quantità sono poi calcolate facendo t_{mis} misure successive e facendo la media (per e ed m) o la varianza (per suscettibilità e calore specifico) dei valori ottenuti. Osserviamo che per taglie $L = 40$ e $L = 70$, la magnetizzazione oscilla intorno alla temperatura critica, mentre è stabile per $L = 100$. Le stesse oscillazioni si osservano per la suscettibilità, dipendendo dalla magnetizzazione. Quindi per ridurre gli effetti di taglia finita abbiamo osservato che per i parametri scelti è sufficiente una taglia $L = 100$. Si può osservare in Fig. 6c e in Fig. 6d che la suscettibilità e il calore specifico all'aumentare della taglia tendono a sviluppare una discontinuità, come ci aspetteremmo nel limite termodinamico.

3.4 Temperatura critica e parametro di Binder

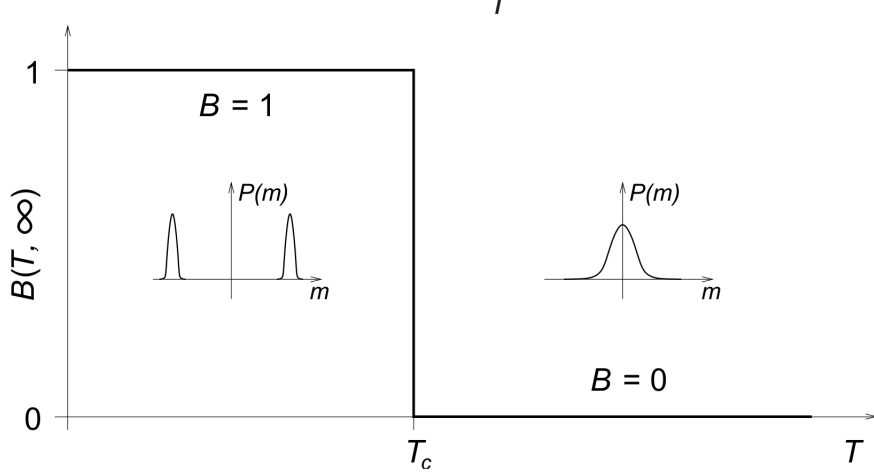


Figura 8: Comportamento parametro di Binder nel limite di tagli infinita

È stato osservato l'andamento delle grandezze termodinamiche al variare della temperatura, da cui è possibile visualizzare la presenza di una transizione di fase. Dalla teoria del modello di Ising 2D è noto che il valore critico della temperatura a cui avviene la transizione è $T_c \simeq 2.269$ in unità $J = k_b = 1$. Nel limite termodinamico dovremmo osservare delle divergenze per la suscettibilità e il calore specifico che nelle simulazioni sono ridotti a dei massimi per la taglia finita del sistema. Inoltre la magnetizzazione m assume valori molto piccoli intorno al valore T_c e le oscillazioni dovute alla taglia finita ne rendono difficile una misura precisa.

Per poter ricavare la temperatura critica è possibile definire una nuova funzione, il parametro di Binder:

$$B(T, L) = \frac{1}{2} \left(3 - \frac{\langle m \rangle^4}{\langle m^2 \rangle^2} \right). \quad (29)$$

Questa definizione con i parametri $1/2$ e 3 è necessaria per avere un andamento nel limite di $L \rightarrow +\infty$ come mostrato in Figura 8, ben distinto dal valore della temperatura critica. Infatti per $T > T_c$ la magnetizzazione è distribuita come una Gaussiana centrata in zero, e sapendo la relazione $\langle m^4 \rangle = 3\langle m^2 \rangle^2$ dalla statistica, si ottiene $B(T > T_c, L \ll 1) = 0$.

Per $T < T_c$ invece la magnetizzazione può assumere i valore $\pm m_0$, perciò in questo caso $\langle m^2 \rangle = m_0^2$ e $\langle m^4 \rangle = m_0^4$, quindi $B(T < T_c, L \ll 1) = 1$. Ciò che facciamo operativamente è di simulare tre sistemi di taglia diversa, $L = [10, 20, 40]$, per un range di temperature $2 < T < 3$, dove si osserva

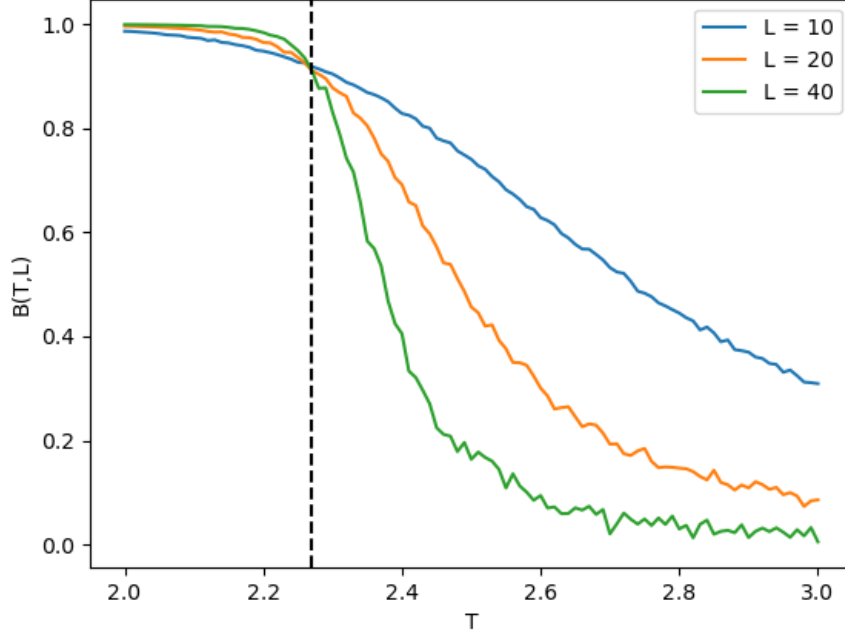


Figura 9: Misura del parametro di Binder per tre sistemi di taglia diversa, in tratteggiato il valore della temperatura critica osservata.

anche dalle immagini della sezione precedente la presenza del comportamento critico, con risoluzione $\Delta T = 0.01$. I risultati in Figura 9, mostrano che le tre curve assumono lo stesso valore in corrispondenza del valore critico. Minimizzando la differenza tra queste curve si trova infatti il valore $T_c = 2.27 \pm 0.01$, quindi compatibile con il valore teorico della temperatura critica. Come ci aspettavamo, si può osservare che l'andamento di B all'aumentare della taglia del sistema tende proprio al limite descritto sopra.

3.5 Esponente critico dinamico

In §3.2 abbiamo visto che in vicinanza di T_c il tempo di correlazione τ_E diverge nel limite termodinamico. Per caratterizzare la divergenza di tale quantità possiamo definire un ulteriore esponente critico, detto dinamico:

$$\tau \sim |t|^{-z\nu} \quad (30)$$

dove ν è l'esponente critico associato alla lunghezza di correlazione definita in (10) e t è la temperatura ridotta $t = \frac{T-T_c}{T_c}$.

L'esponente critico z fornisce un modo per quantificare il rallentamento critico dell'algoritmo usato, nel nostro caso quello di Metropolis; combinando la (10) e la (30) si trova:

$$\tau \sim |\xi|^z \quad (31)$$

Ricordiamo che una transizione di fase può avvenire solo nel limite termodinamico; dunque in un sistema di taglia finita come quelli simulati la lunghezza di correlazione ξ non potrà veramente divergere (nel senso matematico). Fisicamente la lunghezza di correlazione è la distanza tra due spin entro la quale questi sono correlati, nel senso che il valore assunto da uno dipende dall'altro. Avvicinandosi alla temperatura critica gli spin tendono a correlarsi e orientarsi nella stessa direzione in modo tale che il sistema abbia una magnetizzazione finita; in questa maniera si formano dei domini magnetici la cui estensione è descritta da ξ . Evidentemente per un sistema finito di taglia L in dimensione d la massima dimensione di un cluster è L^d e quindi si ha un "cut-off" della lunghezza di correlazione che può essere al massimo L . Allora possiamo riscrivere (31) come:

$$\tau \sim L^z \quad (32)$$

La (32) consente di misurare z se la temperatura critica è nota come nel modello di Ising. Abbiamo quindi eseguito una serie di simulazioni alla temperatura critica per modelli con varie taglie $L = 10, 20, 30, 40, 50, 60, 70, 80, 90, 100$; per ognuna abbiamo valutato il tempo di correlazione τ_E come nella sezione 3.2. In ogni simulazione abbiamo preso $t_{eq} = 1000$ e $t_{mis} = 100000$.

In Tab.2 abbiamo riportato i risultati delle varie simulazioni. In Fig.(10) abbiamo riportando su scala logaritmica il tempo di correlazione τ_E misurato in funzione della taglia L .

L'equazione (32) prevede che i punti ottenuti dalle varie simulazioni si dispongano su una retta il cui coefficiente dovrebbe essere proprio z . Di conseguenza abbiamo effettuato un fit con la la funzione:

$$y = mx + q \quad (33)$$

con m e q parametri del fit.

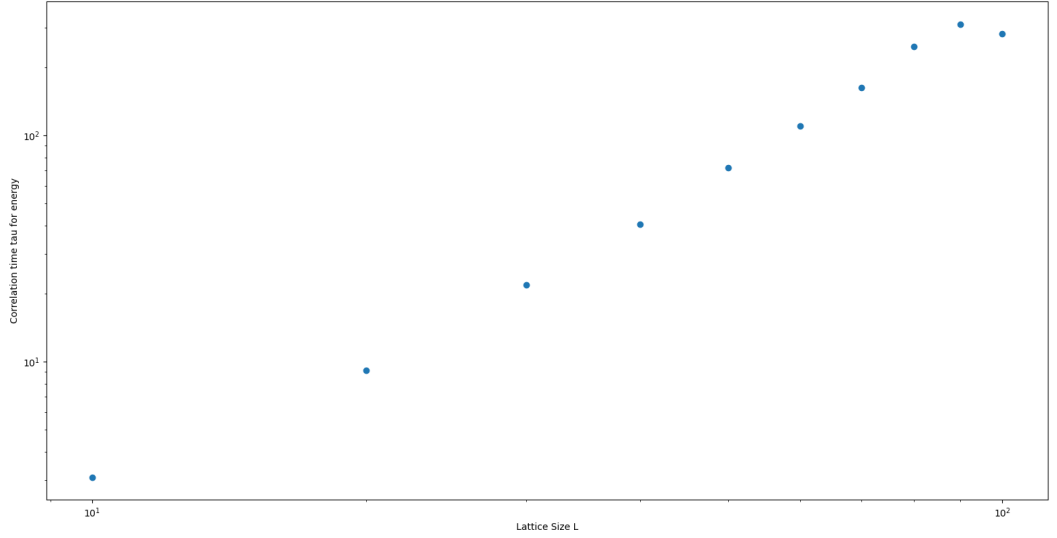


Figura 10: Tempo di correlazione dell'energia in funzione della taglia del sistema

L	$\tau_E(MCS)$
10	3.08714 ± 0.008
20	9.16 ± 0.03
30	21.89 ± 0.07
40	40.57 ± 0.13
50	71.9 ± 0.2
60	110.3 ± 0.4
70	162.0 ± 0.5
80	247.4 ± 0.9
90	310 ± 1
100	280.7 ± 0.9

Tabella 2: Tempi di correlazione per l'energia a $T = 2.3$

Il valore di z così ottenuto, che è la nostra misura dell'esponente critico dinamico per l'algoritmo di Metropolis, è:

$$z_{Metropolis} = 2.12 \pm 0.07 \quad (34)$$

3.6 Problemi dell'algoritmo di Metropolis

Nella sezione §3.2.2 abbiamo osservato come l'algoritmo di Metropolis presenti a T_c un rallentamento critico ("critical slowing down"); questo fenomeno è quello che determina maggiormente la durata di una simulazione. Abbiamo quantificato (vedere §3.5) tale rallentamento misurando l'esponente critico dinamico dell'algoritmo $z_{Metropolis} = 2.12 \pm 0.07$; tale valore è abbastanza alto e ci indica che l'algoritmo di Metropolis, basato su una dinamica di inversione di singolo spin, non è adatto a studiare il comportamento del modello vicino alla transizione di fase. E' importante notare che il fenomeno del rallentamento critico è una proprietà dell'algoritmo usato e nel seguito illustreremo un altro algoritmo che non soffre di tale fenomeno.

4 Algoritmo di Wolff

L'idea alla base dell'algoritmo è quella di individuare "cluster" di spin con la stessa orientazione e invertirli tutti in un unico passo invece di cercare di invertirli uno alla volta come nell'algoritmo di Metropolis. Algoritmi di questo tipo sono detti "cluster-flipping algorithms".

E' necessario dunque individuare i "cluster" di spin che vogliamo flippare; per il modello di Ising sappiamo che ad alte temperature gli spin sono scorrelati e quindi la taglia dei cluster risulta piccola. Al contrario, vicino alla transizione di fase gli spin iniziano a correlarsi su lunghe distanze e quindi la taglia dei "cluster" aumenta. Segue che la dimensione dei "cluster" da flippare debba aumentare al diminuire della temperatura. E' possibile quindi procedere nel modo seguente: iniziamo selezionando casualmente uno spin del reticolo, guardiamo i suoi primi vicini e aggiungiamo al "cluster" quelli con la stessa orientazione di quella di partenza con una probabilità p_{add} che cresce al diminuire della temperatura. Poi guardiamo i primi vicini di quelli aggiunti, li aggiungiamo con probabilità p_{add} e così via fino a quando terminiamo gli spin orientati come quello di partenza che possiamo aggiungere al cluster. Infine flippiamo tutto il cluster costruito.

4.1 Implementazione

Nell'algoritmo di Wolff la probabilità di aggiungere uno spin orientato come quello di partenza è:

$$P_{add} = 1 - \exp\left(-\frac{2}{T}\right) \quad (35)$$

Notiamo che P_{add} cresce diminuendo la temperatura.

Per il modello di Ising l'algoritmo è il seguente:

1. Selezioniamo casualmente uno spin del reticolo, detto seme.
2. Guardiamo i primi vicini del seme. Se sono orientati come lo spin di partenza li aggiungiamo al "cluster" con probabilità P_{add} definita come in (35) .
3. Per ogni spin che è stato aggiunto al passo precedente controlliamo gli spin primi vicini per trovare quelli orientati come il seme e li aggiungiamo con probabilità P_{add} . Ripetiamo questo passo fino a quando non ci sono più spin nel "cluster" i cui primi vicini sono stati considerati per l'inclusione almeno una volta.
4. Invertiamo tutto il cluster.

Abbiamo implementato questo algoritmo in Python all'interno della funzione *wolff* il cui listato riportiamo in appendice.

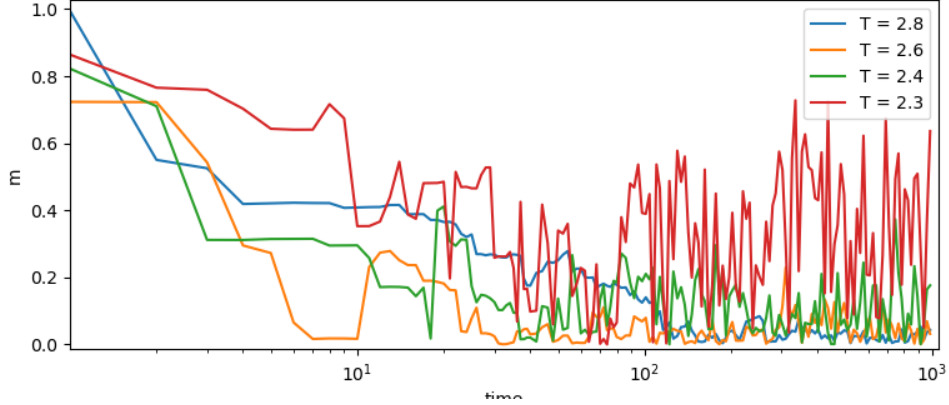


Figura 11: Termalizzazione della configurazione (+), per varie temperature $T \geq T_c$, per $L = 100$

4.2 Tempi di termalizzazione

Per l'algoritmo di Wolff si osservano comportamenti diversi per la termalizzazione rispetto all'algoritmo di Metropolis. In Fig.11 si osserva che il t_{term} è più simile tra le varie temperature, ma aumentano le oscillazioni avvicinandoci alla temperatura critica. Questo è dovuto anche alla natura diversa dell'algoritmo di Wolff, che ad ogni step gira un intero cluster la cui dimensione aumenta diminuendo la temperatura. Questo permette, infatti, di poter esplorare maggiormente lo spazio delle fasi anche sotto la temperatura critica ed essere quindi meno sensibile alla rottura di ergodicità. Riportiamo anche in Fig. 12 i tempi di termalizzazione per varie temperature partendo da configurazioni diverse come in §3.1.1.

Osserviamo che per tutti e tre i sistemi raggiungiamo l'equilibrio di m ed e entro 10^3 step di Wolff. Hanno infatti un andamento che dipende poco dalla temperatura. Ma, come osservato anche sopra, vediamo che al diminuire della temperatura le oscillazioni aumentano, in particolare a $T = T_c$, pur rimanendo intorno al valore corretto. Quindi per qualsiasi temperatura è corretto prendere un tempo $t_{term} \approx 10^3$ per l'algoritmo di Wolff.

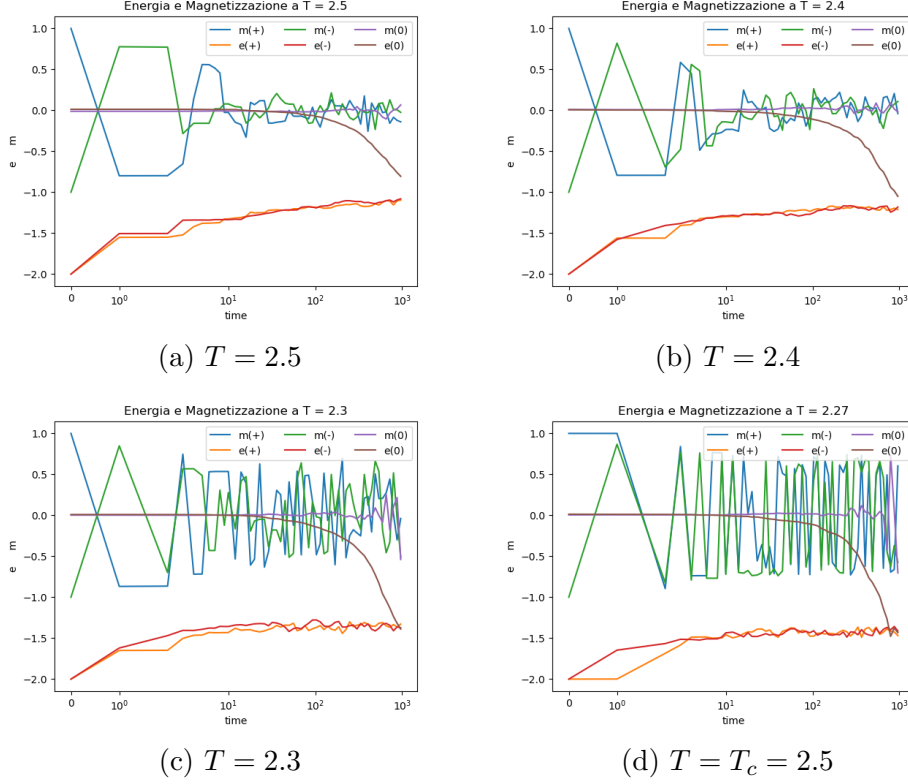


Figura 12: Risultati simulazione per diverse taglie.

4.3 Tempi di correlazione ed esponente critico dinamico

Osserviamo che nel limite di elevata temperatura (intorno a $T = 10$), la taglia dei cluster risulta difficilmente maggiore di uno in quanto la probabilità p_{add} di aggiungere uno spin tende a zero. Ad ogni step dell'algoritmo di Wolff solo il seme viene invertito con probabilità uno; questo è proprio quello che fa l'algoritmo di Metropolis nello stesso regime di temperatura. A basse temperature gli spin tendono ad allinearsi e a formare un dominio magnetico che ha l'estensione del reticolo in modo tale che il sistema abbia una magnetizzazione finita.

La probabilità che lo spin iniziale dell'algoritmo di Wolff appartenga a tale dominio è molto elevata e il cluster che si forma cresce fino ad occupare quasi completamente il dominio. Nel metodo di Wolff tutti gli spin di tale cluster vengono flippati in un'unica volta e quindi in un passo Monte Carlo viene generata una configurazione indipendente dalla precedente.

Nell'algoritmo di Metropolis a basse temperature si deve eseguire un passo

Monte Carlo per ogni sito se si vuole generare uno stato completamente nuovo; quindi per $T \ll T_c$ un passo con l'algoritmo di Wolff è pari a una MCS (Monte Carlo Sweep) con Metropolis. Se vogliamo comparare i tempi di correlazione nei due algoritmi nelle vicinanze della transizione di fase dobbiamo definire correttamente il tempo di correlazione per l'algoritmo di Wolff. Dalla precedente analisi dei due regimi opposti di alta e bassa temperatura, si può intuire che la corretta definizione di τ per l'algoritmo di Wolff é:

$$\tau = \tau_{steps} \frac{\langle n \rangle}{L^2} \quad (36)$$

dove $\langle n \rangle$ è la taglia media dei clusters alla temperatura di simulazione e τ_{steps} è il tempo di correlazione misurato in steps dell'algoritmo di Wolff. Possiamo studiare la dipendenza di $\langle n \rangle$ dalla temperatura e verificare le nostre precedenti intuizioni sul suo comportamento nei due regimi opposti di alte e basse temperature sfruttando la seguente relazione (vedere [3]):

$$\chi = \beta \langle n \rangle \quad (37)$$

con χ suscettibilità magnetica. Usando la (4) possiamo scrivere la (37) come:

$$\langle n \rangle = \frac{1}{N} (\langle M^2 \rangle - \langle M \rangle^2) = \frac{1}{L^2} (\langle M^2 \rangle - \langle M \rangle^2) \quad (38)$$

Abbiamo quindi simulato un sistema di taglia $L = 30$ con $t_{eq} = 1000$ e $t_{mis} = 100000$ variando la temperatura tra $T = 3.5$ e $T = 1$; in Fig.13 abbiamo riportato la taglia media dei "clusters" come una frazione della taglia del modello in funzione della temperatura.

La Fig.13 conferma che ad alte temperature raramente la taglia di un "cluster" è maggiore di uno, mentre molto sotto a T_c essa tende a diventare pari all'intero reticolo.

La definizione (36) permette di confrontare la performance dei due algoritmi vicino alla temperatura critica.

Se misuriamo il tempo in Monte Carlo steps (cioè in semplici inversioni di "cluster") è possibile definire un esponente z_{steps} :

$$\tau_{steps} \approx \xi^{z_{steps}} \quad (39)$$

Possiamo misurare z_{steps} ripetendo l'analisi svolta al §3.5

E' possibile dimostrare (vedere [3]) che z_{steps} è legato al vero esponente critico dinamico z tramite la relazione:

$$z = z_{steps} + \frac{\gamma}{\nu} - 2 \quad (40)$$

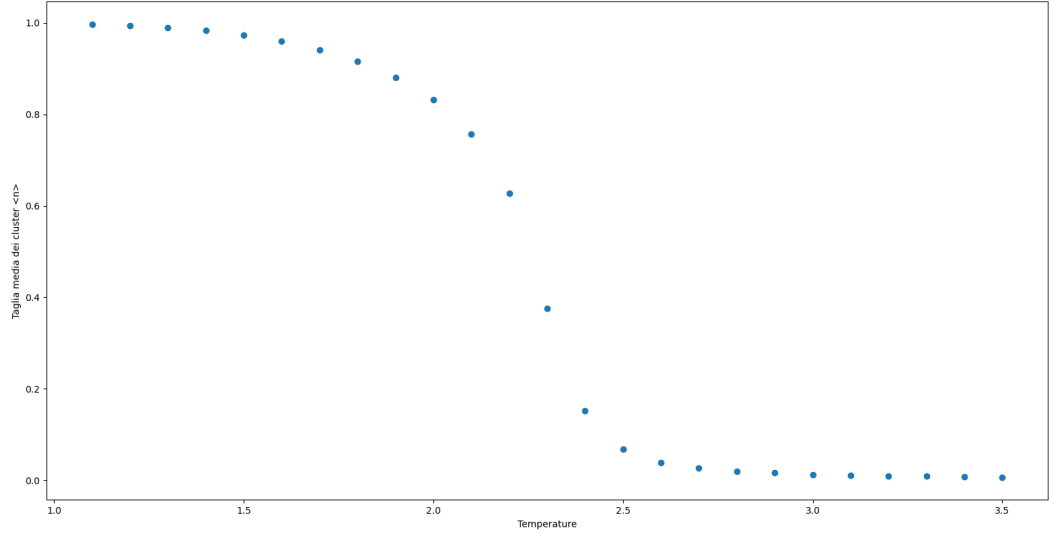


Figura 13: Taglia media dei "clusters" in funzione della temperatura

dove γ e ν sono gli esponenti critici associati alla suscettibilità magnetica e al calore specifico. Dato che nel modello di Ising γ e ν sono noti analiticamente, è possibile (tramite (40)) misurare z per l'algoritmo di Wolff. Abbiamo, come in §3.5, simulato il modello per diverse taglie alla temperatura critica con $t_{eq} = 10000$ e $t_{mis} = 100000$ e misurato per ciascuna il tempo di correlazione dell'energia. I risultati ottenuti sono riportati in Tab.3 e in Fig.14. Abbiamo quindi valutato tramite un fit lineare z_{steps} , che risulta:

$$z_{steps} = 0.460 \pm 0.007 \quad (41)$$

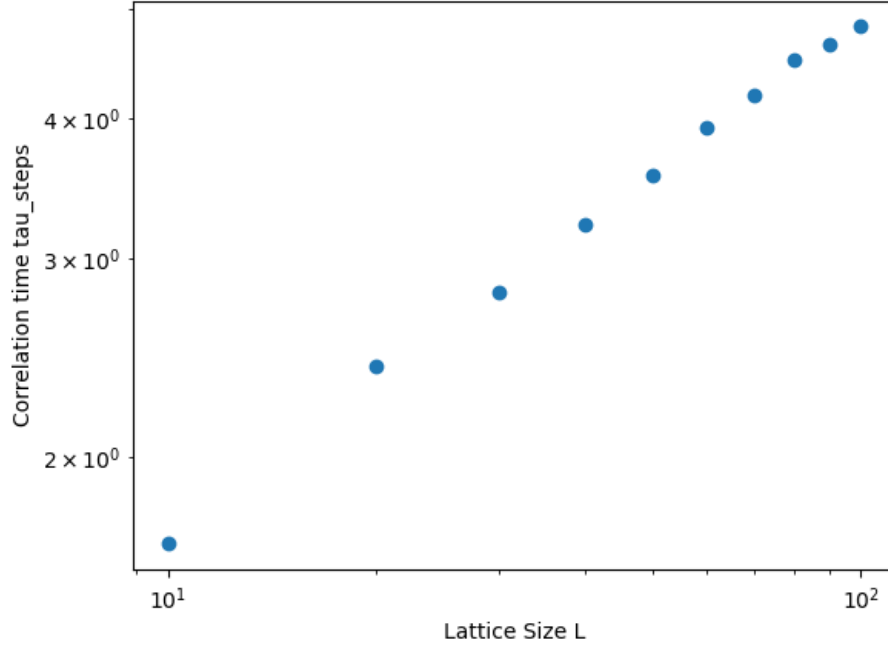


Figura 14: Tempo di correlazione dell'energia in funzione della taglia del sistema

L	$\tau_E(MCS)$
10	1.674 ± 0.013
20	2.41 ± 0.02
30	2.80 ± 0.02
40	3.21 ± 0.03
50	3.56 ± 0.03
60	3.93 ± 0.03
70	4.19 ± 0.04
80	4.51 ± 0.04
90	4.65 ± 0.04
100	4.82 ± 0.04

Tabella 3: Tempi di correlazione per l'energia a $T = 2.3$ con l'algoritmo di Wolff

Usando la (40) possiamo valutare il valore del vero esponente critico

dinamico che risulta:

$$z = 0.210 \pm 0.007 \quad (42)$$

4.4 Osservabili termodinamiche

Per ottenere i risultati in Fig.15 sono stati usati i seguenti parametri:

- tempo di termalizzazione $t_{term} = 10^3$
- tempo di misura $t_{mis} = 10^4$. La misura viene effettuata facendo la media delle m ed e ottenute in questo intervallo.
- la temperatura viene diminuita ad ogni ciclo di $\Delta T = 0.1$ partendo da $T_{max} = 2.5$ fino a $T_{min} = 1.5$
- la taglia del sistema simulato è $L = 50$

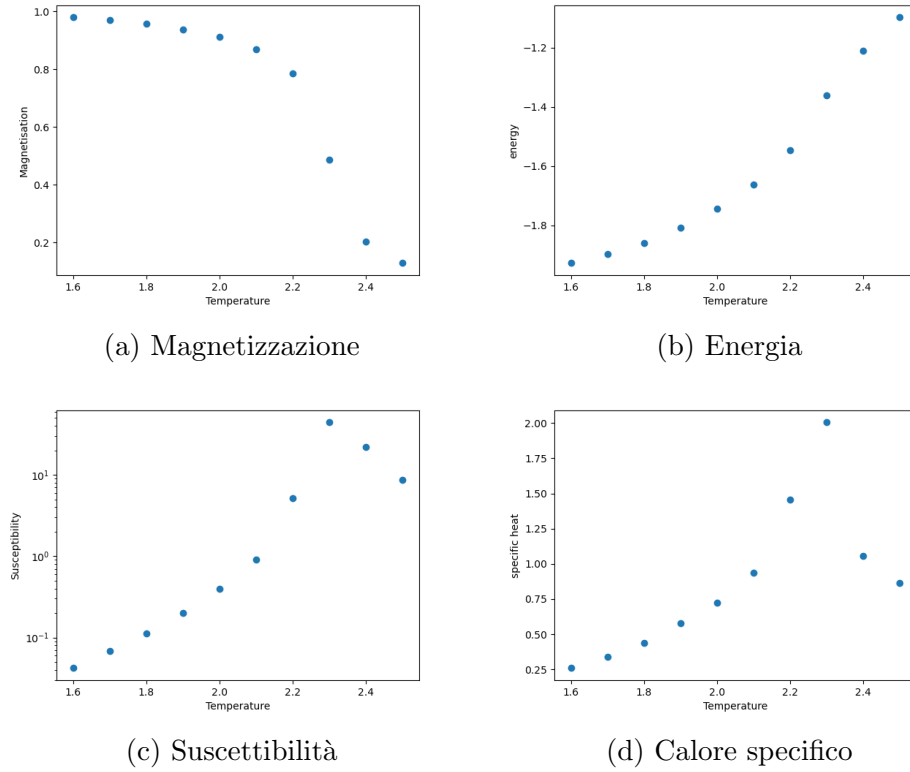


Figura 15: Risultati simulazione con taglia $L = 50$.

Si osserva che si può facilmente riconoscere la presenza di una transizione di fase.

5 Conclusioni

In questo lavoro abbiamo studiato tramite simulazione Monte Carlo il modello di Ising in dimensione $d = 2$; per fare ciò abbiamo usato due algoritmi: Metropolis e Wolff. Per entrambi abbiamo studiato i tempi di termalizzazione e di correlazione la cui conoscenza è necessaria per una corretta simulazione ed analisi dei dati. Per l'algoritmo di Metropolis abbiamo trovato che un tempo $t_{eq} = 1000 \text{ MCS}$ (Monte Carlo Sweep) è sufficiente per tutte le taglie del sistema simulate; anche per Wolff un tempo di termalizzazione pari a 1000 passi Monte Carlo è risultato più che sufficiente.

Dallo studio delle osservabili termodinamiche per entrambi gli algoritmi abbiamo osservato che suscettibilità magnetica e calore specifico presentano dei picchi ad una data temperatura, segnalando, come previsto, una transizione di fase. Per l'algoritmo di Metropolis abbiamo stimato la temperatura critica tramite la misura del parametro di Binder a diverse scale, ottenendo il valore:

$$T_c^{mis} = 2.27 \pm 0.01 \quad (43)$$

che è compatibile con il valore teorico (14).

Abbiamo studiato i tempi di correlazione dell'energia. Per Metropolis abbiamo osservato che essi tendono a divergere in vicinanza della transizione di fase e per quantificare tale comportamento ("critical slowing down") abbiamo misurato l'esponente critico dinamico z che è risultato pari a:

$$z_{Metropolis} = 2.12 \pm 0.07 \quad (44)$$

Per Wolff l'esponente critico dinamico è risultato:

$$z_{Wolff} = 0.210 \pm 0.007 \quad (45)$$

Tali risultati mostrano come l'algoritmo di Wolff non soffra del fenomeno del rallentamento critico e quindi sia preferibile per studiare il comportamento del sistema nella regione critica. Nonostante uno step con Wolff sia computazionalmente più oneroso di un passo con Metropolis, tale algoritmo è particolarmente adatto allo studio nella vicinanza della transizione di fase in quanto ha tempi di correlazione molto piccoli e quindi consente di fare più misure scorrelate in tale regione con meno step. Conviene quindi usare l'algoritmo di Metropolis solo nei limiti di alta e bassa temperatura, lontano dalla temperatura critica.

6 Appendice

```
def wolff2(conf,T):
    N=conf.shape[0]
    i, j = np.random.randint(0,N), np.random.randint(0,N)
    spin_seed= conf[i,j]
    cluster = [[i,j]]
    old_spin = [[i,j]]
    p_add = 1. - np.exp(-2/T)

    while (len(old_spin) != 0) :
        new_spin = []

        for i,j in old_spin:
            nn = [[(i+1)%N,j], [(i-1)%N,j], [i,(j+1)%N], [i,(j-1)%N]]

            for state in nn:
                if conf[state[0],state[1]] == spin_seed and state not in cluster:
                    if np.random.rand() < p_add:
                        new_spin.append(state)
                        cluster.append(state)

        old_spin= new_spin

    for i,j in cluster:
        conf[i,j] *= -1
    return conf
```

Riferimenti bibliografici

- [1] J. J. Binney, N. J. Dowrick, A. J. Fiscer, M. E. J. Newman *The Theory of Critical Phenomena: An Introduction to the Renormalization Group (1992)*, Clarendon Press Oxford
- [2] N. Goldenfeld *Lectures on Phase Transitions and the Renormalization Group (1992)*, Perseus Books
- [3] M. E. J. Newman, G. T. Barkema *Monte Carlo Methods in Statistical Physics (1998)*, Clarendon Press Oxford
- [4] L. M. Barone, E. Marinari, G. Organtini, F. R. Tersenghi *Programmazione Scientifica: Linguaggio C, algoritmi e modelli nella scienza (2018)*, Pearson