



UNIVERSITÀ DEGLI STUDI DI TORINO
DIPARTIMENTO DI FISICA

Tesi di Laurea

Il modello di Ising bidimensionale: soluzione esatta e simulazione Monte Carlo

RELATORE

Prof. Marco Billò

CANDIDATO

Alessandro Tofani

Anno Accademico 2018/2019

Sommario

Nella tesi si analizza il Modello di Ising, che attraverso l'interazione tra gli spin degli atomi disposti su un reticolo d -dimensionale, spiega il fenomeno del ferromagnetismo nei materiali. Si osserva una transizione di fase tra ferromagnete e paramagnete per $d > 1$, quando la temperatura del sistema supera la temperatura critica. Nella tesi si espone la risoluzione analitica del modello in dimensione 1 e 2, e si sviluppa un programma in C++, eseguito con ROOT, che implementa l'algoritmo Metropolis del Metodo Monte Carlo, per simulare il comportamento del sistema e ricavare le osservabili e gli esponenti critici più importanti.

Indice

1	Introduzione	1
2	Il modello di Ising	1
2.1	Transizioni di fase	2
2.2	Esponenti critici	3
3	Metodo Monte Carlo	4
3.1	Cenni storici	4
3.2	Il Metodo	4
3.3	Algoritmo Metropolis	6
3.4	Autocorrelazione	6
4	Il modello in una dimensione	8
4.1	Simulazione Monte Carlo del modello unidimensionale	9
5	Il modello in due dimensioni	10
5.1	Argomento di Peierls	10
6	Relazione di dualità su reticolo quadrato	11
6.1	Sviluppo in serie ad alte temperature	11
6.2	Sviluppo in serie di basse temperature	12
6.3	Autodualità	14
7	Soluzione combinatoria	16

8	Simulazione Monte Carlo del modello bidimensionale	21
8.1	Simulazione Monte Carlo del modello in tre dimensioni	26
1	Estratto del codice sorgente per la simulazione del modello bidimensionale	27

1 Introduzione

Il modello di Ising è un modello ampiamente studiato in meccanica statistica, ed è il sistema ad elementi interagenti più semplice che presenta una transizione di fase. Nella tesi viene presentato il caso unidimensionale, che non ha nessuna transizione di fase. Successivamente si analizza il modello bidimensionale, che presenta una transizione di fase. Con lo sviluppo per alte e basse temperature si trova una stima della temperatura critica, che viene poi determinata in modo esatto, insieme alle altre osservabili fisiche del sistema, con la soluzione combinatoria. Parallelamente alla descrizione teorica del modello, è stato realizzato un programma in C++, eseguito con ROOT, che implementa l'algoritmo Metropolis del metodo Monte Carlo, per misurare le più importanti osservabili del sistema. In conclusione si confrontano i risultati ottenuti dalla simulazione con i risultati esatti.

2 Il modello di Ising

La terza componente dello spin degli atomi è una variabile binaria e può assumere i valori $S_i = \pm 1$. L'Hamiltoniana del sistema è:

$$H = -J \sum_{\langle ij \rangle} S_i S_j - h \sum_i S_i$$

con $\langle ij \rangle$ che rappresenta l'interazione tra primi vicini, h campo magnetico esterno, J costante di accoppiamento tra gli spin. Si vede che per $J > 0$ e spin allineati, il materiale ha un comportamento ferromagnetico. L'energia del sistema diminuisce se gli spin sono allineati col campo esterno. Altresì di ha una diminuzione dell'energia anche se gli spin sono allineati tra di loro. Il numero totale di configurazioni del sistema è 2^N con N numero di atomi nel reticolo.

Usando la distribuzione di Boltzmann $u_i = \frac{e^{-\beta E_i}}{Z}$ con Z funzione di partizione del sistema, e $\beta = \frac{1}{k_B T}$ con k_B costante di Boltzmann, si ottiene:

$$u_i = \frac{1}{Z} e^{-\beta J \sum_{\langle ij \rangle} S_i S_j - \beta h \sum_i S_i}$$

che mostra la competizione tra le fluttuazioni termiche, che spingono il sistema a passare a stati ad entropia maggiore, e la minimizzazione dell'energia. Se il campo magnetico esterno è nullo, allora l'Hamiltoniana è simmetrica. Abbassando la temperatura del sistema si assiste alla rottura spontanea della simmetria, difatti il sistema deve scegliere se magnetizzarsi verso l'alto o verso il basso, rompendo quindi la simmetria iniziale.

2.1 Transizioni di fase

Il modello di Ising è il più semplice sistema a elementi interagenti, che presenta una transizione di fase, da uno stato ordinato, in cui i dipoli magnetici sono orientati nella stessa direzione, ad uno stato disordinato, in cui si formano domini magnetici tali per cui la magnetizzazione totale risulta nulla. La temperatura a cui avviene la transizione di fase è T_c temperatura critica del sistema. Alla temperatura critica si verifica la transizione di fase, cioè si ha un cambio delle proprietà di simmetria del sistema. Pertanto è utile definire il parametro d'ordine di un sistema, cioè la quantità la cui media termica è nulla in una fase del sistema, e diversa da zero nelle altre fasi. Nel modello di Ising un parametro d'ordine locale è il valore medio dello spin:

$$\langle \mathbf{S}_i \rangle = \begin{cases} 0, & \text{se } T > T_c; \\ M \neq 0, & \text{se } T < T_c \end{cases}.$$

Per $T < T_c$ la simmetria del sistema è ristretta alle rotazioni lungo l'asse individuato dal gruppo $O(2)$.

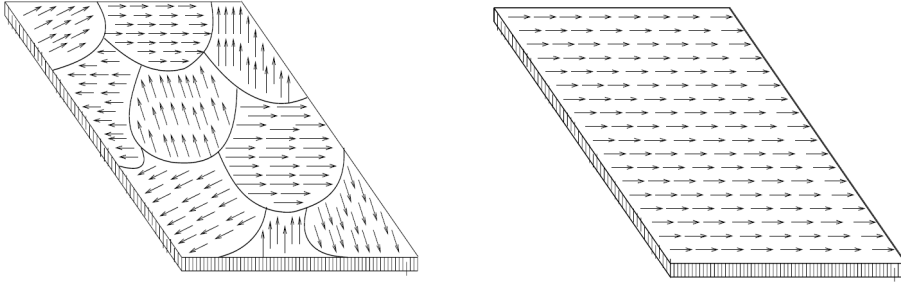


Figura 1: Immagine a sinistra: domini magnetici a temperature superiori alla temperatura critica. Immagine a destra: allineamento degli spin per temperature inferiori alla temperatura critica

La funzione di correlazione a 2 punti degli spin è $G^{(2)}(\mathbf{i}, \mathbf{j}) = \langle \mathbf{S}_i \mathbf{S}_j \rangle$. Se il sistema è isotropo, detta r distanza tra i due spin considerati, $G^{(2)}(r) = G^{(2)}(\mathbf{i}, \mathbf{j})$ che misura il grado di allineamento tra due spin distanti r . Poiché per temperature inferiori alla temperatura critica gli spin puntano prevalentemente nella stessa direzione, per studiare le fluttuazioni del sistema, conviene sottrarre il loro valor medio e introdurre la funzione di correlazione convessa:

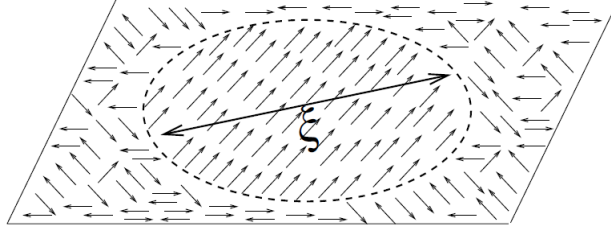


Figura 2: La dimensione dei domini magnetici è data dalla lunghezza di correlazione

$$G_c^{(2)}(r) = \langle (\mathbf{S}_i - \mathbf{M})(\mathbf{S}_j - \mathbf{M}) \rangle = \langle \mathbf{S}_i \mathbf{S}_j \rangle - |\mathbf{M}|^2 \quad .$$

Se la temperatura è superiore alla temperatura critica, allora la funzione di correlazione convessa coincide con la funzione di correlazione. Per temperature diverse dalla temperatura critica, la reciproca dipendenza degli spin vicini si estende fino ad una certa distanza ϵ detta lunghezza di correlazione, che è la dimensione tipica delle regioni in cui gli spin assumono lo stesso valore.

L'andamento asintotico del correlatore è $G_c^{(2)}(r) \simeq e^{-\frac{r}{\epsilon}}$ per temperature diverse dalla temperatura critica, mentre alla temperatura critica si ha un andamento a potenza:

$$G_c^{(2)}(r) \simeq \frac{1}{r^{d-2+\eta}}$$

con d dimensionalità del sistema e η dimensione anomala del parametro d'ordine. Alla temperatura critica l'andamento a potenza di $G_c^{(2)}(r)$ indica che le fluttuazioni del parametro d'ordine risultano significativamente correlate a tutte le scale di distanza. Vicino alla transizione di fase $\epsilon \rightarrow \infty$. Si definisce $t = \frac{T-T_c}{T_c}$ lo spostamento relativo della temperatura dalla temperatura critica, e si ha che

$$\xi(T) = \begin{cases} \xi_+ t^{-\nu}, & \text{se } T > T_c \\ \xi_- (-t)^{-\nu}, & \text{se } T < T_c \end{cases}$$

Si possono unificare gli andamenti di G :

$$G_c^{(2)}(r) = \frac{1}{r^{d-2+\eta}} e^{-\frac{r}{\xi}} \quad .$$

2.2 Esponenti critici

La magnetizzazione del sistema è $M(h, T) = \langle S_i^z \rangle = -\frac{\partial F}{\partial h}$ mentre la magnetizzazione spontanea è $M(T) = \lim_{h \rightarrow 0} M(h, T)$ che sarà:

$$M = M_0(-t)^\beta \text{ con } h \rightarrow 0, \text{ mentre } M(h, T_c) = M_0 h^{\frac{1}{\delta}}$$

con β e δ esponenti critici.

La suscettibilità magnetica, che misura la risposta del sistema a un campo magnetico esterno, è: $\chi(h, T) = \frac{\partial M(h, T)}{\partial h}$ e presenta una singolarità attorno alla temperatura critica, per cui:

$$\chi(0, T) = \begin{cases} \chi_+ t^{-\gamma}, & se T > T_c; \\ \chi_- (-t)^{-\gamma}, & se T < T_c \end{cases} .$$

Allo stesso modo il calore specifico del sistema è:

$$C(T) = \frac{\partial U}{\partial T} = \begin{cases} C_+ t^{-\alpha}, & se T > T_c; \\ C_- (-t)^{-\alpha}, & se T < T_c \end{cases} .$$

Gli esponenti critici non sono quantità indipendenti, ma tra di loro sussistono le seguenti relazioni algebriche:

$$\begin{cases} \alpha + 2\beta + \gamma = 2 \\ \alpha + \beta\delta + \beta = 2 \\ \nu(2 - \eta) = \gamma \\ \alpha + \nu d = 2 \end{cases} .$$

3 Metodo Monte Carlo

3.1 Cenni storici

Il metodo Monte Carlo nasce durante la seconda guerra mondiale, nell'ambito del progetto Manhattan. Una sezione di tale progetto, adibita a calcoli e risoluzione di integrali, immagina l'esistenza di una macchina capace di compiere calcoli elementari e che possa essere programmata per eseguire calcoli più complicati. Viene sviluppato il primo algoritmo, detto Metropolis. Le idee sviluppate da tali scienziati vengono segretate sotto il nome "Monte Carlo". Nel 1952 viene violato il patto di segretezza su tale metodo, ed è pubblicato il primo codice Monte Carlo: Metropolis.

3.2 Il Metodo

Nell'ensemble canonico, il valor medio dell'energia è:

$$\langle E \rangle = \sum_i E_i \frac{e^{-\beta E_i}}{Z}$$

con $Z = \sum_i e^{-\beta E_i}$ funzione di partizione, e l'indice i che etichetta i microstati del sistema. E' possibile mettere assieme gli stati con la medesima energia, che quindi hanno la stessa probabilità. Allora si somma sui livelli energetici, tenendo conto della degenerazione di ogni livello energetico $\Omega(E)$, e si ottiene che il valor medio dell'energia è:

$$\langle E \rangle = \sum_E E \Omega(E) \frac{e^{-\beta E}}{Z} = \sum_E E \frac{e^{-\beta E + \log \Omega(E)}}{Z} \quad .$$

In tal modo, si osserva che il sistema subisce due spinte:

1. Il fattore $-\beta E$ tende a far andare il sistema verso stati ad energia minima;
2. Il fattore $\log \Omega(E)$, che corrisponde all'entropia, a meno di costanti, tende a far andare il sistema verso stati più popolati, ovvero la cui degenerazione è più alta.

Studiando la probability density function, (PDF), si nota che è una delta con larghezza proporzionale alla radice quadrata del numero di gradi di libertà del sistema considerato. Allora il metodo Monte Carlo prevede di costruire un processo stocastico che si muova nello spazio delle distribuzioni, tale per cui qualunque sia il punto di partenza, la distribuzione cercata converga sempre in $\langle E \rangle$. Nel caso del modello di Ising, il Monte Carlo guida per trovare le combinazioni di spin corrette. Ciò è detto "importance sampling", cioè si considerano le regioni dello spazio delle fasi più importanti. Ciò è possibile attraverso le Catene di Markov.

Si considera la matrice $P = (p_{ij})$ con p_{ij} probabilità di transizione dallo stato i allo stato j . La condizione di normalizzazione è $\sum_j p_{ij} = 1$. La matrice P agisce su $u_i(t)$ distribuzione di probabilità al tempo t , con $i = 1, \dots, N$. All'istante $t + 1$ si avrà:

$$u_j(t + 1) = \sum_i u_i p_{ij}$$

che descrive l'evoluzione temporale della distribuzione di probabilità. Se si considera un punto fisso u_i , allora:

$$\sum_j u_j p_{ij} = u_i$$

cioè una volta che il sistema giunge al punto fisso vi permane nel tempo. Praticamente, la condizione del punto fisso risulta essere troppo generale, e quindi si usa una condizione più restrittiva, detta "Detailed balance":

$$u_j p_{ji} = u_i p_{ij}$$

che implica la condizione di punto fisso.

3.3 Algoritmo Metropolis

L'obiettivo è quello di trovare gli elementi p_{ij} . Si definisce la matrice $A = (a_{ij})$, i cui elementi sono le accettanze, ovvero le probabilità che si accetti la transizione da i a j . Sia $P^0 = (p_{ij}^0)$ una matrice contenente probabilità arbitrarie. Allora:

$$p_{ij} = p_{ij}^0 a_{ij} \text{ per } i \neq j \text{ e } p_{ii} = p_{ii}^0 + \sum_{i \neq j} p_{ij}^0 (1 - a_{ij})$$

con $1 - a_{ij}$ probabilità di non accettare la transizione da i a j . Implementando la condizione di bilancio dettagliato e considerando la matrice P^0 simmetrica, si ottiene che deve valere:

$$\frac{a_{ij}}{a_{ji}} = \frac{u_j}{u_i} \implies a_{ij} = F\left(\frac{u_j}{u_i}\right) \quad .$$

Si è perciò trovata la relazione funzionale $z = \frac{F(z)}{F(\frac{1}{z})}$, con $z = a_{ij}$. Allora si ricava che $F(z) = \min(z, 1)$. Considerando come distribuzione di probabilità la distribuzione di Boltzmann, si ottiene:

$$a_{ij} = F(e^{-\beta(E_j - E_i)}) \quad .$$

Si ha tutto per poter descrivere l'algoritmo Metropolis:

1. Si sceglie la configurazione iniziale i e si passa alla configurazione j con probabilità p_{ij}^0 ;
2. Si calcolano le energie E_i ed E_j ;
3. Se $E_i \geq E_j$ cioè l'energia del sistema diminuisce, allora $e^{-\beta(E_j - E_i)} > 1$ e $a_{ij} = 1$, cioè la transizione viene sempre accettata;
4. Se $E_i < E_j$ cioè l'energia del sistema aumenta, la transizione non viene scartata, ma viene accettata con probabilità $e^{-\beta(E_j - E_i)} = a_{ij}$.
5. Si ripete dallo step 2;

La probabilità di accettare la transizione, nello step 4, è confrontata con una probabilità fornita da un generatore di numeri casuali, e se $a_{ij} > \text{random}$ si accetta la transizione.

3.4 Autocorrelazione

Le configurazioni fornite dal metodo Monte Carlo non sono statisticamente indipendenti, perciò occorre fare una stima dell'errore tenendo conto dell'autocorrelazione. Si considera la quantità primaria A . L'autocorrelazione di A è:

$$(A_i A_{i+t}) = \langle A_i A_{i+t} \rangle - \langle A_i \rangle \langle A_{i+t} \rangle = \langle A_i A_{i+t} \rangle - \langle A \rangle^2 = \langle (A_i - \bar{A})(A_{i+t} - \bar{A}) \rangle \quad .$$

Per campioni con un numero molto grande di elementi e con il processo di simulazione indipendente dal tempo, l'autocorrelazione dipende solamente dalla differenza temporale t . Per campioni finiti c'è anche una dipendenza dalla configurazione iniziale. Allora la varianza corretta della media di A è:

$$\sigma_A^2 = \langle \left(\frac{1}{N} \sum_i (A_i - \langle A \rangle) \right)^2 \rangle = \sum_{t=-N}^{t=N} \frac{N-|t|}{N^2} \langle A_i A_{i+t} \rangle$$

$$\sigma_A^2 \xrightarrow{N \rightarrow \infty} \langle A^2 \rangle - \langle A \rangle^2 = \overline{A^2} - \bar{A}^2 \simeq (\overline{A^2} - \bar{A}^2) \frac{2t_{int,A}}{N}$$

con $t_{int,A} = \frac{1}{2} \sum_{t=-\infty}^{t=\infty} \frac{\langle A_i A_{i+t} \rangle}{\langle A^2 \rangle - \bar{A}^2}$ tempo di autocorrelazione integrato, ed il numero di misure indipendenti è $\frac{N}{2t_{int,A}}$.

4 Il modello in una dimensione

In una dimensione, si considera un reticolo periodico, in cui ad ogni sito è associato uno spin.

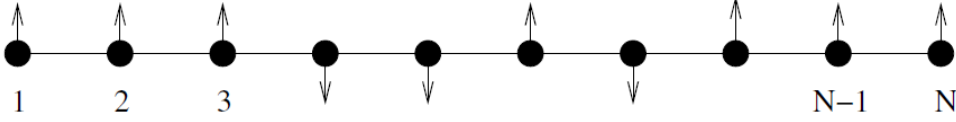


Figura 3: Catena lineare degli N spin

Allora la condizione di interazione tra primi vicini $\sum_{\langle ij \rangle} S_i S_j$ risulta essere $\sum_i S_i S_{i+1}$ con $S_{N+1} = S_1$. La funzione di partizione del sistema è:

$$Z = \sum_{S_i} e^{\sum_i (\beta J S_i S_{i+1} + \beta h S_i)} .$$

Si riscrive la somma $\sum_i S_i = \frac{1}{2} \sum_i (S_i + S_{i+1})$ e la funzione di partizione è:

$$Z = \sum_{S_i} e^{\sum_i (\beta J S_i S_{i+1} + \beta \frac{h}{2} S_i + \beta \frac{h}{2} S_{i+1})} = \sum_{S_i} \prod_i e^{\beta J S_i S_{i+1} + \beta \frac{h}{2} S_i + \beta \frac{h}{2} S_{i+1}}$$

e si definisce la "transfer matrix" $T_{S_i S_{i+1}}$ che può essere rappresentata come:

$$T = \begin{bmatrix} e^{\beta J + \beta h} & e^{-\beta J} \\ e^{-\beta J} & e^{\beta J - \beta h} \end{bmatrix} .$$

Allora:

$$Z = \sum_{S_1} \sum_{S_2} \cdots \sum_{S_N} \prod_i T_{S_i S_{i+1}} = \sum_{S_1} T_{S_1 S_1}^N = \text{tr } T^N$$

dato che la traccia di una matrice è invariante per cambio di base, si diagonalizza la matrice T , i cui autovalori sono λ_+ e λ_- :

$$T^N = \begin{bmatrix} \lambda_+^N & 0 \\ 0 & \lambda_-^N \end{bmatrix}$$

Dato che $Z = \text{tr } T^N = \lambda_+^N + \lambda_-^N = \lambda_+^N (1 + (\frac{\lambda_-}{\lambda_+})^N) \xrightarrow{N \rightarrow \infty} \lambda_+^N$. Dato che la funzione di partizione può essere espressa attraverso l'energia libera F , $Z = e^{-\beta F}$ e $F = -\frac{1}{\beta} \log Z = -\frac{N}{\beta} \log \lambda_+$. Si può ricavare la magnetizzazione che è:

$$M = -\frac{\partial F}{\partial h} \Big|_{T, N} = \langle \sum_i S_i \rangle$$

cioè la magnetizzazione è il valor medio su tutte le configurazioni dello spin. Esplicitando M in funzione di λ_+ si ha che $M = \frac{N}{\beta\lambda_+} \frac{\partial\lambda_+}{\partial h}$. Risolvendo il determinante secolare si ottiene:

$$\lambda_+ = e^{\beta J} (\cosh(\beta h) \pm \sqrt{\cosh^2(\beta h) - 2e^{-\beta J} \sinh(2\beta J)})$$

$$M = \frac{\sinh(\beta h)}{\sqrt{\cosh^2(\beta h) - 1 + e^{-4\beta J}}} \quad .$$

Si vede che M dipende da h campo magnetico esterno, quindi se $h = 0$, non vi è magnetizzazione residua nel materiale e non si ha alcuna transizione di fase.

4.1 Simulazione Monte Carlo del modello unidimensionale

Il numero di siti è $N = 10000$ e sono state imposte condizioni al bordo periodiche, come esposto nella trattazione teorica.

Non vi è alcuna transizione di fase nella magnetizzazione del materiale:

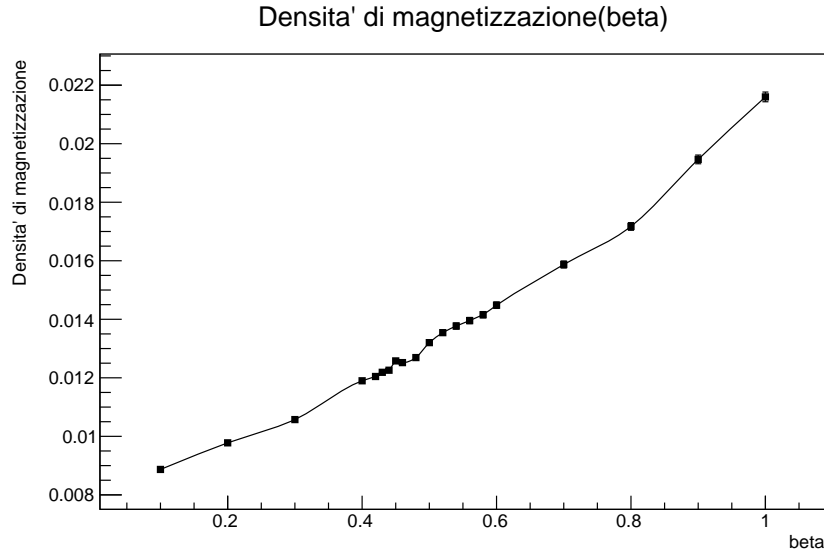


Figura 4: Densità di magnetizzazione in funzione di β

5 Il modello in due dimensioni

5.1 Argomento di Peierls

Con l'argomento di Peierls si può avere una stima ragionevole della temperatura critica nel modello di Ising. Se la dimensione del sistema è maggiore di 1, c'è una regione di basse temperature in cui c'è magnetizzazione spontanea non nulla. In alte temperature il sistema è disordinato. Allora deve esistere un valore critico della temperatura in cui avviene la transizione di fase. Si assume che tutti gli spin siano spin up. Le fluttuazioni termiche creano domini in cui si verifica un'inversione degli spin. Tali domini tendono a destabilizzare questo stato. Il costo energetico per creare tali domini è $\Delta E = 2JL$ con L lunghezza totale dei segmenti di confine. Una stima del numero di queste configurazioni si ottiene immaginando che la linea chiusa sia creata da un moto aleatorio sul reticolo con un numero totale di passi pari a L , cioè si ottengono 2^L configurazioni.

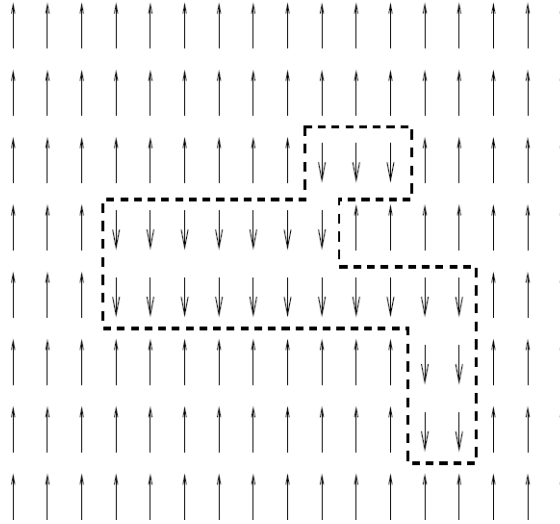


Figura 5: Linea chiusa che racchiude una regione in cui gli spin hanno un valore diverso

La corrispondente variazione di entropia del sistema sarà $\Delta E = k \ln 2^L$ e la variazione di energia libera sarà $\Delta F = \Delta E - T\Delta S = 2JL - kT \ln 2^L = L(2J - kT \ln 2)$. Il sistema è stabile rispetto alla formazione di tali domini di lunghezza L arbitraria se $\Delta F \geq 0$ cioè deve valere:

$$T \leq T_c = \frac{2J}{k \ln 2} = 2.885 \frac{J}{k}$$

che è molto vicino al valore reale della temperatura critica $T_c = 2.269 \frac{J}{k}$.

6 Relazione di dualità su reticolo quadrato

6.1 Sviluppo in serie ad alte temperature

Sia \mathcal{L} un reticolo quadrato, con M segmenti orizzontali e verticali che collegano tra loro i vari siti. Se si è nel limite termodinamico, cioè $M \rightarrow \infty$, allora M coincide con N , numero di siti nel reticolo. Siano J e J' rispettivamente le costanti di accoppiamento orizzontale e verticale. Se il campo magnetico esterno h è nullo, e sia σ_i lo spin nel sito i -esimo, la funzione di partizione è:

$$Z_N = \sum_{\{\sigma\}} e^{[K \sum_{(i,j)} \sigma_i \sigma_j + L \sum_{(i,k)} \sigma_i \sigma_k]}$$

avendo definito $K = \beta J$ e $L = \beta J'$. Ricorrendo all'identità:

$$e^{[x \sigma_i \sigma_l] = \cosh x (1 + \sigma_i \sigma_l \tanh x)}$$

la funzione di partizione può essere scritta come:

$$Z_N = (\cosh K \cosh L)^M \sum_{\{\sigma\}} \prod_{(i,j)} (1 + v \sigma_i \sigma_j) \prod_{(i,k)} (1 + w \sigma_i \sigma_k)$$

avendo definito le variabili $v = \tanh K$; e $w = \tanh L$. I parametri appena definiti sono minori di 1 per ogni valore della temperatura T del sistema, ed in particolare sono piccoli per alti valori di T . Se espando in serie le due produttorie nella funzione di partizione ottengo 2^{2M} termini. Si può rappresentare graficamente tale espansione, associando una linea disegnata nel link orizzontale (i,j) al fattore $v \sigma_i \sigma_j$, e una linea sul link verticale (i,k) al fattore $w \sigma_i \sigma_k$. Non si disegna alcuna linea se c'è il fattore 1. Ripetendo tale operazione per tutti i termini dello sviluppo, si ha una corrispondenza tra tali termini e la configurazione grafica sul reticolo \mathcal{L} . In generale si ha:

$$v^r w^s \sigma_1^{n_1} \sigma_2^{n_2} \sigma_3^{n_3} \dots$$

con r numero totale di linee orizzontali e s numero totale di linee verticali, ed n_i è il numero di linee in cui i è il sito finale. Basta sommare su tutti gli spin del reticolo per ottenere la funzione di partizione. Dato che ogni spin assume valore ± 1 , la somma è non nulla solo se gli n_1, n_2, \dots, n_N sono numeri pari, e in tal caso, il risultato è $2^N v^r w^s$. Allora la funzione di partizione è:

$$Z_N = 2^N (\cosh K \cosh L)^M \sum_P v^r w^s$$

in cui la somma è fatta su tutte le configurazioni di linee su \mathcal{L} aventi un numero pari di linee ad ogni sito, cioè su tutte le configurazioni poligonali chiuse del reticolo \mathcal{L} . Allora la funzione di partizione è data dalla quantità geometrica:

$$\Phi(v, w) = \sum_P v^r w^s \quad .$$

Si determinano i primi termini di tale funzione. Il primo termine è pari ad 1, poichè corrisponde al caso in cui nessun poligono è stato tracciato. Il secondo termine corrisponde al poligono chiuso più piccolo che è possibile tracciare sul reticolo, cioè il quadrato con lato unitario. Il numero di tali quadrati è N , ed ognuno di loro ha un peso di $(vw)^2$, perciò il secondo termine è $N(vw)^2$. Il successivo poligono chiuso è un rettangolo composto da sei linee del reticolo, e di tali rettangoli ce ne sono di due tipi, ognuno con degenerazione pari a N . Allora i primi termini di tale funzione sono:

$$\Phi(v, w) = 1 + N(vw)^2 + N(v^4w^2 + v^2w^4) + \dots$$

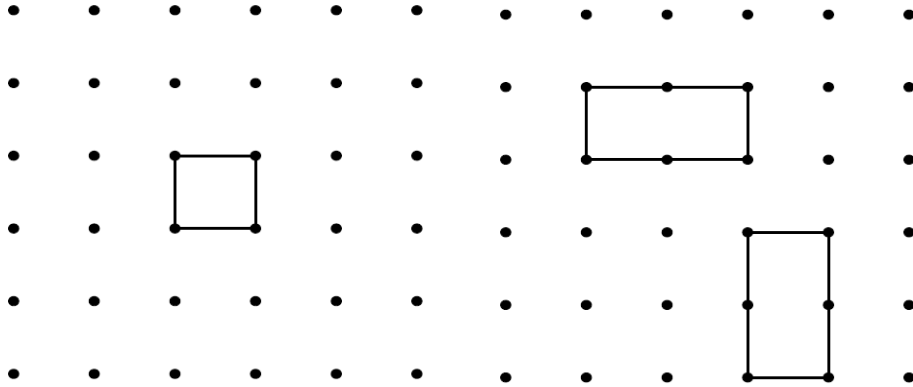


Figura 6: A sinistra: le configurazioni che fanno parte del secondo termine dell'espansione. A destra: le configurazioni che fanno parte del terzo termine dell'espansione

Al fine della relazione di dualità, non è necessario esplicitare tutti gli altri termini della serie.

6.2 Sviluppo in serie di basse temperature

Per basse temperature, gli spin tendono ad allinearsi tra di loro. Per una data configurazione degli spin, siano r ed s rispettivamente il numero di linee (link) verticali e orizzontali in cui due spin adiacenti sono antiparalleli. Dato che M è il numero totale di link verticali e orizzontali, si hanno $(M-r)$ link verticali e $(M-s)$ link orizzontali in cui gli spin adiacenti sono paralleli. Il contributo alla funzione di partizione di tale configurazione è:

$$e^{[K(M-2s)+L(M-2r)]}$$

A meno di costanti, tale espressione dipende solamente dal numero di link in cui gli spin sono antiparalleli.

Si introduce il concetto di reticolo duale: Per ogni reticolo planare \mathcal{L} si può definire \mathcal{L}_D reticolo duale di \mathcal{L} , ottenuto posizionando i suoi siti al centro delle facce del reticolo \mathcal{L} , e congiungendo tra loro i siti relativi alle facce adiacenti, cioè quelle con un segmento in comune.

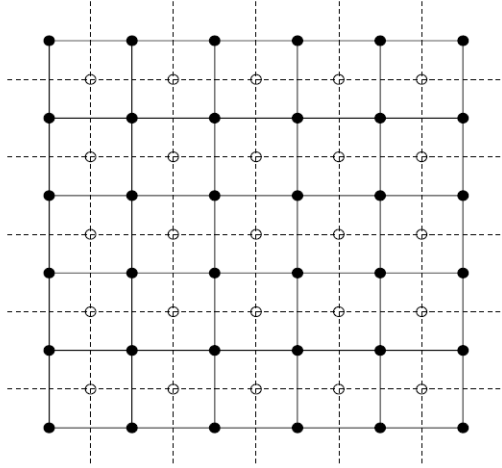


Figura 7: Reticolo duale quadrato

Data una configurazione, si associa ai segmenti antiparalleli, delle linee sul reticolo duale, cioè se due spin vicini sono antiparalleli, allora si traccia una linea lungo il segmento del reticolo duale che passa tra di loro. Allora si avranno r segmenti orizzontali e s segmenti verticali sul reticolo duale. Ci deve essere un numero pari di linee su ciascun sito, poichè vi è sempre un numero pari di cambiamenti successivi di spin tra le 4 facce adiacenti. Allora le linee così tracciate devono raccordarsi a formare dei poligoni chiusi sul reticolo duale.

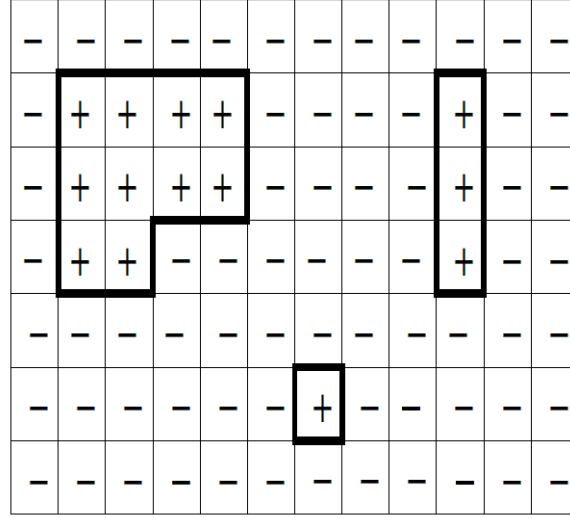


Figura 8: Poligoni che separano i domini con spin -1 e spin +1

I poligoni costruiti in tal modo sono i perimetri dei vari domini magnetici, al cui interno gli spin sono tutti allineati tra di loro. Per un dato insieme di poligoni, ci sono 2 configurazioni corrispondenti, ottenute scambiando il segno di tutti gli spin del reticolo. Allora la funzione di partizione può essere scritta come:

$$Z_N = 2e^{M(K+L)} \sum_{\tilde{P}} \exp[-(2Lr+2Ks)]$$

con la somma fatta su tutti i poligoni chiusi \tilde{P} sul reticolo duale. Quando $T \rightarrow 0$, sia K che L sono grandi, ed i termini dominanti sono dati da piccoli valori di r e s . Allora, la funzione di partizione può essere espressa come:

$$\tilde{\Phi}(e^{-2L}, e^{-2K}) = \sum_{\tilde{P}} e^{[-(2Lr+2Ks)]} \quad .$$

Si considerano i primi termini della serie. Il primo termine è pari ad 1 e corrisponde al caso in cui tutti gli spin hanno lo stesso valore. Il secondo termine corrisponde al caso in cui c'è solo uno spin con valore diverso dagli altri, allora in tal caso ci sono due linee antiparallele orizzontale e due verticali, che formano un quadrato. La degenerazione di tale termine è pari a N , poichè si può scegliere di flippare uno degli N spin dei siti del reticolo. Il terzo termine è dato dal rettangolo con sei segmenti che può estendersi in verticale o in orizzontale. Contando la degenerazione N e l'orientazione del rettangolo, tale termine contribuisce come $N(e^{-4L-8K} + e^{-8L-4K})$. La funzione di partizione è:

$$\tilde{\Phi}(e^{-2L}, e^{-2K}) = 1 + Ne^{-4L-4K} + N(e^{-4L-8K} + e^{-8L-4K}) + \dots$$

Si vede come i termini di $\tilde{\Phi}$ hanno la stessa origine dei termini della funzione Φ .

6.3 Autodualità

Si è visto che i due sviluppi hanno una funzione che ha la stessa origine geometrica, cioè è una somma su tutte le configurazioni poligonali del reticolo originario e del suo duale. Sostituendo K e L con \tilde{K} e \tilde{L} , dati da

$$\tanh \tilde{K} = e^{-2L}; \quad \tanh \tilde{L} = e^{-2K}$$

e confrontando le due funzioni geometriche si ha che

$$\tilde{\Phi}(e^{-2\tilde{K}}, e^{-2\tilde{L}}) = \Phi(v, w) \quad .$$

Ciò implica la seguente identità

$$\frac{Z_N[K, L]}{2^N (\cosh K \cosh L)^N} = \frac{Z_N[\tilde{K}, \tilde{L}]}{2 \exp[N(\tilde{K} + \tilde{L})]}$$

Si possono esprimere le equazioni per \tilde{K} e \tilde{L} in un modo più simmetrico:

$$\sinh 2\tilde{K} \sinh 2L = 1; \quad \sinh 2\tilde{L} \sinh 2K = 1$$

e allora l'identità può essere riscritta come:

$$\frac{Z_N[K, L]}{(\sinh 2K \sinh 2L)^{N/4}} = \frac{Z_N[\tilde{K}, \tilde{L}]}{(\sinh 2\tilde{K} \sinh 2\tilde{L})^{N/4}} \quad .$$

Tali equazioni mostrano l'esistenza di una simmetria nel modello di Ising bidimensionale e definiscono una mappa tra la fase ad alta e bassa temperatura del modello. Si può ora identificare il punto critico del sistema. Si considera il caso isotropico, in cui $K = L$, e $\tilde{K} = \tilde{L}$. Al punto critico la funzione di partizione presenta una divergenza: assumendo che accada al valore K_c , lo stesso deve accadere per $\tilde{K} = K_c$. Dato che esiste un unico punto critico, questi due valori devono coincidere. La condizione che deve soddisfare K_c è:

$$\sinh 2K_c = 1; \quad T_c = 2.26922 \dots J \quad .$$

Nel caso in cui le costanti di accoppiamento siano diverse, combinando le due equazioni per le varie costanti, si ottiene:

$$\sinh 2K \sinh 2L = \frac{1}{\sinh 2\tilde{K} \sinh 2\tilde{L}} \quad .$$

Tale equazione implica che sotto la mappa $(K, L) \rightarrow (\tilde{K}, \tilde{L})$, la regione A (in figura) è trasformata nella regione B e viceversa, lasciando invarianti i punti sulla retta

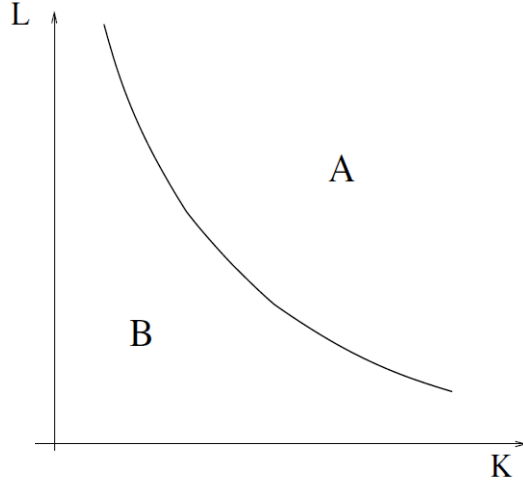


Figura 9: Curva critica

$$\sinh 2K \sinh 2L = 1 \quad .$$

Se c'è una linea di punti fissi in A , ce ne sarà un'altra in B . Assumendo che tale linea sia unica, l'equazione è quella espressa precedentemente. Allora questa è la condizione che racchiude la criticità del modello di Ising con due differenti costanti di accoppiamento.

7 Soluzione combinatoria

La soluzione combinatoria del modello di Ising bidimensionale si rifà all'espansione in alte temperature della funzione di partizione. Per semplicità si analizza il caso omogeneo, in cui c'è solo una costante di accoppiamento e pertanto solamente il parametro $v = \tanh \beta J$ è preso in considerazione. La funzione di partizione sul reticolo quadrato è:

$$Z_N = 2^N (1 - v^2)^{-N} \Phi(v)$$

con

$$\Phi(v) = \sum_r g_r v^r$$

con g_r numero di grafi chiusi, non necessariamente connessi, fatti da un numero r di link. La soluzione si articola in tre step:

-
1. si riduce la somma sui poligoni a una somma su cammini chiusi senza intersezioni;
 2. si trasforma la somma su cammini chiusi senza intersezione in una somma su tutti i possibili cammini chiusi;
 3. si riduce la somma ad un problema di cammini aleatori su un reticolo.

La somma su tutti i poligoni può essere organizzata in una somma sulle parti connesse dei grafi, ma bisogna fare attenzione a non contare più di una volta la stessa configurazione. Ogni grafo è pesato con un fattore $(-1)^n$ con n è il numero totale di autointersezioni di un ciclo. Si può esprimere il peso di un grafo in modo locale. Per i grafi con legami ripetuti $2\pi(l+1)$ è l'angolo totale di rotazione della tangente durante l'aggiramento di una maglia piana, con l intero positivo con stessa parità di ν che è il numero di autointersezioni della maglia considerata. Allora si assegna un fattore di fase $e^{i\alpha/2}$ ad ogni nodo della maglia, in cui si può avere un angolo di rotazione $\alpha = 0, \pm \frac{\pi}{2}$. Dopo l'aggiramento di tutta la maglia il prodotto di tali fattori darà $(-1)^{\nu+1}$. Per un insieme di più maglie si avrà $(-1)^{n+s}$, con $n = \sum \nu$.

Sia f_r la somma su tutte le maglie solitarie composte da r link, con il relativo peso associato ai suoi nodi. La somma su tutte le coppie di maglie con un numero totale r di link, sarà data da

$$\frac{1}{2!} \sum_{r_1+r_2=r} f_{r_1} f_{r_2}$$

con il fattore $2!$ che tiene conto delle permutazioni dei due indici che danno origine alla stessa coppia di maglie. Allora la funzione Φ può essere scritta come:

$$\Phi(v) = \sum_{s=0}^{\infty} (-1)^s \frac{1}{s!} \sum_{r_1, r_2, \dots, r_s=1}^{\infty} v^{r_1+r_2+\dots+r_s} f_{r_1} \dots f_{r_s}$$

dato che Φ comprende gli insiemi di maglie con una lunghezza qualsiasi, la somma può essere scritta come:

$$\sum_{r_1, r_2, \dots, r_s=1}^{\infty} v^{r_1+r_2+\dots+r_s} f_{r_1} \dots f_{r_s} = \left(\sum_{r=1}^{\infty} v^r f_r \right)^s .$$

Allora si ha

$$\Phi(v) = e^{[- \sum_{r=1}^{\infty} v^r f_r]} .$$

Rimane quindi da determinare la quantità f_r . Poichè su un reticolo quadrato si hanno quattro diverse direzioni di spostamento, si etichettano con l'indice $\mu = 1, 2, 3, 4$.

Si introduce la funzione $W_r(i, j, \mu)$ che è la somma su tutte le transizioni di lunghezza r che partono lungo la direzione μ_0 da un nodo iniziale di coordinate (i_0, j_0) e arrivano al nodo di coordinate (i, j) lungo la direzione μ . I cammini considerati sono pesati con il fattore $e^{i\alpha/2}$ per ciascun legame, con α angolo del cambiamento di direzione passando da un legame al successivo. Secondo tale definizione, $W_r(i_0, j_0, \mu_0)$ è la somma su tutte le maglie che partono dal punto di riferimento (i_0, j_0) lungo la direzione μ_0 e che vi ritornano. Allora si ha la seguente identità:

$$f_r = \frac{1}{2r} \sum_{i_0, j_0, \mu} W_r(i_0, j_0, \mu)$$

con il fattore $1/(2r)$ che tiene conto che nella somma su W_r ciascuna maglia può essere aggirata in due direzioni opposte e può avere uno qualsiasi dei suoi nodi r come punto iniziale. La funzione $W_r(i, j, \mu)$ soddisfa le seguenti equazioni:

$$\begin{aligned} W_{r+1}(i, j, 1) &= W_r(i-1, j, 1) + e^{-i\frac{\pi}{4}} W_r(i, j-1, 2) + 0 + e^{i\frac{\pi}{4}} W_r(i, j+1, 4) \\ W_{r+1}(i, j, 2) &= e^{i\frac{\pi}{4}} W_r(i-1, j, 1) + W_r(i, j-1, 2) + e^{-i\frac{\pi}{4}} W_r(i+1, j, 3) + 0 \\ W_{r+1}(i, j, 3) &= 0 + e^{i\frac{\pi}{4}} W_r(i, j-1, 2) + W_r(i+1, j, 3) + e^{-i\frac{\pi}{4}} W_r(i, j+1, 4) \\ W_{r+1}(i, j, 4) &= e^{-i\frac{\pi}{4}} W_r(i-1, j, 1) + 0 + e^{i\frac{\pi}{4}} W_r(i+1, j, 3) + W_r(i, j+1, 4) \end{aligned}$$

Possono essere scritte in forma generale come:

$$W_{r+1}(i, j, \mu) = \sum_{i', j', \mu'} \Lambda(ij\mu|i'j'\mu') W_r(i', j', \mu') \quad .$$

Tale espressione può essere considerata come l'equazione di un processo di Markov relativo a un moto aleatorio su di un reticolo, con probabilità di transizione tra i siti primi vicini pari al corrispondente elemento di matrice di Λ . Si ha che la probabilità di transizione relativa ad un cammino di lunghezza totale r è data dalla matrice Λ^r . In particolare le componenti diagonali di tale matrice sono le probabilità di tornare al punto iniziale dopo aver fatto un cammino chiuso di lunghezza r , cioè sono $W_r(i_0, j_0, \mu_0)$. Allora si ha:

$$\text{Tr } \Lambda^r = \sum_{i_0, j_0, \mu} W_r(i_0, j_0, \mu)$$

cioè

$$f_r = \frac{1}{2r} \text{Tr } \Lambda^r = \frac{1}{2r} \sum_a \lambda_a^r$$

con λ_a autovalori di Λ . Tornando alla funzione Φ si ottiene:

$$\Phi(v) = e^{[-\frac{1}{2} \sum_i \sum_{r=1}^{\infty} \frac{1}{r} v^r \lambda_i^r]} = e^{[\frac{1}{2} \sum_i \log(1-v\lambda_i)]} = \prod_i \sqrt{1-v\lambda_i} \quad .$$

Rimangono da determinare gli autovalori di Λ . La matrice è diagonalizzata rispetto alle coordinate k e l del reticolo usando la trasformata di Fourier, infatti sia:

$$W_r(p, q, \mu) = \sum_{k,l=0}^L e^{-\frac{2\pi i}{L}(pk+ql)} W_r(k, l, \mu)$$

con $N = L^2$, e prendendo la trasformata di Fourier delle equazioni che definiscono W si ottiene:

$$\begin{aligned} W_{r+1}(p, q, 1) &= \epsilon^{-p} W_r(p, q, 1) + \epsilon^{-q} \alpha^{-1} W_r(p, q, 2) + \epsilon^q \alpha W_r(p, q, 4) \\ W_{r+1}(p, q, 2) &= \epsilon^{-p} \alpha W_r(p, q, 1) + \epsilon^{-q} W_r(p, q, 2) + \epsilon^p \alpha^{-1} W_r(p, q, 3) \\ W_{r+1}(p, q, 3) &= \epsilon^{-q} \alpha W_r(p, q, 2) + \epsilon^p W_r(p, q, 3) + \epsilon^q \alpha^{-1} W_r(p, q, 4) \\ W_{r+1}(p, q, 4) &= \epsilon^{-p} \alpha^{-1} W_r(p, q, 1) + \epsilon^p \alpha W_r(p, q, 3) + \epsilon^q W_r(p, q, 4) \end{aligned}$$

con $\epsilon = e^{2\pi i/L}$ e $\alpha = e^{i\pi/4}$. Si ottiene:

$$\Lambda(p, q, \mu | p, q, \mu') = \begin{pmatrix} \epsilon^{-p} & \alpha^{-1} \epsilon^{-q} & 0 & \alpha \epsilon^q \\ \alpha \epsilon^{-p} & \epsilon^{-q} & \alpha^{-1} \epsilon^p & 0 \\ 0 & \alpha \epsilon^{-q} & \epsilon^p & \alpha^{-1} \epsilon^q \\ \alpha^{-1} \epsilon^{-p} & 0 & \alpha \epsilon^p & \epsilon^q \end{pmatrix}$$

Si calcola

$$\prod_{i=1}^4 (1 - v \lambda_i) = \text{Det}(1 - v \Lambda) = (1 + v^2)^2 - 2v(1 - v^2) \left(\cos \frac{2\pi p}{L} + \cos \frac{2\pi q}{L} \right)$$

e ritornando all'espressione iniziale della funzione di partizione si ha:

$$Z_N = 2^N (1 - v^2)^{-N} \prod_{p,q}^L \left[(1 + v^2) - 2v(1 - v^2) \left(\cos \frac{2\pi p}{L} + \cos \frac{2\pi q}{L} \right) \right]^{1/2}$$

con energia libera

$$\begin{aligned} -\frac{F(T)}{kT} &= \log Z_N \\ &= N \log 2 - N \log (1 - v^2) \\ &\quad + \frac{1}{2} \sum_{p,q=0}^L \log \left[(1 + v^2)^2 - 2v(1 - v^2) \left(\cos \frac{2\pi p}{L} + \cos \frac{2\pi q}{L} \right) \right] \quad . \end{aligned}$$

Nel limite $L \rightarrow \infty$, la somma diventa un integrale e può essere espressa come:

$$\begin{aligned} -\frac{F(T)}{kT} &= \log Z_N \\ &= N \log 2 - N \log (1 - v^2) \\ &\quad + \frac{N}{2(2\pi)^2} \int_0^{2\pi} \int_0^{2\pi} \log \left[(1 + v^2)^2 - 2v(1 - v^2) (\cos \omega_1 + \cos \omega_2) \right] d\omega_1 d\omega_2 \quad . \end{aligned}$$

L'espressione mostra che $F(T)$ è una quantità estensiva, proporzionale al numero N di siti del reticolo. A parte per $v = 1$, corrispondente a $T = 0$, F ha un punto singolare per un valore finito di T quando l'argomento del logaritmo dell'integrale si annulla, Allora per $\cos \omega_1 = \cos \omega_2 = 1$, si ha:

$$(1 + v^2)^2 - 4v(1 - v^2) = (v^2 + 2v - 1)^2 \quad .$$

Tale espressione ha un minimo in cui si annulla, per il valore (positivo):

$$v = v_c = \sqrt{2} - 1 \quad .$$

La corrispondente temperatura critica è:

$$\tanh \frac{J}{kT_c} = v_c, \quad kT_c = 2.26922 \dots J$$

che determina il punto di transizione di fase.

Si riportano i risultati relativi alle altre osservabili del sistema. Il calore specifico attorno la singolarità è

$$C \sim B \log |T - T_c|$$

che corrisponde all'esponente critico

$$\alpha = 0 \quad .$$

Vicino alla temperatura critica, la lunghezza di correlazione è

$$\xi \simeq |T - T_c|^{-1}$$

e l'esponente critico

$$\nu = 1 \quad .$$

La magnetizzazione spontanea è:

$$M^2 = \begin{cases} \left[1 - \left(\frac{1-v^2}{2v} \right)^4 \right]^{1/4}, & T < T_c \\ 0, & T > T_c \end{cases}$$

perciò il valore dell'esponente critico è

$$\beta = \frac{1}{8} \quad .$$

Alla temperatura critica il correlatore decade algebricamente come

$$\langle \sigma_i \sigma_j \rangle \simeq \frac{1}{|i-j|^{1/4}}$$

e l'esponente critico è

$$\eta = \frac{1}{4} \quad .$$

I rimanenti esponenti critici possono essere ricavati dalle relazioni di scala:

$$\delta = 15; \quad \gamma = \frac{7}{4} \quad .$$

8 Simulazione Monte Carlo del modello bidimensionale

Si riportano di seguito i risultati ottenuti con il programma di simulazione. Si ricava l'andamento della densità di magnetizzazione in funzione della variabile β per differenti valori di $L = \{10, 20, 30, 40, 60, 80\}$, lato del reticolo quadrato.

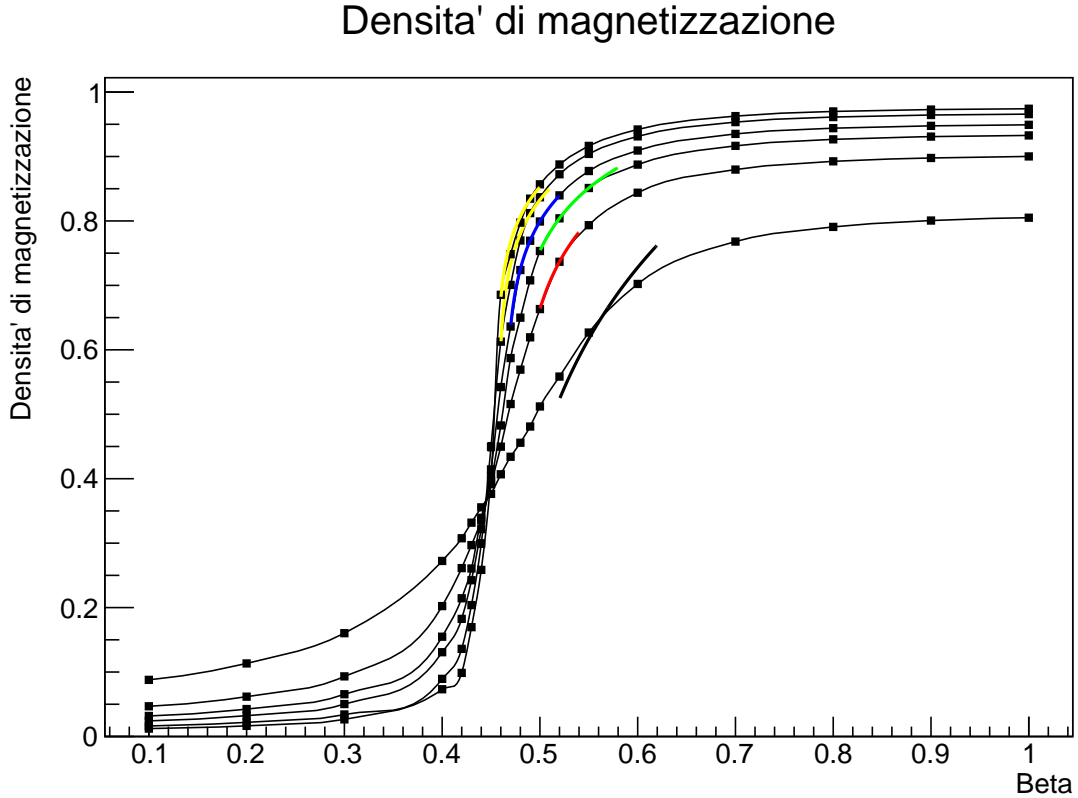


Figura 10: Densità di magnetizzazione in funzione di β .

Si vede che aumentando il lato del reticolo la densità di magnetizzazione tende ad 1, cioè corrisponde al caso in cui tutti gli spin sono allineati, cioè il materiale ha un comportamento puramente ferromagnetico. Da tale grafico è possibile ricavare la β_c in funzione della dimensione del lato, che sarà utile in seguito per ricavare la β_c per $N \rightarrow \infty$. Nel limite termodinamico la magnetizzazione in funzione della temperatura presenta il seguente andamento:

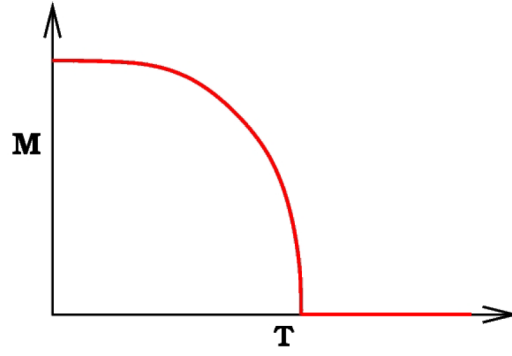


Figura 11: Magnetizzazione in funzione della temperatura nel limite termodinamico.

La capacità termica e la suscettività magnetica sono state calcolate con il Teorema di fluttuazione dissipazione, come:

$$\chi(\beta) = \beta (\langle M^2 \rangle - \langle M \rangle^2)$$

$$C(\beta) = \frac{\partial U(\beta)}{\partial \beta} = \beta^2 (\langle E^2 \rangle_\beta - \langle E \rangle_\beta^2) \quad .$$

Dalla simulazione è stato possibile ricavare anche la densità di energia in funzione di β per diversi L . La curva più in alto è riferita al lato $L = 10$.

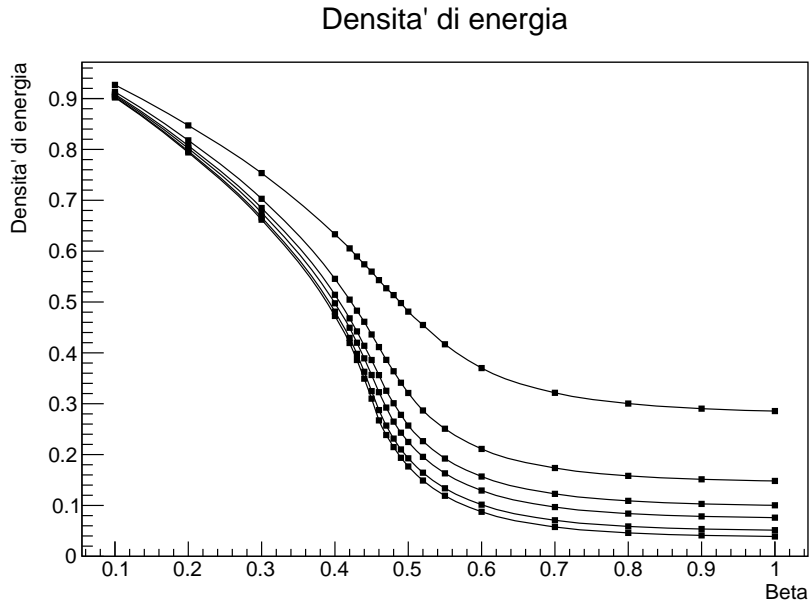


Figura 12: Densità di energia in funzione di β per diversi valori del lato del reticolo.

Sono riportati di seguito gli andamenti della capacità termica e della suscettività magnetica in funzione di β per diversi valori di L .

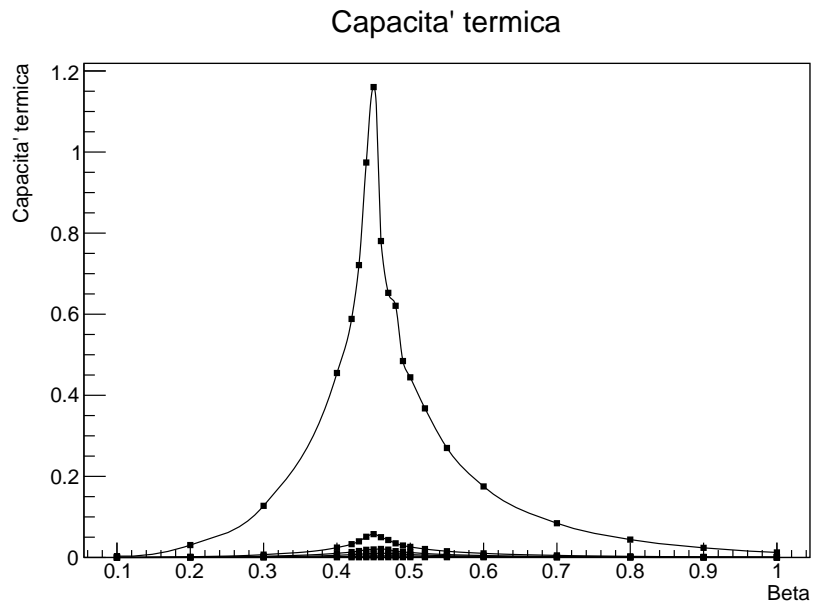


Figura 13: Capacità termica in funzione di β per diversi valori del lato.

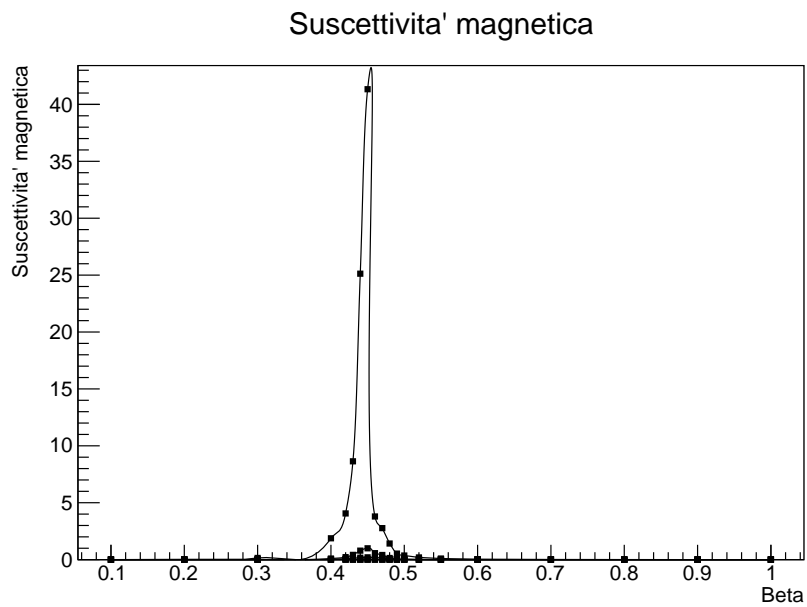
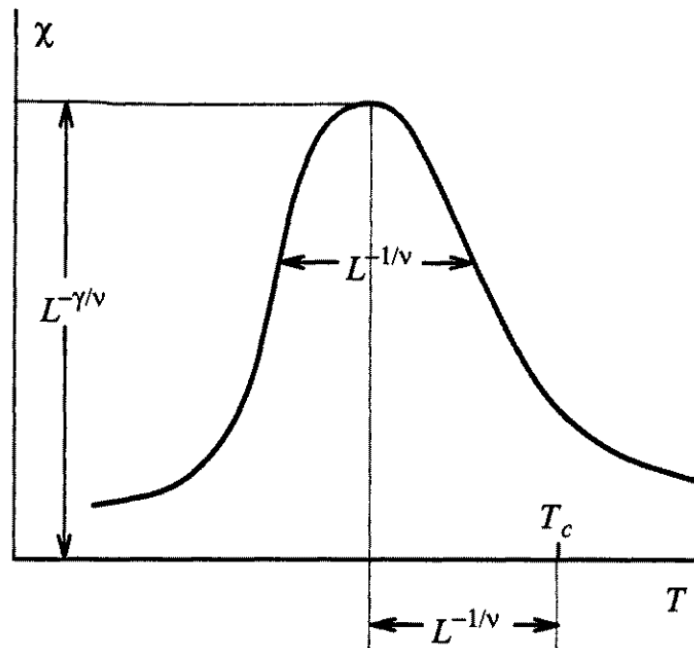


Figura 14: Suscettività magnetica in funzione di β per diversi valori del lato.

Attraverso il "finite-size scaling" è possibile ricavare l'esponente critico ν dalla larghezza a metà altezza della capacità termica o della suscettività magnetica.



Misurando le varie larghezze a metà altezza della suscettività magnetica, si ricava il valore dell'esponente critico ν :

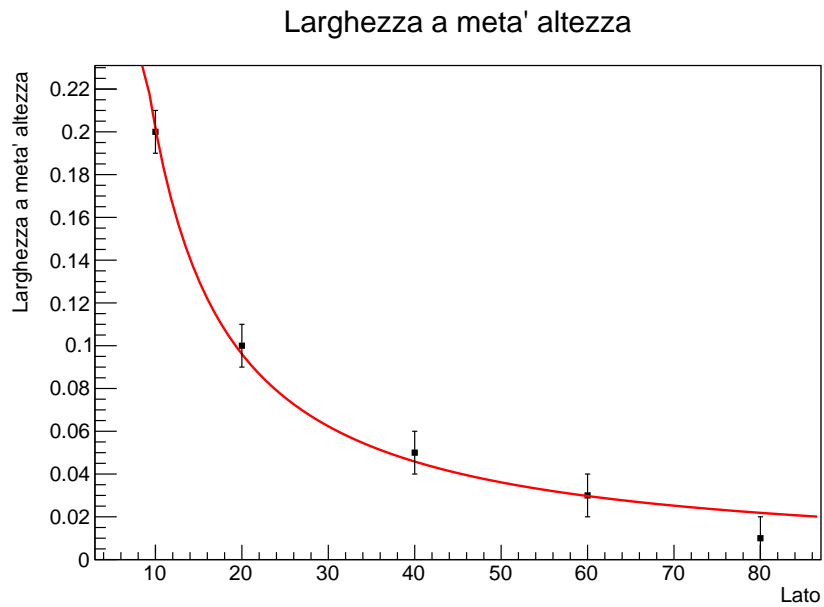


Figura 15: Larghezza a metà altezza in funzione del lato del reticolo

Nel fit si è utilizzata la funzione $\frac{a}{L^\nu}$ e si ottiene

$$a = (2.4 \pm 0.6), \nu = (1.1 \pm 0.1)$$

$$p - value = 62\%$$

$$\chi^2 = 1.75 \text{ con 3 gradi di libertà.}$$

Si ottiene che il valore trovato per l'esponente critico è compatibile col valore teorico $\nu = 1$. Ricavato ν , è possibile estrapolare la β_c nel limite termodinamico, poichè l'andamento della β_c in funzione del lato è il seguente:

$$\beta_c(L) = \beta_c + bL^{-\frac{1}{\nu}}$$

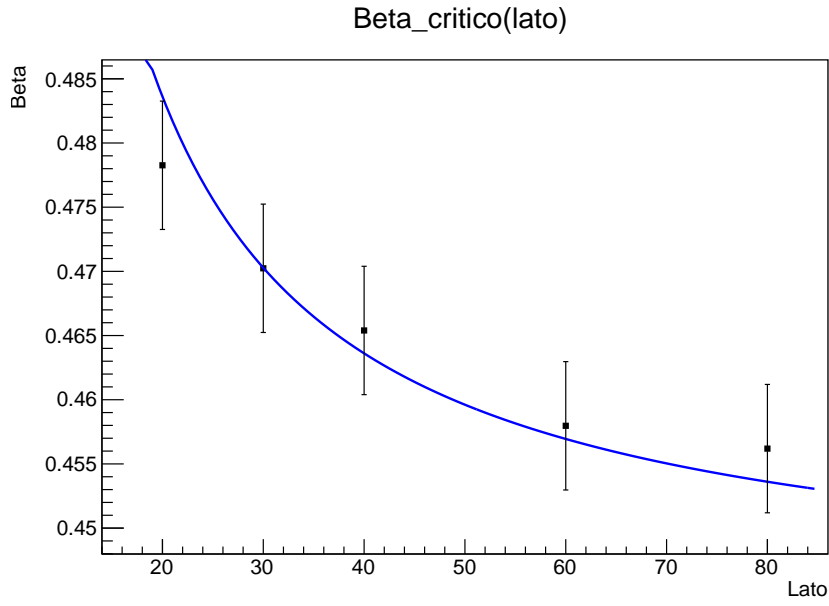


Figura 16: β_c in funzione del lato del reticolo

Si ottiene:

$$b = (0.8 \pm 0.1) \frac{1}{J}$$

$$\beta_c = (0.44 \pm 0.01) \frac{1}{J}, \beta_{c,teorica} = 0.44 \frac{1}{J}$$

$$p - value = 66\%$$

$$\chi^2 = 1.58 \text{ con 3 gradi di libertà.}$$

E' stato possibile ottenere anche l'esponente β_{exp} dalla densità di magnetizzazione in funzione di β , nel reticolo con lato 40, ottenendo:

$$\beta_{exp} = (0.117 \pm 0.001) \text{ con } \beta_{exp,teorico} = 0.125$$

$$\chi^2 = 113 \text{ con 3 gradi di libertà.}$$

8.1 Simulazione Monte Carlo del modello in tre dimensioni

E' stato possibile estendere il processo di simulazione anche in tre dimensioni, ricavando la β_c con le medesime considerazioni sul finite-size scaling nel caso bidimensionale.

Si riporta di seguito l'andamento della densità di magnetizzazione per reticoli cubici di lato $L = \{5, 10, 15, 20, 30\}$.

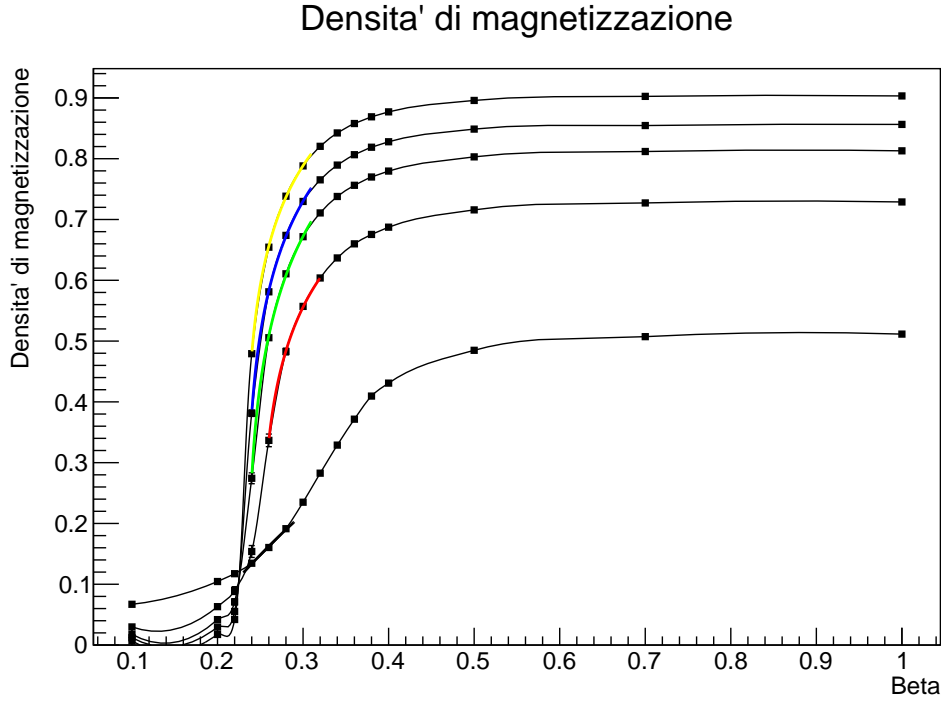


Figura 17: Densità di magnetizzazione in funzione di β

Assumendo noto l'indice critico $\nu = 0.63$, l'andamento della β_c in funzione di L , lato del reticolo, sarà:

$$\beta_c(L) = \beta_c + bL^{-\frac{1}{\nu}} \quad .$$

Si ottiene:

$$\begin{aligned} b &= (0.05 \pm 0.01) \frac{1}{J} \\ \beta_c &= (0.228 \pm 0.001) \frac{1}{J}, \quad \beta_{c,teorica} = 0.221 \frac{1}{J} \\ p-value &= 56\% \\ \chi^2 &= 0.34 \text{ con 1 grado di libertà.} \end{aligned}$$

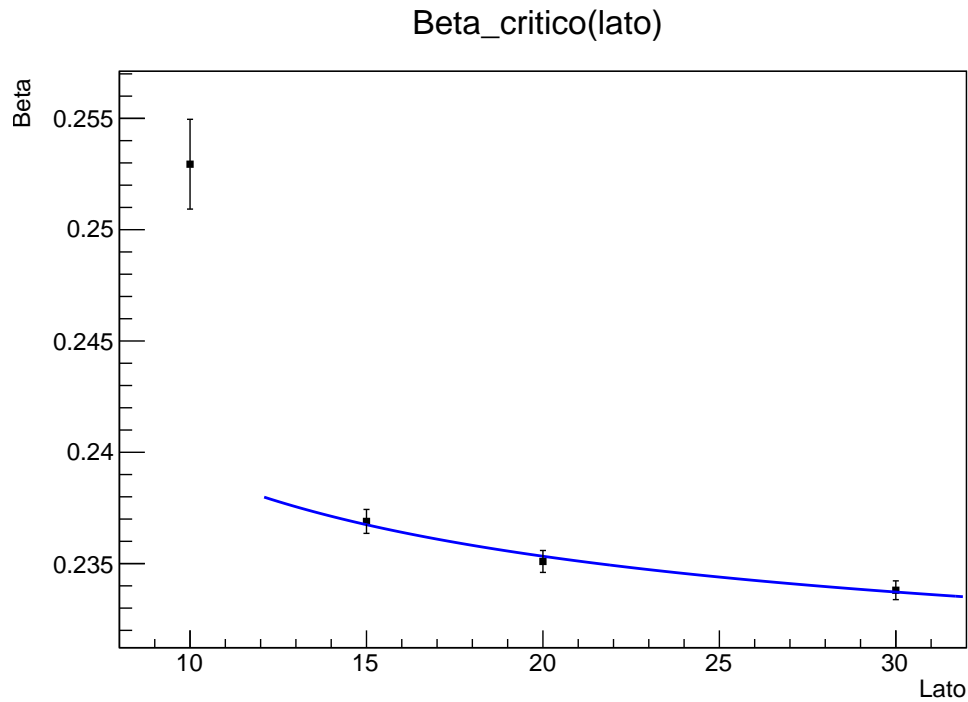


Figura 18: β_c in funzione del lato del reticolo

Appendice 1 Estratto del codice sorgente per la simulazione del modello bidimensionale

```
#include <iostream>
#include <TGraphErrors.h>
#include <TAxis.h>
#include <TCanvas.h>
#include <TF1.h>
#include <iomanip>
#include <TMath.h>
#include <ctime>
#include <time.h>
#include <cmath>

using namespace std;

int avanti(int posizione, int lato) {
    posizione = posizione + 1;
```

```

if (posizione == lato)
posizione = 0;
return posizione;
}

int indietro(int posizione, int lato) {
posizione = posizione - 1;
if (posizione == -1)
posizione = 0;
return posizione;
}

double** set_spin(double **s,double lato) { //setta gli spin in modo random

    for (int i = 0; i < lato; i++) {
for (int j = 0; j < lato; j++) {

        s[i][j] = rand()%2;
        if(s[i][j] == 0){ s[i][j] = -1;}
    }
}
return s;
}

bool** set_spin_flippato(bool **spin_flippato,double lato) {
for (int i = 0; i < lato; i++) {
for (int j = 0; j < lato; j++) {

spin_flippato[i][j] = false;
}
}
return spin_flippato;

}

double* binning(double* magnetizzazione, int &iterazioni){

```

```

    int iterazioni_new = iterazioni / 2;
    double* magnetizzazione_new = new double [iterazioni_new];
    for(int i = 0; i < iterazioni_new; i++){
        magnetizzazione_new[i] = (magnetizzazione[2*i] + magnetizzazione[2*i+1])/2;}
    iterazioni = iterazioni/2;
    for(int i = 0; i<iterazioni; i++){
        magnetizzazione[i] = magnetizzazione_new[i];}

    return magnetizzazione;

}

double media(double* x, double misure){ //calcola la media di x
    double valore_medio = 0;
    for(int i = 0; i<misure;i++){
        valore_medio += x[i]/misure;
    }
    return valore_medio;}

double varianza_(double* x, double x_medio, double misure){ //calcola la varianza
sulla media di x rispetto a x medio su un numero di misure dato
    double varianza_x = 0;
    double somma_quadrati = 0;

    for(int i = 0; i<misure; i++){
        somma_quadrati += pow(x[i],2);}
    for(int i = 0; i<misure; i++){
        varianza_x = (somma_quadrati/ misure - pow(x_medio,2))/misure;
    }
    return varianza_x;}

void check_magnetizzazione(double magnetizzazione_finale, double counter){
    if( magnetizzazione_finale > 1 || magnetizzazione_finale<-1)
    { counter = counter -1;} }

void check_varianza(double &varianza){
    if(varianza<0){

```

```

        varianza = -varianza;}}
void check_positivo(double &x){
    if(x<0){ x= -x;}
}

//ciclo creato nell'heap
// passo alla funzione gli indirizzi di Ed e m per poterli modificare
double** Ciclo_Monte_Carlo(double **s, bool** spin_flippato, int lato,
    double beta,
    double h, double &Ed, double &m){

    double M_totale = 0;
    double E_totale = 0;
    double spin_corrente = 0;
    double somma_vicini = 0;
    double forza_efficace = 0;
    double new_spin = 0;
    double random = 0;
    int avanti_x, avanti_y, indietro_x, indietro_y;
    int siti_visitati = 0;
    int i = 0;
    int j = 0;

    while(siti_visitati<pow(lato,2)){
        i = rand()%lato;
        j = rand()%lato;
        if( spin_flippato[i][j] == false){
            spin_corrente = s[i][j];
            avanti_x = avanti(i, lato);
            avanti_y = avanti(j, lato);
            indietro_x = indietro(i, lato);
            indietro_y = indietro(j, lato);
            if (avanti_x == 0) {
                s[avanti_x][j] = 0;
            }
            if (avanti_y == 0) {
                s[i][avanti_y] = 0;
            }
        }
    }
}

```

```

    if (indietro_x == 0) {
        s[indietro_x][j] = 0;
    }
    if (indietro_y == 0) {
        s[i][indietro_y] = 0;
    }
    somma_vicini = s[i][avanti_y] + s[i][indietro_y] +
        s[avanti_x][j] + s[indietro_x][j];
    forza_efficace = somma_vicini + h;
    random = ((double) rand()/RAND_MAX); // random tra 0 e 1
    if (exp(-(beta * forza_efficace * spin_corrente * 2)) > random) {
        new_spin = -spin_corrente;
    }
    else new_spin = spin_corrente;
    s[i][j] = new_spin;
    M_totale = M_totale + new_spin;
    E_totale = E_totale -(0.5 * somma_vicini + h) * new_spin;
    m = M_totale / pow(lato, 2);
    Ed = E_totale / pow(lato, 2);
    siti_visitati++;
    spin_flippato[i][j] = true;
}

}

return s;
}

void Fit()
{

    const int pmisure = 20; // sono le magnetizzazioni finali che trova
    const int bin = 8;

```

```

bool binned = false; // se true, fa il binning
int iterazioni = 100000; // numero di volte che ripete il ciclo monte carlo
int iterazioni_iniziali = iterazioni;
int lato = 30; //con 30 o 40 viene
double Iterazioni = iterazioni; //serve quando calcolo le medie
int random = 0;
double beta = 0.5;
//double beta_finale[pmeasure]; // serve per fare il grafico
double sbeta_finale[pmeasure];
double h = 0;
double Ed = 0;
double m = 0;
int seed = 0;
int iterazione = 0;
double* store_m = new double [iterazioni];
double* magnetizzazione = new double [iterazioni];
double* Energyd = new double [iterazioni];
double magnetizzazione_finale[pmeasure]; //media delle magnetizzazioni
    dopo le iterazioni
double smagnetizzazione_finale[pmeasure];
double Energyd_finale[pmeasure];
double sEnergyd_finale[pmeasure];
double Energia[pmeasure];
double sEnergia[pmeasure];
double Pmeasure = pmeasure;
double std[pmeasure];
double varianza[pmeasure];
double std_Energyd[pmeasure];
double varianza_Energyd[pmeasure];
int iterazioni_new = iterazioni / 2;
    double* magnetizzazione_new = new double [iterazioni_new];

//autocorrelazione
double t_auto[pmeasure]; //tempo di autocorrelazione
double step_1[pmeasure];
double t_auto_energia[pmeasure]; //tempo di autocorrelazione
double step_1_energia[pmeasure];
double errore_magnetizzazione[pmeasure];

```

```

double errore_energia[pmisure];
double varianza_corretta[pmisure];
double varianza_corretta_energia[pmisure];

//capacita' termica
double capacita_termica[pmisure];
double scapacita_termica[pmisure];

//suscettibiita' magnetiza
double X[pmisure];
double sX[pmisure];
double Magnetizzazione[pmisure];
double sMagnetizzazione[pmisure];

srand (time(NULL)); //seed del random generator

double beta_finale[] = {0.1, 0.2, 0.3, 0.4, 0.42, 0.43, 0.44 , 0.45 , 0.46,
0.47, 0.48, 0.49, 0.5, 0.52, 0.55, 0.6, 0.7, 0.8, 0.9, 1}; //20 misure

for(int i = 0; i<pmisure; i++){
    magnetizzazione[i] = 0;
    Energyd[i] = 0;
    magnetizzazione_finale[i] = 0;
    smagnetizzazione_finale[i] = 0;
    Energyd_finale[i] = 0;
    sEnergyd_finale[i] = 0;
    Energia[i] = 0;
    sEnergia[i] = 0;
    std[i] = 0;
    varianza[i] = 0;
    std_Energyd[i] = 0;
    varianza_Energyd[i] = 0;
    t_auto[i] = 0;
    step_1[i] = 0;
    t_auto_energia[i] = 0;
    step_1_energia[i] = 0;

```

```

    errore_magnetizzazione[i] = 0;
    errore_energia[i] = 0;
    varianza_corretta[i] = 0;
    varianza_corretta_energia[i] = 0;
    store_m[i] = 0;
    capacita_termica[i] = 0;
    scapacita_termica[i] = 0;
    X[i] = 0;
    sX[i] = 0;
    Magnetizzazione[i] = 0;
    sMagnetizzazione[i] = 0;
}

for(int i = 0; i<iterazioni; i++){ //inizializzo il vettore
    magnetizzazione[i] = 0;
    Energyd[i] = 0;
}

    double** s; //contiene una configurazione del sistema
s = new double* [lato];
for (int i = 0; i < lato; i++) {
s[i] = new double[lato];
}

bool** spin_flippato;
spin_flippato = new bool* [lato];
for (int i = 0; i < lato; i++) {
spin_flippato[i] = new bool[lato];
}

set_spin(s,lato);
set_spin_flippato(spin_flippato,lato);

for(int counter = 0; counter < pmisure; counter++){
    beta = beta_finale[counter];
    cout<<" beta = "<<beta<<endl;
    iterazioni = iterazioni_iniziali;
}

```

```

for (int k = 0; k < iterazioni; k++) {

    Ciclo_Monte_Carlo(s, spin_flippato, lato, beta, h, Ed, m);
    set_spin_flippato(spin_flippato,lato);
    store_m[k] = m;
    while( m > 1 || m < -1){
        m = store_m[k-1];
        Ciclo_Monte_Carlo(s, spin_flippato, lato, beta, h, Ed, m);}
    if(m>0){ magnetizzazione[k] = m;}
        else magnetizzazione[k] = -m;
    Energyd[k] = Ed;
}

if(binned){
for( int i = 0; i<bin; i++){
    binning(magnetizzazione,iterazioni);
    Iterazioni = iterazioni;
}}

magnetizzazione_finale[counter] = media(magnetizzazione, Iterazioni);
varianza[counter] = varianza_(magnetizzazione, magnetizzazione_finale[counter],
Iterazioni); //varianza della media
check_varianza(varianza[counter]);
std[counter] = sqrt(varianza[counter]);

Energyd_finale[counter] +=media(Energyd, Iterazioni);
varianza_Energyd[counter] += varianza_(Energyd, Energyd_finale[counter],
Iterazioni); //varianza della media
check_varianza(varianza_Energyd[counter]);
std_Energyd[counter] = sqrt(varianza_Energyd[counter]);

//calcolo tempo di autocorrelazione integrato

for( int t = 0; t<iterazioni; t++){
    for(int i = 0; i<iterazioni;i++){
        if(i+t<iterazioni){

```

```

step_1[counter] += (magnetizzazione[i]*magnetizzazione[i+t] -
    Iterazioni*pow(magnetizzazione_finale[counter],2))/Iterazioni;
step_1_energia[counter] += (Energyd[i]*Energyd[i+t] - Iterazioni*
    pow(Energyd_finale[counter],2))/Iterazioni;

    }}}

t_auto[counter] = 0.5 * (step_1[counter])/(varianza[counter]);
//tempo di autocorrelazione integrato
t_auto_energia[counter] = 0.5 * (step_1_energia[counter])/
    (varianza_Energyd[counter]);
//tempo di autocorrelazione integrato
check_positivo( t_auto[counter]);
check_positivo( t_auto_energia[counter]);
//calcolo l'errore corretto sul valore medio
for(int t = 0; t<iterazioni; t++){
    for( int i = 0; i<iterazioni; i++){
        if( i+t<iterazioni){
varianza_corretta[counter] += (1/pow(Iterazioni,2)) * (Iterazioni -
    abs(t))*(magnetizzazione[i]*magnetizzazione[i+t] -
    magnetizzazione_finale[counter])
    /Iterazioni;
varianza_corretta_energia[counter] += (1/pow(Iterazioni,2)) * (Iterazioni -
    abs(t))*(Energyd[i]*Energyd[i+t] - Energyd_finale[counter])/Iterazioni;
        }}}

check_varianza(varianza_corretta[counter]);
check_varianza(varianza_corretta_energia[counter]);

//errore corretto sui valori medi
errore_magnetizzazione[counter] = sqrt(varianza_corretta[counter]);
errore_energia[counter] = sqrt(varianza_Energyd[counter]*2*
    t_auto_energia[counter]/Iterazioni); //sqrt(varianza_corretta_energia[counter]);

//stampo i risultati
    cout<<" magnetizzazione = "<<magnetizzazione_finale[counter]<<endl;
    cout<<" smagnetizzazione = "<<std[counter]<<endl;

```

```

cout<<" Densita' di energia = "<<Energyd_finale[counter]<<endl;
cout<<" sDensita' di energia = "<<std_Energyd[counter]<<endl;
cout<<" tempo di autocorrelazione = "<<t_auto[counter]<<endl;
cout<<" tempo di autocorrelazione energia = "<<t_auto_energia[counter]<<endl;
check_magnetizzazione(magnetizzazione_finale[counter],counter);

} //fine del ciclo for con counter -> pmisure

for(int i = 0; i<pmisure; i++){
    Energia[i] =( Energyd_finale[i] +2 ) * pow(lato,2) ; // metto il +2 per
    scalare tutte le energie e mettere lo zero corrispondente all'energia minore
    sEnergia[i] = std_Energyd[i] * pow(lato,2);
    Magnetizzazione[i] = magnetizzazione_finale[i] * pow(lato,2);
    sMagnetizzazione[i] = std[i] * pow(lato,2);

}

//calcolo la capacita' termica del sistema

for(int i=0; i<pmisure;i++){
    capacita_termica[i] = pow(sEnergia[i]*beta_finale[i],2);
    scapacita_termica[i] = pow(sEnergia[i]/sqrt(2*(Iterazioni-1)) , 2);
    X[i] = pow(sMagnetizzazione[i]*beta_finale[i],2);
    sX[i] = pow(sMagnetizzazione[i]/sqrt(2*(Iterazioni-1)) , 2);
    cout<<"Beta = "<<beta_finale[i]<<endl;
    cout<<" Capacita' termica =( "<<capacita_termica[i]<<" +- "
    <<scapacita_termica[i]<<" )" <<endl;
    cout<<" Suscettibilita' magnetica =( "<<X[i]<<" +- "<<sX[i]<<" )" <<endl;

}

for(int i = 0; i<pmisure; i++){
    sbeta_finale[i] = 0;}

cout << "\n\n --- Relazione tra beta e densita' di magnetizzazione ---" <<endl;

```

```

// Grafico 1
TCanvas *c1 = new TCanvas("c1","Densita' di magnetizzazione(beta)",
200,10,600,400);
//c1->SetLogx();
c1->SetFillColor(0);
c1->cd();
TGraphErrors *g1 = new TGraphErrors(pmisure,beta_finale,
magnetizzazione_finale, sbeta_finale,std);
g1->SetMarkerSize(0.6);
g1->SetMarkerStyle(21);
g1->SetTitle("Densita' di magnetizzazione(beta)");
g1->GetXaxis()->SetTitle("beta");
g1->GetYaxis()->SetTitle("Densita' di magnetizzazione ");
g1->Draw("ACP");

// Fit 1
cout << "\n\n Ipotesi funzione: pow([1]-x,1/8) \n" <<endl;
TF1 *funz1 = new TF1("funz1","pow((1/[0] - 1/x), [1]) ",0.5,0.65);
funz1->SetParameter(0,0.48);
funz1->SetParameter(1,0.120);
funz1->SetLineColor(1);
g1->Fit(funz1,"RM+");
cout<<endl;
float beta_critico = funz1->GetParameter(0);
float sbeta_critico = funz1->GetParError(0);
double beta_critico_teorico = 1/2.27;
cout<<"Beta critico =( "<<beta_critico<<" +- "<<sbeta_critico<<" )" <<endl;
cout<<"Beta critico teorico = "<<beta_critico_teorico<<endl;

float Esponente_b = funz1->GetParameter(1);
float sEsponente_b = funz1->GetParError(1);
double esponente_b_teorico = 0.125;
cout<<"Esponente critico =( "<<Esponente_b<<" +- "<<sEsponente_b<<" )" <<endl;
cout<<"Esponente critico teorico = "<<esponente_b_teorico<<endl;

```

```

cout << "Chi^2:" << funz1->GetChisquare() << ", number of DoF: "
<< funz1->GetNDF() << " (Probability: " << funz1->GetProb() << ")." << endl;

//relazione tra Energia e beta

// Grafico 2
TCanvas *c2 = new TCanvas("c2","Energia(beta)",200,10,600,400);
c2->SetFillColor(0);
c2->cd();
TGraphErrors *g2 = new TGraphErrors(pmisure,beta_finale,Energia
,sbeta_finale,sEnergia);
g2->SetMarkerSize(0.6);
g2->SetMarkerStyle(21);
g2->SetTitle("Energia(beta)");
g2->GetXaxis()->SetTitle("beta");
g2->GetYaxis()->SetTitle("Energia");
g2->Draw("ACP");

//Cpacita' Termica
cout<<"Capacita' termica in funzione di beta"<<endl;

// Grafico 3
TCanvas *c3 = new TCanvas("c3","Capacita' termica(beta)",200,10,600,400);
c3->SetFillColor(0);
c3->cd();
TGraphErrors *g3 = new TGraphErrors(pmisure,beta_finale,
capacita_termica,sbeta_finale, scapacita_termica);
g3->SetMarkerSize(0.6);
g3->SetMarkerStyle(21);
g3->SetTitle("Capacita' termica(beta)");
g3->GetXaxis()->SetTitle("beta");
g3->GetYaxis()->SetTitle("Capacita' termica ");
g3->Draw("ACP");

```

```

//Suscettivita' magnetica
cout<<endl;
cout<<"Suscettivita' magnetica in funzione di beta"<<endl;

// Grafico 4
TCanvas *c4 = new TCanvas("c4","Suscettivita' magnetica(beta)",200,10,600,400);
c4->SetFillColor(0);
c4->cd();
TGraphErrors *g4 = new TGraphErrors(pmisure,beta_finale,X,sbeta_finale,sX);
g4->SetMarkerSize(0.6);
g4->SetMarkerStyle(21);
g4->SetTitle("X(beta)");
g4->GetXaxis()->SetTitle("beta");
g4->GetYaxis()->SetTitle("X ");
g4->Draw("ACP");

// Fit 4
cout << "\n\n Ipotesi funzione: [0]-pow(abs((-1/0.45 + 1/x)), [1]) \n" <<endl;
TF1 *funz4 = new TF1("funz4","[0]-pow(abs((-1/[2] + 1/x)), [1]) ",0.4,0.54);
funz4->SetParameter(0,0.8);
funz4->SetParameter(1,1.75);
funz4->SetParameter(2,0.46);
funz4->SetLineColor(1);
g4->Fit(funz4,"RM+");
cout<<endl;

float gamma= funz4->GetParameter(1);
float sgamma = funz4->GetParError(1);
double gamma_teorico = 1.75;
cout<<"Esponente critico =( "<<gamma<<" +- "<<sgamma<<" )"<<endl;
cout<<"Esponente critico teorico = "<<gamma_teorico<<endl;

cout << "Chi^2:" << funz4->GetChisquare() << ", number of DoF: "
<< funz4->GetNDF() << " (Probability: " << funz4->GetProb() << ")." << endl;}
```

Riferimenti bibliografici

- [1] István Montvay and Gernot Münster. *Quantum fields on a lattice*. Cambridge University Press, 1997.
- [2] Giuseppe Mussardo et al. *Il modello di Ising: introduzione alla teoria dei campi e delle transizioni di fase*. Bollati Boringhieri, 2007.
- [3] Giorgio Parisi. A short introduction to numerical simulations of lattice gauge theories. *Field Theory, Disorder And Simulations*, 49:407, 1992.
- [4] John Cardy. *Scaling and renormalization in statistical physics*, volume 5. Cambridge university press, 1996.