

Parte VII

Metodi stocastici

In quest'ultima parte del corso vediamo come possiamo usare metodi basati sulla generazione di numeri casuali per risolvere alcune delle equazioni della fisica. Vedremo che non saremo limitati alla trattazione di soli processi stocastici (o casuali).

26 Generazione di numeri casuali

Ci interessa generare dei numeri casuali nell' intervallo $[0, 1]$. Per un numero casuale r si intende che la densità lineare di probabilità di generare tale numero è:

$$p(r) = 1 \quad (281)$$

Anche se in pratica è possibile la generazione di *veri* numeri casuali, ad esempio tirando un dado o misurando il decadimento radioattivo di un isotopo, essa è fuori portata per le applicazioni di fisica computazionale. Quindi dobbiamo ricorrere a metodi *pseudo-casuali* in cui generiamo dei numeri con una probabilità simile a quella ideale. Tali algoritmi generano serie di numeri $\{r_i\}$ nell'intervallo $[0, 1]$. Il numero di $\{r_i\}$ diversi che vengono generati è determinato dall'algoritmo usato e prende il nome di *periodo massimo*. Vediamo ora alcuni algoritmi:

26.1 Generatori LCG

Gli algoritmi (LCG) (linear congruential generators) sono basati sul seguente algoritmo:

1. Inizializzo $x_0 \in \mathbb{N}$ detto seme o *seed*.
2. scelgo due parametri $a, c \in \mathbb{N}$ e $m \in \mathbb{N}$ con $m > a, c$
3. uso l'iterazione $x_{i+1} = (ax_i + c) \bmod m$
4. I numeri finali sono ottenuti come $r_i = \frac{x_i}{m}$

Di solito i parametri a, c, m sono tenuti costanti mentre il seme x_0 viene cambiato tra un *run* del programma ed il successivo in maniera da non generare la stessa sequenza di numeri. Per cambiare x_0 un metodo frequentemente usato è quello di ricavarlo dall'orologio di sistema del computer.

Una scelta di parametri usata frequentemente è quella di PARK-MILLER: $a = 7^5$, $c = 0$, $m = 2^{31} - 1$ che ha quindi un periodo di 2^{31} . Per ottenere periodi maggiori e quindi numeri pseudo-casuali di qualità maggiore è frequentemente usata le seguenti varianti che prende il nome di *shuffling*:

1. Scelgo un parametro $N > 0 \in \mathbb{N}$ e i quattro parametri x_0, a, c, m
2. con l'algoritmo LCG calcolo i primi N numeri $x_i \ i = 1, N$
3. pongo $k = N, l = 1$
4. Calcolo il numero intero $j = 1 + \text{floor}(N * x_k / M)$ nell'intervallo $[1, N]$
5. Pongo $r_l = x_j / m$
6. Pongo $x_j = x_k = (ax_i + c)$
7. Pongo $k = j$ e $l = l + 1$
8. Chiudo il loop tornando al punto 4

26.2 Test statistici sui numeri casuali

Per un insieme di numeri distribuiti secondo la distribuzione di Eq. 281 mi aspetto che il calcolo dei loro momenti sia in accordo con quello per la funzione $p(x)$:

$$\langle x^k \rangle = \int_0^1 x^k p(x) dx = \int_0^1 x^k dx = \frac{1}{k+1} \quad (282)$$

questi li confronto con i valori medi ottenuti dai miei numeri pseudo casuali:

$$\overline{x^k} = \frac{1}{N} \sum_{i=1,N} r_i^k \quad (283)$$

inoltre posso andare ad investigare se i numeri pseudo-casuali sono *scorrelati* ossia se, dato $l > 0$:

$$\langle r_i r_{i+l} \rangle = \langle r_i \rangle \langle r_{i+l} \rangle = \frac{1}{2} \frac{1}{2} = \frac{1}{4} \quad (284)$$

per calcolare la funzione di correlazione per i miei numeri uso:

$$\overline{r_i r_{i+l}} = \frac{1}{N-l} \sum_{i=1,N-l} r_i r_{i+l} \quad (285)$$

Infine un test grafico della correlazione può essere effettuato plottando i punti (r_i, r_{i+1}) su di un grafico cartesiano e andando a vedere se ci sono delle periodicità evidenti.

27 Sampling di funzioni densità di probabilità

Ora vogliamo che i nostri numeri casuali siano distribuiti secondo una funzione densità di probabilità $p(x)$ con $x \in [a, b]$ normalizzata

$$\int_a^b p(x) dx = 1 \quad (286)$$

ossia vogliamo che dati $y_1 < y_2$ se generiamo N numeri casuali $\{x_i\}$ e N_{y_1, y_2} è il numero di numeri $\in [y_1, y_2]$ nel limite di grande N si verifichi:

$$\frac{N_{y_1, y_2}}{N} \rightarrow \int_{y_1}^{y_2} p(x) dx \quad (287)$$

inoltre vogliamo generare i numeri (pseudo-)casuali $\{x_i\}$ a partire dai numeri $\{r_i\}$ uniformemente distribuiti nell'intervallo $[0, 1]$. Per questo compito sono disponibili alcuni metodi:

27.1 Metodo della trasformata inversa

Cominciamo con il calcolare la funzione comulante:

$$P(x) = \int_a^x p(x) dx \quad (288)$$

essa è una funzione monotona (p è positiva) con $P(a) = 0$ $P(b) = 1$ quindi (almeno in principio) invertibile. Supponiamo ora di generare N numeri casuali distribuiti uniformemente tra 0 e 1 e sia N_{r_1, r_2} il numero di numeri casuali tra $0 \leq r_1 < r_2 \leq 1$. Consideriamo ora i numeri casuali ottenuti dalla trasformazione $x_i = P^{-1}(r_i)$ corrispondenti a quelli $\in [r_1, r_2]$ essi saranno tutti compresi nell'intervallo $[x_1 = P^{-1}(r_1), x_2 = P^{-1}(r_2)]$ dove $r_1 = P(x_1)$ e $r_2 = P(x_2)$ da cui:

$$N_{x_1, x_2} = N_{r_1, r_2} = \frac{r_2 - r_1}{1} N = [P(x_2) - P(x_1)] N = \left[\int_a^{x_2} p(x) dx - \int_a^{x_1} p(x) dx \right] N = N \int_{x_1}^{x_2} p(x) dx \quad (289)$$

quindi i numeri $x_i = P^{-1}(r_i)$ sono distribuiti secondo la funzione densità di probabilità $p(x)$. Il metodo della trasformata inversa può essere applicato però solo nel caso in cui la funzione P è invertibile.

27.2 Sampling della funzione Gaussiana

Vediamo ora come trovare numeri casuali distribuiti secondo la densità di probabilità Gaussiana: $p(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$. In questo caso l'applicazione diretta del metodo della precedente sezione non è possibile visto che non riusciamo ad avere una formula analitica per la funzione comulante. Riformulando il problema si arriva alla seguente ricetta: si prendono due numeri casuali r_1 e r_2 . Poi viene ottenuto un numero casuale z distribuito secondo la densità di probabilità Gaussiana usando la seguente formula:

$$z = \cos(2\pi r_2) \sqrt{-2 \ln(r_1)} \quad (290)$$

27.3 Metodo accetta/nega

Abbiamo già visto questo metodo nella Sezione 15. Consideriamo una densità di probabilità $p(x)$ con $x \in [a, b]$ indichiamo con $p_{max} = \max(p)$. L'algoritmo è il seguente:

1. Generiamo un numero casuale r_1 nell'intervallo $[0, 1]$
2. Consideriamo il corrispondente numero casuale $x = (b - a)r_1 + a$ distribuito uniformemente in $[a, b]$
3. Generiamo un numero casuale r_2 nell'intervallo $[0, 1]$
4. Solo se $r_2 < p(x) / p_{max}$, aggiungiamo il numero casuale x ai numeri generati dall'algoritmo
5. Torna al numero 1

Si vede facilmente che i numeri generati dall'algoritmo soddisfano la distribuzione richiesta

28 Integrazione con numeri casuali o Monte Carlo

Cominciamo con considerare l'integrazione di una funzione generica $f(x)$ sull'intervallo $[a, b]$:

$$\int_a^b f(x) dx = \bar{f}(b-a) \quad (291)$$

dove il valore medio \bar{f} della funzione è definito come:

$$\bar{f} = \frac{\int_a^b f(x) dx}{b-a} \quad (292)$$

Però noi possiamo calcolare \bar{f} con una certa precisione generando N numeri casuali x_i distribuiti uniformemente su $[a, b]$ e mediando gli $f_i = f(x_i)$, dalle formule di teoria della probabilità si trova:

$$\bar{f} = \frac{1}{N} \sum_{i=1,N} f_i \pm \sqrt{\frac{\text{var}(f)}{N}} \quad (293)$$

dove la varianza è definita come:

$$\text{var}(f) = \overline{(f - \bar{f})^2} = \bar{f}^2 - 2\bar{f}\bar{f} + \bar{f}^2 = \bar{f}^2 - \bar{f}^2 \quad (294)$$

con

$$\bar{f}^2 = \frac{1}{N} \sum_{i=1,N} f_i^2 \quad (295)$$

la varianza $\text{var}(f)$ può quindi essere calcolata esattamente come:

$$\text{var}(f) = \frac{\int_a^b (f(x))^2 dx}{b-a} - \left(\frac{\int_a^b f(x) dx}{b-a} \right)^2 \quad (296)$$

quindi l'accuratezza di integrazione è di ordine $O\left(\frac{1}{\sqrt{N}}\right)$. Ci ricordiamo che il metodo dei rettangoli naif ha un'accuratezza $O\left(\frac{1}{N}\right)$ mentre quello dei rettangoli e quello dei trapezi hanno un'accuratezza $O\left(\frac{1}{N^2}\right)$. Quindi per integrare funzioni definite su \mathbb{R} usare numeri casuali non è conveniente.

Il discorso cambia invece se consideriamo funzioni definite su \mathbb{R}^M . Supponiamo quindi di usare il metodo dei rettangoli (o trapezi) e di valutare la funzione in N punti, distribuiti in maniera regolare, cosicché lungo ogni dimensione di \mathbb{R}^M io ho $N^{\frac{1}{M}}$ punti. L'accuratezza con cui integriamo (coi rettangoli-trapezi) è determinata dal numero di punti lungo le singole dimensioni quindi risulta essere $O\left(\frac{1}{N^{2/M}}\right)$.

Usando invece il metodo dei numeri casuali, l'accuratezza è sempre di ordine $O\left(\frac{1}{\sqrt{N}}\right)$. Quindi per $M > 4$ il metodo dei numeri casuali (o Monte Carlo) risulta essere più accurato a parità di costo computazionale.

28.1 Integrazione di funzione composite

Consideriamo ora l'integrazione (per semplicità di sola notazione consideriamo funzioni su \mathbb{R}) di una funzione del tipo $f(x)p(x)$ su $[a, b]$ dove $p(x)$ è una densità di probabilità definita sullo stesso intervallo $[a, b]$. E' facile vedere che posso calcolare:

$$\int_a^b f(x)p(x)dx = (b-a) \left(\frac{1}{N} \sum_{i=1,N} f_i \pm \sqrt{\frac{\text{var}(f)}{N}} \right) \quad (297)$$

dove gli N numeri casuali x_i sono distribuiti secondo la distribuzione $p(x)$ e $f_i = f(x_i)$

Infatti consideriamo un numero N grande di numeri casuali distribuiti secondo $p(x)$. Il numero dN di numeri casuali in dx è pari a:

$$dN = Np(x) \frac{dx}{b-a} \quad (298)$$

e il contributo al termine di destra in Eq. 297 è:

$$(b-a) \frac{dN}{N} f(x) = (b-a) f(x) \frac{Np(x) \frac{dx}{b-a}}{N} = f(x)p(x)dx \quad (299)$$

quindi pari al termine che compare nell'integrale nel membro di sinistra di Eq. 297.

29 L'algoritmo di Metropolis

Spesso è richiesto di calcolare integrali del tipo:

$$\int dx O(x)p(x) \quad (300)$$

dove la densità di probabilità $p(x)$ è definita come:

$$\begin{cases} p(x) &= \frac{q(x)}{Z} \\ Z &= \int dx q(x) \end{cases} \quad (301)$$

dove con x indichiamo un elemento di uno spazio generico, O è una genrica funzione corrispondente ad un osservabile e $q(x)$ è una funzione densità di probabilità NON normalizzata. La costante di normalizzazione è Z cosicchè $p(x)$ è una buona funzione densità di probabilità.

Ad esempio per un sistema di particelle classiche, come quelli che abbiamo visto in Sezione 9, il valore medio di un osservabile O in condizioni di equilibrio termico alla temperatura T viene calcolato come:

$$\langle O \rangle_T = \int dq_1 \dots \int dq_N \int dp_1 \dots \int dp_N O(q_1, \dots, q_N, p_1, \dots, p_N) \frac{e^{-\frac{1}{k_B T} E(q_1, \dots, q_N, p_1, \dots, p_N)}}{Z} \quad (302)$$

con la funzione di partizione Z definita come:

$$Z = \int dq_1 \dots \int dq_N \int dp_1 \dots \int dp_N e^{-\frac{1}{k_B T} E(q_1, \dots, q_N, p_1, \dots, p_N)} \quad (303)$$

Quindi per calcolare $p(x)$ abbiamo bisogno di Z ma per calcolare Z dobbiamo calcolare un integrale della stessa complessità. Quindi integrare con il metodo di Monte Carlo le formule di Eqs. 300,302 non è possibile a meno di generare i punti casuali $\{x_i\}$ secondo la distribuzione $p(x)$ e poi usare il metodo di Sezione 28.1. Siccome non conosciamo $p_{max} = \max(p(x))$ il metodo accetta-nega può essere utilizzato solo usando un valore massimo p'_{max} sufficientemente grande da essere sicuri che $p'_{max} > p_{max}$. Questo può risultare altamente inefficiente.

Metropolis invece introduce un algoritmo efficiente per generare i punti casuali $\{x_i\}$ secondo la distribuzione $p(x)$ usando solo la funzione $q(x)$. L'algoritmo è il seguente:

1. Generiamo casualmente il primo punto x_1 , poniamo $i = 1$
2. Generiamo casualmente il punto x_{trial}
3. Se $p(x_{trial}) > p(x_i)$ ossia se $q(x_{trial}) > q(x_i)$, aggiungiamo il punto ai numeri generati: $x_{i+1} = x_{trial}$, $i = i + 1$, TORNIAMO A 2
4. Altrimenti generiamo casualmente un numero $r \in [0, 1]$

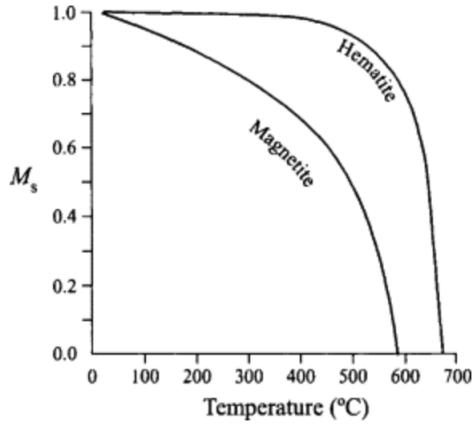


Figura 13: Andamento della magnetizzazione relativa rispetto alla temperatura.

5. Se $\frac{p(x_{trial})}{p(x_i)} = \frac{q(x_{trial})}{q(x_i)} > r$ aggiungiamo il punto ai numeri generati: $x_{i+1} = x_{trial}$, $i = i + 1$, TORNIAMO A 2
6. Altrimenti aggiungiamo x_i ai numeri generati: $x_{i+1} = x_i$, $i = i + 1$, TORNIAMO A 2

L'algoritmo di Metropolis definisce una funzione probabilità $c(x)$ che un punto venga generato dato un certo numero di passi. Inoltre soddisfa la condizione di bilancio dettagliato ossia il numero medio di x_j generato subito dopo (o a partire da) x_i è pari al numero di punti x_i generati a partire da x_j . Questa proprietà è indicata nel lavoro originale di Metropolis e collaboratori come dovuta alla semplice condizione di equilibrio in un sistema ergodico (e invariante per inversione temporale) e si può dimostrare in maniera rigorosa tramite la teoria delle catene di Markov. Ora dimostriamo che il bilancio dettagliato implica il fatto che $c(x)$ è proporzionale a $p(x)$ come auspicato. Indichiamo con $p(x_j|x_i)$ la probabilità di generare x_j da x_i :

$$p(x_j|x_i) = \begin{cases} 1 & p(x_j) > p(x_i) \\ \frac{p(x_j)}{p(x_i)} = \frac{q(x_j)}{q(x_i)} & p(x_j) \leq p(x_i) \end{cases} \quad (304)$$

Allora il bilancio dettagliato viene scritto come:

$$p(x_j|x_i)c(x_i) = p(x_i|x_j)c(x_j) \quad (305)$$

se siamo nel caso $p(x_j) > p(x_i)$, il bilancio dettagliato diviene:

$$c(x_i) = \frac{p(x_i)}{p(x_j)}c(x_j) \quad (306)$$

mentre se $p(x_j) \leq p(x_i)$, il bilancio dettagliato diviene:

$$\frac{p(x_j)}{p(x_i)}c(x_i) = c(x_j) \quad (307)$$

In entrambi i casi si ottiene:

$$\frac{c(x_i)}{c(x_j)} = \frac{p(x_i)}{p(x_j)} \quad (308)$$

quindi la funzione $c(x)$ che posso scrivere come $c(x) = \frac{p(x)}{p(x_1)}c(x_1)$ è come voluto proporzionale a $p(x)$.

30 Modello di Ising per il magnetismo

Il magnetismo nei materiali è dovuto ai gradi di libertà elettronici e in particolare alla loro componente di spin. I materiali chiamati ferromagnetici esibiscono magnetizzazione spontanea (ossia presente anche in assenza di campo magnetico esterno) fino ad una certa temperatura critica chiamata temperatura di Curie. Il tipico andamento temperatura magnetizzazione è raffigurato in Figura 13. Alla temperatura di Curie i materiali esibiscono una transizione di fase di secondo ordine, ossia la magnetizzazione è continua ma con derivata discontinua. La magnetizzazione è dovuta alla presenza nel materiale di atomi di metalli di transizione (es. Fe) che contengono una shell di orbitali d incompleta con un numero di elettroni dispari. Siccome tali orbitali sono localizzati è sensato introdurre un semplice modello per descrivere l'energia (o

meglio l'operatore di Hamilton) del sistema. Gli unici gradi di libertà sono l'orientazione degli spin che indichiamo che con σ_l dove l indica il sito atomico di appartenenza. La variabile σ_l può assumere solo i valori ± 1 . La funzione Hamiltoniana modello che descrive il sistema è data da:

$$H = -\frac{1}{2} \sum_{l \neq l'} J_{ll'} \sigma_l \sigma_{l'} - h \sum_l \sigma_l \quad (309)$$

dove i coefficienti $J_{ll'}$ rendono conto del termine di scambio tra il sito l e quello l' e abbiamo $J_{ll'} = J_{l'l}$. L'ultimo termine nel membro di destra tiene conto dell'accoppiamento con un eventuale campo magnetico esterno e abbiamo $h = -\mu_B g B / 2$ con μ_B momento magnetico dell'elettrone e dove il fattore g , detto di Landé rende conto del fatto della perturbazione introdotta dagli altri elettroni del cristallo.

Se $O(\{\sigma\})$ è un generico osservabile, il suo valore di aspettazione alla temperatura T è dato da:

$$\langle O \rangle_T = \frac{\sum_{\{\sigma\}} O \{\sigma\} e^{-\frac{1}{k_B T} H(\{\sigma\})}}{\sum_{\{\sigma\}} e^{-\frac{1}{k_B T} H(\{\sigma\})}} \quad (310)$$

dove con $\{\sigma\} = (\sigma_1, \dots, \sigma_{N_{siti}})$ indichiamo una configurazione di spin per tutti gli N atomi o siti del nostro sistema. Gli osservabili che ci interesseranno maggiormente sono:

l'energia per unità di sito:

$$E(\{\sigma\}) = \frac{H(\{\sigma\})}{N_{siti}} \quad (311)$$

e la magnetizzazione per sito:

$$M(\{\sigma\}) = \frac{1}{N} \sum_l \sigma_l \quad (312)$$

Inoltre si può mostare che i termini di scambio che contribuiscono maggiormente sono quelli le coppie di siti più vicini (detti *primi-vicini*). Allora possiamo ulteriormente semplificare il nostro modello, che diviene il noto modello di Ising:

$$H = -\frac{1}{2} \widetilde{\sum}_{l \neq l'} J \sigma_l \sigma_{l'} - h \sum_l \sigma_l \quad (313)$$

dove con $\widetilde{\sum}$ indichiamo che la somma è limitata ai primi vicini. Si vede subito che se $J > 0$ l'ordine ferromagnetico è preferito ossia con gli spin con medesimo orientazione mentre se $J < 0$ l'ordine antiferromagnetico è preferito ossia con gli spin con orientazioni opposte.

Il modello di Ising ammette soluzioni analitiche nei casi 1D e 2D per il limite di numero di siti infiniti. Nel caso 1D non è osservata alcuna transizione di fase mentre il modello 2D mostra una transizione di fase analoga a quella misurata.

Nel caso 3D invece non è stato possibile ancora trovare soluzioni analitiche. Pertanto il modello di Ising (e modelli simili) vegono studiati con metodi Monte Carlo ed in particolare con l'algoritmo di Metropolis.

Notiamo che per $h = 0$ il sistema è simmetrico (ossia l'energia è invariante) per inversione di tutti gli spin. Quindi ci aspettiamo che per ogni temperatura T : $\langle M \rangle_{T,h=0} = 0$. In questa maniera però non riusciamo studiare la magnetizzazione permanente. Il problema si risolve andando a definire e poi a studiare:

$$\langle M \rangle_{T,h=0^+} = \lim_{h \rightarrow 0^+} \langle M \rangle_{T,h} \quad (314)$$

Nel applicare il metodo di Metropolis al modello di Ising di solito ci si limita la generazione della configurazione di prova (σ_{trial}) all'inversione di uno solo spin scelto a caso rispetto alla configurazione precedente. Questo da un lato rende l'algoritmo molto più efficiente dall'altro *tende a bloccare* il sistema in un sottospazio dello spazio delle configurazioni con maggior parte degli spin aventi medesima orientazione tanto che per temperature basse e per numero di configurazioni generate relativamente limitato si trova $\overline{M}_{h=0} \neq 0$ e $|\overline{M}_{h=0}| \cong \langle M \rangle_{T,h=0^+}$.

Nelle Figure 14,15,16 riportiamo l'energia per sito e la magnetizzazione per sito nel caso del modello 2D con condizioni al contorno periodiche e una griglia di 50×50 spin. Notiamo che lo stato fondamentale ad energia più bassa corrisponde a tutti gli spin aventi la medesima orientazione ed è quindi doppiamente degenero. Per tale stato l'energia per sito è $-2J$ e la magnetizzazione per sito è ± 1 .

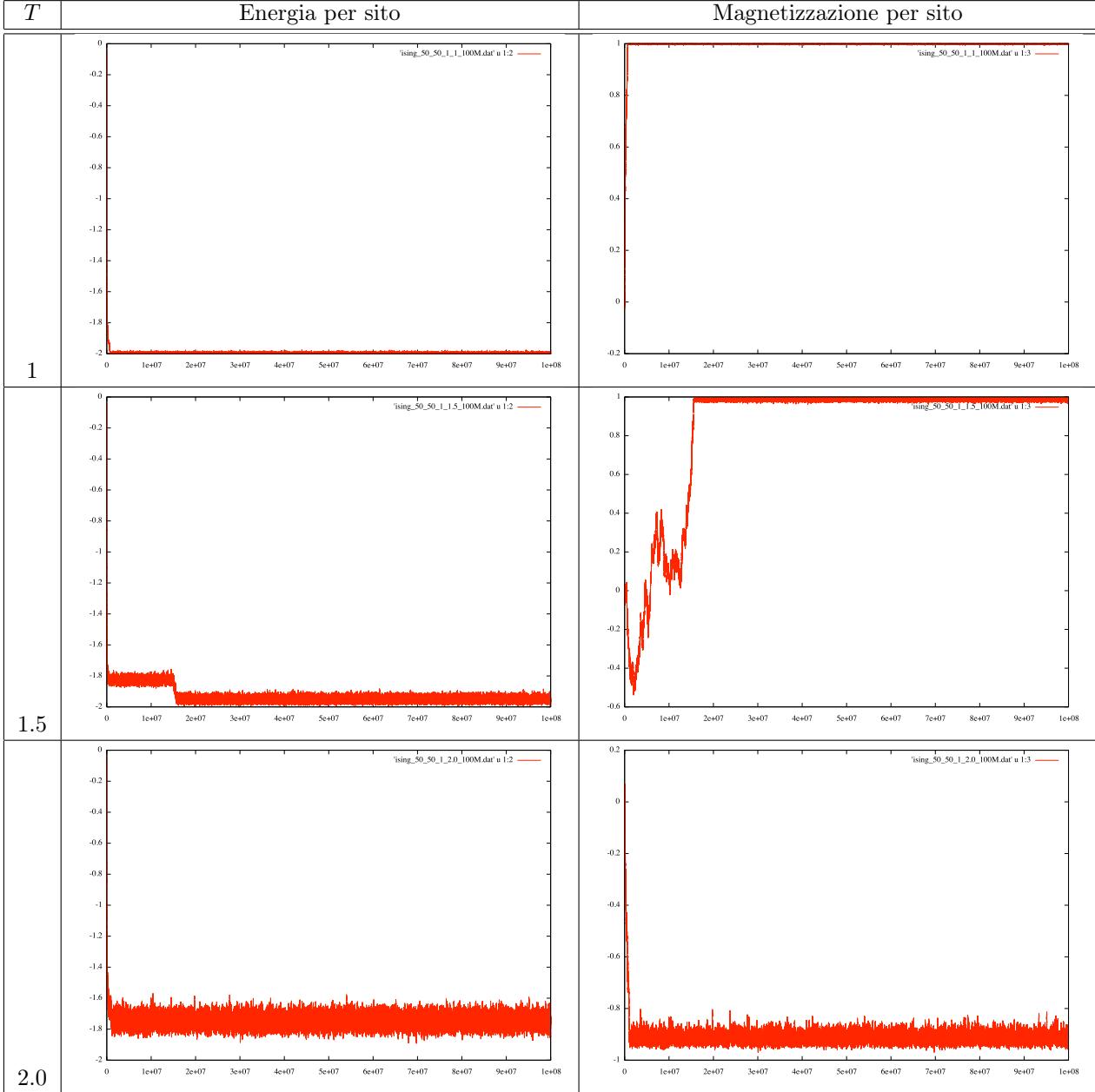


Figura 14: Simulazione Monte Carlo con l'algoritmo di Metropolis per il modello di Ising 2D con dimensioni 50x50 e condizioni al contorno periodiche. $J = 1$ e $h = 0$. Ogni simulazione consta di $100 \cdot 10^6$ di passi.

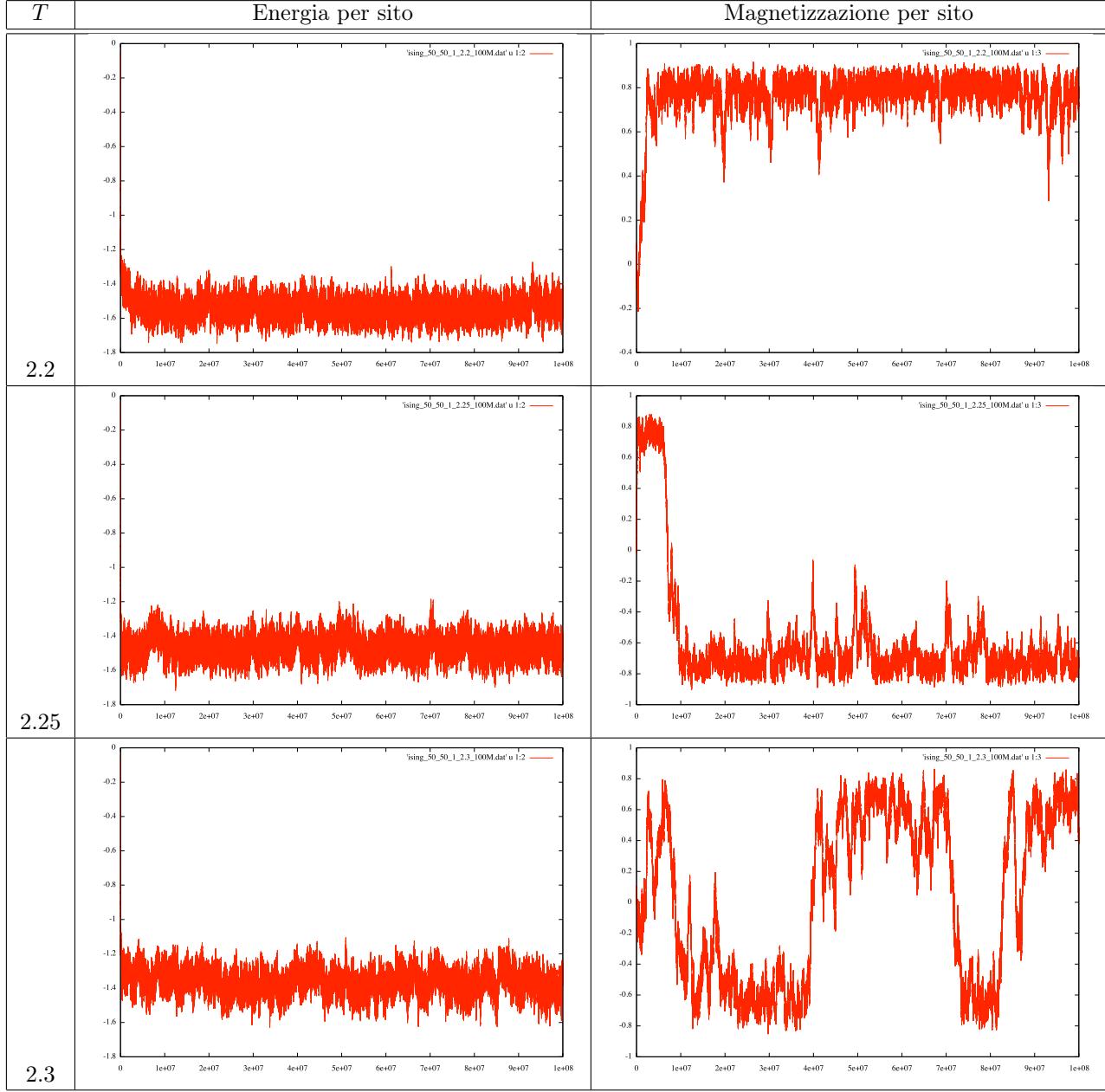


Figura 15: Simulazione Monte Carlo con l'algoritmo di Metropolis per il modello di Ising 2D con dimensioni 50x50 e condizioni al contorno periodiche. $J = 1$ e $h = 0$. Ogni simulazione consta di $100 \cdot 10^6$ di passi.

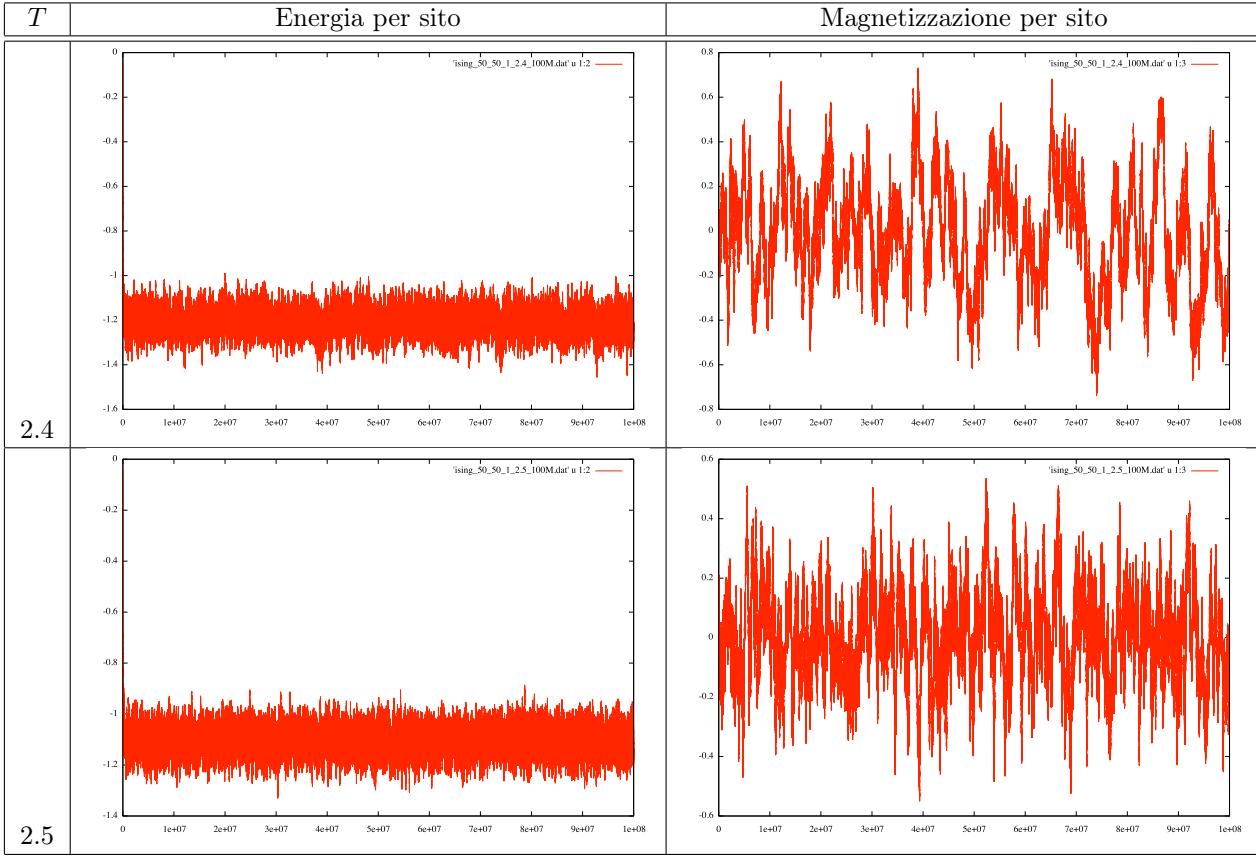


Figura 16: Simulazione Monte Carlo con l'algoritmo di Metropolis per il modello di Ising 2D con dimensioni 50x50 e condizioni al contorno periodiche. $J = 1$ e $h = 0$. Ogni simulazione consta di $100 \cdot 10^6$ di passi.

Parte VIII Algoritmi di ottimizzazione

Consideriamo (brevemente) il problema di trovare il minimo (o i minimi) di una generica funzione $F(\lambda_1, \dots, \lambda_N)$ rispetto ai parametri $\lambda_1, \dots, \lambda_N$.

31 Metodi basati sul gradiente

Supponiamo sia possibile calcolare:

$$\nabla F(\lambda_1, \dots, \lambda_N) = \begin{pmatrix} \frac{\partial}{\partial \lambda_1} F(\lambda_1, \dots, \lambda_N) \\ \frac{\partial}{\partial \lambda_2} F(\lambda_1, \dots, \lambda_N) \\ \vdots \\ \frac{\partial}{\partial \lambda_N} F(\lambda_1, \dots, \lambda_N) \end{pmatrix} \quad (315)$$

allora possiamo trovare il minimo spostandoci lungo la direzione opposta al gradiente. Secondo il seguente algoritmo detto *Steepest descend*.

1. Dati i valori iniziali $\lambda_1^i, \dots, \lambda_N^i$, calcolo $\nabla F(\lambda_1^i, \dots, \lambda_N^i)$
2. Calcolo il parametro a tale che $F\left(\lambda_1^i - a \frac{\partial}{\partial \lambda_1} F(\lambda_1^i, \dots, \lambda_N^i), \dots, \lambda_N^i - a \frac{\partial}{\partial \lambda_N} F(\lambda_1^i, \dots, \lambda_N^i)\right)$ sia minimo
3. Pongo per il valore di a trovato $\lambda_1^{i+1} = \lambda_1^i - a \frac{\partial}{\partial \lambda_1} F(\lambda_1^i, \dots, \lambda_N^i), \dots, \lambda_N^{i+1} = \lambda_N^i - a \frac{\partial}{\partial \lambda_N} F(\lambda_1^i, \dots, \lambda_N^i)$ e ritorno al punto 2

Talvolte è conveniente usare un valore di a prefissato senza ricavarlo ad ogni iterazione.

Una metoda affine basato sui gradienti è quello del *Gradiente Coniugato* che permette di scegliere la direzione lineare di minizzazione in maniera più efficiente.