

Università degli studi di Trieste
Corso di studi magistrale in Fisica

Percolazione e L'algoritmo di Newmann-Ziff

Luigi Brancati
Curriculum di Fisica Teorica
matricola: SM2300304
email: `luigi.brancati93@gmail.com`

a.a. 2016/2017

1 Introduzione

La percolazione è un processo che interessa molti sistemi fisici, in quanto permette di studiare le caratteristiche di fenomeni critici come il passaggio di fase di un sistema e come le quantità che lo descrivono variano in prossimità del suo punto critico. Ciò che si andrà a studiare nel seguito è in particolare la *site percolation*, che consiste in un sistema composto da una griglia a celle quadrate o *siti*, ognuna delle quali può trovarsi in uno di due stati, occupata con probabilità p o libera con probabilità $1 - p$. Il sistema può sembrare semplice, ma il suo studio ha portato benefici e ha avuto applicazioni in molti campi che esulano dalla fisica e in particolare dallo studio di sistemi termodinamici vicini al punto critico: questo modello è utilizzato ad esempio per lo studio della diffusione degli incendi in un'area boschiva o delle epidemie in differenti condizioni abitative, oppure anche per testare la solidità della rete Internet e di altre reti locali o per studiare la percolazione (cioè la diffusione) di un fluido in un materiale poroso, e proprio da quest'ultimo fenomeno deriva il nome del processo.

Nel seguito, si presenta una trattazione della *site percolation* e uno studio delle quantità più importanti. Nella sezione 2 si introducono queste quantità e se ne discute il comportamento, con particolare attenzione al loro comportamento intorno al punto critico; nella sezione 3 è invece presentato l'algoritmo di Newmann-Ziff, descritto nei singoli passaggi e nell'implementazione del codice, e nella sezione 4 sono presentati i dati misurati utilizzando quest'algoritmo. Si conclude con alcune considerazioni finali nella sezione 5.

2 Site percolation e quantità di interesse

Come già accennato, la *site percolation* consiste nello studiare una griglia quadrata (la griglia può in realtà avere differenti topologie, ma noi ci limiteremo al caso della griglia quadrata) di dimensione lineare L , composta di celle ognuna delle quali può essere occupata o libera, con probabilità di occupazione che indicheremo con p . Un algoritmo semplice, che permette di avere un'idea di come si comporta un sistema di questo tipo, consiste nello stabilire per p un valore fisso e generare, per ognuno dei siti, un numero randomico $0 \leq r \leq 1$ che determinerà se il sito è occupato ($r \leq p$) oppure no ($r > p$). Una volta controllati tutti i siti, si otterrà una certa configurazione della griglia, con alcuni siti occupati ed altri liberi; i siti occupati, se p è sufficientemente grande, tenderanno ad agglomerarsi in *clusters* di siti vicini (si considerano siti vicini soltanto due celle con un lato in comune, la diagonale non è considerata). Ad una certa p di osserverà uno *spanning cluster*, cioè un cluster che attraversa tutta la griglia: in questo caso si dice che il sistema ha *percolato*. La comparsa di uno *spanning cluster* determina un cambiamento significativo nello stato del sistema, se consideriamo sistemi infiniti ($L \rightarrow \infty$) determina un passaggio di fase del sistema in questione, ad esempio da uno stato isolante allo stato conduttore, e nel limite infinito è possibile identificare in maniera univoca un valore p_c , detto probabilità *critica*, tale che la probabilità di avere uno *spanning cluster* sia nulla per $p < p_c$, e sia invece maggiore di 0 per $p = p_c$, mentre per $p > p_c$ si riscontrerà sempre la presenza di uno *spanning cluster*.

La probabilità critica, o probabilità di transizione, è quindi la prima quantità che permette di caratterizzare il fenomeno della percolazione. Nel nostro caso,

potendo simulare solo sistemi finiti, la probabilità critica dipenderà da L , $p_c(L)$, e i criteri che permettono di definire questa quantità così come il concetto stesso di spanning cluster soffrono di una certa arbitrarietà: lo spanning cluster può essere infatti definito come un cluster che attraversa la griglia da destra a sinistra, o da sopra a sotto, o in entrambe le direzioni; la p_c invece può essere definita come la media delle probabilità alle quali, per varie configurazioni, si osserva apparire lo spanning cluster, oppure come la probabilità alla quale lo spanning cluster appare esattamente in metà delle configurazioni. La probabilità di ottenere uno spanning cluster in una configurazione ad una certa p verrà indicata con $P_{span}(p)$: si noti che mentre per un sistema infinito questa quantità è sempre 0 per $p < p_c$ e cresce in maniera monotona per $p \geq p_c$, nel caso finito si comporta diversamente, come molte altre quantità che studieremo nel seguito, ed è sempre maggiore di 0 anche per $p < p_c$; in particolare, per una griglia di dimensione L e probabilità di occupazione p si avrà che $0 \leq P_{span} \leq p^L$, e questa quantità andrà a zero, come atteso, per $L \rightarrow \infty$, essendo $p \leq 1$. Nella sezione 4 misureremo queste quantità, e stabiliremo in maniera chiara i criteri scelti per la loro misura.

Un'altra quantità legata allo spanning cluster e molto importante ai nostri fini è la probabilità che un sito occupato appartenga allo spanning cluster, definita semplicemente come segue:

$$P_\infty(p) = \frac{\text{siti nello spanning cluster}}{\text{siti totali occupati}}$$

per un sistema infinito ci si aspetta che

$$P_\infty = \begin{cases} 0 & \text{se } p < p_c \\ 1 & \text{se } p = 1 \end{cases} \quad \text{per } L \rightarrow \infty$$

mentre per $p_c < p < 1$ la $P_\infty(p)$ sarà una funzione monotona crescente. Si noti che questa quantità è un *parametro d'ordine* per il passaggio di fase, ci permette cioè di distinguere uno stato che ha superato il punto critico ($p > p_c$) da uno stato che non l'ha invece superato ($p < p_c$). Un parametro d'ordine è ad esempio la magnetizzazione totale per un sistema di Ising in 2D.

Una quantità più generale, che riguarda tutti i cluster e permette di avere un'idea più chiara sulla configurazione in cui si trova il sistema ad una certa p è la distribuzione delle dimensioni dei cluster. Per dimensione di un cluster tipicamente si intende il numero dei siti che lo compongono, ma è possibile associare ad ogni cluster anche una dimensione spaziale, mediando su tutte le posizioni spaziali dei siti che compongono il cluster e calcolando il "raggio" dello stesso. Noi intenderemo nel seguito la versione più semplice, e la dimensione di un cluster sarà determinata dal numero dei siti che lo compongono. Definiamo quindi la distribuzione delle dimensioni dei cluster

$$n_s(p) = \frac{\text{numero di cluster di dimensione } s}{\text{numero totale di siti}}$$

dove non viene però considerato lo spanning cluster per $p \geq p_c$, ma soltanto i cluster di dimensione finita, in quanto lo spanning cluster ha dimensione infinita per sistemi infiniti. L'andamento di $n_s(p)$ sarà studiato nel seguito ed è più complicato delle quantità definite fin'ora, in quanto dipende da due parametri s e p ed è in generale più difficile da prevedere al variare di p . In via generale,

per sistemi infiniti la distribuzione sarà una funzione fortemente decrescente di s , dimensione dei cluster, ad una certa p fissata, e per $p \neq p_c$ decrescerà più rapidamente rispetto al caso $p = p_c$. Ciò è dovuto al fatto che per $p = p_c$ sono presenti cluster di tutte le dimensioni, mentre per $p > p_c$ lo spanning cluster, certamente presente e non considerato in n_s , tenderà ad inglobare alcuni dei cluster maggiori, riducendo la n_s corrispondente e facendo diminuire n_s più velocemente per s crescente; per $p < p_c$ si avranno pochi cluster di grandi dimensioni.

Una seconda quantità legata a n_s e alla dimensione dei cluster è la dimensione media dei cluster nella griglia, definita come

$$S(p) = \frac{\sum_s s^2 n_s}{\sum_s s n_s}$$

dove la quantità a numeratore è detta anche momento quadro della griglia, e si fa notare che $N s n_s$ è il numero di siti occupati appartenenti a cluster di dimensione s . Si noti che, essendo definita tramite n_s , nella media delle dimensioni non è considerato lo spanning cluster. Il suo andamento è descritto nella sottosezione successiva.

Infine, come già accennato è possibile associare ai cluster una dimensione spaziale. Definiamo il raggio associato ad un cluster di dimensione s

$$R_s^2 = \frac{1}{s} \sum_{i=1}^s (\mathbf{r}_i - \bar{\mathbf{r}})^2$$

con \mathbf{r}_i posizione dell' i -esimo sito e $\bar{\mathbf{r}}$ centro di massa del cluster

$$\bar{\mathbf{r}} = \frac{1}{s} \sum_{i=1}^s \mathbf{r}_i$$

quindi R_s non è altro che la distanza media dei siti dal centro di massa del cluster. La lunghezza di connessione $\xi(p)$, cioè la lunghezza associata ai cluster di un sistema è definita semplicemente come la media dei raggi di tutti i cluster del sistema, ad eccezione dello spanning cluster. Se chiamiamo $w_s = s n_s$ la probabilità che un sito appartenga a un cluster di dimensione s e $\langle R_s^2 \rangle$ la media del raggio quadro su tutti i cluster di dimensione s , e ricordando che ogni sito in un cluster è affiancato da altri $s - 1$ siti dello stesso cluster, allora possiamo scrivere

$$\xi^2(p) = \frac{\sum_s (s - 1) w_s \langle R_s^2 \rangle}{\sum_s (s - 1) w_s}$$

per il quadrato della lunghezza di connessione dei cluster nel sistema, dove è ovviamente escluso lo spanning cluster.

2.1 Esponenti critici e Finite size scaling

Abbiamo quindi visto che un sistema infinito presenta un punto critico corrispondente alla probabilità p_c , intorno la quale ci si aspetta che le proprietà del sistema cambino in maniera significativa. In particolare, si osserva che per un qualunque sistema si ha che $\xi(p = p_c) \sim L$, e ci si aspetta quindi che la ξ diverga per $L \rightarrow \infty$ in un intorno di p_c . Un comportamento divergente è tipico

di un passaggio di fase, e al fine di studiare questo fenomeno è quindi naturale concentrare lo studio del sistema in una regione $|p - p_c| \ll 1$, detta *regione critica*.

Per descrivere l'andamento, nella regione critica, delle quantità di interesse fin'ora discusse, è necessario introdurre gli *esponenti critici*. Si suppone, in sostanza, che le quantità nell'intervallo critico abbiano un andamento che può essere descritto da una semplice potenza della differenza $(p - p_c)$, la cui esatta forma può essere dedotta dall'andamento della quantità corrispondente nell'intorno di p_c . Ad esempio, per quanto riguarda P_∞ abbiamo già discusso di come essa sia 0 per $p < p_c$ e sia una funzione crescente di p per $p \geq p_c$, e per questo potrà essere scritta come segue

$$P_\infty(p) \sim \begin{cases} 0 & \text{per } p < p_c \\ (p - p_c)^\beta & \text{per } p \geq p_c \end{cases}, \text{ in } |p - p_c| \ll 1$$

dove β è appunto l'esponente critico legato a P_∞ . Per n_s ci si aspetta un andamento del tipo

$$n_s(p = p_c) \sim s^{-\tau}$$

decescente per s e calcolato per $p = p_c$.

Per quanto riguarda invece $S(p)$, il suo comportamento all'infinito può essere estrapolato da quello finito. Per $p < p_c$ è una funzione crescente, in quanto per p crescente aumenterà il numero di cluster di dimensioni sempre maggiori, fino a raggiungere un massimo a $p = p_c$ e poi decrescere velocemente per $p > p_c$, quando lo spanning cluster tenderà ad inglobare molti dei cluster più grandi escludendoli dal conteggio della media. In particolare si è osservato in molte simulazioni che

$$S_{max} = S(p = p_c) \propto N$$

con N numero totale di siti, e perciò nel limite infinito $L \rightarrow \infty, N \rightarrow \infty$ la $S(p)$ crescerà avvicinandosi a p_c fino a divergere nel punto p_c , e per $p > p_c$ avrà invece un andamento decrescente. Per questi motivi, si suppone che il suo andamento nell'intervallo critico sia del tipo

$$S(p) \sim |p - p_c|^{-\gamma}$$

coiè divergente per $p \rightarrow p_c$. Infine, per ξ si ha un andamento crescente per $p < p_c$ e decrescente per $p > p_c$ molto simile a quello di $S(p)$, e abbiamo inoltre osservato che diverge nel punto critico, perciò la forma attesa sarà anche in questo caso

$$\xi(p) \sim (p - p_c)^{-\nu}$$

appunto divergente per $p = p_c$, con ν esponenziale critico associato a ξ . Nella tabella sottostante sono riportati i valori esatti degli esponenti critici trattati in questa sottosezione

Quantità	Esponente	Valore esponente
$P_\infty \sim (p - p_c)^\beta$	β	5/36
$S(p) \sim p - p_c ^{-\gamma}$	γ	43/18
$\xi(p) \sim p - p_c ^{-\nu}$	ν	4/3
$n_s(p = p_c) \sim s^{-\tau}$	τ	187/91

Come già accennato all'inizio della sottosezione, si ha che la lunghezza $\xi(p = p_c) \sim L$, mentre assume valori $\xi \ll L$ per $p < p_c$ e $p > p_c$ fuori dall'intervallo critico. Il fatto che la lunghezza ξ sia paragonabile alla dimensione lineare del sistema indica che nella regione critica gli effetti dovuti alla limitatezza del sistema si fanno sentire maggiormente e provocano le maggiori deviazioni nel comportamento del sistema finito da quello atteso per un sistema infinito, mentre per p fuori dall'intervallo critico i due casi non sono distinguibili. Questi effetti possono però essere efficacemente usati per fare ulteriori stime sul comportamento delle quantità legate al sistema. Si osservi anzitutto che $\xi \sim L$ implica che la regione critica si comporti come

$$|p - p_c| \sim L^{-\frac{1}{\nu}}$$

cioè l'ampiezza della regione critica si ridurrà all'aumentare di L . Da questa proprietà è possibile derivare il comportamento delle altre quantità in relazione ad L

$$P_\infty(p = p_c) \sim L^{-\beta/\nu}$$

$$S(p = p_c) \sim L^{\gamma/\nu}$$

e quindi, misurando l'andamento delle quantità al variare di L , sarà possibile stimare gli esponenti critici, sfruttando quindi gli effetti dovuti alla finitezza della griglia. La strategia che consiste nel misurare le osservabili del sistema studiando la loro dipendenza da L per stimare gli esponenti critici è detta *finite size scaling*.

3 Algoritmo di Newmann-Ziff

L'algoritmo utilizzato per studiare un tale sistema è l'algoritmo di Newmann-Ziff. Quest'algoritmo permette di calcolare tutte le quantità di interesse discusse nella sezione precedente a vari valori di n , numero di siti occupati: l'idea alla base è che partendo da uno stato con n siti occupati, è possibile ottenere una configurazione con $n+1$ siti occupati semplicemente aggiungendo in maniera casuale un sito allo stato precedente, laddove invece si sarebbe generata una nuova configurazione; si parte quindi da uno stato con $n = 0$ e si occupano i siti della griglia fino a riempirla tutta, aggiornando ad ogni passo le quantità che si intende misurare nel nuovo stato con $n+1$. Questo algoritmo ha il vantaggio di non dover quindi generare una nuova configurazione ad ogni n , e inoltre aggiornare le quantità ogni volta che si aggiunge un sito richiede uno sforzo computazionale minimo, in quanto si aggiunge appunto un solo sito al sistema. Lo svantaggio è che ovviamente le quantità così come le configurazioni del sistema ai vari n non saranno statisticamente indipendenti, ma presenteranno invece forti correlazioni: ciò non costituisce però un problema in quanto in gran parte dei casi di interesse ciò che importa è che le configurazioni siano indipendenti l'una dall'altra allo stesso n ma per *esecuzioni diverse* dell'algoritmo, cosa che ovviamente avviene e garantisce che la statistica relativa ad ogni n non sia alterata da eventuali correlazioni.

Per quest'algoritmo si è seguito il codice fornito nel testo H. Gould, J. Tobochnik, W. Christian, *An introduction to Computer Simulation Methods Application to Physical System*, 2016 e le indicazioni contenute nell'articolo M.E.J. Newmann, N.J. Ziff, *A fast montecarlo algorithm for site or bond percolation*, Phys. Rev. E 64, 2001. I punti salienti dell'algoritmo sono riportati di seguito:

1. Prima di tutto, si deve stabilire l'ordine in cui verranno occupati i siti della griglia, partendo da uno stato con nessun sito occupato fino ad occupare l'intera griglia. Il metodo più efficiente per fare ciò, soprattutto nel caso in cui L sia molto grande, consiste nel considerare un array, che chiameremo `order[]`, con tutti i numeri da 1 a N , con N numero totale di siti, e considerare una permutazione casuale dei siti:

- si considera `order[i]`, partendo da $i = 1$
- si prende un numero casuale intero j , con distribuzione uniforme, nel range $i \leq j \leq N$ e si considera l'elemento `order[j]`
- si scambiano i valori di `order[i]` e `order[j]`
- si ripete il tutto per $i + 1$, fino a $i = N$

la posizione nell'array `order[]` in cui i siti della griglia compaiono indica l'ordine in cui verranno occupati.

2. Si passa quindi alla parte centrale dell'algoritmo, contenuta nel metodo `occupy_site()`. Viene occupato un sito, seguendo l'ordine stabilito in precedenza in `order[]`, e questo sito viene considerato inizialmente come root di un cluster di dimensione 1, e vengono quindi aggiornate tutte le quantità relative per tenerne conto. I cluster vengono gestiti tramite un array di interi, che chiameremo `parent[]`, utilizzando una struttura logica ad albero: ogni cluster ha un unico sito root che lo identifica, e ogni sito appartenente al cluster conterrà un puntatore, contenuto nell'array `parent[]`, ad un altro sito del cluster in modo tale che ripercorrendo tali puntatori si raggiungerà sempre il sito root del cluster; il sito root invece che un puntatore conterrà, nella posizione corrispondente dell'array `parent[]`, l'opposto della dimensione del cluster di cui è root, salvata come numero negativo per distinguerla dai puntatori degli altri siti. L'array `parent[]`, che quindi identifica la griglia e la struttura dei cluster in essa contenuti, potrà avere i seguenti valori interi

Valore	Caso
EMPTY	se il sito non è occupato
$-[1, N]$	se il sito è root di un cluster, contiene l'opposto della dimensione del cluster
$[0, N]$	se il sito appartiene ad un cluster, punta alla posizione di un altro sito del cluster

dove EMPTY è un valore intero associato ad un sito non occupato, nel nostro caso sarà `std::numeric_limits<int>`, il più piccolo intero negativo rappresentabile dal tipo `int`, ma si può anche porre pari a $-(N + 1)$.

3. Una volta occupato il nuovo sito, si controllano i suoi vicini, che nel caso della griglia quadrata sono 4, e quelli già occupati si fondono uno dopo l'altro con il nuovo sito a formare un cluster più grande. Questa parte dell'algoritmo, contenuta nel metodo `mergeclusters(int, int)`, consta di un processo *Union/Find*, un tipo di algoritmo molto studiato. Nel nostro caso si userà l'algoritmo "weighted union/find with path compression" dovuto a Fischer, come indicato dallo stesso Newmann nell'articolo, e che consta dei seguenti passaggi:

- **Find:** si consideri uno dei vicini del nuovo sito, diciamo a destra, e supponiamo che questo sito appartenga già ad un cluster. In questo caso, si ripercorrerà la struttura ad albero del cluster attraverso i puntatori contenuti nell'array `parent[]` e partendo appunto dal sito considerato fino ad arrivare al sito root del cluster cui appartiene. Durante questo processo si esegue la "path compression": una volta trovato il sito root, il tragitto fatto nel trovarlo è ripercorso a ritroso e ognuno dei siti attraversati viene fatto puntare direttamente al sito root, invece che ad un altro sito del cluster. Ciò permette di risparmiare molto tempo nel caso in cui si dovesse ripetere questo processo con uno dei siti già interessati dalla compressione, in quanto questo punterà immediatamente al sito root. Questa parte dell'algoritmo è contenuta nel metodo `findroot(int)`, presentato di seguito data la sua brevità, ed è realizzata sfruttando la ricorsività

```
int findroot(int s){
    if(parent[s]<0){
        //se parent[s]<0, s e' un sito root,
        //torna direttamente la sua posizione
        return s;
    }
    //ricorsivita' e compressione
    return parent[s]=findroot(parent[s]);
}
```

- **Union:** una volta trovato il root del cluster considerato, e sapendo che il nuovo sito è un cluster di dimensione 1 che contiene solo sè stesso, si fondono i due cluster. La fusione avviene semplicemente facendo puntare il root di uno dei cluster al root dell'altro cluster, rendendolo quindi un ramo del cluster cui è aggiunto. Questo processo non avviene in maniera causale, ma secondo un criterio di peso ("weighted"): il cluster più piccolo è sempre inglobato nel cluster più grande. Una volta avvenuta la fusione, vengono aggiornate tutte le quantità di interesse, in particolare la dimensione del cluster più grande viene aggiornata aggiungendoci la dimensione del cluster più piccolo, in modo da tener conto dei nuovi siti acquisiti a seguito della fusione.

l'algoritmo fonde quindi due cluster, aggiungendo semplicemente il più piccolo al più grande, e aggiorna le quantità interessate, perchè tengano conto della fusione di due cluster in uno più grande. Inoltre, controlla che il nuovo cluster sia uno spanning cluster, e in tal caso salva la sua dimensione.

4. il procedimento descritto nei punti 1-3 è ripetuto finchè non viene occupata tutta la griglia.
5. La procedura descritta nei punti precedenti è ripetuta molte volte, in modo da accumulare dati e mediarli su molte configurazioni diverse, ottenute cambiando di volta in volta il seed.

In figura 1 i passaggi dell'algoritmo sono illustrati tramite una breve animazione.

(a) *Prime fasi* (b) *Completo*

Figura 1: **a)** Prime fasi dell'algoritmo di Newmann-Ziff. I cluster sono identificati dai vari colori, mentre i siti isolati e i siti root di ogni cluster sono colorati sempre in rosso. Nell'animazione sono illustrati i passaggi dell'algoritmo con particolare attenzione al caso in cui si debbano fondere due siti in un solo cluster di dimensione 2 o più siti isolati in un cluster più grande. Sopra ognuno dei siti è riportato il corrispondente valore contenuto nell'array `parent[]`, ad eccezione dei siti vuoti. **b)** Algoritmo completo, non sono visualizzati in dettaglio i singoli passaggi.

4 Analisi dati

Utilizzando l'algoritmo appena discusso, è stato possibile misurare e stimare le varie quantità di interesse. La prima e più importante quantità da determinare, fondamentale per poter passare ad uno studio approfondito delle altre, è la probabilità critica $p_c(L)$, dove è stata esplicitato il fatto che, a causa del fatto che simuliamo un sistema finito, il suo valore dipenderà dalla grandezza del sistema, tendendo al valore corretto per $L \rightarrow \infty$. Come già accennato, p_c determina la comparsa di uno spanning cluster nel sistema, il quale altera notevolmente le sue proprietà. In una simulazione finita, la definizione di spanning cluster e di p_c sono arbitrarie, perciò è necessario definire a priori in base a quali criteri si svolgerà lo studio: nel nostro caso, uno spanning cluster attraverserà la griglia in direzione orizzontale; per la p_c , si seguiranno entrambi i procedimenti presentati all'inizio della sezione 2, partendo dalla stima di $p_c(L)$ come media delle probabilità alle quali si osserva apparire uno spanning cluster per ogni configurazione, per poi passare alla stima di $p_c(L)$ tramite P_{span} .

Per la prima misurazione, sono state quindi considerate 100 000 configurazioni differenti ad un determinato L , variando di volta in volta il seed. Per ogni configurazione, è stato considerato il valore n e la corrispondente $p = n/N$ alla quale si osserva apparire lo spanning cluster, e la $p_c(L)$ è stata quindi stimata come la media di questi valori. Per risparmiare tempo, l'esecuzione dell'algoritmo per ogni configurazione viene arrestata nel momento in cui si osserva uno spanning cluster, permettendo di salvare il 40 – 50% del tempo di esecuzione. Nella tabella a seguire sono riportati i valori di $p_c(L)$, insieme alla deviazione standard delle p indicata come σ_p e anch'essa dipendente da L , per vari valori di L :

L	$p_c(L)$	σ_p
8	0.590 388	0.000 262
16	0.590 752	0.000 177
32	0.592 103	0.000 112
64	0.592 414	6.98×10^{-5}
128	0.592 607	4.25×10^{-5}
256	0.592 769	2.58×10^{-5}
512	0.592 712	1.76×10^{-5}

Il valore di p_c atteso nel caso infinito dipende molto dalla geometria della griglia che si sta considerando: differenti griglie si distinguono per il numero di *nearest neighbour* (n.n.) per ogni sito, e ciò influenza l'identificazione e formazione dei cluster. Nel nostro caso, con una griglia quadrata abbiamo 4 n.n. per ogni sito, e per $L \rightarrow \infty$ la probabilità critica assume il valore

$$p_c \approx 0.592741 \quad (1)$$

il valore (1) sarà preso come valore di riferimento nel seguito. In tabella, si può osservare come il valore della $p_c(L)$ stimata tenda, per L crescente, al valore esatto di p_c . Per quanto riguarda la velocità di convergenza del valore stimato al valore p_c esatto, essa dipende dal tipo di stimatore utilizzato per calcolarne il valore, ma in generale l'andamento sarà regolato da una legge del tipo

$$|p - p_c| \propto L^{-\frac{1}{\nu}} \xrightarrow{d=2} L^{-3/4} = L^{-0.75}$$

dove l'esponente ν nel caso di un sistema in due dimensioni ha valore $\nu = 4/3$, e quindi ci si aspetta che la convergenza della stima vada come $L^{-3/4}$. L'utilizzo di stimatori differenti permette di ottenere una convergenza ulteriore del tipo L^{-1} o nei casi migliori L^{-2} , e la stima convergerà quindi ripetitivamente con andamento $L^{-7/4}$ e $L^{-11/4}$. In figura 2 è riportato un log-log plot dei valori $|p_c(L) - p_c|$ al variare di L . Si osservi che l'andamento in scala logaritmica è di tipo lineare, il che implica $|p_c(L) - p_c| \propto L^\alpha$. Come mostrato in figura 2, il valore dell'esponente è $\alpha = (-1.2008 \pm 0.1019)$, quindi lontano dal valore atteso -0.75 per questo tipo di stima. Le motivazioni di ciò verranno spiegate nella sezione successiva.

Ai nostri fini è interessante notare anche che la deviazione dei valori di p ottenuti nelle varie configurazioni, σ_p , diminuisce con l'aumentare di L , il che significa che il valore p_c risulta essere ben definito per L grandi, mentre per L piccolo le fluttuazioni sono molto elevate. In figura 3 è riportato un plot del logaritmo della varianza sul logaritmo di L . L'andamento log-log è di tipo lineare, il che indica che l'andamento effettivo è regolato da una legge di potenza del tipo $\sigma_p \sim L^\beta$. Nel plot di figura 3 sono riportati i valori del fit lineare effettuato, che ci permettono di stimare il valore dell'esponente $\beta \approx (-0.666 \pm 0.008)$. Si noti in questo caso come il valore ottenuto sia molto vicino al valore $1/\nu = -0.75$, il che è ovvio se si considera che così come l'intervallo critico si ridurrà con andamento $L^{-\frac{1}{\nu}}$, ci si aspetta che anche l'errore, che individua un intervallo intorno al valore stimato, sia regolato dalla stessa legge di potenza per L .

Per la seconda stima di $p_c(L)$, si seguirà il testo di riferimento prendendo come valore di p_c il valore di p corrispondente ad una $P_{span}(p)$ pari a circa il

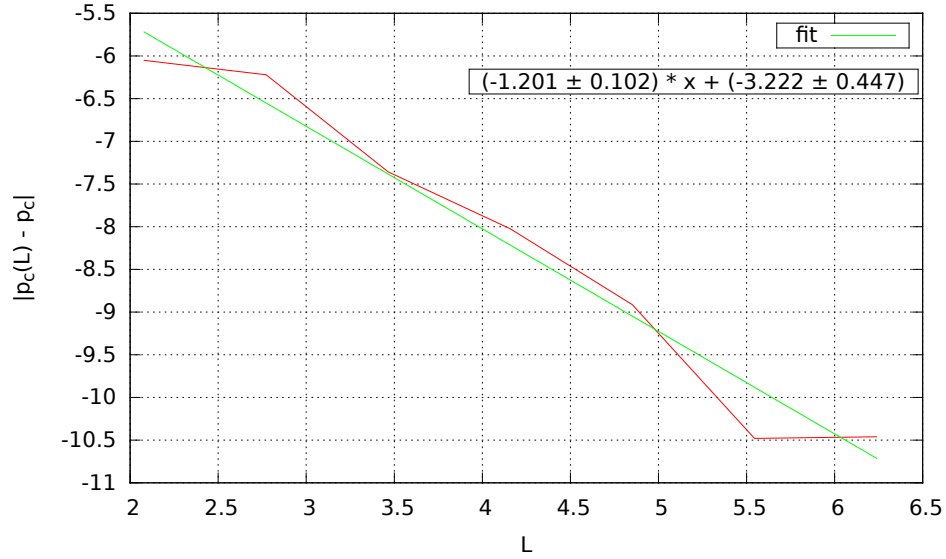


Figura 2: Log-log plot di $|p_c(L) - p_c|$ al variare di L . Nella legenda sono riportati i parametri ottenuti con il fit lineare.

50%, dove si ricorda che $P_{span}(p)$ è la probabilità di trovare uno spanning cluster, ad una certa $p = n/N$, in una configurazione.

Si è quindi misurata, come prima cosa, la P_{span} e il suo andamento al variare di p : se n^* è il valore di n al quale appare uno spanning cluster, allora la P_{span} viene posta uguale a 0 per ogni $n < n^*$ e uguale a 1 per ogni $n \geq n^*$. Il suo valore viene poi mediato, per ogni p , su 100 000 configurazioni. L'andamento è riportato in figura 4 per alcuni valori di L , e si può osservare come esso tenda, per L crescente, all'andamento atteso nel caso infinito: infatti, per L piccola si ha che $P_{span} > 0$ già per valori di p molto bassi e cresce lentamente fino a raggiungere il valore $P_{span} = 1$ per p prossima ad 1; per L grandi, invece, P_{span} si mantiene molto vicina allo 0 per $p < p_c$ e inizia a crescere soltanto per $p \sim p_c$, crescendo rapidamente fino al valore 1 in un intervallo molto ristretto: si noti come l'intervallo in cui si ha $0 < P_{span} < 1$ decresce in ampiezza con l'aumentare di L , in accordo con la legge $|p - p_c| \sim L^{-\frac{1}{\nu}}$ vista nel paragrafo 2.1. Il comportamento osservato per L grandi è tipico dei fenomeni critici, con le quantità caratteristiche del sistema che variano molto rapidamente in un intervallo ristretto intorno al punto critico.

Nella seguente tabella sono riportati i valori stimati di $p_c(L)$ per ogni valore di L considerato, e nella colonna accanto i corrispondenti valori di $P_{span} \approx 0.5$ con relativo errore:

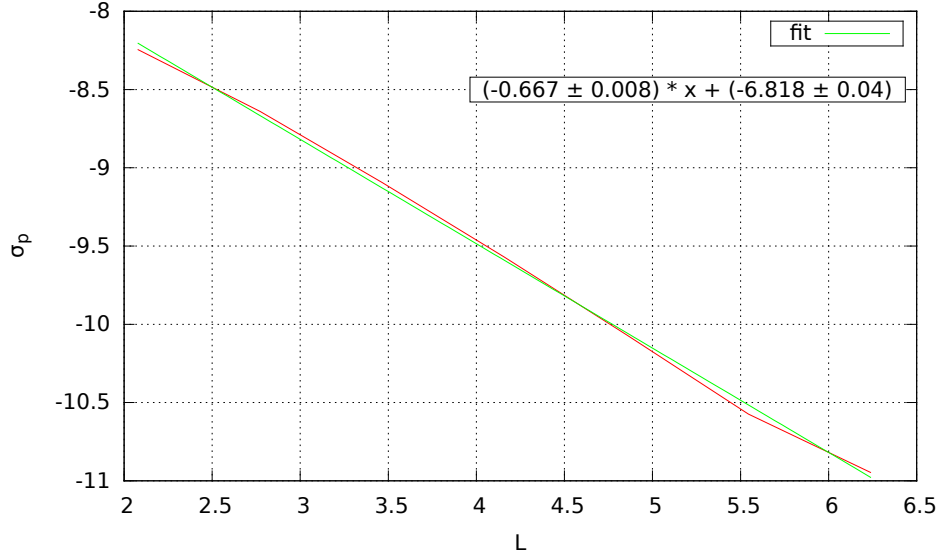


Figura 3: Plot log-log della varianza delle probabilità in relazione ad L , dimensione lineare della griglia. Nella legenda sono riportati i parametri del fit lineare con il relativo errore.

L	$p_c(L)$	P_{span}
8	0.578 125	0.46480 ± 0.00158
16	0.589 844	0.49629 ± 0.00158
32	0.591 797	0.49995 ± 0.00158
64	0.592 529	0.50075 ± 0.00158
128	0.592 590	0.49918 ± 0.00158
256	0.592 712	0.49977 ± 0.00158
512	0.592 754	0.49987 ± 0.00158

Si noti anzitutto che i valori di P_{span} non raggiungono mai il valore 0.5, e per ogni L è stato considerato un valore differente: per ogni L si ha infatti soltanto un insieme discreto di probabilità $p = n/N$ mentre P_{span} è una funzione continua di p e ciò comporta che nella nostra stima abbiamo accesso soltanto ad alcuni valori di P_{span} . Si è potuto quindi scegliere soltanto il valore più vicino a 0.5 tra quelli disponibili, e ciò comporta un'approssimazione e un'errore nella stima; una soluzione a questo problema verrà discussa in seguito, quando verrà presentata una stima alternativa basata sullo stesso criterio. Si noti inoltre come l'errore di P_{span} non dipenda da L , al contrario dell'errore ottenuto nella stima precedente che diminuiva all'aumentare di L : ciò è dovuto al fatto che P_{span} può effettivamente assumere due soli valori, 0 e 1, a seconda che lo spanning cluster sia presente o meno, ed il suo errore non potrà quindi dipendere da L . La successione di valori 0 e 1 su varie configurazioni sarà descritta da una distribuzione binomiale con probabilità appunto $P_{span}(p_c)$, e quindi la deviazione

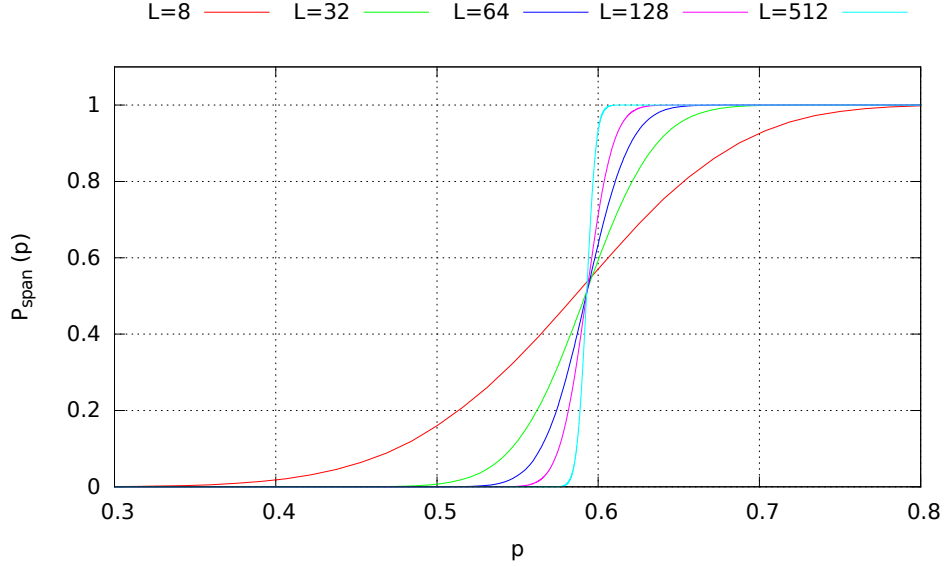


Figura 4: Andamento di P_{span} al variare di p per vari valori di L .

standard associata alla media sarà quella della distribuzione binomiale

$$\sigma = \sqrt{\frac{P_{span}(p_c)(1 - P_{span}(p_c))}{trials}}$$

dove *trials* è il numero di configurazioni indipendenti utilizzate per calcolare la media. L'errore dipenderà quindi solo dal numero di *trials*, e possiamo porre $P_{span}(p_c) = 0.5$ per semplificare il calcolo ed aggirare l'approssimazione prima discussa per quanto riguarda la scelta di $P_{span}(p_c)$.

Si noti come anche in questo caso il valore di $p_c(L)$ tenda al valore $p_c \approx 0.592741$ con l'aumentare di L . Come nel caso precedente, si è calcolata la velocità di tendenza di questa stima: in figura 5 sono riportati in grafico i valori di $|p_c(L) - p_c|$ in relazione ad L , in scala logaritmica per entrambi gli assi.

Come si può notare dal grafico in figura 5, l'andamento lineare indica che la dipendenza è del tipo $|p_c(L) - p_c| \propto L^\alpha$ come atteso, e l'esponente ottenuto tramite il fit lineare è pari a $\alpha = (-1.652 \pm 0.090)$. La stima tende quindi molto più rapidamente al valore atteso rispetto alla stima precedente ottenuta usando il valor medio, ed inoltre l'esponente calcolato ha un valore compatibile con la tendenza attesa per questa stima, che è del tipo

$$|p_c(L) - p_c| \propto L^{-1-\frac{1}{\nu}} \xrightarrow{d=2} L^{-\frac{7}{4}} = L^{-1.75}$$

dove ν è l'esponente critico di ξ introdotto nella sottosezione 2.1 ed è pari a $\nu = 4/3$ per una griglia di dimensione 2. Lo svantaggio rispetto alla stima fatta utilizzando la media è che questa stima si discosta molto dal valore atteso per L piccoli ed è quindi più soggetta agli effetti dovuti alla finitezza del sistema, mentre la stima in media tende più lentamente ma permette di avere valori abbastanza accurati anche per L non eccessivamente grandi.

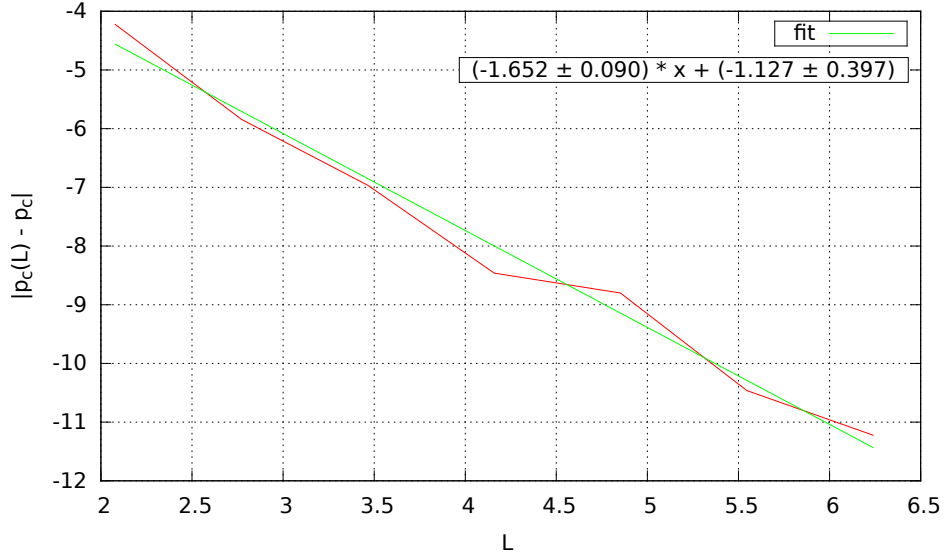


Figura 5: Log-log plot della distanza tra la stima e il valore atteso per vari valori di L . Nella legenda sono riportati i parametri del fit lineare con il relativo errore.

La discussione fatta fin'ora ha portato a confrontare due stime fatte utilizzando criteri differenti. Il criterio della media è un criterio abbastanza semplice da applicare, ma è tra i meno precisi e converge molto lentamente, mentre il criterio in cui si è cercata la $p_c(L)$ come soluzione dell'equazione (approssimata) $P_{span} \approx 0.5$ converge più rapidamente, ma come si è già accennato implica un'approssimazione che verrà discussa nel seguito. Nel seguito verrà preso come riferimento l'articolo M.E.J. Newmann, N.J. Ziff, *A fast montecarlo algorithm for site or bond percolation*, Phys. Rev. E 64, 2001 , già citato in precedenza.

Si consideri una griglia, le variabili che determinano lo stato di questa griglia sono il numero di siti occupati n , la probabilità di occupazione p e ovviamente il numero di siti totali N . Se fissiamo p ed n , potremmo considerare gli stati possibili della griglia come microstati di un ensemble microcanonico, in cui N ha lo stesso ruolo del volume, n il ruolo dell'energia e p quello della temperatura e i microstati possibili saranno dati dal coefficiente binomiale tra N ed n , con distribuzione di probabilità uniforme data dall'inverso di quest'ultimo valore. Ora, immaginiamo di tenere fissati soltanto N e p e variare n , saremo nel caso dell'ensemble canonico e la probabilità di trovarsi in uno dei microstati con numero di siti occupati pari ad n sarà data da una distribuzione binomiale

$$B(N, n, p) = \binom{N}{n} p^n (1 - p)^{N-n}$$

per ognuno degli n possibili. La distribuzione binomiale ha valore massimo per $n_{max} = pN$, e decresce molto rapidamente per valori lontani dal massimo, come mostrato in figura 6. Per calcolare il valore di una qualunque osservabile $Q(p)$ nell'ensemble canonico, basterà misurarne quindi il valore nell'ensemble microcanonico per ogni valore di n possibile e pesare i valori ottenuti, che

chiameremo Q_n , con la distribuzione binomiale vista sopra, ottenendo quindi

$$Q(p) = \sum_n B(N, n, p) Q_n = \sum_n \binom{N}{n} p^n (1-p)^{N-n} Q_n$$

valore di Q nell'ensemble canonico con p fissata. Misurando quindi i valori Q_n dell'osservabile nell'ensemble microcanonico per ogni valore di n , potremo ottenere il valore della stessa osservabile a qualunque valore di p eseguendo semplicemente la convoluzione con la distribuzione binomiale, come mostrato sopra.

A questo punto è possibile approfondire la discussione della P_{span} . Consideriamo la probabilità $R_L(p)$ che ad una certa p nell'ensemble canonico una griglia di dimensione lineare L sia attraversata da uno spanning cluster. In base alla discussione precedente, questa quantità può essere calcolata come:

$$R_L(p) = \sum_n B(N, n, p) S(n)$$

dove $B(N, n, p)$ è la distribuzione binomiale e $S(n)$ è la probabilità di avere uno spanning cluster per una configurazione con esattamente n siti occupati. Viene da sé che la $S(n)$ non è altro che la P_{span} calcolata in precedenza per i vari valori di n . Nella stima fatta utilizzando la P_{span} l'approssimazione consisteva quindi nell'ignorare la convoluzione con la distribuzione binomiale e considerare quindi come valore di $R_L(p)$, per ogni probabilità di occupazione $p = n/N$, il valore di P_{span} calcolato sull'ensemble microcanonico più probabile, quello tale che $n = n_{max} = pN$. Non solo quindi la stima precedente permetteva di calcolare R_L soltanto per certi valori discreti di p , ma inoltre per ogni valore di p veniva considerato soltanto il contributo per $n = n_{max}$ e ignorati tutti gli altri contributi per n diverso dal valore massimo. Data la rapidità con cui la distribuzione binomiale decresce per n lontano dal massimo, l'approssimazione può andare bene per misure non troppo precise e per N piccoli, ma è richiesta la convoluzione se si vuole raggiungere un livello di precisione più elevato, soprattutto per N più elevati dove la distribuzione ha un'ampiezza maggiore, come si può vedere in figura 6.

A fini illustrativi e di confronto, si è proceduto con un calcolo della $R_L(p)$ seguendo le linee guida dettate da Newmann e Ziff nel loro articolo. Prima di tutto, le due quantità da calcolare sono la $S(n)$ e la distribuzione binomiale $B(N, n, p)$: il calcolo della $S(n)$ è identico a quello già eseguito per la P_{span} , ma ci si è limitati a considerare la media su 10 000 configurazioni, per via del fatto che per N grandi l'algoritmo richiede molto tempo. Il calcolo della $B(N, n, p)$ risulta essere molto semplice ma bisogna adottare i dovuti accorgimenti: per N grandi non è infatti possibile calcolare direttamente i fattoriali da cui poi derivare il coefficiente binomiale; è possibile approssimare la distribuzione binomiale con una gaussiana, ma l'approssimazione non è molto buona. Per risolvere il problema, si farà ricorso ad un metodo proposto dallo stesso Newmann nell'articolo citato in precedenza: per p fissata, la distribuzione binomiale ha valore massimo nel punto $n_{max} = pN$, come già accennato. Poniamo questo valore pari ad 1 e calcoliamo i valori della distribuzione per gli altri n usando il seguente metodo iterativo:

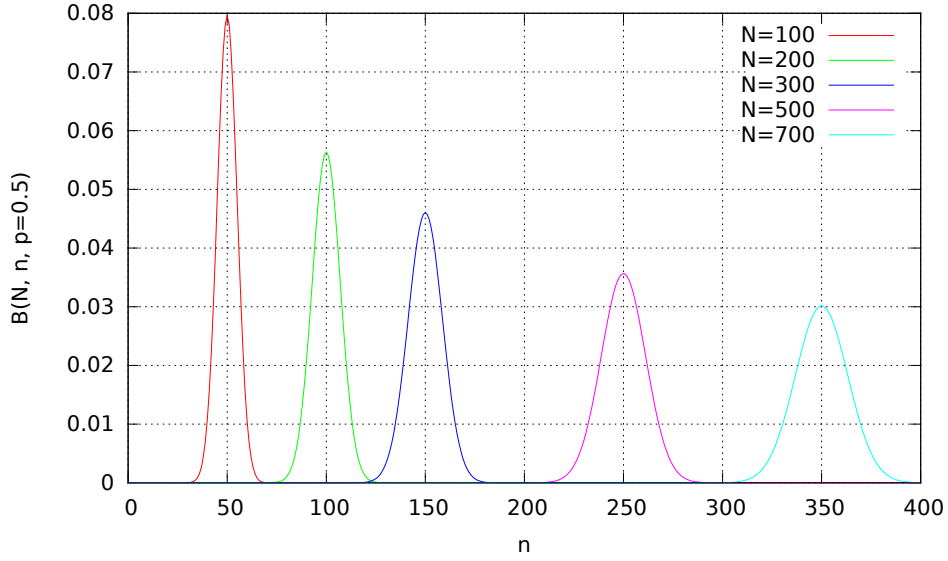


Figura 6: Distribuzione binomiale calcolata con il metodo iterativo per alcuni valori di N e per $p = 0.5$.

$$B(N, n, p) = \begin{cases} B(N, n-1, p) \frac{N-n+1}{n} \frac{p}{1-p} & \text{per } n > n_{max} \\ B(N, n+1, p) \frac{n+1}{N-n} \frac{1-p}{p} & \text{per } n < n_{max} \end{cases}$$

Avendo posto $B(N, n_{max}, p) = 1$ la distribuzione non sarà ovviamente normalizzata, per normalizzarla basterà dividere ognuno dei valori calcolati per la costante di normalizzazione $\sum_n B(N, n, p)$. In figura 6 è riportata la distribuzione binomiale calcolata usando questo metodo per alcuni valori di N .

Si è quindi calcolata la $R_L(p)$ per vari valori di p usando la convoluzione di $S(n)$ con $B(N, n, p)$. La convoluzione è un calcolo impegnativo dal punto di vista del tempo di computazione, e per velocizzare il calcolo ci si è limitati a valori di p compresi nell'intervallo $[0.56, 0.6]$, in quanto in nessuno dei casi trattati la $p_c(L)$ è stata mai osservata fuori da questo intervallo; considerare un intervallo così ristretto ha permesso di calcolare $R_L(p)$ a step di p dell'ordine di 1.0×10^{-6} , e il suo andamento per vari valori di L è riportato in figura 7. Si osservi che l'andamento di $R_L(p)$ ha le stesse caratteristiche di quello già osservato per P_{span} , con la differenza che $R_L(p)$ è ora una funzione continua di p . Per la stima di $p_c(L)$ si è utilizzato lo stesso criterio usato per P_{span} : si è presa come $p_c(L)$ la p tale che $R_L(p_c) = R_\infty(p_c) = 0.5$, dove $R_\infty(p_c)$ è il valore atteso nel caso infinito. I valori ottenuti sono riportati nella tabella a seguire:

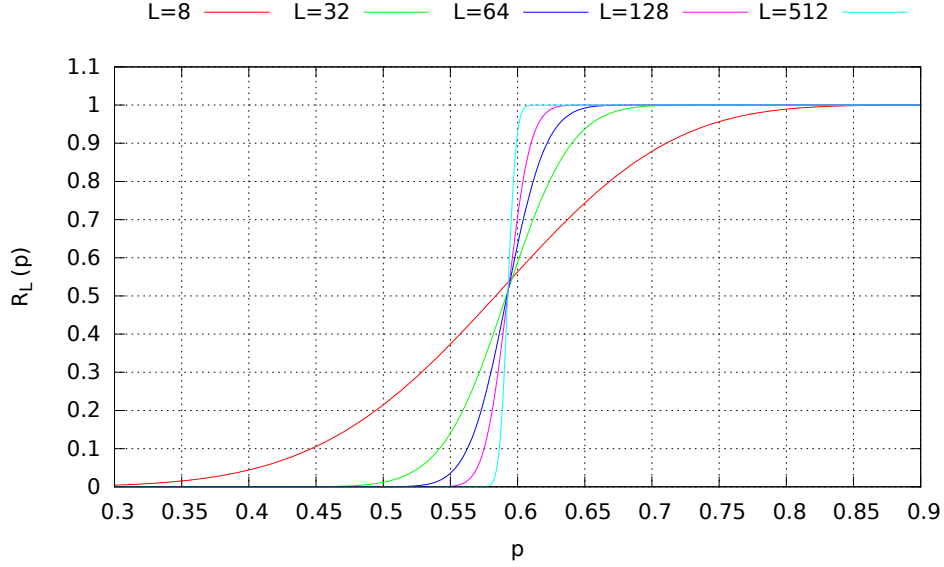


Figura 7: Andamento calcolato di $R_L(p)$ per vari valori di L .

L	p_c	$R_L(p_c)$
8	0.585 158	0.500 001
16	0.590 037	0.500 002
32	0.591 694	0.499 999
64	0.592 143	0.500 002
128	0.592 456	0.499 987
256	0.592 622	0.499 981
512	0.592 744	0.499 977

L'errore sui valori $R_L(p)$ ha le stesse caratteristiche dell'errore di P_{span} e può essere quindi calcolato ponendo nella formula dell'errore vista in precedenza $R_L(p_c) = 0.5$ e $trials = 10\,000$, in questo caso. L'errore ottenuto è

$$\sigma_R = 0.005$$

Non resta che calcolare la velocità di convergenza di questa stima. In figura 8 è riportato l'andamento di $|p_c(L) - p_c|$ al variare di L , il tutto in scala logaritmica. L'andamento è quindi di tipo potenza $|p_c(L) - p_c| \propto L^\alpha$ e l'esponentiale calcolato è $\alpha = (-1.6000 \pm 0.2579)$, valore molto vicino a quello ottenuto per la stima precedente fatta utilizzando P_{span} e compatibile con il valore -1.75 atteso. Nell'articolo citato in precedenza, Newmann e Ziff applicano questo metodo con $R_\infty = 0.521058$ e riscontrano come questo stimatore abbia una velocità di convergenza del tipo L^{-2} , il che implica che la stima di p_c tenderà al valore per il caso infinito con una legge del tipo

$$|p_c(L) - p_c| = L^{-2 - \frac{1}{\nu}} \xrightarrow{d=2} L^{-11/4} = L^{-2.75}$$

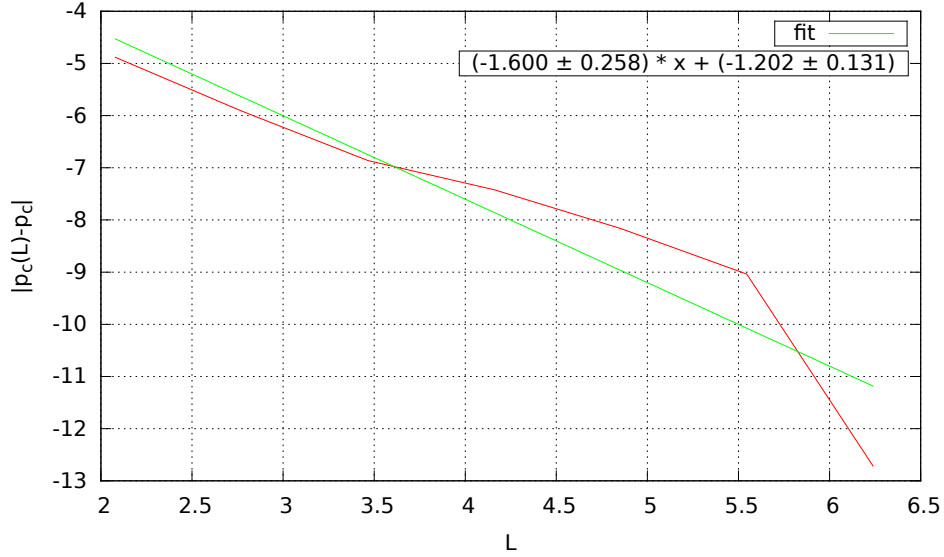


Figura 8: Log-log plot di $|p - p_c|$ per vari valori di L . Nella legenda sono riportati i parametri ottenuti con il fit lineare e con il relativo errore.

questa tendenza è molto più rapida rispetto a qualunque altra fin'ora osservata, e non è stata misurata nel nostro caso.

Infine, come passo conclusivo della discussione, si sono misurate le varie quantità presentate nella sezione 2 e se ne è osservato l'andamento al variare di L . Non si sono potuti però stimare gli esponenti critici per mancanza di tempo, quindi la discussione nel seguito sarà puramente qualitativa.

La P_{span} è già stata discussa, passiamo quindi alla P_∞ , probabilità che un sito appartenga allo spanning cluster. In figura 9 è riportato il suo andamento al variare di p . Si noti che da qui in avanti $p = n/N$, cioè nessuna di queste quantità sarà pesata con la distribuzione binomiale, quindi si osserveranno in effetti soltanto i valori calcolati sull'ensemble microcanonico, ma sarà sufficiente per una discussione qualitativa. L'andamento in figura 9 è quello atteso per P_∞ : per L piccoli si ha che $P_\infty > 0$ già a valori bassi di p e cresce molto lentamente al valore 1 in un intervallo molto ampio intorno a p_c ; per L crescente si osserva che l'intervallo in cui P_∞ cresce tende a ridursi avvicinandosi al valore critico p_c , e la P_∞ cresce molto velocemente in un intervallo sempre più piccolo, per poi mantenersi $P_\infty = 1$ per $p > p_c$.

Un'altra quantità di interesse è $S(p)$, la dimensione media dei cluster nella griglia, da non confondersi con $S(n)$ utilizzata in precedenza per la convoluzione di $R_L(p)$. In figura 10 è riportato l'andamento di $S(p)$ per vari valori di L . L'andamento osservato in figura è quello atteso e già discusso nella sottosezione 2.1: $S(p) \approx 1$ per valori di p piccoli e cresce per avvicinandosi a p_c , fino a raggiungere il massimo nel punto critico; allontanandosi dal punto critico, $S(p)$ decresce rapidamente fino ad annullarsi. Si noti come il massimo cada sempre in prossimità del valore critico $p_c \approx 0.592$ e come il massimo cresca all'aumentare di L , in linea con la divergenza prevista nel caso $L \rightarrow \infty$. Per calcolare questa

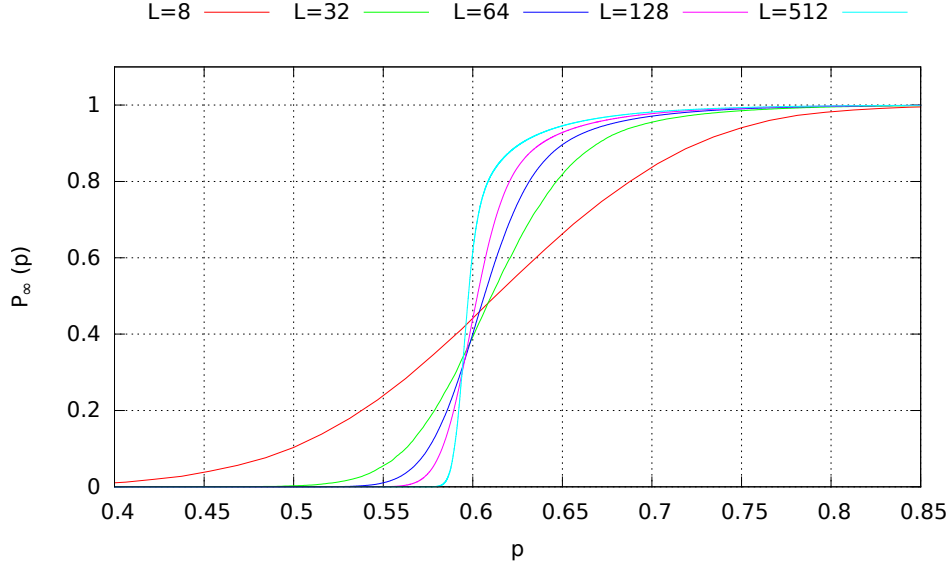


Figura 9: Andamento di P_∞ per p .

quantità è necessario accumulare, nell'algoritmo, la dimensione al quadrato dei vari cluster e sommare su tutti i cluster: ciò, per $N \geq 256$, richiede l'utilizzo di una variabile di tipo `long long int`, per evitare errori di overload che darebbero valori negativi.

In ultimo, si è considerato $n_s(p)$, la distribuzione della dimensione s dei cluster per ogni p . In figura 11 è riportato l'andamento di n_s al variare di s , per due diversi valori di p . Si noti come la n_s decresce molto più rapidamente per $p \neq p_c$; questo comportamento è facilmente osservabile in figura 12, dove è riportato un log-log plot di n_s calcolata per alcuni valori di p .

Come si può osservare in figura 12, $n_s(p)$ decresce in generale con l'aumentare di s , ma per $p \neq p_c$ decresce molto più velocemente rispetto al caso con $p = p_c$. Inoltre, per $p = p_c$ il logaritmo della distribuzione ha un andamento lineare rispetto a $\log(s)$, il che indica un andamento del tipo $n_s(p = p_c) = s^{-\tau}$, con $\tau = (1.512 \pm 0.006)$, cioè n_s decresce come una potenza di s . Lo stesso andamento lineare non è osservato per $p \neq p_c$ se non per piccoli intervalli, e si può osservare come l'intervallo in cui l'andamento è approssimativamente lineare cresce per $p \rightarrow p_c$, raggiungendo un massimo per $p = p_c$. Inoltre, in figura si può osservare l'esistenza di oscillazioni molto ampie per s grandi, che vanno a disturbare l'andamento lineare: la giustificazione sta nel fatto che per s abbastanza grandi la probabilità, ad una certa p , di avere cluster della dimensione considerata è molto bassa ma non nulla, e quindi la statistica per s vicini sarà soggetta a forti oscillazioni e spesso per s vicini si otterranno gli stessi valori; inoltre, per $p > p_c$ si hanno dei picchi per s molto grandi, ad indicare una configurazione in cui esiste un cluster che occupa gran parte della griglia senza però attraversarla (lo spanning cluster non è considerato in n_s).

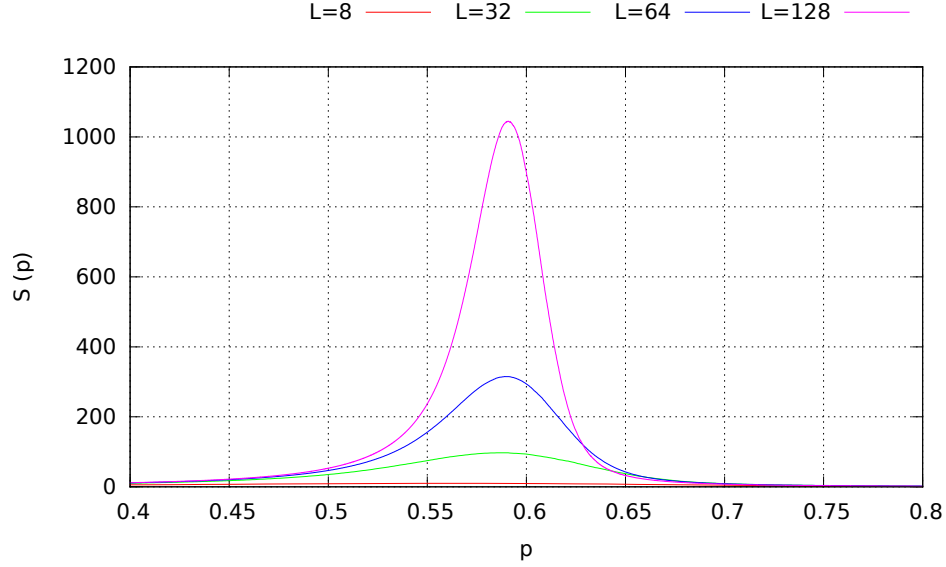


Figura 10: Andamento di $S(p)$ per p .

5 Conclusioni

In questa relazione ci si è quindi concentrati sullo stimare il valore di p_c confrontando due stime differenti. In entrambe le stime, il valore ottenuto tende a p_c per L crescente e la velocità di tendenza dipende dalla stima utilizzata: per la stima effettuata utilizzando il valor medio di p cui si osserva uno spanning cluster, l'andamento non è risultato in accordo con l'andamento atteso, e in particolare l'esponente stimato risulta essere più piccolo di quello atteso, risultando in una stima molto più rapida di quanto riportato nella letteratura sulla percolazione; per la stima fatta utilizzando la P_{span} , questa stima ha dato buoni risultati sia per quanto riguarda il valore di p_c che per quanto riguarda l'andamento, in quanto l'esponente calcolato è compatibile con il valore atteso. Si è poi presentata brevemente, seguendo l'articolo di Newmann e Ziff, una procedura per ottenere una stima più precisa, e si sono riportati i risultati per p_c ottenuti con lo stesso criterio di P_{span} : nonostante il numero molto limitato di iterazioni (10 000), si sono ottenuti risultati accettabili sia per l'esponente che per la stima di p_c .

Le motivazioni che potrebbero giustificare i risultati ottenuti e le deviazioni di alcuni di essi dai valori attesi sono principalmente due, se si escludono errori nell'algoritmo utilizzato (l'algoritmo è stato testato per $L = 2$ e $L = 3$ confrontandolo con i valori attesi per, ad esempio, P_{span} , e non si sono riscontrate anomalie): il numero di iterazioni e le condizioni al contorno. Come argomentato da Newmann e Ziff nell'articolo presentato nel 2001, considerando una griglia con N , se ogni sito può assumere due stati il numero totale di stati possibili è 2^N , numero enorme anche per N molto piccoli; è quindi importante mediare i propri dati su un numero di iterazioni dell'algoritmo che permetta di avere un buon campione di stati, e che quindi non sia troppo piccolo. Si è scelto di utilizzare 100 000 iterazioni come compromesso tra il tempo di esecuzione, che per N

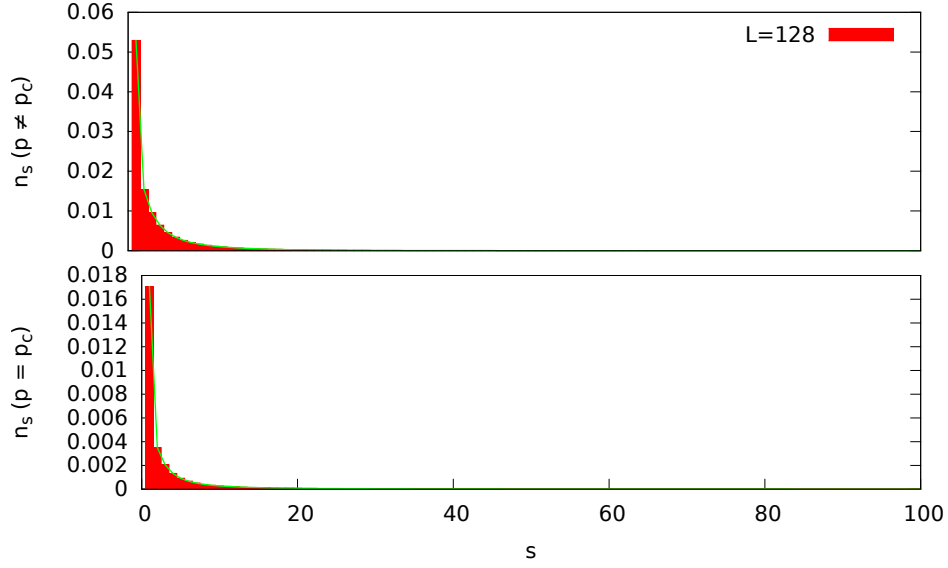


Figura 11: Andamento di $n_s(p)$ per s , per $p = 0.4$ in alto e per $p = p_c = 0.592741$ in basso. I due plot hanno lo stesso asse x.

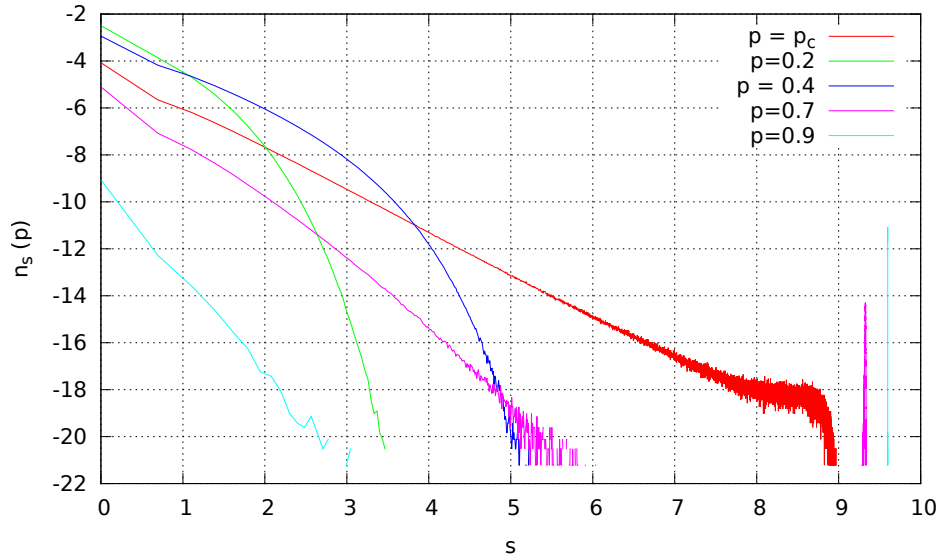


Figura 12: Log-log plot di n_s su s per vari valori di p . Tutti i valori sono calcolati per $L = 128$.

grandi tende a diventare proibitivo, e un numero di iterazioni sufficiente ad aver un buon campione. Nell'articolo gli autori dichiarano di aver usato circa 7×10^9 iterazioni soltanto per la griglia $L = 128$, ma una simulazione di questa entità esula dagli scopi di questa relazione. Gli effetti di questa considerazione possono essere osservati in figura 5 e soprattutto in figura 8, dove è possibile osservare come l'andamento dei valori osservato per N piccoli quando si considerano N più grandi è maggiormente soggetto a deviazioni e tende ad allontanarsi da un andamento strettamente lineare.

Per quanto riguarda le condizioni al contorno, nella stesura dell'algoritmo si sono utilizzate le condizioni di contorno aperto (obc), seguendo le linee guida del testo di riferimento. Per lo studio di un sistema come una griglia di dimensione finita, le condizioni al contorno più adatte e utilizzate sono le condizioni al contorno periodiche (pbc), e tali condizioni sono state utilizzate da Newmann e Ziff nel loro studio. L'utilizzo di queste condizioni implica però delle modifiche all'algoritmo e al metodo di identificazione di uno spanning cluster che non si è fatto in tempo ad implementare, e per le quali si rimanda all'articolo di Newmann e Ziff. L'utilizzo delle pbc migliora notevolmente la statistica e permette di simulare meglio il comportamento che il sistema dovrebbe avere all'infinito, riducendo gli effetti di bordo che nel caso delle obc sono molto più influenti, soprattutto se non si utilizza un campione di grandezza adeguata.