

Laboratorio di Fisica Computazionale

Luca Cassia - [MAT. 728341]

Spring 2012

Indice

1	Integrazione Numerica	2
1.1	Newton-Cotes	2
1.2	Quadrature Gaussianhe	3
1.2.1	Integrazione Composta	5
1.3	Monte Carlo	5
1.3.1	Campionamento di Importanza	5
1.4	Esempi	6
2	Algoritmo Metropolis	8
2.1	Oscillatore Armonico	9
2.1.1	Azione e Termalizzazione	10
2.1.2	Autocorrelazione	10
2.1.3	Funzioni di Correlazione	11
2.1.4	Calcolo di Grandezze Quanto-Meccaniche Secondarie	13
3	Metodo Runge-Kutta	15
3.1	Sistemi Dinamici	16
4	Metodo Implicito	17
4.1	Equazione di Schrödinger	18
	Bibliografia	20

Capitolo 1

Integrazione Numerica

In analisi numerica, l'integrazione numerica consiste in una serie di metodi che stimano il valore di un integrale definito, senza dover calcolare la primitiva della funzione integranda. In questa sezione si illustrano alcuni dei principali metodi deterministici e non.

1.1 Newton-Cotes

Le regole di quadratura di Newton-Cotes sono formule che consistono nel valutare l'integrando in punti equispaziati dell'intervallo di integrazione.

Si assume che il valore di una funzione $f : [a, b] \in \mathbb{R} \rightarrow \mathbb{R}$ sia noto nei punti x_i , per $i = 0, \dots, n$ tali che

$$x_i = a + \left(\frac{b-a}{n} \right) i$$

La formula di Newton-Cotes di grado n si ottiene interpolando f nei punti x_i con i polinomi della base di Lagrange, e integrando la polinomiale risultante, $L(x)$, nell'intervallo $[a, b]$.

$$\int_a^b f(x) dx \approx \int_a^b L(x) dx = \int_a^b \left(\sum_{i=0}^n f(x_i) \ell_i(x) \right) dx = \sum_{i=0}^n f(x_i) \underbrace{\int_a^b \ell_i(x) dx}_{w_i}$$

dove gli $\ell_i(x)$ sono i polinomi di Lagrange così definiti

$$\ell_i(x) := \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - x_j}{x_i - x_j}$$

$$\ell_i(x_j) = \delta_{ij}$$

La formula di Newton-Cotes assume così la semplice forma di media dei valori $f(x_i)$ pesati sui coefficienti w_i (indipendenti da f)

$$\int_a^b f(x) dx \approx \sum_{i=0}^n w_i f(x_i)$$

Si dimostra che l'errore dell'interpolazione di f con un polinomio è

$$E(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi(x)) \prod_{i=0}^n (x - x_i)$$

per un certo $\xi \in [a, b]$ dipendente da x . Non avendo tuttavia alcuna informazione su come individuare il punto ξ in genere si effettua solo una stima del limite superiore sull'errore $E(x)$ per ξ tale che $f(\xi) = \max_{x \in [a, b]} f(x)$.

Al primo ordine dell'approssimazione Newton-Cotes la formula si riduce al cosiddetto metodo dei "trapezi":

$$\int_a^b f(x) dx = \frac{b-a}{2} [f(a) + f(b)] + \underbrace{\frac{1}{2} \int_a^b f''(\xi(x))(x-a)(x-b)}_{E_1}$$

mentre al secondo ordine non è altro che la regola di Simpson:

$$\int_a^b f(x) dx = \frac{b-a}{3} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] + \underbrace{\frac{1}{6} \int_a^b f'''(\xi(x))(x-a)\left(x - \frac{a+b}{2}\right)(x-b)}_{E_2}$$

1.2 Quadrature Gaussian

La regola di quadratura Gaussiana a n -punti è un metodo di integrazione numerica costruito in modo tale da fornire un risultato esatto per polinomi di grado inferiore a $2n$, attraverso una scelta appropriata di punti x_i e pesi w_i , per $i = 1, \dots, n$. Sul dominio di integrazione convenzionale $[-1, 1]$ la regola è così formulata

$$\int_{-1}^1 f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

L'accuratezza del risultato è tanto più grande quanto meglio f è approssimata da un polinomio. Se tuttavia la funzione integranda può essere scritta come $f(x) = W(x)g(x)$, dove $g(x)$ è approssimativamente polinomiale e $W(x)$ è nota, allora esistono w'_i tali che

$$\int_{-1}^1 f(x) dx = \int_{-1}^1 W(x)g(x) dx \approx \sum_{i=1}^n w'_i g(x_i)$$

$W(x)$ viene detta funzione peso, mentre i punti x_i sono le radici di un polinomio appartenente alla classe dei polinomi ortogonali.

Nel caso considerato $W(x) = 1$ ed i polinomi associati sono i polinomi ortogonali di Legendre $P_n(x)$. Il peso i -esimo associato al nodo Gaussiano x_i è dato da

$$w_i = \frac{2}{(1 - x_i^2) [P'_n(x_i)]^2}$$

Analogamente alle regole di Newton-Cotes, l'errore teorico del metodo della quadratura Gaussiana è

$$E(x) = \frac{1}{n!} f^{(n)}(\xi(x)) \prod_{i=1}^n (x - x_i)$$

che può tuttavia essere ridotto a

$$E(x) = \frac{1}{(2n)!} f^{(2n)}(\xi(x)) \prod_{i=1}^n (x - x_i)^2$$

semplicemente considerando due volte ogni punto di interpolazione x_i .

Infine, se si vuole calcolare l'integrale su $[a, b]$ invece che sull'intervallo $[-1, 1]$, si deve effettuare il cambio di variabile

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right) dx$$

ed applicando il metodo di Gauss si ottiene

$$\int_a^b f(x) dx \approx \frac{b-a}{2} \sum_{i=1}^n w_i f\left(\frac{b-a}{2}x_i + \frac{a+b}{2}\right)$$

I nodi ed i pesi per il polinomio di Legendre di quinto grado sono riportati in tabella:

x_i	w_i
0	$\frac{128}{255}$
$\pm \frac{1}{3} \sqrt{5 - 2\sqrt{10/7}}$	$\frac{322+13\sqrt{70}}{900}$
$\pm \frac{1}{3} \sqrt{5 + 2\sqrt{10/7}}$	$\frac{322-13\sqrt{70}}{900}$

1.2.1 Integrazione Composta

Una tecnica utile a migliorare la precisione del calcolo dell'integrale numerico consiste nello spezzare l'intervallo d'integrazione in n sottointervalli

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} \int_{\frac{b-a}{n}i}^{\frac{b-a}{n}(i+1)} f(x)dx$$

Dopodiché si applica uno dei metodi di quadratura numerica descritto ad ogni intervallino ed infine si somma ad ottenere il risultato voluto.

1.3 Monte Carlo

L'integrazione Monte Carlo, a differenza dei metodi di quadratura precedentemente descritti, fa uso di sampling casuali e per questo motivo rientra nella categoria dei metodi non deterministici.

Nella sua versione più semplice l'algoritmo consiste nell'estrarre uniformemente punti dalla regione di integrazione per stimare l'integrale ed il relativo errore. Si supponga che il sample sia costituito da N punti x_1, \dots, x_N appartenenti alla regione di integrazione di misura V , allora la stima dell'integrale è data da

$$I \approx E_N \equiv V \frac{1}{N} \sum_{i=1}^N f(x_i) = V \langle f \rangle$$

Poiché $\{x_i\}$ è una sequenza di punti equidistribuiti in V , si può dimostrare che $I = \lim_{N \rightarrow \infty} E_N$. Tenendo presente che la varianza della funzione integranda è

$$\text{Var}(f) \equiv \sigma_N^2 = \frac{1}{N-1} \sum_{i=1}^N (f(x_i) - \langle f \rangle)^2$$

la varianza di E_N è quindi

$$\text{Var}(E_N) = \frac{V^2}{N^2} \sum_{i=1}^N \text{Var}(f) = V^2 \frac{\text{Var}(f)}{N} = V^2 \frac{\sigma_N^2}{N}$$

Dal momento che le considerazioni appena fatte rimangono vere anche nel caso multidimensionale, quello che si deduce è che l'errore sulla stima dell'integrale scala come $1/\sqrt{N}$, indipendentemente dal numero di dimensioni.

1.3.1 Campionamento di Importanza

Dal punto di vista matematico, il campionamento di importanza corrisponde al cambio di variabile

$$\int f(x) dx = \int \frac{f(x)}{p(x)} p(x) dx = \int \frac{f(x)}{p(x)} dP(x)$$

con

$$p(x) = \frac{\partial^d}{\partial x_1 \dots \partial x_d} P(x)$$

Se si restringe $p(x)$ ad essere una funzione a valori non negativi normalizzata all'unità, allora si può interpretare $p(x)$ come una densità di probabilità. Se poi si ha a disposizione un generatore di numeri casuali corrispondente alla distribuzione $P(x)$ si può anche stimare l'integrale da un sample x_1, \dots, x_N di numeri casuali distribuiti secondo $P(x)$

$$E_N = \frac{1}{N} \sum_{n=1}^N \frac{f(x_n)}{p(x_n)}$$

L'errore statistico dell'integrazione Monte Carlo è dato da $\sigma(f/p)/\sqrt{N}$.

Il campionamento di importanza è efficace se si sceglie $p(x)$ tale che approssimi bene $|f(x)|$ e tale che si sia capaci di generare numeri casuali con distribuzione di probabilità $P(x)$.

1.4 Esempi

Si dimostrano i metodi descritti nei paragrafi precedenti applicandoli al calcolo dell'integrale

$$\int_1^2 \log(1+x) dx$$

In figura sono rappresentati gli errori rispetto al risultato analitico dei quattro metodi analizzati in funzione del numero di sottointervalli utilizzati per i metodi di quadratura numerica o del numero di iterazioni per il metodo Monte Carlo.

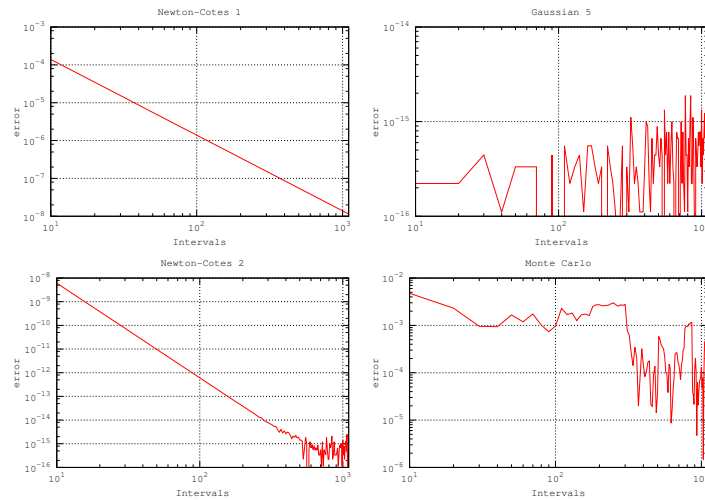


Figura 1.1: Errore calcolato

Dei metodi di quadratura l'algoritmo di Gauss-Legendre di grado 5 è il più preciso, con errori dell'ordine di 10^{-15} anche per n piccolo. Al di sotto di 10^{-15} l'errore diventa instabile a causa della precisione finita del calcolatore. La precisione delle formule di Newton Cotes invece è bassa per n piccolo ma aumenta all'aumentare di n e del grado dell'approssimazione.

Come previsto, il metodo Monte Carlo risulta il più impreciso, con un errore di molti ordini di grandezza superiore a quello dei metodi di quadratura. Il Monte Carlo è infatti impiegato prevalentemente in quei casi in cui la dimensione dell'insieme di integrazione è troppo grande per poter impiegare algoritmi di tipo deterministico.

La distribuzione dell'errore MC in funzione del numero di samples, calcolata sull'integrale

$$\int_{\Omega} d\mu = \frac{\pi}{4}$$

dove Ω indica il primo quarto di circonferenza goniometrica, verifica la relazione $\sigma_{MC} = 1/\sqrt{N}$ (figura 1.2).

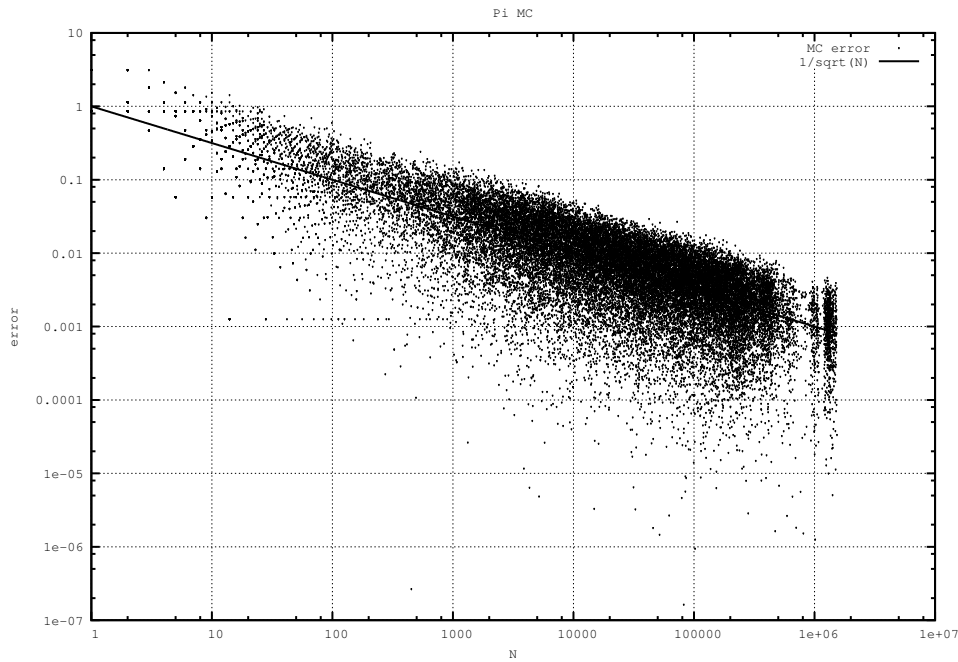


Figura 1.2: Errore MC in funzione del numero di samples

Capitolo 2

Algoritmo Metropolis

Il Metropolis è l'algoritmo più influenziale fra quelli appartenenti alla classe dei metodi Monte Carlo. Supportato da una profonda teoria, questo algoritmo costituisce uno strumento fondamentale per la scienza della computazione.

In questa sezione si propone di sviluppare un algoritmo Metropolis per simulare un oscillatore armonico quantistico e confrontare i risultati numerici con la teoria.

Il metodo Metropolis nasce dalla necessità pratica di dover generare numeri casuali distribuiti con una densità di probabilità $p(x_1, \dots, x_d)$, che non necessariamente fattorizza. Sia il vettore $\phi = (x_1, \dots, x_d)$ uno stato dell'ensamble, che vogliamo generare. L'algoritmo consiste nel partire da uno stato iniziale ϕ_0 , e sostituire iterativamente uno stato vecchio con uno nuovo, in maniera tale da ottenere la distribuzione corretta nel limite di un gran numero di iterazioni. La distribuzione che viene raggiunta all'equilibrio è indipendente dallo stato iniziale ϕ_0 . Una volta che tale distribuzione è stata raggiunta, l'applicazione ripetuta dell'algoritmo mantiene il sistema all'interno dello stesso ensamble. In altre parole, la distribuzione desiderata costituisce l'unico punto fisso dell'algoritmo.

Due condizioni importanti devono essere soddisfatte per poter applicare il Metropolis: Ergodicità e bilancio dettagliato. La condizione di bilancio dettagliato afferma che le probabilità di transizione $W(\phi_1 \rightarrow \phi_2)$ e $W(\phi_2 \rightarrow \phi_1)$ rispettano l'equazione

$$p(\phi_1)W(\phi_1 \rightarrow \phi_2) = p(\phi_2)W(\phi_2 \rightarrow \phi_1)$$

L'ergodicità invece richiede che ogni stato dell'ensamble possa essere raggiunto in un numero finito di steps.

Dato uno stato iniziale ϕ_1 , un'iterazione dell'algoritmo consiste delle seguenti istruzioni:

- generare casualmente un nuovo candidato ϕ'

- calcolare $\Delta S = -\log(p(\phi')/p(\phi_1))$
- se $\Delta S < 0$ impostare il nuovo stato $\phi_2 = \phi'$
- se $\Delta S > 0$ accettare il nuovo candidato con probabilità $p(\phi')/p(\phi_1)$, altrimenti mantenere ϕ_1
- passare all'iterazione successiva

L'algoritmo Metropolis tuttavia presenta alcuni svantaggi, tra i quali il fatto che quando si raggiunge la distribuzione d'equilibrio, gli stati successivi sono in genere molto correlati. Se si vuole ottenere un unbiased sample di stati ϕ_i si possono quindi trascurare un numero τ_d di stati prima di passare a quello successivo. Il numero di steps τ_d viene denominato tempo di decorrelazione ed è dell'ordine di ξ^2 , dove ξ è una tipica distanza di correlazione del sistema. Ciò è dovuto al fatto che il Metropolis aggiorna gli stati localmente in maniera casuale. L'algoritmo consiste quindi nell'eseguire un random walk attraverso lo spazio delle configurazioni, il che richiede ξ^2 passaggi per effettuare uno spostamento di una distanza ξ in una certa direzione.

2.1 Oscillatore Armonico

Il sistema è costituito da una particella che si muove in uno spazio unidimensionale e in un reticolo temporale finito di passo a e lunghezza N con condizioni di periodicità al contorno. Le variabili del sistema sono quindi le coordinate x_t della particella ai vari istanti di tempo t , con la condizione $x_N \equiv x_0$. La particella interagisce inoltre con un potenziale armonico della forma $V(x) = \frac{m}{2}\omega^2 x^2$.

In meccanica quantistica la stima di un osservabile può essere estratta dalle funzioni di correlazione fra gli stati del sistema. Si calcola quindi il correlatore delle variabili l -esima e k -esima $\langle x_l x_k \rangle$, definito dalla formula

$$\langle x_l x_k \rangle = \frac{\int Dx x_l x_k e^{-S_E}}{\int Dx e^{-S_E}}$$

Calcolare il correlatore significa calcolare un *path integral* usando come ampiezza di probabilità il fattore e^{-S_E} . Il ruolo dell'algoritmo Metropolis è dunque quello di generare configurazioni del sistema con distribuzione di ampiezza di probabilità e^{-S_E} .

Dopo aver ottenuto una stima per il correlatore delle variabili x_l e x_k per ogni combinazione di (l, k) , si ricavano infine i valori di $\Delta E = (\tilde{E}_1 - \tilde{E}_0)$ e dell'elemento di matrice $W = \langle \tilde{E}_0 | \hat{x} | \tilde{E}_1 \rangle$ attraverso la relazione

$$\langle x_l x_k \rangle = 2|\langle \tilde{E}_0 | \hat{x} | \tilde{E}_1 \rangle|^2 \exp\left(-\frac{Na}{2}(\tilde{E}_1 - \tilde{E}_0)\right) \cosh\left[a\left(\frac{N}{2} - |l - k|\right)(\tilde{E}_0 - \tilde{E}_1)\right].$$

2.1.1 Azione e Termalizzazione

Partendo da uno stato ϕ_0 casuale, è necessario lasciare del tempo affinché sistema si porti alla *pdf* di equilibrio, dove, per l'ipotesi di ergodicità, tutti gli stati accessibili sono equiprobabili. Tale processo, denominato termalizzazione, richiede in questo caso non più di 500 cicli di Metropolis, eseguiti i quali sarà possibile estrarre le configurazioni con la distribuzione desiderata. Come condizione iniziale si possono inizializzare tutte le variabili a zero (cold start) oppure a valori random (hot start).

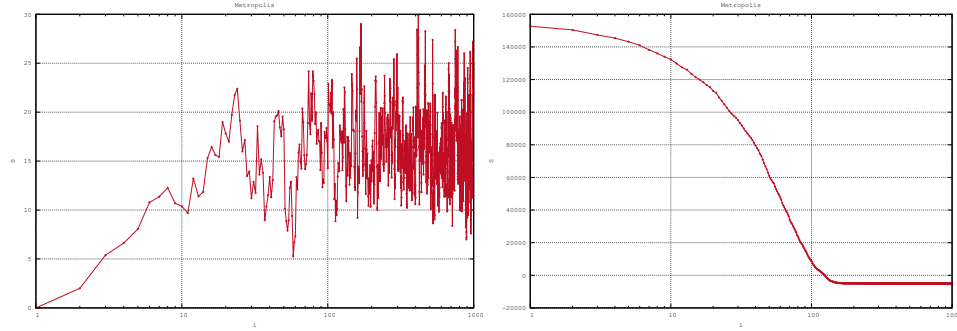


Figura 2.1: Cold start e hot start

Per il calcolo dell'azione euclidea si usa la formula

$$S = a \sum_{i=0}^{N-1} \mathcal{L}(x_i, x_{i+1})$$

dove

$$\mathcal{L}(x_i, x_{i+1}) = \frac{m}{2} \left(\frac{x_i - x_{i+1}}{a} \right)^2 - \frac{1}{2} V(x_i) - \frac{1}{2} V(x_{i+1})$$

Tuttavia, per calcolare $\Delta S = S' - S$ si può tenere presente il fatto che ad ogni estrazione solo una variabile di sistema viene modificata e quindi tutti i termini delle due sommatorie che non contengono quella variabile si elidono nella differenza. Pertanto si utilizza la formula ridotta

$$\begin{aligned} \Delta S_i &= a[\mathcal{L}(x_{i-1}, x'_i) + \mathcal{L}(x'_i, x_{i+1}) - \mathcal{L}(x_{i-1}, x_i) - \mathcal{L}(x_i, x_{i+1})] \\ &= \frac{m}{a} [(x_i - x'_i)(x_{i+1} + x_{i-1}) + (x_i'^2 - x_i^2)] + aV(x'_i) - aV(x_i) \end{aligned}$$

È facile verificare che l'algoritmo e i risultati possono essere resi indipendenti dal fattore di reticolo a se si ridefiniscono propriamente i parametri del problema, il che è equivalente a porre $a = 1$.

2.1.2 Autocorrelazione

Si studia ora la correlazione tra configurazioni generate a step consecutivi dell'algoritmo per ottenere una stima del tempo di decorrelazione τ_d del

sistema. Tale dipendenza fra elementi della catena viene anche denominata *autocorrelazione*.

La formula per il correlatore di due stime dell'osservabile O calcolate da stati estratti a istanti di tempo distanti k è

$$R(k) = \frac{\langle O_i O_{i+k} \rangle - \langle O_i \rangle^2}{\langle O_i^2 \rangle - \langle O_i \rangle^2} = \frac{\frac{\sum_{i=0}^{n-k} O_i O_{i+k}}{n-k} - \langle O_i \rangle^2}{\langle O_i^2 \rangle - \langle O_i \rangle^2}$$

Il grafico dell'autocorrelazione $R(k)$ per l'osservabile $O(|l-k|) = \langle x_l x_k \rangle$ evidenzia la perdita esponenziale di correlazione nel segnale.

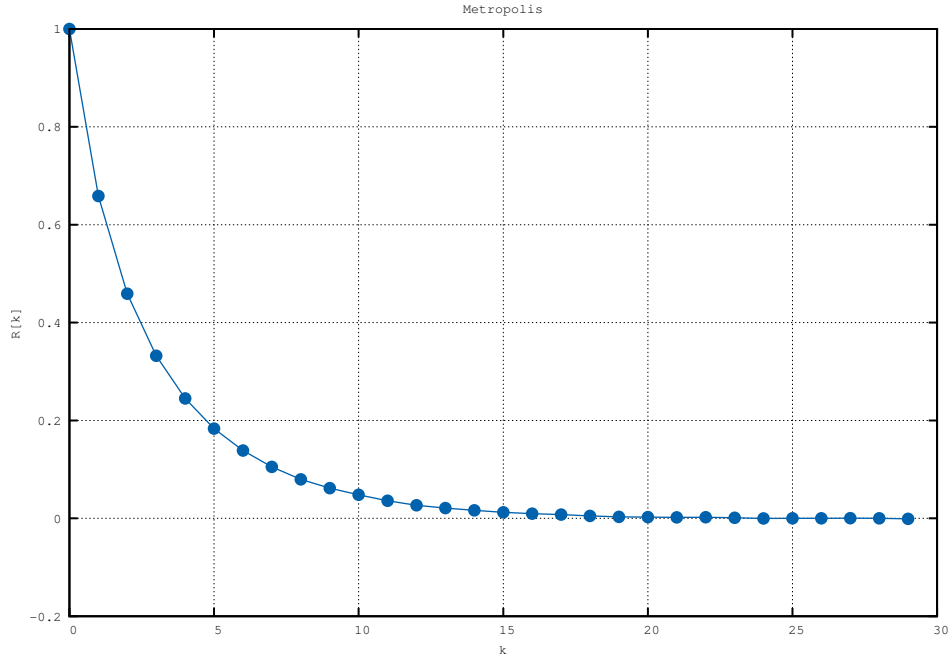


Figura 2.2: Autocorrelazione dell'osservabile $\langle x_l x_k \rangle$ per $|l-k| = 1$

Il tempo di decorrelazione τ_d è dell'ordine delle unità (~ 3) e quindi, per $k > 5\tau_d$ il segnale può essere considerato praticamente scorrelato.

2.1.3 Funzioni di Correlazione

Si calcola numericamente l'integrale $\int Dx x_l x_k e^{-S_E}$ applicando i metodi Monte Carlo e Metropolis in congiunzione. Affinchè le configurazioni utilizzate siano statisticamente indipendenti, si raggruppano i sample in *bin* (*clusterization*) e si utilizza l'insieme delle medie su ogni *bin* come ensemble di configurazioni per il calcolo MC. In questo modo, se la dimensione dei *bin* è molto maggiore di τ_d , la media sul *bin* attenua considerevolmente

gli effetti di correlazione e perciò il nuovo ensamble può essere considerato statisticamente libero.

La media di un generico osservabile O calcolata in questo modo è analiticamente uguale a quella calcolata per mezzo di ogni sample di ogni bin , tuttavia, per calcolare la varianza, bisogna ricorrere alla tecnica del *cluster jackknife* descritta di seguito.

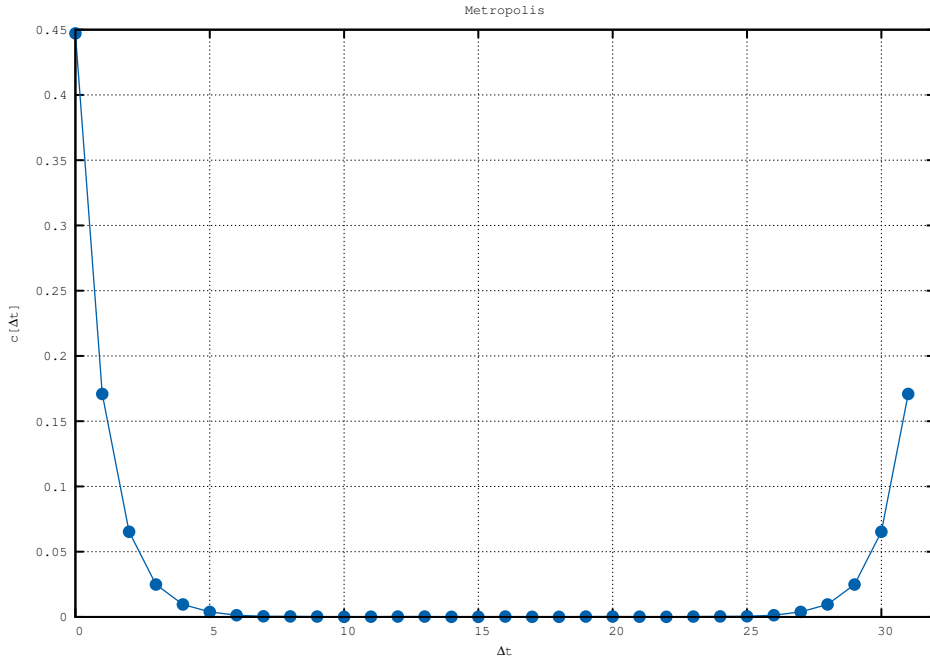


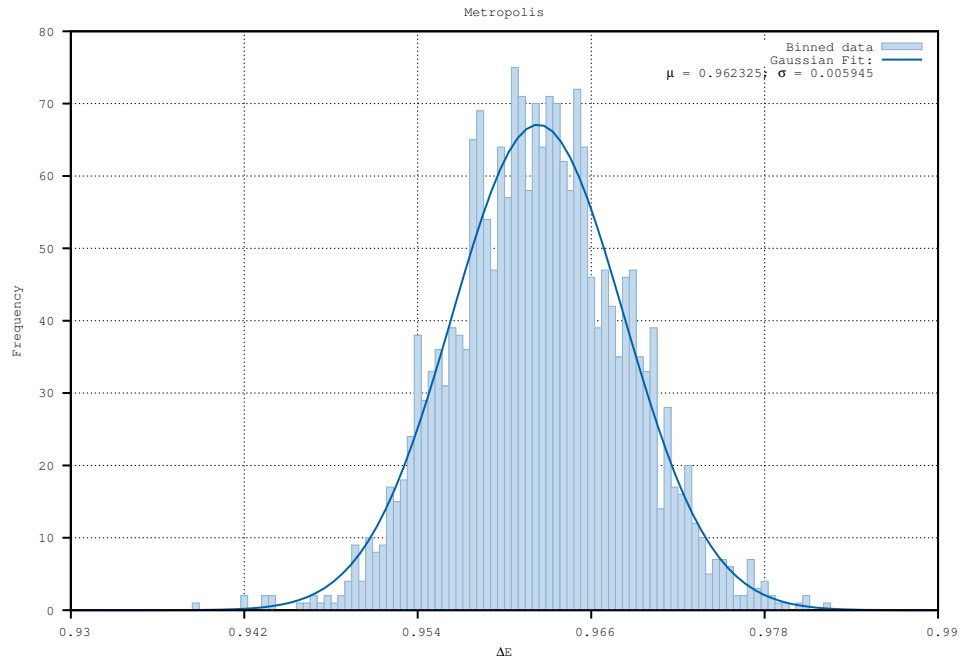
Figura 2.3: L'andamento del correlatore delle coordinate in funzione della loro separazione nel reticolo temporale è quello di un coseno iperbolico centrato in $|l - k| = \frac{N}{2}$.

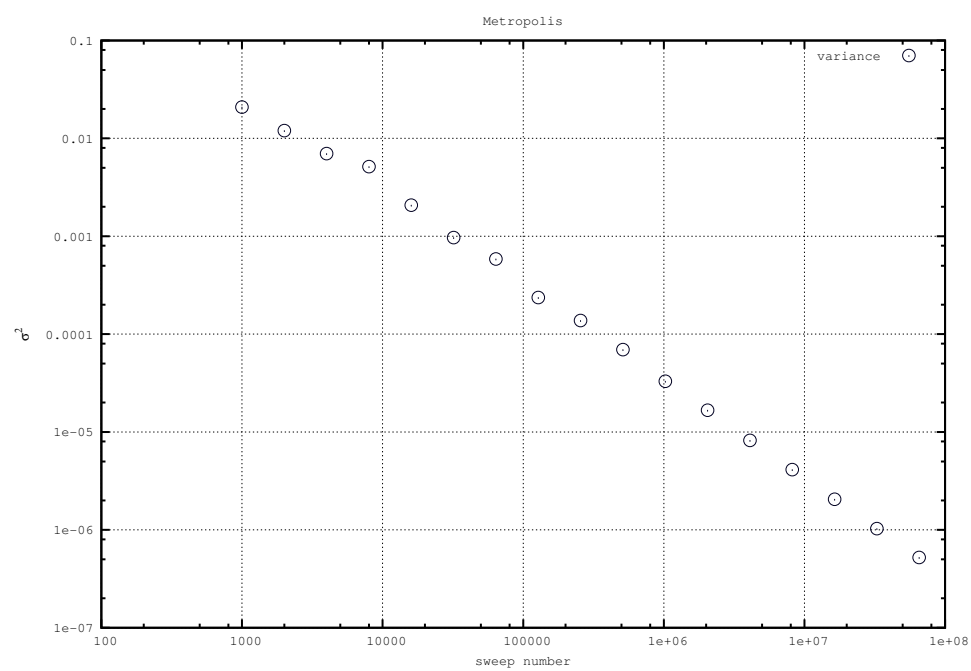
Si definisce o_i la stima semplice (calcolata su una singola estrazione) dell'osservabile O allo step i . La tecnica *cluster jackknife* consiste nel ricavare le grandezze derivate o^i definite come

$$o^i = \langle o \rangle + \frac{\langle o \rangle - o_i}{N_{bin} - 1}$$

dove N_{bin} indica il numero totale di bin .

2.1.4 Calcolo di Grandezze Quanto-Meccaniche Secondarie

Figura 2.4: Distribuzione di ΔE

Figura 2.5: Varianza di ΔE

Capitolo 3

Metodo Runge-Kutta

In analisi numerica i metodi Runge-Kutta sono una famiglia di metodi iterativi impliciti ed espliciti per la risoluzione approssimata di equazioni differenziali ordinarie (ODE). Il più comune di questi metodi è il cosiddetto “RK4” o anche Runge-Kutta del quarto ordine.

Sia dato il problema di Cauchy

$$\dot{y} = f(t, y), \quad y(t_0) = y_0.$$

Si assume che il tempo sia discretizzato in istanti t_n equidistanziati di un intervallo h .

Il metodo RK4 per questo problema è allora dato dalle seguenti equazioni:

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4) \\ t_{n+1} &= t_n + h \end{aligned}$$

dove y_{n+1} è l'approssimazione RK4 di $y(t_{n+1})$, e

$$\begin{aligned} k_1 &= hf(t_n, y_n), \\ k_2 &= hf(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1), \\ k_3 &= hf(t_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2), \\ k_4 &= hf(t_n + h, y_n + k_3). \end{aligned}$$

Il valore della funzione y all'istante t_{n+1} è uguale quindi al suo valore all'istante t_n incrementato della media ponderata di quattro incrementi k , dove ogni incremento è il prodotto della dimensione dell'intervallo h ed una stimata pendenza specificata dalla funzione f .

- k_1 è l'incremento basato sulla pendenza di f all'estremo sinistro dell'intervallo, calcolato in y_n (metodo di Eulero);
- k_2 è l'incremento basato sulla pendenza nel punto medio dell'intervallo, calcolato in $y_n + \frac{1}{2}k_1$;

- k_3 è ancora l'incremento basato sulla pendenza nel punto medio, calcolato però in $y_n + \frac{1}{2}k_2$;
- k_4 è l'incremento basato sulla pendenza all'estremo destro dell'intervallo, calcolato in $y_n + k_3$.

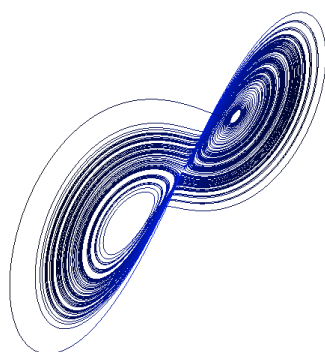
Si nota dalla formula per y_{n+1} che peso maggiore viene assegnato all'incremento al centro dell'intervallo. Inoltre, se $f = f(t)$ cioè non dipende da y , il metodo RK4 si riduce alla regola di integrazione di Simpson.

RK4 è un metodo del quarto ordine e quindi l'errore ad ogni step è dell'ordine di h^5 , mentre l'errore totale accumulato è dell'ordine di h^4 .

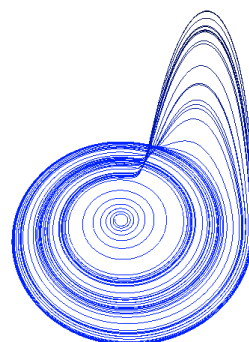
3.1 Sistemi Dinamici

Per mezzo del metodo RK4 è possibile simulare l'evoluzione di sistemi dinamici, in particolare di quelli non lineari, le cui soluzioni non sono note analiticamente.

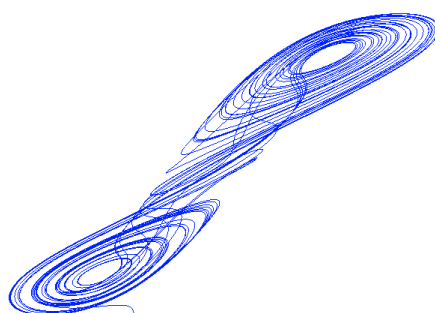
Esempi di sistemi dinamici che presentano comportamenti non lineari sono i sistemi di Lorenz, Rossler, Chua e di Rabinovich-Fabrikant.



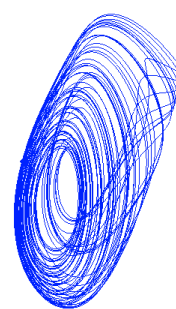
(a) Lorenz



(b) Rossler



(c) Chua



(d) Rabinovich-Fabrikant

Figura 3.1: Soluzioni numeriche di sistemi dinamici non lineari

Capitolo 4

Metodo Implicito

Per sistemi di equazioni alle derivate parziali (PDE) esistono altri metodi oltre a quelli RK. In questa sezione si illustra e implementa il cosiddetto metodo implicito per risolvere equazioni agli operatori lineari.

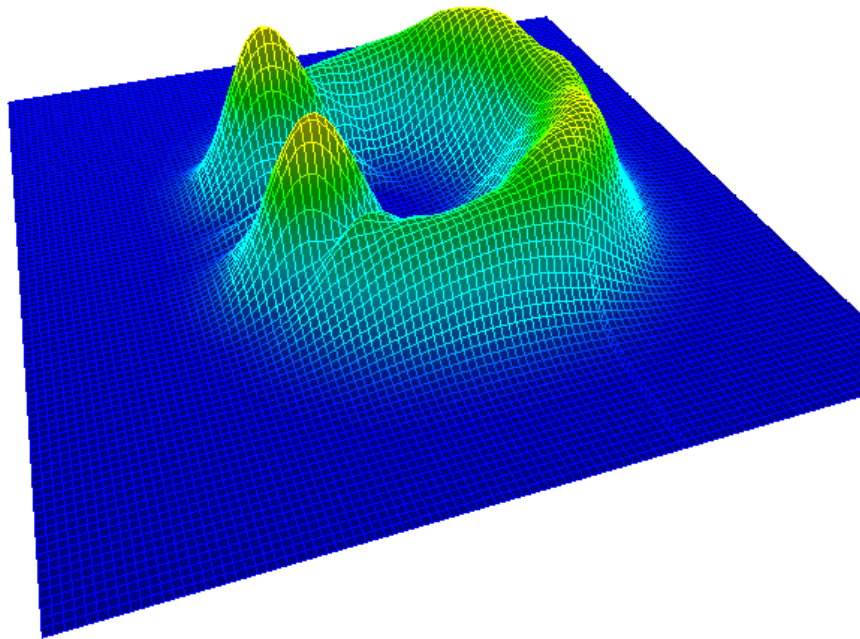


Figura 4.1: Scattering di un pacchetto gaussiano con una buca di potenziale

4.1 Equazione di Schrödinger

L'equazione che si intende risolvere è l'equazione di Schrödinger dipendente dal tempo

$$i\hbar \frac{\partial}{\partial t} \psi(x, t) = -\frac{\hbar^2}{2m} \nabla^2 \psi(x, t) + V(x) \psi(x, t)$$

A tal fine si discretizza lo spazio in modo tale che $x = na$ con $n = 0, \dots, N-1$ ed a passo del reticolo spaziale. $\psi(x, t)$ diventa quindi un vettore di valori complessi $\psi_n(t)$. Si pongono inoltre condizioni di periodicità al contorno di modo che $\psi_N \equiv \psi_0$.

Con questa discretizzazione si sostituisce

$$\nabla^2 \psi(x) \rightarrow \frac{\psi_{n+1} + \psi_{n-1} - 2\psi_n}{a^2}$$

pertanto l'equazione diventa

$$i\hbar \frac{\partial}{\partial t} \psi_n = -\frac{\hbar^2}{2m} \frac{\psi_{n+1} + \psi_{n-1} - 2\psi_n}{a^2} + V_n \psi_n = \sum_m A_{n,m} \psi_m$$

La soluzione dell'equazione è la funzione esponenziale

$$\psi(t) = \exp\left(-\frac{i}{\hbar} A t\right) \psi(0)$$

dove $U(t) = \exp\left(-\frac{i}{\hbar} A t\right)$ rappresenta l'evolutore temporale del sistema. Sviluppando ora $U(t)$ intorno all'unità per $t \rightarrow 0$ e trascurando gli ordini superiori in t si ottiene la formula approssimata

$$\psi(t_0 + t) = \psi(t_0) - \frac{i}{\hbar} A t \psi(t_0)$$

Questa prima approssimazione tuttavia porta all'insorgere di numerosi problemi dovuti al fatto che l'iterazione del precedente passaggio non rispetta la condizione di unitarietà dell'operatore evolutore temporale. Bisogna quindi ricorrere ad una strategia più astuta.

Il **metodo implicito** consiste nell'approssimare l'operatore esatto $U(t)$ attraverso la formula

$$\psi(t + dt) = \left(1 + \frac{i}{2\hbar} A dt\right)^{-1} \left(1 - \frac{i}{2\hbar} A dt\right) \psi(t)$$

Utilizzando il fatto che A è Hermitiano, è facile dimostrare che l'operatore della precedente formula è unitario proprio come $U(t)$. Per implementare questo algoritmo è necessario risolvere ad ogni step il sistema lineare

$$\left(1 + \frac{i}{2\hbar} A dt\right) \psi(t + dt) = \left(1 - \frac{i}{2\hbar} A dt\right) \psi(t)$$

Infine si moltiplicano entrambi i membri dell'equazione per l'aggiunto dell'operatore a primo membro, in modo tale da ottenere

$$(1 + \frac{1}{4\hbar^2} A^2 dt^2) \psi(t + dt) = (1 - \frac{i}{2\hbar} A dt)^2 \psi(t)$$

dove ora $(1 + \frac{1}{4\hbar^2} A^2 dt^2)$ è Hermitiano e si può quindi risolvere rispetto a $\psi(t + dt)$ utilizzando il *Conjugate Residual Method*.

L'idea che sta alla base di questo metodo è che se in un intervallo di tempo dt la funzione $\psi(t)$ evolve in $\psi(t + dt)$, allora facendo evolvere $\psi(t)$ in avanti nel tempo per $\frac{dt}{2}$ e $\psi(t + dt)$ indietro nel tempo sempre di $\frac{dt}{2}$ le due funzioni dovranno coincidere.

Bibliografia

- [1] M. Abramowitz, I. A. Stegun. “§25.4, Integration”, *Handbook of Mathematical Functions (with Formulas, Graphs, and Mathematical Tables)*, 1972
- [2] M. Hjorth-Jensen. “§9, Random walks and the Metropolis algorithm”, *Computational Physics*, 2009
- [3] Y. Saad. “§6.8, The Conjugate Residual Method”, *Iterative Methods for Sparse Linear Systems*, 2003
- [4] I. Beichl, F. Sullivan. *Computing in Science and Engineering*, 2000
- [5] C. Rebbi. *Lecture Notes on Advanced Computing in Physics*, 1996
- [6] S. Weinzierl. *Introduction to Monte Carlo methods*, 2000
- [7] M. Creutz, B. Freedman. *A Statistical Approach to Quantum Mechanics*, 1980
- [8] C. Morningstar. *The Monte Carlo Method in Quantum Mechanics*, 2004