Corso di Laurea in Fisica
Prova di esame - Laboratorio di Calcolo e Statistica

1 settembre 2025

Indicazioni generali

Si risolva il seguente esercizio, scrivendo un programma in `C++` o in `Python` ed organizzando il codice sorgente in modo che le funzioni utilizzate risultino implementate in librerie separate del programma principale. Ai fini della valutazione, il primo criterio che deve essere soddisfatto è che il codice sia eseguibile senza errori (inclusi quelli di compilazione, nel caso del `C++`) realizzando le funzionalità richieste dal testo. Per la valutazione sarà inoltre tenuto in considerazione il fatto che i codici sorgente siano scritti con ordine, utilizzando opportunamente l'**indentazione** e i **commenti**. Per gli svolgimenti in `C++`, si richiede infine di iniziare i codici con una riga di commento contenente il comando necessario per creare l'eseguibile.

L'algoritmo di Metropolis

Arianna Rosenbluth e Nicholas Metropolis, due fisici che hanno lavorato ai primi sviluppi della fisica nucleare, a metà del secolo scorso parteciparono alla scrittura di un algoritmo di generazione di numeri pseudo-casuali ad alta efficienza, molto utile in particolare nel caso di distribuzioni casuali $f(\underline{x})$ definite su domini multi-dimensionali ($\underline{x} \in \mathbb{R}^n$). A differenza di numeri realmente casuali, i numeri generati sono matematicamente correlati a primi vicini, perché ogni numero $\underline{x}_{k+1}$ si calcola sulla base di quello precedente $\underline{x}_k$ in questo modo:

a. si genera un numero pseudo-casuale $\underline{x}'$ distribuito uniformemente in un intorno di $\underline{x}_k$;
b. si calcola il rapporto $\alpha = f(\underline{x}')/f(\underline{x}_k)$;
c. si genera un numero pseudo-casuale reale u distribuito uniformemente fra 0 e 1;
d. se $\alpha > u$, si aggiunge $\underline{x}'$ alla sequenza generata (cioè $\underline{x}_{k+1} = \underline{x}'$).

Reiterando questa procedura si genera un campione di punti $\{\underline{x}_k\}$ distribuiti secondo la distribuzione di densità di probabilità multi-dimensionale $f(\underline{x})$.

1. Si prepari una libreria che contenga una funzione che calcola una densità di probabilità bidimensionale $g(x, y)$ definita in $\mathbb{R}^2$ come il prodotto di due Gaussiane monodimensionali, ciascuna con la propria media e sigma:

$$\mu_x = 1 \quad \sigma_x = 0.5$$
$$\mu_y = 2 \quad \sigma_y = 1.2$$

2. Si implementi l'algoritmo di Metropolis seguendo il ciclo (a, ..., d) descritto sopra e lo si utilizzi in un programma principale per generare una collezione di 10,000 eventi pseudo-casuali distribuiti secondo $g(x, y)$.
3. Si disegni l'istogramma della distribuzione monodimensionale delle coordinate x e y separatamente.
4. Si esegua il fit gaussiano di ciascuna distribuzione utilizzando `iMinuit` e si verifichi la compatiblità di media e sigma ottenute dal fit con quelle scelte al punto 1.
5. Si calcoli la covarianza del campione generato: il valore ottenuto corrisponde alle aspettative?

Nello svolgimento del punto 2, i punti casuali $\underline{x}'$ andranno generati secondo una distribuzione uniforme multi-dimensionale, cioè ogni coordinata dovrà essere generata in modo uniforme. Si consiglia di scegliere la dimensione delle distribuzioni uniformi utilizzate proporzionalmente alle sigma della distribuzione $g(x, y)$.

Gli studenti affetti da disturbi specifici dell'apprendimento (DSA) potranno tralasciare i punti 3 e 5 o, alternativamente, svolgere tutto il compito in 4 ore di tempo, dichiarando la propria preferenza all'inizio della prova. Questi dovranno anche consegnare, oltre allo svolgimento del tema, una copia del proprio Progetto Universitario Individualizzato (P.Uo.I).