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

16 settembre 2024

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.

Numeri quasi casuali

L'ottimizzazione dell'integrazione numerica con il metodo Monte Carlo si ottiene, fra le altre cose, con una scelta oculata delle coordinate x dei punti generati casualmente. Infatti, più essi ricoprono in maniera ottimale l'insieme di definizione della funzione da integrare, migliore è la precisione ottenuta nella sua stima, a parità di punti generati. La sequenza s_n generata secondo il seguente algoritmo:

$$s_{n+1} = (s_n + \alpha) \bmod 1 \tag{1}$$

produce un insieme di punti, distribuiti fra 0 ed 1, che hanno la proprietà di ben riempire questo insieme di definizione, in particolare se $\alpha = (\sqrt{5} - 1)/2$.

1. Si scriva una libreria che contenga una classe di python, chiamata `additive_recurrence`, che generi la sequenza di numeri s_n dell'equazione (1), che abbia come variabili membro il parametro α, il numero di partenza della sequenza e l'ultimo numero generato, che assegni un valore ad α durante l'inizializzazione della classe ed implementi i metodi seguenti:
   - `get_number` per ottenere un numero della sequenza
   - `set_seed` per inizializzare la sequenza
2. Si faccia un test del funzionamento della classe generando una sequenza di 1000 numeri e scrivendone i primi 10 a schermo.
3. Si aggiunga alla libreria una funzione chiamata `MC_mod` che calcoli l'integrale definito di $f(x) = 2x^2$ nell'intervallo $(0, 1)$, utilizzando il metodo *hit or miss* dove la generazione dei punti lungo l'asse x non sia fatta in modo pseudo-casuale, ma utilizzando la classe `additive_recurrence`.
4. Utilizzando il metodo dei *toy experiment*, si determini l'incertezza del calcolo dell'integrale in funzione del numero totale `N_points` di punti generati per la stima di un singolo integrale, disegnandone l'andamento dell'errore in funzione di `N_points` al variare fra 10 e 25000.
5. Si rifaccia il medesimo test con l'algoritmo *hit or miss* studiato a lezione e si confrontino i due risultati: quale è più efficiente?

Gli studenti affetti da disturbi specifici dell'apprendimento (DSA) potranno tralasciare il punto 5. Questi dovranno anche consegnare, oltre allo svolgimento del tema, una copia del proprio Progetto Universitario Individualizzato (P.Uo.I).