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

13 gennaio 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.

Una Gaussiana asimmetrica

Per poter riprodurre dati sperimentali con asimmetria non nulla, si può utilizzare un modello che generalizza la distribuzione Gaussiana, con le code a sinistra e destra del massimo caratterizzate da parametri σ diversi.

1. Si scriva una libreria di Python che contenga l'implementazione di una distribuzione di densità di probabilità, chiamata `double_Gaus`, definita sull'asse reale. La funzione dovrà possedere un massimo e due code Gaussiane, a destra e sinistra del massimo rispettivamente, con sigma differenti (σ_{sx} e σ_{dx}), ricordando che la funzione deve essere continua su tutto l'asse reale.
2. Si scriva un programma in Python che ne faccia un disegno per controllare il risultato.
3. Si controlli che la funzione è normalizzata, utilizzando il metodo di integrazione *hit-or-miss*.
4. Si generi un campione di 1000 punti pseudo-casuali distribuiti secondo la pdf `double_Gaus` utilizzando il metodo *try-and-catch*, se ne disegni l'istogramma scegliendone con un algoritmo appropriato minimo, massimo e numero di bin e si stampino a schermo la media e la mediana del campione ottenuto.
5. Si assuma che $\sigma_{\text{sx}} < \sigma_{\text{dx}}$ e si trovi una formula che ricavi il rapporto $\sigma_{\text{dx}}/\sigma_{\text{sx}}$ a partire dalla differenza fra media e mediana del campione, utilizzando il metodo dei *toy experiment* per determinarla empiricamente. Si provi ad utilizzare un *fit* per ottenere il risultato: assumendo che l'incertezza sulla mediana sia uguale a quella sulla media, quale variabile viene utilizzata come indipendente? Come si può quantificare l'affidabilità della formula?

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).

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

10 ottobre 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.

Il metodo di Box-Mueller per generare numeri pseudo-casuali Gaussiani

Secondo l'algoritmo di Box-Müller, dati due numeri pseudo-casuali x_1 ed x_2 generati uniformemente nell'intervallo $(0, 1)$, si dimostra che i due numeri g_1 e g_2 calcolati con le equazioni seguenti:

$$g_1 = \sqrt{-2\log(x_1)}\ \cos\left(2\pi x_2\right) \tag{1}$$

$$g_2 = \sqrt{-2\log(x_1)}\ \sin\left(2\pi x_2\right) \tag{2}$$

possano essere considerati due numeri pseudo-casuali distribuiti secondo una distribuzione di densità di probabilità normale.

1. Si scriva una funzione chiamata `generate_gaus_bm` che generi coppie di numeri pseudo-casuali distribuiti secondo una densità di probabilità Gaussiana utilizzando l'algoritmo di Box-Müller, implementata in una libreria dedicata.
2. Si generino $N = 1000$ numeri pseudo-casuali utilizzando la funzione appena sviluppata e li si disegni in un istogramma, scegliendone con un algoritmo opportuno gli estremi ed il binnaggio.
3. Si determinino media e varianza della distribuzione ottenuta e relativi errori.
4. Si mostri graficamente che, al variare del numero N di eventi generati, la sigma della distribuzione non cambia, mentre l'errore sulla media si riduce.
5. Si trasformi l'algoritmo in modo che generi numeri pseudo-casuali con densità di probabilità Gaussiana con media $\mu = 5$ e varianza $\sigma^2 = 4$. Si generi un nuovo campione di $N = 1000$ eventi con il nuovo algoritmo e se ne disegni la distribuzione, sempre scegliendo in modo opportuno gli estremi ed il binnaggio dell'istogramma corrispondente.

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).

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 *crude Montecarlo* 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 *crude Montecarlo* 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).

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

8 luglio 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.

Random Walk

In una torrida giornata di luglio, in un villaggio sperduto in Armorica, il druido Panoramix sbagliò ricetta ed invece della solita pozione magica produsse, nel suo calderone, la grappa più alcolica mai distillata in Gallia.

1. Si scriva una funzione che simuli il cammino degli abitanti del villaggio dopo aver bevuto la grappa, assumendo che si spostino in piano, che ogni passo abbia direzione casuale uniforme angolarmente ed una lunghezza distribuita secondo una distribuzione Gaussiana con media 1 e larghezza 0.2, troncata a valori positivi.
2. Immaginando che il calderone si trovi alle coordinate $(0, 0)$ sul piano, si scriva una funzione che calcoli la posizione (x, y) raggiunta da Asterix dopo $N = 10$ passi e si disegni il suo percorso.
3. Si consideri ora l'intera popolazione: si determini la posizione (x, y) di ogni abitante dopo $N = 10$ passi a partire dal calderone e si disegni le distribuzione della distanza raggiunta dal punto di partenza, assumendo la popolazione totale composta da 10000 persone.
4. Si determinino media, varianza, asimmetria e curtosi della distribuzione ottenuta.
5. Se la lunghezza dei passi è costante uguale ad 1, la distribuzione delle distanze r dopo N passi segue una distribuzione di Rayleigh:

   $$f(r) = \frac{2r}{N} e^{-r^2/N} \ . \tag{1}$$

   Si utilizzi un fit per determinare, a partire dalla distribuzione di distanze costruita in queste ipotesi, il numero di passi effettuati, sapendo che la distribuzione di Rayleigh è presente in `scipy` come `scipy.stats.rayleigh` e che per ottenere la forma funzionale di interesse per il problema questa distribuzione ha come parametri $\texttt{loc} = 0$ e $\texttt{scale} = \sqrt{N/2}$ (dove N è il numero di passi).

Gli studenti affetti da disturbi specifici dell'apprendimento (DSA) potranno tralasciare il punto 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).

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

24 giugno 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.

La verosimiglianza e il suo profilo

La scoperta di nuove particelle instabili, come nel caso del bosone di Higgs, si basa molto spesso sullo studio dell'istogramma della massa invariante dei loro prodotti di decadimento, per trovare un eccesso rispetto al rumore di fondo localizzato in un intorno di un valore specifico. In questo tema vi è richiesto di costruire una simulazione di questa ricerca, utilizzando un modello esponenziale di rumore di fondo ed uno Gaussiano per il segnale.

1. Si generi un campione $\{x_i\}$ di `N_exp`=2000 eventi distribuiti secondo una distribuzione di densità di probabilità esponenziale, con $\lambda = 1/200$, compresi fra 0 e 3 volte τ, ed uno $\{x_j\}$ di `N_gau`=200 eventi distribuiti secondo una distribuzione di densità di probabilità Gaussiana, con $\mu = 190$ e $\sigma = 20$.
2. Si costruisca un campione pari all'unione dei due precedenti e se ne disegni l'istogramma, scegliendo opportunamente il numero di bin.
3. Si effettui un fit del campione per determinare i parametri del modello.
4. Si costruisca una funzione che calcoli il logaritmo della verosimiglianza associata al campione, dato il seguente modello di densità di probabilità:

$$f(x) = a * \text{Exp}(x, \lambda) + b * \text{Gaus}(x, \mu, \sigma) \tag{1}$$

5. Fissati i parametri del modello al risultato ottenuto dal fit, si calcoli il valore del logaritmo della verosimiglianza per il campione dato il modello, variando il valore del parametro μ fra 30 e 300 con passo costante e se ne disegni l'andamento.
6. Si determini il massimo della funzione di verosimiglianza in funzione del parametro μ, utilizzando l'algoritmo della sezione aurea.

Gli studenti affetti da disturbi specifici dell'apprendimento (DSA) potranno tralasciare i punti 5 e 6 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).

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

19 febbraio 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.

Un Dataset da Nobel

L'espansione dell'universo fu determinata da Edwyn Hubble nel 1929 osservando, nelle galassie visibili dalla TERRA, un legame matematico fra la loro distanza D_L ed il loro *redshift* z, che è uno spostamento Doppler verso frequenze più basse del loro colore naturale [1]. Oggi queste misure vegono svolte osservando un tipo particolare di supernova, dette A1 e le misure correnti vengono utilizzate per decidere se l'universo sia destinata a continuare o se prima o poi tornerà a contrarsi, con decelerazione q. In assenza di decelerazione, l'espansione dell'universo è descritta da un andamento lineare:

$$D_L = \frac{z \cdot c}{H_0} \tag{1}$$

dove $c = 3 \cdot 10^5$ km/s è la velocità della luce nel vuoto, mentre H_0 è una costante di proporzionalità detta costante di Hubble.

1. Si legga il file `SuperNovae.txt` e si salvino i dati in tre liste (o array). La prima colonna è il redshift, la seconda la distanza e la terza l'errore sulla distanza.
2. Si faccia il grafico dei dati mettendo sull'asse x il *redshift* z e sull'asse y la distanza D_L, includendo gli errori nel grafico.
3. Si esegua un fit dei dati utilizzando il modello lineare e si stampi la costante di Hubble incluso il suo errore.
4. Si esegua il fit dei dati utilizzando un modello che preveda decelerazione dell'universo:

$$D_L = \frac{c}{H_0} \cdot \left(z + \frac{1}{2} \left(1 - q\right) z^2 \right) \tag{2}$$

   facendo un grafico con i dati ed i due modelli sovrapposti (utilizzando *ax.legend()* per identificarli), decidendo quale dei due si adatti meglio ai dati. Si determinino il valore della costante di Hubble ed il valore medio della densità dell'universo Ω_m ed il loro errore, sapendo che:

$$q = \frac{3 \cdot \Omega_m}{2} - 1$$

[1] Il *redshift* è definito come il rapporto fra lo spostamento doppler e la frequenza a riposo $z = \Delta f / f$

5. Utilizzando la generazione di numeri casuali uniformi, si generi un elenco di 30 indici nelle liste dei dati in ingresso a partire dal quali si costruisca un loro sotto-campione, sul quale rifare il fit con il modello 2.
6. Ripetendo il punto precedente 50 volte si costruisca la distribuzione attesa del parametro q e della costante di Hubble.
7. (facoltativo) Si costruisca, utiizzando la stessa tecnica del punto precedente, la distribuzione congiunta dei parametri nel piano (q, H_0).

Gli studenti affetti da disturbi specifici dell'apprendimento (DSA) potranno tralasciare i punti 5 e 6 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).

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

5 febbraio 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.

Test di bontà di fit

1. Si definisca una funzione $\varphi(x, a, b, c)$ che traccia un andamento parabolico in funzione di x e se ne disegni disegni l'andamento nell'intervallo $(0, 10)$:
$$\varphi(x, a, b, c) = a + bx + cx^2$$
con:
$$\begin{cases} a &= 3 \\ b &= 2 \\ c &= 1 \end{cases}$$
2. Si generino N = 10 punti x_i distribuiti in modo pseudo-casuale secondo una distribuzione uniforme sull'intervallo orizzontale e si associ a ciascuno di essi una coordinata
$$y_i = \varphi(x_i, a, b, c) + \varepsilon_i,$$
dove ε_i è un numero pseudo casuale generato, con il metodo del teorema centrale del limite, secondo una distribuzione Gaussiana di media 0 e deviazione standard $\sigma_y = 10$.
3. Si faccia un fit della funzione $\varphi(x, a, b, c)$ sul campione così generato (che tecnica bisogna utilizzare?).
4. Si costruisca la distribuzione del Q^2 a partire dal fit effettuato, ripetendolo molte volte utilizzando toy experiment.
5. Si svolgano i punti precedenti generando gli scarti ε_i secondo una distribuzione uniforme che abbia la stessa deviazione standard della Gaussiana, disegnando poi la distribuzione del Q^2 così ottenuto sovrapposta a quella precedente (per una visualizzazione migliore, si può utilizzare l'opzione `histtype='step'`).
6. In funzione della distribuzione ottenuta per il Q^2, si determini la soglia oltre la quale rigettare il risultato del fit, dato il suo valore di Q^2, per ottenere un p-value maggiore o uguale di 0.10.

Gli studenti affetti da disturbi specifici dell'apprendimento (DSA) potranno tralasciare il punto 6 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).

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

18 settembre 2023

Indicazioni generali

Si risolva il seguente esercizio, scrivendo un programma in `C++`. Ai fini della valutazione, il primo criterio che deve essere soddisfatto è che il codice compili senza errori ed esegua realizzando le funzionalità richieste dal testo. Per la valutazione sarà inoltre tenuto in considerazione il fatto che i codici siano scritti con ordine, utilizzando opportunamente l'**indentazione** e i **commenti**. Si richiede infine di iniziare i codici con una riga di commento contenente il comando necessario per creare l'eseguibile.

La distribuzione di Cauchy

La distribuzione di Cauchy ha la seguente forma funzionale:

$$f(x) = \frac{1}{\pi} \frac{\Gamma}{(x-M)^2 + \Gamma^2} ,$$

dove M e Γ sono i suoi due parametri notevoli.

1. Si implementi una libreria che generi punti pseudo-casuali distribuiti secondo una distribuzione di Cauchy in un intervallo simmetrico attorno ad M utilizzando il metodo *try-and-catch*, dove la funzione `f_cauchy` che produce ciascun numero casuale prenda come parametri in ingresso i due parametri, M e Γ e la semi-larghezza dell'intervallo di generazione.
2. Si scriva un programma `main.cpp` che verifichi il funzionamento della libreria generando N numeri casuali fra $(M - 3\Gamma)$ ed $(M + 3\Gamma)$, prendendo i valori di N, M e Γ come parametri in ingresso a linea di comando al momento della chiamata del programma.
3. Si accumulino questi numeri in un istogramma di tipo `TH1F` con minimo, massimo e numero di bin scelti algoritmicamente sulla base di N, M e Γ e lo si disegni in un'immagine di tipo `png`.
4. Si calcolino media e sigma di numeri pseudo-casuali generati secondo una distribuzione di Cauchy sull'intervallo $(M - i_\Gamma\Gamma,\ M + i_\Gamma\Gamma)$ al variarare di i_Γ fra 1 e 100 e si rappresenti l'andamento delle due quantità in funzione di i_Γ su due `TGraph`, disegnati in due immagini di tipo `png`.
5. Si implementi una funzione `rand_TCL_cauchy` che generi numeri pseudo-casuali con la tecnica del teorema centrale del limite a partire da numeri generati con la funzione `f_cauchy`. Si trovi un modo di verificare se i numeri generati con `rand_TCL_cauchy` siano effettivamente distribuiti secondo una Gaussiana: lo sono? Perché?

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

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

4 settembre 2023

Indicazioni generali

Si risolva il seguente esercizio, scrivendo un programma in `C++`. Ai fini della valutazione, il primo criterio che deve essere soddisfatto è che il codice compili senza errori ed esegua realizzando le funzionalità richieste dal testo. Per la valutazione sarà inoltre tenuto in considerazione il fatto che i codici siano scritti con ordine, utilizzando opportunamente l'**indentazione** e i **commenti**. Si richiede infine di iniziare i codici con una riga di commento contenente il comando necessario per creare l'eseguibile.

Incertezze statistiche e sistematiche in un fit

La tecnica di stima dei parametri basate sui minimi quadrati assume che le incertezze che affliggono le misure siano di carattere statistico, cioè che il loro valore per ogni punto di una legge $y = \varphi(x)$ sia indipendente dagli altri.

1. Si scriva un programma `main.cpp` che generi generi 6 coppie (x_i, y_i) di punti pseudo-casuali lungo l'andamento della funzione
$$f(x) = 2 \cdot \sin(0.5 \cdot x + 0.78) + 0.8$$
con x_i fissati ai seguenti valori:
$$\{x_i\}_{i=1...N} = \{0.5,\ 2.5,\ 4.5,\ 6.5,\ 8.5,\ 10.5\}$$
e ciascun y_i distribuito attorno ad $f(x_i)$ secondo una densità di probabilità Gaussiana di sigma 0.3 inserita dall'utente a riga di comando al momento della chiamata del programma. I punti siano generati con una funzione che produca i numeri pseudo-casuali Gaussiani sfruttando il teorema centrale del limite, prendendo come parametri in ingresso la media e la sigma della Gaussiana.
2. Si rappresenti il campione di punti così generato con un `TGraphErrors` e lo si disegni su un file di tipo `gif`.
3. Dopo aver definito una funzione di `ROOT` di tipo `TF1` a partire dall'espressione di $f(x)$, avente 4 parametri liberi p_i corrispondenti ai coefficienti presenti nella definizione di $f(x)$:
$$f_{fit}(x) = p_0 \cdot \sin(p_1 \cdot x + p_2) + p_3$$
si esegua il fit del `TGraphErrors` con la TF1 costruita, controllando se il fit abbia avuto successo e stampando a schermo il valore del Q^2 e del *p-value* corrispondenti. Perché il fit abbia successo potrebbe essere necessario inizializzarne qualche parametro: per fare questo si utilizzino algoritmi basati sui punti pseudo-casuali come informazione in ingresso.
4. Si esegua una nuova generazione di punti (x_i, y_i), che aggiunga al valore di y_i, oltre all'incertezza casuale ε_i generata nel punto 1 del compito, una nuova incertezza δ_i, completamente correlata fra tutti i punti. Si ipotizzi che la stima dell'incertezza dei singoli punti derivi dalla somma in quadratura dei termini ε e δ e si disegni in un'immagine di tipo `gif` il `TGraphErrors` corrispondente a questo nuovo insieme di punti.
5. Si esegua il fit del nuovo `TGraphErrors` e, dal valore di Q^2 minimo ottenuto, si deduca una stima della incertezza statistica associata alle misure; si utilizzi il metodo dei *toy experiment* per verificare il funzionamento di questa tecnica.

6. Che effetto ci si aspetta sul risultato del fit, dopo l'aggiunta del termine δ_i?

Gli studenti affetti da disturbi specifici dell'apprendimento (DSA) potranno tralasciare i punti 5 e 6. Questi ultimi dovranno anche consegnare, oltre allo svolgimento del tema, una copia del proprio Progetto Universitario Individualizzato (P.Uo.I).

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

3 luglio 2023

Indicazioni generali

Si risolva il seguente esercizio, scrivendo un programma in `C++`. Ai fini della valutazione, il primo criterio che deve essere soddisfatto è che il codice compili senza errori ed esegua realizzando le funzionalità richieste dal testo. Per la valutazione sarà inoltre tenuto in considerazione il fatto che i codici siano scritti con ordine, utilizzando opportunamente l'**indentazione** e i **commenti**. Si richiede infine di iniziare i codici con una riga di commento contenente il comando necessario per creare l'eseguibile.

Bontà di un fit

Una tecnica per la determinazione della bontà di un *fit* si basa sul calcolo del valore del *p-value* a partire dal risultato dell'interpolazione.

1. Si scriva un programma `main.cpp` che generi generi 4 coppie (x_i, y_i) di punti pseudo-casuali lungo l'andamento della funzione
$$f(x) = (x-2)^3 + 3$$
con x_i fissati ai seguenti valori:
$$\{x_i\}_{i=1...N} = \{0.5,\ 1.5,\ 2.5,\ 3.5\}$$
e ciascun y_i distribuiti attorno ad $f(x_i)$ secondo una densità di probabilità Gaussiana di sigma 0.2, utilizzando una funzione che generi i numeri pseudo-casuali Gaussiani sfruttando il teorema centrale del limite prendendo come parametri in ingresso la media e la sigma della Gaussiana.
2. Si rappresenti il campione di punti così generato con un `TGraphErrors` e lo si disegni su un file di tipo `gif`.
3. Dopo aver definito una funzione di `ROOT` di tipo `TF1` a partire dall'espressione di $f(x)$, avente due parametri liberi p_i corrispondenti alle traslazioni orizzontale e verticale presenti nella definizione di $f(x)$:
$$f_{fit}(x) = (x-p_0)^3 + p_1$$
si esegua il fit del `TGraphErrors` con la TF1 costruita, controllando se il fit abbia avuto successo e stampando a schermo il valore del Q^2 e del *p-value* corrispondenti. Si incapsuli lo svolgimento di questo punto in una funzione di `C++` dedicata.
4. Utilizzando un ciclo opportuno, si riproducano N *toy experiment* dei punti precedenti (utilizzando la funzione di `C++` sviluppata nel punto precedente) e si riempiano due istogrammi di tipo `TH1F`, contenenti rispettivamente la distribuzione del Q^2 e del *p-value* attesi dal fit.
5. Si svolga l'esercizio del punto precedente generando i punti casuali secondo la funzione $f(x)$ e fittandoli con un andamento di tipo differente:
$$g_{fit}(x) = (x-p_0)^2 + p_1$$
confrontando visivamente la distribuzione del Q^2 atteso con le due diverse ipotesi di fit (può essere utile per questo esercizio normalizzare gli istogrammi in modo che la loro altezza massima sia 1,

sapendo che si può ottenere l'altezza massima di un istogramma `histo` chiamando: `histo.GetBinContent (histo.GetMaximumBin ())` ed utilizzando il metodo `TH1F::Scale (double val)` per riscalare l'altezza di tutti i bin di un valore fissato).

6. Si determini come vari la differenza di risultati fra i due fit all'aumentare della sigma della distribuzione Gaussiana utilizzata per generare i valori y_i: come si potrebbe quantificare questa differenza?

Gli studenti affetti da disturbi specifici dell'apprendimento (DSA) potranno tralasciare il punto 5. Questi ultimi dovranno anche consegnare, oltre allo svolgimento del tema, una copia del proprio Progetto Universitario Individualizzato (P.Uo.I). Il punto 6 è facoltativo, cioè lo svolgimento corretto e completo dei soli primi cinque punti permette di raggiungere il massimo punteggio nella valutazione.

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

15 febbraio 2023

Indicazioni generali

Si risolva il seguente esercizio, scrivendo un programma in C++. Ai fini della valutazione, il primo criterio che deve essere soddisfatto è che il codice compili senza errori ed esegua realizzando le funzionalità richieste dal testo. Per la valutazione sarà inoltre tenuto in considerazione il fatto che i codici siano scritti con ordine, utilizzando opportunamente l'**indentazione** e i **commenti**. Si richiede infine di iniziare i codici con una riga di commento contenente il comando necessario per creare l'eseguibile.

Un'applicazione del teorema centrale del limite

Il teorema centrale del limite può essere applicato per ottenere una sequenza di numeri pseudo-casuali distribuiti secondo una Gaussiana, a partire da distribuzioni note.

1. Si prepari una funzione `rand_TCL_unif`, implementata in una libreria scritta in coppia di *file* chiamati `montecarlo.h` e `montecarlo.cc`, che generi numeri pseudo-casuali distribuiti secondo una distribuzione di densità di probabilità Gaussiana nell'intervallo $(1, 3)$ utilizzando il teorema centrale del limite, a partire da numeri pseudo-casuali distribuiti secondo una distribuzione di densità di probabilità uniforme, si scriva un `main.cpp` dove vengano generati 10000 numeri con questa funzione, li si disegni in un file di tipo `png` tramite un istogramma di tipo `TH1F`.
2. Si aggiunga alla libreria del punto precedente una funzione `rand_TAC` che generi numeri pseudo-casuali nell'intervallo $(1, 3)$ distribuiti secondo la seguente distribuzione di densità di probabilità parabolica $f(x)$:

   $$f(x) = -(x-2)^2 + 1 \;;$$

   nel programma principale `main.cpp` la generazione di 10000 numeri con questa funzione, li si disegni in un file di tipo `png` tramite un istogramma di tipo `TH1F`.
3. Si aggiunga alla libreria del primo punto una funzione `rand_TCL_para` che generi numeri pseudo-casuali nell'intervallo $(1, 3)$ distribuiti secondo una distribuzione di densità di probabilità Gaussiana utilizzando il teorema centrale del limite, a partire da numeri pseudo-casuali distribuiti secondo $f(x)$ e li si disegni in un file di tipo `png` come nel punto precedente.
4. Si crei una nuova libreria che calcoli asimmetria e curtosi di un campione di eventi salvato in un `std::vector`, e si calcolino queste due qantità per un campione di 10000 eventi generato con `rand_TCL_unif` ed per uno generato con `rand_TCL_para`.
5. Utilizzando quattro `TGraph`, si traccino in due file di tipo `png` l'andamento di asimmetria e curtosi per campioni generati con `rand_TCL_unif` e `rand_TCL_para` rispettivamente, al variare del numero di eventi pseudo-casuali generati all'interno di queste due ultime funzioni. Dopo quanti numeri le due funzioni possono essere considerate equivalenti in termini di prestazioni? Perché?

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

Corso di Laurea in Fisica
Esame di Laboratorio II – I Modulo

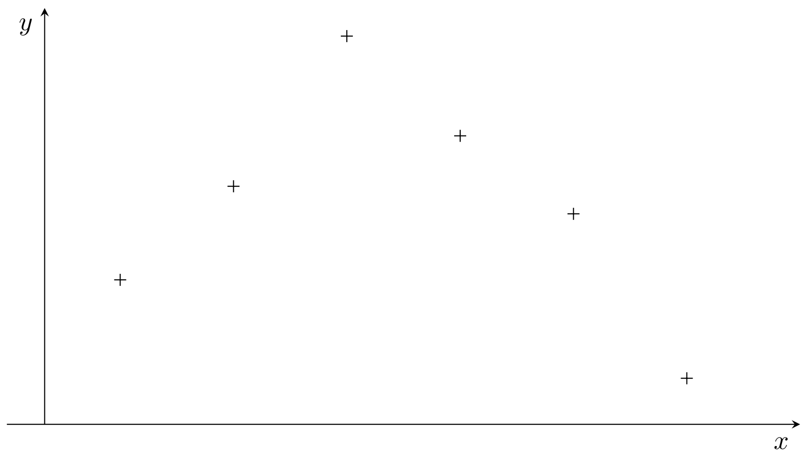
26 settembre 2022

Indicazioni generali

Si risolva il seguente esercizio, scrivendo un programma in `C++`. Ai fini della valutazione, il primo criterio che deve essere soddisfatto è che il codice compili senza errori ed esegua realizzando le funzionalità richieste dal testo. Per la valutazione sarà inoltre tenuto in considerazione il fatto che i codici siano scritti con ordine, utilizzando opportunamente l'**indentazione** e i **commenti**. Si richiede infine di iniziare i codici con una riga di commento contenente il comando necessario per creare l'eseguibile.

Traiettoria parabolica

Un punto materiale si muove dall'origine di un sistema di coordinate cartesiano (x, y) lungo una traiettoria parabolica, che viene misurata in sei punti, dando origine alle coordinate riportate nel file `coordinate.txt`.



1. Si scriva un programma che legga il file di coordinate e ne trascriva il contenuto in due `std::vector`.
2. Si disegnino a schermo i punti letti tramite un `TGraph` di `ROOT`.
3. Utilizzando le funzionalità di fit di `ROOT`, si determinino i parametri della traiettoria.
4. Utilizzando i parametri così trovati, si determini il valore della gittata ed il suo errore.
5. Si aggiunga alla libreria una funzione che trovi il massimo di una `TF1` di `ROOT` e la si applichi alla funzione di fit trovata nei punti precedenti, scrivendo poi a schermo le coordinate $(x_{\max}, y_{\max})$ della massima altezza raggiunta dal punto materiale.

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