

Laboratorio di Calcolo, Esercitazione 5, 10-14 novembre 2024

Canale Pet-Z, Docenti: Shahram Rahatlou, Fabio Bellini, Sibilla Di Pace

Stima di π con l'ago di Buffon: Lo scopo di questa esercitazione è di stimare il valore di π con un metodo iterativo noto dal XVIII secolo che utilizza l'estrazione dei numeri casuali tramite le funzioni `srand48()` e `lrand48()`.

Supponiamo di avere un piano percorso da linee parallele distanti d tra di loro e un ago di lunghezza L con $L < d$. Lanciando l'ago sul piano, essa ha una probabilità $2L/\pi d$ di incrociare una linea del piano. Sia x la distanza tra il centro dell'ago e la linea più vicina allago e θ l'angolo acuto tra l'ago e le linee. L'ago incrocerà una delle linee se è verificata la condizione $x < (L/2) \sin \theta$. Effettuando N_{lanci} lanci e indicando con S il numero di volte che l'ago incrocia una linea si ha che $S/N_{\text{lanci}} = 2L/(\pi d)$ da cui possiamo ottenere la stima $\pi = 2LN_{\text{lanci}}/(Sd)$.

In questa prova vogliamo ripetere la stima $N_{\text{exp}} = 1000$ di volte, ripetendo ogni volta procedura descritta sopra.

► Cartella di lavoro

Fare login sulla postazione utilizzando le credenziali del vostro gruppo, `lcsrNNN`, dove `NNN` è il vostro numero di gruppo, ad esempio `098`. Creare una cartella `LCSR5` nella *home directory* con il comando `mkdir` in cui scrivrete i programmi di oggi. Tutti i file di codice sorgente in C e in python dovranno trovarsi in questa cartella per essere visualizzati. **Le cartelle create sulla scrivania (Desktop) o in altre sotto-cartelle non verranno considerate.**

► Stima di π

Creare un programma `buffon.c` nella cartella `LCSR5` utilizzando l'editor di testo `emacs`, per eseguire le seguenti operazioni:

1. acquisire dall'utente il valore delle variabili L , d , N_{lanci} e verificarne individualmente la validità.
Informare l'utente che i valori di L e d devono essere in cm e minori di 5 cm, oltre che $L < d$, e $N_{\text{lanci}} < 10000$;
2. creare un array `pigreco` di lunghezza N_{exp} per immagazzinare i valori di π stimati in ciascuna prova;
3. aprire un file di testo `pigreco.txt` per la scrittura dei dati

```
1 FILE* fp;
2 fp = fopen("pigreco.txt", "w");
```

Listato 1: Apertura del file in scrittura

4. con un opportuno ciclo simulare N_{lanci} lanci dell'ago dove ciascun lancio consiste in
 - generare un valore casuale di θ compreso tra 0 e $\pi/2$ (usare `M_PI` della libreria matematica);
 - generare un valore casuale di x compreso tra 0 e $d/2$.
5. per ciascun lancio determinare se l'ago incrocia o meno una linea;
6. contare il numero S di lanci in cui l'ago ha incrociato una linea e calcolare il valore di π utilizzando l'espressione fornita ed immagazzinarlo nell'array `pigreco`;

7. con un opportuno ciclo ripetere i passi 4-6 per N_{exp} volte;
8. al termine del ciclo di esperimenti, determinare il minimo π_{min} , il massimo π_{max} , e il valore medio $\langle \pi \rangle$ dei valori immagazzinati nell'array `pigreco`; si ricorda che il valore medio di N misure $\{x_1, \dots, x_N\}$ è definito come $\langle x \rangle = \sum_{j=1}^N x_j / N$;
9. stampare sullo schermo i valori di π_{min} , π_{max} , e $\langle \pi \rangle$ con 12 cifre decimali ;
10. scrivere nel file `pigreco.txt` il numero dell'esperimento (indice dell'array) e il valore di π ottenuto; ci devono essere quindi N_{exp} righe con solo 2 valori numerici per riga (nessun testo aggiuntivo) con il formato "%5d \t %.12f\n" usando la funzione `fprintf()`:

```
1   fprintf(fp, "%d %.12f\n", iexp, pigreco[iexp]);
```

Listato 2: Scrittura di dati sul file

11. Al termine del ciclo di scrittura, chiudere il file di output prima di terminare il programma

```
1   fclose(fp);
```

Listato 3: Chiusura del file

12. infine, creare un istogramma dei valori scritti nel file utilizzando il codice in python fornito in seguito.

Si ricorda che per creare l'eseguibile utilizzando la libreria matematica dovete usare il comando `gcc -Wall -o app.exe programma.c -lm` dalla riga di comando nella shell.

Si consiglia di scrivere il programma in modo incrementale, verificando la corretta compilazione e l'esecuzione almeno dopo ciascuno dei passi indicati nel testo.

► Opzionale solo se avete finito tutto

Oltre alla media $\langle \pi \rangle$ potete calcolare e stampare sullo schermo l'incertezza $\delta\pi$ su questa media come $\delta\pi = \sigma / \sqrt{N_{exp}}$, dove σ è la deviazione standard campionaria dei valori ottenuti.

Per N misure $\{x_1, \dots, x_N\}$ la deviazione standard campionaria è definita come $\sigma^2 = \sum_{i=1}^N (x_i - \langle x \rangle)^2 / (N - 1) = \langle x^2 \rangle - \langle x \rangle^2$, con $\langle x \rangle = \sum_{j=1}^N x_j / N$ e $\langle x^2 \rangle = \sum_{j=1}^N x_j^2 / N$

► Iistogramma con python

Vogliamo graficare

1. l'andamento delle stime in funzione del numero di esperimenti;
2. la distribuzione dei valori stimati

con il seguente codice che deve essere scritto in un file chiamato `pigreco.py` e salvato nella cartella LCSR5. Per girare il programma in python, dovete eseguire il comando `python3 pigreco.py` nel terminale, assicurandovi di trovarvi nella cartella LCSR5.

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 import math as m
4
5 # exp sono numero di esperimenti nella prima colonna
6 # pi sono le stime nella seconda colonna
7 exp, pi = np.loadtxt('pigreco.txt', unpack=True)
8
9 # crea grafica di pi stimato in funzione del numero di esperimento
10 plt.title("Andamento di  $\pi$  in funzione di esperimento")
11 plt.plot(exp, pi, '.', label='stima di  $\pi$  con ago di buffon')
12
13 # specifica limite inferiore superiore per asse x e y con numeri
14 # opportuni in base ai vostri dati
14 plt.xlim(0, 1000)
15 plt.ylim(2., 4.)
16 # aggiungi legenda per gli assi
17 plt.xlabel('#esperimento')
18 plt.ylabel('$\pi$ stimato')
19
20 # linea orizzontale rossa: valore libreria matematica
21 plt.axhline(y=m.pi, color = 'red', linestyle='--', label='valore
    libreria matematica')
22 # mostra griglia x, y
23 plt.grid()
24 # nome del file in cui salvare il grafico
25 plt.savefig("piexp.png")
26 # mostra grafico
27 plt.show()
28
29 ##### istogramma della distribuzione di pi greco
30 plt.title("Distribuzione delle stime di  $\pi$  greco")
31 plt.hist(pi, color='green')
32 plt.xlabel("$\pi$ stimato")
33 plt.ylabel("# esperimenti")
34 # linea verticale: valore libreria matematica
35 plt.axvline(x=m.pi, color = 'blue', linestyle='-', label='valore
    libreria matematica')
36 # legenda per il grafico
37 plt.legend()
38 plt.grid()
39 plt.savefig("pihist.png")
40 plt.show()

```

Listato 4: Programma pigreco.py