

# Laboratorio di Calcolo, Esercitazione 8, 5-6 dicembre 2024

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

**Interazione di particelle con un bersaglio di atomi:** Lo scopo di questa esercitazione è di simulare l'interazione di un fascio di particelle con un bersaglio di forma ellittica composto da atomi. È necessario implementare alcune funzioni ed utilizzare gli array per immagazzinare dati da passare alle funzioni.

## ► Cartella di lavoro

Fare login sulla postazione utilizzando le credenziali user-id `studente` e password `informatica`. Creare una cartella **LCSR8** nella *home directory* con il comando `mkdir` in cui scriverete 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 copiate né valutate.**

## ► Nozioni utili

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 nel terminale.

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

## ► Prima parte

Quando una particella colpisce un bersaglio, interagisce con una certa probabilità  $p_{\text{int}}$  con gli atomi che compongono il bersaglio. Per simulare questo fenomeno si possono immaginare gli atomi del bersaglio come una serie di punti distribuiti a caso su una superficie  $S$  che rappresenta la sezione trasversale del bersaglio. Assumiamo che una particella che incide sul bersaglio nelle coordinate  $(x_p, y_p)$ , interne alla superficie  $S$ , abbia una probabilità di interazione  $p_{\text{int}} = 70\%$  di interagire con un atomo che si trova a una distanza  $R_{\text{int}}$  da essa.

Lo scopo del programma di oggi è di simulare l'interazione di un fascio di  $N_p$  particelle incidenti, lungo l'asse  $z$ , su un bersaglio ellittico con il semiasse maggiore  $A$  lungo l'asse  $x$  e il semiasse minore  $B$  lungo l'asse  $y$ , nel quale sono contenuti  $M_A$  atomi. Si ricorda che l'ellisse è descritta dall'equazione  $(x/A)^2 + (y/B)^2 = 1$ .

Creare un file `collision-NNN.c`, dove `NNN` è il vostro numero di gruppo, ad esempio `098`, nella cartella **LCSR8** utilizzando l'editor di testo `emacs`, per eseguire le seguenti operazioni:

1. chiedere all'utente di inserire i valori per le variabili  $N_p$ ,  $M_A$ ,  $R_{\text{int}}$ ,  $A$ , e  $B$  (le ultime tre variabili in unità arbitrarie  $ua$ ), scegliendo variabili di tipo opportuno, ed assicurandosi che siano rispettate le condizioni  $R < 0.1$ ,  $B < A < 10$ , ed  $N_p < M_A < 10000$ . Si ricorda che ciascuna variabile va acquisita separatamente e l'utente deve essere informato dei limiti per il valore da inserire.
2. implementare una funzione `ellisse()` di tipo `void`, con opportuni argomenti, che generi in modo uniforme le coordinate  $x_A$  e  $y_A$  di un punto all'interno di un ellisse; il valore dei semiassi devono essere passati come argomenti della funzione.

3. implementare una funzione `atomi()` per generare le coordinate di  $M_A$  atomi contenuti all'interno dell'ellisse e memorizzare queste coordinate in un array bidimensionale `bersaglio`. Al suo interno, questa funzione deve chiamare la funzione `ellisse` implementata al punto precedente;
4. con un ciclo opportuno, generare le coordinate di  $N_p$  particelle incidenti sul bersaglio. Per ciascun proiettile verificare se c'è un atomo a distanza  $R_{\text{int}}$ , e tenendo conto della probabilità  $p_{\text{int}}$  di interazione, determinare il numero  $N_{\text{int}}$  di atomi con cui interagisce.
5. al termine del ciclo dei proiettili, stampare sullo schermo il numero medio di interazioni per particella incidente, ed il numero massimo di interazioni avvenute per una data particella incidente

► **Seconda parte**

1. Modificare `collision-NNN.c` per chiedere all'utente di inserire il nome del file in cui scrivere le posizioni  $x_A$  ed  $y_A$  degli atomi (un atomo per riga,  $M_A$  righe in totale);
2. scrivere un programma `bersaglio-NNN.py` in python per leggere i valori dal file di dati e graficare la posizione degli  $M_A$  atomi usati nella simulazione;
3. aggiungere opportuni titoli agli assi per comprendere il grafico;
4. **opzionale:** disegnare un'ellisse con i valori di  $A$  e  $B$  inseriti.