

Laboratorio di Calcolo per Fisici, Prova pratica del 07/07/2023

Tutti i canali — A.A. 2022–2023

Nome _____ Cognome _____
Matricola _____ Ritirato/a

Lo scopo di questa prova d'esame è di scrivere un programma in C e uno script in python seguendo la traccia riportata di seguito. Si tenga presente che:

1. Per svolgere il compito avete a disposizione 3 ore.
2. Si possono usare libri di testo, prontuari e gli appunti, ma non è ammesso parlare con nessuno né utilizzare cellulari, tablet o laptop, pena l'annullamento del compito.
3. Seguite pedissequamente le istruzioni che trovate nel testo (nomi delle funzioni, dei file, delle variabili, formati di stampa, *ecc*).
4. Il programma va scritto e salvato esclusivamente sul computer del laboratorio, a cui si deve accedere utilizzando come username **studente** e come password **informatica**
5. Il programma va creato all'interno di un file di nome `NOME_COGNOME.c`, salvato esclusivamente in una cartella di nome `NOME_COGNOME` creata all'interno della home directory. **Tutto ciò che non si trova all'interno della cartella suddetta non verrà valutato.** Lo script python e la figura corrispondente andranno salvati su due file di nome `NOME_COGNOME.py` e `NOME_COGNOME.png` rispettivamente. All'inizio dei file c e python inserire una riga di commento contenenti il vostro nome, cognome e numero di matricola.
6. Per consegnare il compito (o dietro richiesta del docente) dovete copiare tutto il contenuto della cartella con il seguente comando: `cp * /media/sf_esame/`
7. Consegnate il presente testo indicando nome, cognome e numero di matricola, barrando la casella "Ritirato/a" se ci si vuole ritirare, ovvero se non si vuole che l'elaborato venga valutato.

► Background

I polimeri sono macromolecole composte da un numero N di unità (che nel caso più semplice sono tutte uguali) dette *monomeri*. Sono molto importanti sia da un punto di vista tecnologico (essendo alla base delle materiale plastiche) che da un punto di vista teorico, per le tante proprietà peculiari che posseggono. Molte delle proprietà dei polimeri possono essere spiegate e capite tramite modelli semplici, trattabili anche numericamente. Nel caso più semplice, un polimero può essere modellizzato come un *random walk* (*cammino aleatorio*) su di un reticolo bidimensionale quadrato (illimitato). In questo modo si stabilisce un'analogia formale tra un cammino casuale composto da $N - 1$ passi e un polimero composto da N monomeri.

Lo scopo di questa prova è quella di generare NC polimeri (modellizzati come *random walk* su un reticolo bidimensionale quadrato) composti da N monomeri, per valori di N diversi. Si assume che più monomeri possano occupare la stessa posizione sul reticolo. Dato un valore di N ($N = 15$ nella figura di esempio in basso), dovete generare NC polimeri. Ognuno degli N monomeri è caratterizzato da due coordinate intere, $\vec{r}_i = (x, y)$. Per semplicità si pone il primo monomero nell'origine, $\vec{r}_0 = (0, 0)$. La posizione del secondo monomero varrà $\vec{r}_1 = \vec{r}_0 + \Delta\vec{r}_1$, dove $\Delta\vec{r}_1$ è un passo di dimensione 1 in una direzione casuale. Iterando questa procedura, dato il monomero i -esimo, la posizione dell' $(i + 1)$ -esimo è data da $\vec{r}_{i+1} = \vec{r}_i + \Delta\vec{r}_{i+1}$, dove il passo $\Delta\vec{r}_{i+1}$ è scelto casualmente tra i seguenti:

- $\Delta\vec{r}_{i+1} = (1, 0)$ indica un passo verso destra
- $\Delta\vec{r}_{i+1} = (-1, 0)$ indica un passo verso sinistra

- $\Delta\vec{r}_{i+1} = (0, 1)$ indica un passo verso l'alto
- $\Delta\vec{r}_{i+1} = (0, -1)$ indica un passo verso il basso

Si noti che, avendo assunto che più monomeri possano occupare la stessa posizione sul reticolo, non c'è bisogno di controllare che il punto di arrivo, \vec{r}_{i+1} , sia stato o meno già visitato. Una volta che gli N monomeri sono stati posizionati, definiamo la dimensione del polimero come la distanza tra le estremità (anche nota come distanza *end-to-end*), cioè la distanza tra il monomero 0 e quello $(N - 1)$ -esimo:

$$r_{ee} = |\vec{r}_{N-1} - \vec{r}_0| \quad (1)$$

Questa quantità va mediata su ognuno degli NC polimeri di taglia N generati.

La figura in basso mostra l'idea base dell'algoritmo da utilizzare e la definizione di r_{ee} , da applicare una volta generato completamente il polimero (cioè, dopo $N - 1$ passi).

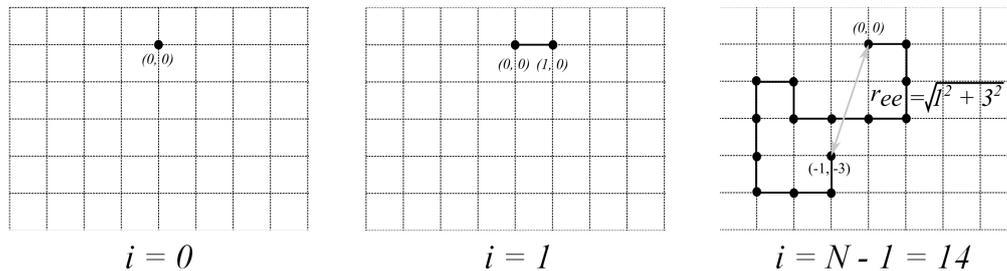


Figura 1: Come costruire un polimero composto da $N = 15$ monomeri utilizzando un cammino aleatorio su di un reticolo bidimensionale composto da $N - 1 = 14$ passi. Anche se la figura non lo mostra, in linea di principio il cammino generato può tornare su di un punto già visitato.

► Prima parte

Scrivete un codice `NOME_COGNOME.c` che generi $NC = 1000$ polimeri (*random walk*) su di un reticolo quadrato bidimensionale per valori di N che vanno da 10 a `NMAX`. In particolare il programma deve

1. definire, attraverso un'opportuna direttive `#define`, le costanti `NMAX` ed `NC`;
2. definire all'interno del main un array `polimero[NMAX][2]` che conterrà un cammino generato aleatoriamente;
3. contenere una funzione `genera_polimero()` che prende in ingresso la taglia del polimero N e l'array `polimero`. La funzione inizializza a $(0,0)$ la prima casella dell'array e poi esegue $N - 1$ passi casuali sul reticolo. Ad ogni passo la casella i -esima prende il valore di quella $(i - 1)$ -esima più un passo casuale scelto tra questi quattro casi: sinistra, $(-1, 0)$, destra, $(+1, 0)$, basso $(0, -1)$ o alto $(0, +1)$.
4. contenere una funzione `lunghezza_polimero()` che prende in ingresso N e `polimero` e restituisce la distanza tra il primo e l'ultimo punto del polimero. **Nota Bene:** si trova numericamente che il valore medio della distanza *end-to-end* per polimeri generati come descritto sopra valga circa $0.89 \cdot (N - 1)^{0.5}$, cioè ≈ 2.7 per $N = 10$ e ≈ 3.9 per $N = 20$. Utilizzate questi valori per testare le vostre funzioni.
5. effettuare un ciclo per N che va da 10 a 100 (estremi compresi), e per ogni N calcolare r_{ee} , mediata sugli `NC` polimeri;
6. stampare su di un file `ree.dat` i valori di N e r_{ee} per ogni valore di N .

► Seconda parte

Scrivete uno script Python `NOME_COGNOME.py` che grafichi la distanza *end-to-end* in funzione di N contenuta nel file `ree.dat`. Lo script deve salvare il grafico, che dovrà contenere una legenda e opportuni *label* sugli assi, nel file `NOME_COGNOME.png`.