

Laboratorio di Informatica

Corso di laurea triennale in Fisica

CARLO MEREGHETTI

ATTENZIONE!!

Per questo progetto non verrà dato alcun suggerimento sulla strutturazione del codice, esattamente come all'esame. Sta a voi prevedere funzioni utili da mettere in una o più opportune librerie. Siete caldamente invitati a NON scrivere un unico `main` — *pena una bassa votazione finale* — vista anche la complessità del progetto.

In più, siete caldamente invitati a costruire un `makefile` che consenta un'agevole gestione del progetto mediante compilazione separata e tale che i comandi `make compila` e `make esegui` lancino, rispettivamente, la compilazione e l'esecuzione del vostro programma. L'assenza di un siffatto `makefile` genererà una *bassa votazione finale all'esame*.

In questo progetto cominceremo ad utilizzare RooT per rappresentare graficamente dati.

Tutti i file (sorgenti ed eseguibili) prodotti andranno salvati sotto la directory `Lab9`.

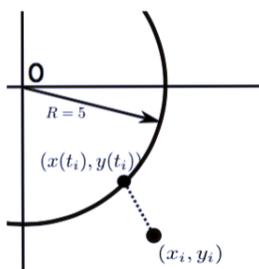
MOTO CIRCOLARE

Si consideri una massa che si muove con moto circolare uniforme intorno all'origine di un sistema di riferimento cartesiano assegnato. La posizione della massa è misurata a diversi istanti di tempo e registrata nel file `posizioni.in` nella forma di terne (t_i, x_i, y_i) , dove t_i , x_i e y_i sono `float` che indicano, rispettivamente, il tempo a cui è stata effettuata la misura e le coordinate della posizione registrata. Le coordinate della posizione sul piano della massa è descritta dalle leggi orarie

$$x(t) = 5 \cos(2t)$$

$$y(t) = 5 \sin(2t).$$

La seguente figura esemplifica la situazione qui esposta.



Al tempo t_i la misura della posizione della massa restituisce le coordinate (x_i, y_i) mentre le leggi orarie prevedono che la massa si trovi alla posizione $(x(t_i), y(t_i))$. Le differenze, coordinate per coordinate, tra questi due risultati sono le componenti lungo gli assi della *discrepanza* tra esperimento e teoria e riempiono i campi `dscrx` e `dscry` della struct `misura` (vedi sotto). Il campo `de` di tale struct, invece, contiene la *discrepanza*, cioè distanza tra i punti (x_i, y_i) e $(x(t_i), y(t_i))$.

Scrivere un programma che svolga le seguenti le seguenti operazioni:

1. Conti il numero di terne (misure) presenti nel file `posizioni.in` e le carichi in un array di

```
struct misura {
    float t;
    float x;
    float y;
    float dscrx;
    float dscry;
    float de;
};
```

stampando a video solo il numero di terne caricate.

2. Per ogni terna, calcoli le discrepanza $dscrx_i = x_i - x(t_i)$ e $dscry_i = y_i - y(t_i)$ e la registri nei corrispondenti campi `dscrx` e `dscry`. (Vedi la descrizione alla pagina precedente per il significato di discrepanza: in sostanza, stiamo parlando di scostamento tra la posizione misurata dal nostro strumento e quella prevista dalle leggi orarie descritte.) Dev'essere inoltre calcolata la distanza $de_i = \sqrt{(dscrx_i)^2 + (dscry_i)^2}$ la quale va registrata nel corrispondente campo `de`.

Successivamente, stampi a video le 5-tuple $(t_i, x_i, y_i, dscrx_i, dscry_i)$ riferite ai primi 10 elementi dell'array caricato.

3. Elimini dall'array ottenuto al punto precedente (ricordate il metodo di scrematura dei dati in array adottato alla scorsa esercitazione) tutti gli elementi relativi a posizioni nel primo quadrante e stampi a video le 6-tuple $(t_i, x_i, y_i, dscrx_i, dscry_i, de_i)$ riferite ai soli elementi eliminati.
4. Ordini i soli elementi sopravvissuti alla precedente operazione di scrematura in ordine crescente secondo `de`. Successivamente, stampi a video le 6-tuple $(t_i, x_i, y_i, dscrx_i, dscry_i, de_i)$ degli ultimi 10 elementi del segmento ordinato dell'array.
5. Disegni con RooT uno scatterplot (grafico di dispersione) dei punti (x_i, y_i) degli elementi sopravvissuti all'operazione di scrematura. Sullo stesso scatterplot, operare il fitting con un polinomio di nono grado.

Tutti i risultati (eccetto lo scatterplot al punto 5), oltre che stampati a video *con opportune diciture*, devono essere salvati nel file `risultati.out` corredati dalle stesse diciture.