Laboratorio di Calcolo per Fisici, Decima esercitazione

Canale D-K, Docente: Dott.ssa Livia Soffi,
Esercitatori: Prof. S. Rahatlou e Dott.ssa Giulia D'Imperio

Lo scopo della decima esercitazione di laboratorio è di esercitarsi con il metodo hit and miss Montecarlo e l'I/O da file.

Le leggi di Keplero descrivono il moto dei pianeti su orbite ellittiche attorno al sole. Con la seconda legge si osserva che: "il raggio vettore che unisce il Sole con il pianeta, copre aree uguali in tempi uguali". Quindi, conoscendo il tempo t che la Terra impiega a coprire una determinata area a si può ricavare la velocità areolare del pianeta:

$$v = \frac{a}{t}$$

che è una costante del moto. Poichè l'eccentricità dell'orbita terrestre è prossima allo zero, si può assumere in questo caso una traiettoria puramente circolare di raggio medio pari a $R = 149.5 \cdot 10^6$ km.

La tabella di seguito riassume gli intervalli angolari delle due regioni in figura e i corrispondenti tempi che impiega il raggio vettore del pianeta Terra a percorrerli.

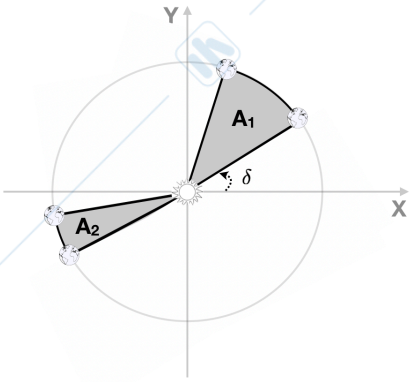
Regione	Intervallo angolare [rad]	Tempo di percorrenza [s]	Velocità areolare [km^2/s]
A_1	$0.7 \leq \delta \leq 1.3$	$t_1 = 3.5 \cdot 10^6$	$v_1 =$
A_2	$3.3 \leq \delta \leq 3.5$	$t_2 = 9.8 \cdot 10^5$	$v_2 =$

Utilizzando i tempi di percorrenza forniti dalla tabella, e misurando empiricamente l'area delle due regioni, è possibile stimare la velocità areolare per ciascuna regione e mediare i due risultati ottenuti:

$$\bar{v} = \frac{v_1 + v_2}{2}$$

L'errore associato alla media si può calcolare con la formula di dispersione massima:

$$Err = \frac{|v_1 - v_2|}{2}$$



Prima parte:

Scrivere un programma **keplero.c** per calcolare utilizzando il metodo *hit or miss* l'area delle regioni A_1 e A_2 in figura, necessarie per il calcolo della velocità areolare della Terra. Il programma deve:

- spiegare sinteticamente (max 1 riga) quello che sta per fare
- chiamare una funzione di nome **fill** di tipo *void* che generi opportunamente 5000 punti (x,y) all'interno della circonferenza di raggio R. La funzione deve prendere in ingresso due array unidimensionali vuoti, x[5000] e y[5000], e riempirli con le coordinate generate.
- chiamare per ciascuno dei due settori illustrati in tabella, la funzione **area** che prenda in ingresso gli array x[5000] e y[5000] e i valori angolari che delimitano il settore (δ_{MIN} e δ_{MAX}) in esame. La funzione ne deve restituire l'area. Per farlo:
  - contare quanti dei 5000 punti generati all'interno della circonferenza appartengono al settore delimitato da δ_{MIN} e δ_{MAX}.

  - tramite l'opportuna proporzione fornita dal metodo *hit or miss*, ricavare l'area del settore in esame

- sfruttando le relazioni decritte nell'introduzione, calcolare il valore delle v_1 e v_2 e della velocità areolare media e stamparlo su schermo nel formato:

$$\bar{v} + / - Err[km^2/s]$$

Formulario:

- Equazione della circonferenza: $x^2 + y^2 = R^2$
- Trasformazione coordinate cartesiane in coordinate polari:
  $x = Rcos(\delta)$
  $y = Rsin(\delta)$
  $tan(\delta) = \frac{y}{x}$ con $\delta \in [-\pi/2, \pi/2]$, con opportuna attenzione al quadrante e alla periodicità della funzione tangente
- Si ricorda che nel metodo *hit or miss*, dati N punti uniformemente distribuiti in un area A di riferimento, si contano quanti di questi, N_{hit}, appartengono ad una sottoregione di A la cui area è ignota. Il rapporto tra N_{hit} e N tenderà al rapporto delle aree corrispondenti.
- Si consiglia di definire l'area della circonferenza nelle direttive al preprocessore. Conviene inoltre utilizzare le coordinate polari nel metodo hit and miss.

Sarà elemento ulteriore di valutazione la struttura delle funzioni usate, l'utilizzo di funzioni aggiuntive e la minimizzazione del numero di operazioni da compiere.

Laboratorio di Calcolo per Fisici, Decima esercitazione - Addendum

Canale D-K, Docente: Dott.ssa Livia Soffi,
Esercitatori: Prof. S. Rahatlou e Dott.ssa Giulia D'Imperio

Seconda parte:
Aggiungere le seguenti funzionalità al programma:

- Modificare la funzione **fill**. Al suo interno scrivere su un file **circonferenza.txt** le coordinate dei 5000 punti generati all'interno della circonferenza.
- Modificare la funzione **area**. Fare in modo che questa prenda in ingresso anche una stringa di caratteri (diversa per ciascuna chiamata della funzione). All'interno della stessa poi, salvare le coordinate (x, y) dei punti che appartengono al settore in esame, su un file il cui nome sarà dato dalla stringa in ingresso alla funzione.

Terza parte:

- Tramite uno script **grafico.py** graficare i punti appartenenti alla circonferenza e poi quelli appartenenti ai settori di circonferenza considerati usando colori diversi al fine di ottenere un grafico simile a quello in figura. In questo grafico deve essere presente una legenda, un titolo e dei nomi opportuni per gli assi. Salvare infine l'immagine in un file chiamato **grafico.png**.

PROGRAMMA IN C ESEGUIBILE*

*__ATTENZIONE!__ Il programma comprende già le modifiche effettuate a seguito dell'Addendum, quindi se stai leggendo e non capisci qualcosa prova a controllare se quel "qualcosa" deriva dalla seconda consegna.

*__ATTENZIONE!__ Nota anche che, affinché la funzione fopen non dia il messaggio di errore, devi creare il file .txt** vuoto nella stessa cartella in cui hai creato il file.c.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<time.h>
#define R 149.5e6
#define A (M_PI*R*R) //km^2
#define N 5000
#define dmin1 0.7
#define dMax1 1.3
#define dmin2 3.3
#define dMax2 3.5
#define t1 3.5e6 //s
#define t2 9.8e5 //s

void welcome();
void azzera(double az[], double bz[]);
void fill(double a[], double b[]);
double area(char filename[], double c[], double d[], double dmin, double dMax, int MIN, int
MAX);  //NOTA: dichiaro 2 int come gli intervalli del quadrante che scelgo
double areolare(double areasett, double t);
double medspeed(double sp1, double sp2);
double error(double s1, double s2);



//FUNZIONE main
int main(){
 double x[N];
 double y[N];
double Asett1, Asett2;
double v1, v2, speed, err;
int seed;

seed=time(0);     //INIZIALIZZAZIONE seed
srand48(seed);

 //CHIAMATA FUNZIONI
 welcome();
 azzera(x, y);
fill(x, y);
Asett1 = area("area1.txt", x, y, dmin1, dMax1, 0, R);
Asett2 = area("area2.txt", x, y, dmin2, dMax2, (-R), 0);

   printf("\n");

v1 = areolare(Asett1, t1);
v2 = areolare(Asett2, t2);

printf("\n");

speed = medspeed(v1, v2);
```

```
err = error(v1, v2);

printf("\n\n");

return 0;
}



//FUNZIONE welcome
void welcome(){
printf("\n\nBenvenuto. Questo programma calcola la velocità areolare della Terra rispetto
        ad una specifica area.\n");
}



//FUNZIONE azzera
void azzera(double az[], double bz[]){
 int h;

 for(h=0; h<N; h++){
 az[h]=0;
bz[h]=0;
}
}



//FUNZIONE fill
void fill(double a[], double b[]){
int n;
double a1=0;
double b1=0;
FILE *fp;

if((fp=fopen("circonferenza.txt", "r+"))==NULL){ //**
 printf("Errore apertura file.\n");
exit(1);
 }


fp=fopen("circonferenza.txt", "w+");
fprintf(fp, "\t#n\t#x\t\t\t\t\t\t\t#y\n");

for(n=0; n<N; n++){
a1 = ((R+R)*lrand48()/RAND_MAX)-R;
b1 = ((R+R)*lrand48()/RAND_MAX)-R;

if(pow(a1,2)+pow(b1,2)<pow(R,2)){
a[n]=a1;
b[n]=b1;

fprintf(fp, "%4.0d\t%e\t\t\t%e\n", n+1, a[n], b[n]);

} else {
 n--;
```

```
}
}

fclose(fp);

}



//FUNZIONE area
double area(char filename[], double c[], double d[], double dmin, double dMax, int MIN, int
MAX){
 int i;
 int Nhit=0;
double delta, area;
FILE *fp;


 if((fp=fopen(filename, "r+"))==NULL){
printf("Errore apertura file.\n");
exit(1);
 }

fp=fopen(filename, "w+");
fprintf(fp, "#x\t\t\t\t\t\t\t#y\n");


for(i=0; i<N; i++){
delta = atan(d[i]/c[i]);

if((c[i]>=MIN && c[i]<=MAX) && (d[i]>=MIN && d[i]<=MAX)){  //Controllo se (x, y)>0 oppure
<0 in base al quadrante, poi applico i controlli sul delta
 if(tan(delta)<=tan(dMax) && tan(delta)>=tan(dmin)){

 Nhit++;
fprintf(fp, "%e\t\t%e\n", c[i], d[i]);

          }
      }
}

 fclose(fp);

area = (double)Nhit*A/(double)N;
printf("-Area settore [delta min.=%2.1lf, delta max.=%2.1lf] = %e km^2.\n", dmin, dMax,
area);

return area;
}



//FUNZIONE areolare
double areolare(double areasett, double t){
double v;

v = areasett/t;
```

```
printf("-Velocità areolare settore con area=%e km^2, con tempo=%e s  =  %lf km^2/s.\n",
areasett, t, v);

return v;
}



//FUNZIONE medspeed
double medspeed(double sp1, double sp2){
double m;

m = (sp1+sp2) / 2.;

printf("-Velocità media con relativo errore.\n+%e km^2/s / ", m);

return m;
}



//FUNZIONE error
double error(double s1, double s2){
double e;

 e = fabs(s1-s2) / 2.;

printf("-%e km^2/s.\n", e);

return e;
}
```

PROGRAMMA IN PYTHON

```
#This Python file uses the following encoding: utf-8


import matplotlib.pyplot as plt
import numpy as np


#PUNTI CIRCONFERENZA
x,y=np.loadtxt('circonferenza.txt', unpack=True, usecols=(1,2))
plt.title('Area con metodo "hit or miss"')
plt.xlim(-1.5e8, 1.5e8)
plt.ylim(-1.5e8, 1.5e8)
plt.xlabel('coordinate x')
plt.ylabel('coordinate y')
plt.plot(x,y,'co', label='Punti circonferenza')
plt.legend()

#PUNTI AREA1
x,y=np.loadtxt('area1.txt', unpack=True, usecols=(0,1))
plt.plot(x,y,'mo', label='Punti settore 1')
plt.legend()

#PUNTI AREA2
x,y=np.loadtxt('area2.txt', unpack=True, usecols=(0,1))
plt.plot(x,y,'bo', label='Punti settore 2')
plt.legend()

plt.savefig('grafico.png')
plt.show()
```

ALLEGATO DI COME DOVREBBE USCIRTI IL GRAFICO