Esercitazione 4

funzioni, minimi quadrati, chi quadrato

29-30-31 ottobre 2018

Corso di Laboratorio 1A, A.A. 18-19
Laurea triennale in Fisica, Università di Genova

Lo scopo di questa esercitazione è imparare ad usare le funzioni ed applicarle al calcolo di alcune grandezze di interesse per la trattazione statistica dei dati nelle esperienze di Lab1B.
Il primo esercizio, svolto, ha lo scopo di rendere chiari i concetti relativi al passaggio di parametri alle funzioni, in particolare il passaggio per valore e per reference.
Il secondo esercizio, parzialmente svolto, è un esempio di funzione per leggere un file con un numero arbitrario (ma noto a priori) di colonne e un numero arbitrario (e non noto a priori) di righe.
Il terzo esercizio, da svolgere in laboratorio, richiede di implementare delle funzioni che calcolano delle quantità di rilevanza statistica a partire da dei vettori di double in ingresso.
Il quarto esercizio, da svolgere in laboratorio, utilizza le funzioni sviluppate nel terzo esercizio per l'applicazione del metodo dei minimi quadrati ad un set di dati legati da una relazione lineare.
Il quinto esercizio, da svolgere come compito a casa, chiede di estendere il programma dell'esercizio 4 per fare in modo che calcoli il chi quadro corrispondente ai parametri del fit.

- Svolgete gli esercizi secondo la traccia riportata nel resto di questo documento
- Seguite le istruzioni riportate qui per la consegna degli esercizi di laboratorio
- IMPORTANTE: la consegna deve essere effettuata da entrambi i componenti del gruppo

Esercizio 1

Esercizio svolto. Esempio del passaggio di parametri per valore e per reference.
Compilare ed eseguire il seguente programma:

```
#include <iostream>
using namespace std;

double PassByValue(int val) {
   cout << "In PassByValue() &val " << &val << endl;
   double result = 0.45*val;
   val = 0;
   return result;
}

double PassByReference(int& val) {
   cout << "In PassByReference() &val " << &val << endl;
   double result = 0.45*val;
   val = 0;
   return result;
}

int main(int argc, char** argv) {
   // Changes to a parameter passed by value have no effect outside the
   // body of the function
   cout << endl;
   int var1 = 9;
   cout << "Before PassByValue() var1 = " << var1
       << ", &var1 = " << &var1 << endl;
   double ret1 = PassByValue(var1);
   cout << "After PassByValue() var1 = " << var1
       << ", &var1 = " << &var1 << ", ret1 = " << ret1 << endl;

   // Changes to a parameter passed by reference also affect the variable
   // outside the body of the function
   cout << endl;
   int var2 = 9;
   cout << "Before PassByReference() var2 = " << var2
       << ", &var2 = " << &var2 << endl;
   double ret2 = PassByReference(var2);
   cout << "After PassByReference() var2 = " << var2
       << ", &var2 = " << &var2 << ", ret2 = " << ret2 << endl;

   return 0;
}
```

Materiale da consegnare: esercizio1.cpp

Esercizio 2

Esercizio parzialmente svolto. Lettura da file di una matrice di dati. Creare un programma che, all'interno del main(), usi la funzione ReadFile() (la cui implementazione è data nella prossima pagina) per leggere il file dati_esp4.dat, contenente 4 colonne (x, y, ex, ey), e che stampi il numero di righe contenute nel file, e tutti i valori della colonna ey.

La funzione ReadFile() prende in ingresso il nome di un file da leggere, un "vettore di vettori di double", che verrà riempito con i dati, e il numero di colonne attese nel file di input. Ciascun vettore di double contenuto nel vettore in uscita verrà riempito con una delle colonne presenti nel file. La funzione ritorna una variabile bool che indica se la lettura del file è andata a buon fine.

Nota: per poter "riempire" il vettore passato come argomento, questo DEVE essere passato per reference. Invece il fatto che il parametro fileName sia passato per reference invece che per valore è molto meno importante. Anche passando il parametro fileName per valore si sarebbe ottenuto lo stesso risultato, a meno di una perdita di efficienza, del tutto trascurabile, legata al fatto che nel caso di passaggio per reference la funzione evita di fare una copia inutile della stringa

Nota: la funzione ReadFile() proposta in questo esercizio è di carattere piuttosto generale, e con modifiche minime può essere usata per leggere la quasi totalità dei file che incontrerete nel corso di Lab1.

Suggerimenti

Abbiamo visto a lezione che si può usare una funzione anche senza sapere come fa a svolgere un compito, ciò che conta è che sia chiaro qual è la sua interfaccia, e come vada usata. In un primo momento concentratevi sull'interfaccia della funzione e cercate di capire:

1. Qual è il significato di ciascun parametro da passare alla funzione?
2. In particolare, che cos'è "vector< vector<double> >& data"?
3. Cosa ritorna la funzione, e come posso usare il suo return value?

In un primo momento i dettagli relativi all'implementazione della funzione possono essere ignorati, e l'esercizio può essere svolto semplicemente avendo chiari i punti elencati qui sopra.

Nella lezione sulle funzioni di venerdi 26/10 abbiamo visto un esempio simile a quello proposto in questo esercizio, ma più semplice, perché in quel caso il programma leggeva un file contenente una sola colonna di dati. Una versione semplificata di quel programma è disponibile qui.
Sapendo che in quel caso si voleva leggere un file con una sola colonna, mentre ora si vuole leggere un file con 4 colonne, potete confrontare le due funzioni ReadFile(), identificare le differenze tra i due casi e capire come usare la versione proposta in questo esercizio.

```cpp
#include <vector>
#include <string>
#include <iostream>
#include <fstream>

bool ReadFile(string& fileName, vector< vector<double> >& data, int nColumns)
{
   ifstream file(fileName.c_str());
   if(!file.good()) { // if open fails, print message and return false
      cerr << "Failed to open file: " << fileName << endl;
      return false;
   }

   // Clear content of data and make its size match nColumns. This latter step
   // ensures that we can use data[i] without risking to access invalid memory
   data.clear();
   data.resize(nColumns);

   while(file.good()) { // loop over the rows of the file
      vector<double> row(nColumns); // temp vector used to read the file

      for(int i = 0; i < nColumns; i++) // read one row from file
        file >> row[i];

      if(!file.good()) // break if the file is not good (we are at end of file)
        break;

      // fill data with the values read from the file
      for(int i = 0; i < nColumns; i++)
        data[i].push_back(row[i]);
   }
   file.close();
   return true;
}
```

Materiale da consegnare: esercizio2.cpp

Esercizio 3

Esercizio da svolgere in laboratorio. Scrivere tre funzioni che calcolino le seguenti quantità

1. $S_0 = \sum_{i=1}^{N} \frac{1}{\sigma_i^2}$
2. $S_1 = \sum_{i=1}^{N} \frac{x_i}{\sigma_i^2}$
3. $S_2 = \sum_{i=1}^{N} \frac{x_i y_i}{\sigma_i^2}$

Le interfacce delle tre funzioni devono essere le seguenti:

```
double S0(vector<double>& sigma);
double S1(vector<double>& V1, vector<double>& sigma);
double S2(vector<double>& V1, vector<double>& V2, vector<double>&
sigma);
```

dove sigma, V1 e V2 sono dei vettori di double di dimensione N e che contengano rispettivamente i valori delle σ_i , delle x_i e delle y_i.

Per verificare il corretto funzionamento delle funzioni, creare un programma in grado di leggere il file dati_esp4.dat (riusare la funzione dell'esercizio 2), e che calcoli e stampi su schermo:

1. S_0(C4), ovvero applichi la funzione S_0 con sigma=C4
2. S_1(C1, C4), ovvero applichi la funzione S1 con V1 = C1, sigma = C4
3. S_2(C1, C1, C4), ovvero applichi la funzione S2 con V1 = V2 = C1, sigma = C4

dove con C1..C4 si sono indicate la prima...quarta colonna del file di input.
Se le funzioni sono implementate correttamente, i valori che si ottengono usando il file dati_esp4.dat sono i seguenti:

1. S_0(C4) = 10006.4
2. S_1(C1, C4) = 43.0185
3. S_2(C1, C1, C4) = 287.676

Materiale da consegnare: esercizio3.cpp

Esercizio 4

Esercizio da svolgere in laboratorio. Scrivere un programma per calcolare i parametri di un fit lineare e le incertezze ad essi associate utilizzando il metodo dei minimi quadrati. Usate come punto di partenza una copia dell'esercizio3, che poi andrete a modificare
Usate il comando cp: *cp esercizio3.cpp esercizio4.cpp*

Un fit lineare consiste nel determinare i parametri m e q della retta $y = mx + q$ che meglio approssima un set di dati sperimentali (x_i, y_i) ai quali sono associate le incertezze $(\sigma_{xi}, \sigma_{yi})$. Si dimostrerà più avanti nel corso di Lab1B che, nel caso in cui l'incertezza sui valori x sia trascurabile, la miglior stima di m e q si ottiene dalle seguenti formule:

$$m = \frac{S_{xy}S_0 - S_xS_y}{S_{xx}S_0 - S_xS_x} \qquad q = \frac{S_yS_{xx} - S_xS_{xy}}{S_{xx}S_0 - S_xS_x}$$

$$\sigma^2_m = \frac{S_0}{S_{xx}S_0 - S_xS_x} \qquad \sigma^2_q = \frac{S_{xx}}{S_{xx}S_0 - S_xS_x}$$

in cui, con riferimento alle funzioni dell'esercizio 3:
$S_0 = S_0(\sigma_y)$, $S_x = S_1(x, \sigma_y)$, $S_y = S_1(y, \sigma_y)$, $S_{xx} = S_2(x, x, \sigma_y)$ e $S_{xy} = S_2(x, y, \sigma_y)$

Si consideri l'espressione del periodo di un pendolo semplice: $T = 2\pi\sqrt{\frac{L}{g}}$, dove L è la lunghezza del pendolo e g è l'accelerazione di gravità. Elevando al quadrato entrambi i membri si ottiene

$$T^2 = \frac{4\pi^2}{g}L$$

A questo punto si possono fare le seguenti associazioni (linearizzazione):
$T^2 \Leftrightarrow y$, $L \Leftrightarrow x$, $\frac{4\pi^2}{g} \Leftrightarrow m$, $0 \Leftrightarrow q$

Si immagini di aver effettuato diverse misure del periodo T di oscillazione di un pendolo semplice al variare della sua lunghezza L, e di aver calcolato i quadrati dei periodi ottenuti e le incertezze ad essi associate mediante la propagazione degli errori. Si potrà quindi avere un file di dati in cui la prima colonna contiene i valori di L, la seconda i valori di T^2, la terza le incertezze sulle misure di L (che trascureremo) e la quarta le incertezze sui valori di T^2. Con un fit lineare sarà possibile ricavare i valori di m e q della retta che meglio approssima i dati.

1. Verificare che usando i dati contenuti nel file pendolo.dat si ottengono i seguenti valori
   m = 4.049 ± 0.066, q = -0.006 +- 0.016
2. Si noti che il valore dell'intercetta è compatibile con zero e che il valore di $4\pi^2/m$ è compatibile con il valore dell'accelerazione gravitazionale g
3. Fate un grafico del fit. Aprite il file pendolo.C ed eventualmente mettete i valori giusti di m, q e del nome del file da leggere, poi eseguite da terminale:
   ~$ root pendolo.C

Materiale da consegnare: esercizio4.cpp

Esercizio 5 (per casa)

Estendere il programma dell'esercizio 4 aggiungendo il calcolo del χ^2 e del χ^2 ridotto a partire dal file pendolo.dat

Data una serie di dati del tipo (x_i, y_i, σ_{yi}), che vengono modellizzati con la forma funzionale

y = f(x, p)

dove p rappresenta un insieme di parametri (m e q nel caso di una retta), il χ^2 è definito come

$$\chi^2 = \sum_{i=1}^{N} \frac{(y_i - f(x_i, p^*))^2}{\sigma_{yi}^2}$$

dove f(x, p*) sarebbe la forma funzionale valutata in x e per il set di parametri p* ottimali che sono stati ottenuti come risultato del fit. Ad esempio nel caso della retta f(x, p*) = m*x + q, dove m e q sono i valori ottenuti dal fit, e quindi

$$\chi^2 = \sum_{i=1}^{N} \frac{(y_i - (mx_i + q))^2}{\sigma_{yi}^2}$$

Il χ^2 ridotto è invece il rapporto tra il χ^2 e il numero di gradi di libertà, che sono dati dal numero di punti misurati meno il numero di parametri del fit (2 nel caso della retta).

Materiale da consegnare: compito4.cpp (AulaWeb entro il 2018-11-19)