Esame di Laboratorio di Fisica Computazionale

23 giugno 2011, ore 13.30

shell scripting

Nella directory `dati/` trovate cinque files `.dat`, ciascuno contenente un set di valori numerici, preceduti da una riga con il valore di una sezione d'urto.

Preparate uno script che ottenga i seguenti risultati:

- raccogliere in un unico file `xsec.res` le cinque righe con le cinque sezioni d'urto;
- cancellare la prima riga con la sezione d'urto, rinominando allo stesso tempo il file da `nomefile.dat` a `nomefile.res`.

Mathematica

1. Si calcolino gli autovalori della matrice

$$M = \begin{pmatrix} 1 & 9 & 5 \\ 8 & 8 & 2 \\ 3 & 4 & 4 \end{pmatrix} \tag{1}$$

Si chiami $X = (x, y, z)$ il vettore delle incognite e $K = (1, 2, 3)$ quello del termine noto; si risolva il sistema di equazioni

$$M \cdot X = K \tag{2}$$

2. Si definisca una funzione $f(k)$ che, per dato valore di k, disegna nello stesso grafico, nell'intervallo $x \in [-3, 3], y \in [-3, 3]$ le due superfici definite da:

$$\begin{aligned} z &= x + y + k \\ z &= x^2 + y^2 \end{aligned} \tag{3}$$

Si disegnino in particolare i grafici con $k = 2$ e con $k = -2$.
(facoltativo)
Si utilizzi Mathematica per determinare il valore di k per cui le due superfici sono tangenti in un solo punto.

3. Si risolva l'equazione differenziale:

$$r'(\theta) = b\, r(\theta), \qquad r(0) = a \tag{4}$$

Si tracci il grafico parametrico, con $\theta \in [0, 8\pi]$, della curva il cui punto generico è individuato dalla coppia $(x, y) = (r(\theta)\cos\theta, r(\theta)\sin\theta)$; si generino due grafici, ponendo in un primo caso $a = 0.1, b = 0.1$ e poi in un secondo caso $a = 0.01, b = 0.01$.

4. Si definisca una funzione $g(n)$ che, per dato n, generi una lista di n coppie di punti (x_i, y_i), con:

$$x_i = \frac{y_{i-1} - x_{i-1}\tan\left(\theta_{i-1} + \frac{\pi}{2}\right)}{\tan\theta_i - \tan\left(\theta_{i-1} + \frac{\pi}{2}\right)} \tag{5}$$

$$\begin{aligned} y_i &= x_i \tan\theta_i \\ \theta_i &= \theta_0 + \frac{2\pi}{n}\, i \\ \theta_0 &= \frac{\pi}{180}, \qquad x_0 = 0.01, \qquad y_0 = x_0 \tan\theta_0 \end{aligned} \tag{6}$$

Si utilizzi il comando `ListPlot` con l'opzione `PlotJoined->True` per congiungere la sequenza di punti così generata, con $n = 10$.

C++

Nell'esercizio di C^{++} è prevista l'implementazione di una classe *Histogram*, per la gestione e creazione di istogrammi, partendo da una lista di valori.

Nella prima parte dell'esame verrà chiesto di implementare un *contenitore* per i dati che servirà come base per l'istogramma. Nella seconda parte verrà utilizzato il contenitore dei dati per la creazione dell'istogramma vero e proprio.

Ogni parte è suddivisa in punti i quali possono essere obbligatori o facoltativi. I punti obbligatori debbono essere svolti in sequenza per il corretto funzionamento della libreria mentre i punti facoltativi (se non esplicitamente indicato) possono essere svolti in qualsiasi ordine. Il codice deve essere **correttamente compilabile** e accompagnato dal rispettivo **Makefile** funzionante (se il Makefile è assente saranno decurtati dei punti). Non saranno accettati codici non compilabili (abbiate cura di commentare la parte di codice che non riuscite sviluppare. In caso di indecisione sul voto tale parte sarà comunque parzialmente valutata). Il codice deve rispondere correttamente alle richieste previste nel main. In sede di valutazione il codice sarà comunque testato per verificarne l'affidabilità.

I punti obbligatori sono:

I Parte (a.1), (a.2), (a.3), (b.1), (b.2), (b.3)

II Parte (c.1), (c.2), (c.3), (c.4), (c.5), (c.6), (c.7)

1 I Parte: classe Vector

La classe **Vector** rappresenterà un generico vettore di **double** a dimensione variabile (che conterrà i dati raccolti da usare per la creazione vera e propria dell'istogramma). La classe **Vector** sarà composto da un array di **double** (dichiarato nella classe come puntatore), da un **int** (che rappresenterà la dimensione dell'array) e da una serie di metodi atti a inserire, gestire e modificare dinamicamente questo array.

(a) Costruttori/Distruttori:

(1 - Obb) Implementare il costruttore che accetta in ingresso un intero che rappresenta la dimensione dell'array (che conterrà i dati) e alloca correttamente lo spazio per il puntatore all'array.

(2 - Obb) Implementare il costruttore di copia.

(3 - Obb) Implementare il distruttore.

TIP 1 Ricordarsi di allocare (deallocare) correttamente l'array di double nel costruttore (distruttore).

(b) Operatori:

(1 - Obb) Implementare il metodo **size()** che mi restituisce la dimensione dell'array

(2 - Obb) Implementare l'operatore di assegnamento tra due oggetti **Vector**.

(3 - Obb) Implementare l'operatore [] che dato un intero i mi permette di accedere all'elemento $i-$esimo dell'array.

(4) Implementare il metodo **resize** che accetta un intero *new_size* e modifica la dimensione dell'array a *new_size*.

(5) Implementare il metodo **push_back** che mi permette di inserire un elemento in coda all'array.

(6) Implementare il metodo **pop_back** che mi permette di eliminare l'ultimo elemento in coda dell'array.

TIP 1 Ricordarsi che i due vettori possono avere dimensioni differenti. In tal caso è necessario riallocare correttamente la memoria.

TIP 2 L'operatore in (3) deve funzionare sia in lettura

$$\begin{array}{c} \textbf{Vector } v(10); \\ \textbf{double } a = v[3]; \end{array} \tag{7}$$

sia in scrittura

$$\begin{array}{c} \textbf{Vector } v(10); \\ \textbf{double } a = 2.5; \\ v[1] = a; \\ v[0] = 3.8; \end{array} \tag{8}$$

TIP 3 Ignorare i controlli sulla validità dei dati passati ai metodi.

2 II Parte: Histogram

In questa parte sarà chiesto di implementare una classe per la raccolta di dati (**double**) che verranno rappresentati sotto forma di istogramma. Si ricorda che un istogramma si costruisce dividendo un intervallo (x_{min}, x_{max}) in n_{bin} sotto-intervalli e contando quante volte in un certo sotto-intervallo cadono i valori del set di dati di cui si vuole avere un istogramma. Utilizzeremo la classe **Vector** implementata nel punto precedente per *conteggiare* il numero di uscite in ogni sotto-intervallo. Chiaramente, il numero di elementi che dovrà avere l'oggetto **Vector** sarà il numero di sotto-intervalli n_{bin}. Nella classe sarà presente anche una variabile **int** *total* che indicherà il numero di dati totali che sono stati inseriti nell'istogramma.

Numeriamo da 0 a $n_{bin} - 1$ il numero degli intervalli e sia **Vector** $v(nbin)$ dove $v[i]$ è il numero di conteggi nell'intervallo i. Se x è un **double** nell'intervallo $x \in (x_{min}, x_{max})$, il numero dell'intervallo a cui appartiene x è

$$\textbf{int } i = (\textbf{int})((x - xmin)/(xmax - xmin));$$

(c) Classe Histogram:

(1 - Obb) Implementare il costruttore che accetta in ingresso i **double** x_{min}, x_{max}, e l'**int** n_{bin} e inizializzare in modo opportuno la classe **Vector** e la variabile *total*.

(2 - Obb) Implementare il costruttore di copia.

(3 - Obb) Implementare il distruttore.

(4 - Obb) Implementare l'operatore di assegnamento.

(5 - Obb) Implementare gli operatori **Xmin**, **Ymin**, **NBins** e **NumData** che restituiscono rispettivamente il valore minimo e massimo dell'intervallo considerato, il numero di suddivisioni ed il numero totale di dati inseriti.

(6 - Obb) Implementare il metodo **Add** che accetta in ingresso un **double** x ed incrementa di uno il rispettivo intervallo ed il numero totale dei conteggi.

(7 - Obb) Implementare il metodo **Count** che accetta in ingresso un **int** i e restituisce il conteggio dell'intervallo i-esimo.

(8) Implementare l'operatore [] che accetta in ingresso un **double** x e restituisce 0 se $x < x_{min}$ o $x > x_{max}$, oppure un **int** che indica il numero di conteggi nell'intervallo che conterrebbe x.

(9) Implementare l'operatore () che accetta in ingresso un **double** x ed incrementa di uno il rispettivo intervallo ed il numero totale dei conteggi.

(10) Implementare il metodo **CenterBin** che accetta in ingresso un **int** i e restituisce il centro dell'intervallo i-esimo.

TIP 1 Ignorare i controlli sulla validità dei dati passati ai metodi.

3 III Parte: Main

(d) Main:

(1) Creare un file main che crei un oggetto di tipo **Vector**.

(2) Creare un oggetto di tipo **Histogram**.

(3) Creare un istogramma prendendo 2000 double estratti nell'intervallo $[0, 1)$ e sommandoli a blocchi di 5. Il risultato deve essere simile a quello riportato nella seguente figura:

```
0.00000 | |0
0.02000 | |0
0.04000 | |0
0.06000 | |0
0.08000 | |0
0.10000 |** |2
0.12000 |**** |4
0.14000 |*** |3
0.16000 |*** |3
0.18000 |****** |6
0.20000 |****** |6
0.22000 |**************** |16
0.24000 |********************** |22
0.26000 |*************************** |27
0.28000 |********************************************* |45
0.30000 |********************************************** |46
0.32000 |********************************************* |45
0.34000 |******************************************************************* |67
0.36000 |************************************************************************** |74
0.38000 |*********************************************************************************** |83
0.40000 |************************************************************************************************************ |108
0.42000 |**************************************************************************************** |88
0.44000 |******************************************************************************************************************** |116
0.46000 |******************************************************************************************************************** |116
0.48000 |************************************************************************************************************************ |120
0.50000 |******************************************************************************************************************** |116
0.52000 |***************************************************************************************************************************** |125
0.54000 |*********************************************************************************************************************** |119
0.56000 |***************************************************************************************************************** |113
0.58000 |********************************************************************************************************** |106
0.60000 |********************************************************************* |69
0.62000 |**************************************************************************** |76
0.64000 |****************************************************************** |66
0.66000 |********************************************* |45
0.68000 |***************************************** |41
0.70000 |*********************************************** |47
0.72000 |*********************** |23
0.74000 |***************** |17
0.76000 |**************** |16
0.78000 |******* |7
0.80000 |******* |7
0.82000 |****** |6
0.84000 |** |2
0.86000 | |0
0.88000 |* |1
0.90000 | |0
0.92000 |* |1
0.94000 | |0
0.96000 | |0
0.98000 | |0
```