Esame di Laboratorio di Fisica Computazionale

4 maggio 2011, ore 13.30

shell scripting

Nella directory `lettere/` trovate cinque files `.tex` con delle brevi lettere.
Scrivete un ciclo `for` per esaminare tutti i files e

- usate `sed` per modificare la data di ciascuna lettera da 15 marzo a 4 maggio
  Suggerimento: assegnate al file modificato un nuovo nome, p.es `nomefile2.tex`
- compilate il file con `latex` e trasformatelo in `.pdf` con il comando `dvipdf`.

Mathematica

Risolvere i seguenti problemi

1. Risolvere la seguente equazione:

$$x^3 + x^2 + x + 1 = 0 \tag{1}$$

   e presentare i risultati come una lista con le tre radici.

2. Risolvere il seguente sistema di equazioni:

$$\begin{cases} y = & x^3 + x^2 + x + 1 \\ y = & 4x + 3 \end{cases} \tag{2}$$

   e presentare i risultati come una lista con le tre coppie di radici.
   Disegnare in un unico grafico le due curve definite dalle due equazioni del sistema, con $x \in [-3, 3]$.

3. Disegnare la superficie parametrica 3D definita, con $r = 5,\ \theta \in [0, 2\pi],\ \phi \in [0, 2\pi]$, dalle seguenti equazioni:

$$\begin{aligned} x &= (r + \cos\theta)\ \cos\phi \\ y &= (r + \cos\theta)\ \sin\phi \\ z &= \sin\theta \end{aligned} \tag{3}$$

4. Risolvere la seguente equazione differenziale

$$y'(x) = -x^2 y(x)^2 \tag{4}$$

   Creare una lista di funzioni, generate, a partire dalla soluzione dell'equazione, assegnando alla costante di integrazione tutti i valori interi compresi tra -10 e 10.
   Disegnare la lista di funzioni in un solo grafico.

Risolvere almeno uno dei due seguenti problemi:

1. Risolvere il problema precedente discretizzando la soluzione nell'intervallo $[-3, 3]$.

   - Si scelga come punto iniziale $(x_0, y_0) = \{-3, 1\}$. Si scelga come incremento $\Delta x = 0.03$

- Si usi il secondo membro della eq.(4) per valutare la pendenza della soluzione nel punto x_0.
- Si calcoli $(x_1, y_1) = (x_0 + \Delta x, y_0 + y'(x_0)\Delta x)$.
- Si iteri il procedimento, per determinare (x_2, y_2), prendendo questa volta come punto di partenza (x_1, y_1) e utilizzando questi valori per valutare $y'(x_1)$.
  Si proceda fino al calcolo di (x_{200}, y_{200})
- Si utilizzi `ListPlot` per disegnare i punti, con l'opzione `Joined->True`.
- Si ripeta il calcolo scegliendo come punto iniziale $(x_0, y_0) = \{-3, 1 - 0.1 * i\}$ con $i \in [1, 20]$

2. Si consideri una hamiltoniana di oscillatore armonico con una correzione perturbativa anarmonica (massa m, frequenza ω e $\hbar$ sono scelte in modo da riassorbire tutte le costanti numeriche).
$$H = p^2 + x^2 + \varepsilon p^4 \equiv H_0 + \varepsilon H_1 \tag{5}$$
Gli operatori di innalzamento e abbassamento sono definiti
$$\begin{aligned} a &= \frac{1}{2}(p - ix) \\ a^\dagger &= \frac{1}{2}(p + ix) \end{aligned} \tag{6}$$
Si definisca una funzione T di più argomenti il cui significato sarà quello di elemento di matrice del termine di perturbazione anarmonico tra due autostati dell'oscillatore imperturbato: in notazione usuale (non di `Mathematica`!) $T(i, H_1, j) = \langle i|H_1|j\rangle$
Il primo e l'ultimo argomento della funzione T indicano gli stati imperturbati su cui si applica la perturbazione. Gli argomenti intermedi, contengono le quattro potenze della perturbazione p^4 espressa in termini degli operatori a e $a^\dagger$. Si rispetti l'ordinamento degli operatori che, in generale, non commutano tra loro.
La definizione della funzione deve esplicitare l'azione di innalzamento o abbassamento degli operatori $a^\dagger$ e a, e cioè
$$a|n\rangle = \sqrt{n}|n-1\rangle, \quad a^\dagger|n\rangle = \sqrt{n+1}|n+1\rangle \quad . \tag{7}$$
La definizione della funzione, in assenza di operatori tra gli stati, deve esplicitare la relazione di ortogonalità degli autostati imperturbati, e cioè
$$\langle i|j\rangle = \delta_{ij} \tag{8}$$

Suggerimento importante:
dal momento che ogni potenza di p contiene una combinazione di a e $a^\dagger$, per ottenere tutti i termini della perturbazione si utilizzi la definizione
`t[i_,x___,j_] := Distribute[T[i,x,j]]`.
Tutte le definizioni richieste devono essere date per la funzione `T`*, ma ad essere invocata nei calcoli espliciti è la funzione* `t`.

- Si calcoli l'elemento di matrice $\langle 5^{(0)}|p^4|3^{(0)}\rangle$, dove $|3^{(0)}\rangle$ e $|5^{(0)}\rangle$ sono due autostati dell'oscillatore armonico imperturbato.
- Si calcoli la correzione al primo ordine $E_3^{(1)}$ all'autovalore E_3 dell'hamiltoniana imperturbata, secondo la formula:

$$\lambda_3^{(1)} = \langle 3^{(0)}|p^4|3^{(0)}\rangle \tag{9}$$

- Si calcoli la correzione al primo ordine $|3^{(1)}\rangle$ al terzo stato eccitato dell'hamiltoniana imperturbata, secondo la formula

$$|3^{(1)}\rangle = \sum_{j\neq 3} \frac{\langle j^{(0)}|p^4|3^{(0)}\rangle}{\lambda_j^{(0)} - \lambda_3^{(0)}} \, |j^{(0)}\rangle \quad , \tag{10}$$

  introducendo un simbolo `Ket[j,0]` per rappresentare gli stati imperturbati e ricordando che, con la scelta fatta per le costanti, $\lambda_j^{(0)} = 4\,(j + 1/2)$.

3. Si determini un quadrato magico `4x4` in cui la somma degli elementi di tutte le righe, di tutte le colonne e delle due diagonali sia uguale a 34.
   Suggerimento:

   - *si scrivano le dieci equazioni di vincolo per gli elementi generici a_{ij} della matrice incognita;*
   - *si risolva il sistema di queste dieci equazioni, che lascia 9 elementi espressi in termini degli altri 7;*
   - *con un ciclo* `Do` *con 7 iteratori si assegnino valori interi agli elementi rimanenti*
   - *si verifichi che tutti e 16 gli elementi sono positivi e, se questa condizione è soddisfatta, si scriva il quadrato magico sullo schermo e si interrompa il ciclo* `Do` *con un'istruzione* `Abort[]`

C++

Nell'esercizio di C^{++} è prevista l'implementazione di una classe *Matrice*, rappresentante matrici *reali* di dimensioni $N \times M$ (con N ed M arbitrari).

Nella prima parte dell'esame sarà richiesta l'implementazione degli operatori più importanti che caratterizzano l'algebra delle matrici $N \times M$. Nella seconda parte, verrà chiesto di implementare alcuni metodi per calcolare alcune proprietà fondamentali delle matrici.

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), (b.4), (b.5)

II Parte (c.1), (c.2), (c.3)

1 I Parte: classe Matrice

La classe **Matrice** rappresenta una generica matrice di dimensioni $N \times M$. Sarà dunque necessario allocare spazio sufficiente a contenere l'intera matrice e le due dimensioni. Le dimensioni della matrice non potranno essere modificate una volta istanziato l'oggetto.

(a) Costruttori/Distruttori:

(1 - Obb) Implementare il costruttore che accetta in ingresso due interi M ed N ed un $a0$ (settato a zero di default). Allocare lo spazio necessario per una matrice di dimensioni $M \times N$ e settare ogni elemento della matrice a $a0$.

(2 - Obb) Implementare il costruttore di copia.

(3 - Obb) Implementare il distruttore.

TIP 1 La dichiarazione di un array di array di **double** può essere fatta come

double **_array;

TIP 2 L'allocazione dinamica di un array di array avviene prima allocando le "righe"

_array = **new double**[nr];

dove *nr* è il numero delle righe, e successivamente lo spazio per *ogni* riga

_array[i] = **new double**[nc];

dove i è l'$i - esima$ riga e *nc* è il numero di colonne.

TIP 3 Ricordarsi che la deallocazione dinamica si esegue in modo inverso rispetto all'allocazione, ovvero si procede eliminando lo spazio per ogni riga e successivamente lo spazio allocato per le righe.

(b) Operatori:

(1 - Obb) Implementare i metodi **M()**, **N()** che restituiscono rispettivamente il numero di righe e di colonne.

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

(3 - Obb) Implementare l'operatore "()" che accetta in ingresso due interi $i,\ j$ e restituisce la *referenza* all'elemento della matrice in posizione $(i,\ j)$.

(4 - Obb) Implementare l'operatore "$*$" tra matrici.

(5 - Obb) Implementare l'operatore "^ " che accetta in ingresso un **unsigned int** k e restituisce la matrice elevata alla potenza k.

(6) Implementare l'operatore "$*$" che accetta in ingresso un **double** α e restituisce la matrice $C = \alpha B$.

(7) Implementare l'operatore "+" e l'operatore "+ =" tra matrici.

(8) Implementare l'operatore "==" e l'operatore "! =" tra matrici.

(9) Implementare l'operatore "−" e l'operatore "− =" tra matrici.

(10) Implementare gli operatori unari "+" e "−" necessari per la sintassi $C = +B$ e $C = -B$.

TIP 1 Nel punto (3) è fondamentale che il metodo restituisca una referenza. In caso contrario, tale elemento non potrebbe essere modificato dall'esterno con una sintassi del tipo

$$\mathrm{A}(0,\, 0) = 1;$$

TIP 2 Ricordarsi che $A^k = \prod_{i=1}^{k} A$ e dunque la $k - esima$ potenza di una matrice (quadrata) può essere calcolata iterativamente usando il prodotto tra matrici implementato nel punto (4).

TIP 3 Per quei metodi che richiedono una matrice quadrata, non preoccuparsi di verificare se realmente la matrice passata come argomento sia quadrata.

2 II Parte: Proprietà delle matrici

In questa sezione provvederemo all'implementazioni di alcuni metodi per calcolare alcune proprietà fondamentali delle matrici. In particolar modo sarà richiesto il calcolo della $p-$Norma di una matrice che definiamo come

$$\|A\|_p = \left(Tr\left[|A|^p\right]\right)^{1/p},$$

dove la matrice $|A|$ è ottenuta prendendo i valori assoluti della matrice A.

(c) Classe Particella (Madre):

(1 - Obb) Implementare il metodo **friend Tr** che accetta in ingresso una matrice e ne calcola la traccia.

(2 - Obb) Implementare il metodo **friend Abs** che accetta in ingresso una matrice e restituisce la matrice dei valori assoluti.

(3 - Obb) Implementare il metodo **friend Norm** che accetta in ingresso una matrice ed un **unsigned int** e ne calcola la $p-$Norma.

(4) Implementare il metodo **friend Trasposta** che accetta in ingresso una matrice e ne restituisce la trasposta.

(5) Implementare il metodo **friend Randomize** che accetta in ingresso una matrice e assegna agli elementi di tale matrici reali casuali tra $(-1, 1)$.

TIP 1 Per quei metodi che richiedono una matrice quadrata, non preoccuparsi di verificare se realmente la matrice passata come argomento sia quadrata.

3 III Parte: Main

(d) Main:

(1) Creare un file main che crei una matrice di dimensione 3×3.

(2) Assegnare dei valori alla matrice.

(3) Calcolare la $3-$Norma.