Esame di Laboratorio di Fisica Computazionale

16 aprile 2019, ore 9.00

1 Mathematica

1. • • •○ Si studi la seguente equazione differenziale che descrive un pendolo di frequenza caratteristica ω_0, smorzato da un termine proporzionale a γ e forzato con intensità f_0 e frequenza ω.

$$\begin{cases} x''(t) + \gamma x'(t) + \omega_0^2 x(t) = f_0 \cos(\omega t) \\ x'(0) = 1, \qquad x(0) = 0 \end{cases} \tag{1}$$

   (a) Si studi il caso puramente smorzato ($f_0 = 0$) e si disegnino per $t \in [0, 20]$ tre soluzioni sotto, sopra e al punto critico, con $\gamma = \omega_0/10$, $10\omega_0$, $2\omega_0$ rispettivamente.
   La soluzione per il valore critico di γ va calcolata separatamente.

   (b) Si studi il caso puramente forzato ($\gamma = 0$) e si disegnino per $t \in [0, 100]$ tre soluzioni sotto, sopra e al punto critico, con $\omega = \omega_0/10$, $10\omega_0$, ω_0 rispettivamente, avendo posto $f_0 = 1$.
   La soluzione per il valore critico di ω va calcolata separatamente.

   (c) Fissati f_0 e γ diversi da zero, fissato inoltre ω_0, si studi il comportamento dell'oscillatore per $t \in [0, 100]$, al variare di ω.

2. • ○ ○○ Data la matrice

$$A = \begin{pmatrix} -5 & -9 & 8 \\ 9 & 3 & -9 \\ 10 & 2 & 7 \end{pmatrix} \tag{2}$$

   se ne calcolino autovalori e autovettori usando i comandi nativi di `Mathematica`.

3. • • ○○ Si calcoli il polinomio caratteristico della matrice A e quindi si determinino le sue radici.

4. • • ○○ Usando i risultati del punto precedente, calcolare gli autovettori di A, senza usare il comando `Eigenvectors`.
   Una componente deve essere assegnata dall'utente.

5. • • • • Scrivere una procedura che calcoli gli autovettori di una generica matrice $n \times n$, senza usare il comando `Eigenvectors`, ma ricorrendo al comando `Module` per organizzare il codice.

6. • • • ○ Si scomponga la matrice A nel prodotto $A = U \cdot S \cdot O^T$, dove U, O sono matrici ortogonali, mentre S è una matrice diagonale, seguendo i passaggi indicati qui di seguito:

   (a) introdurre la matrice ausiliaria $M = A^T \cdot A$ e calcolarne autovalori e autovettori; si chiami V la matrice degli autovettori restituita da `Mathematica` (autovettori riga); si

chiami S^2 la matrice diagonale avente come elementi lungo la diagonale gli autovalori di M; valutare numericamente le espressioni con 8 cifre significative, salvando il risultato in questa approssimazione;

(b) utilizzare la procedura di ortonormalizzazione di Gram-Schmidt per creare, a partire da V, una matrice O contenente tre vettori ortonormali (autovettori riga);

(c) determinare U utilizzando l'equazione $U = A \cdot O \cdot S^{-1}$ (essendo $S^2 = S \cdot S$)

(d) verificare la scomposizione $A = U \cdot S \cdot O^T$

7. • • •• Si scriva una procedura che calcoli la derivata rispetto a una variabile z (a scelta dell'utente) di un polinomio in z di grado arbitrario.

2 C++: Esercizio

Si risolva l'esercizio proposto. Per facilitare la correzione, se possibile includere tutto in un unico file sorgente. La sufficienza è raggiunta risolvendo correttamente i primi tre punti.

Esercizio

Si vuole scrivere un semplice database di pubblicazioni scientifiche.

1. Si scriva una classe `Pub`, che rappresenterà una pubblicazione. Si mettano tra i membri *private* il titolo (una variabile di tipo `std::string`) e l'anno di pubblicazione (un intero). Si scriva un opportuno costruttore che inizializzi i due membri, con l'anno 2019 come *valore di default*.

2. Si scriva il *copy constructor* di `Pub` (usando la sintassi della *lista di inizializzazione*) e una funzione `Print`, che scriva su `cout` il titolo, seguito dall'anno tra parentesi tonde.

3. Si scriva un particolare costruttore che prenda due pubblicazioni e ne costruisca una nuova uguale alla piú recente tra le due.

   Bonus: si usi la sintassi della lista di inizializzazione, scrivendo un costruttore con corpo vuoto {}.

4. Si scriva il *move constructor* di `Pub`. Nota: `std::string` è una classe che gestisce una risorsa, dunque la sua semantica di spostamento è diversa da quella di copia. Si provi la funzionalità di questo costruttore con il seguente `main`:

   ```
   Pub pub1("C++");
   Pub pub2(std::move(pub1));
   pub1.print(); pub2.print();
   ```

   Si commenti brevemente: quale funzione si è occupata di modificare lo stato della stringa membro di `pub1`, in modo da trasferire l'*ownership* della risorsa?

5. Si scriva una classe `Researcher`, che aggregherà le pubblicazioni di un ricercatore. Tra i membri *private* si metta un `std::vector` di pubblicazioni. Si scriva un costruttore che prenda come argomento un `std::vector` di pubblicazioni e lo usi per inizializzare il vettore membro.

6. Si scriva una funzione membro `n_pub` che restituisca il numero di pubblicazioni (senza controllare la presenza di eventuali ripetizioni). Si scriva inoltre una *access function* `get_pubs` che restituisca il vettore privato, ma solo in lettura. Si commenti brevemente: come può essere modificata la funzione per permettere l'accesso anche in scrittura?

7. Si scriva una classe `Database`, che avrà come membri privati un `std::vector` di ricercatori, e un `std::map`. Quest'ultimo conterrà il numero di citazioni di ogni pubblicazione. Perché la mappa funzioni sarà necessario implementare una ulteriore funzionalità tra i membri di `Pub`, che permetta di confrontare due pubblicazioni: lo si faccia in modo che due `Pub` siano considerate uguali se hanno lo stesso titolo, a prescindere dall'anno di pubblicazione. (Non è richiesto scrivere i costruttori di `Database`.)

8. Si scriva una funzione `update` (membro di `Database`), che prenda come argomento un ricercatore e aggiorni entrambe le strutture dati. Dovrà cioè aggiungere il ricercatore nel vettore (senza fare controlli su eventuali duplicati) e aggiungere ogni sua pubblicazione tra le chiavi della mappa. Nota: si può fare affidamento sul fatto che `map::operator[]` quando non trova una chiave la inserisce e inizializza il valore corrispondente a 0.

9. Si scriva una funzione membro `cite`, che prenda una pubblicazione e un intero n, e aumenti di n il numero di citazioni della pubblicazione, ma solo se questa esiste nel database (altrimenti non deve fare nulla).

10. Si scriva una funzione globale `common_pubs`, che prenda come parametri due ricercatori e restituisca il numero di pubblicazioni che hanno in comune. Attenzione: nel caso in cui i vettori dei due `Researcher` contengano duplicati questi non devono essere contati piú di una volta. A questo scopo si consiglia di usare degli oggetti d'appoggio di tipo `std::set<Pub>`.

    Bonus: si utilizzi l'algoritmo `set_intersection` [difficile].

3 Shell scripting

Con uno script si cancellino le linee pari dal file allegato `dati.dat`.