Corso di python novembre-dicembre 2020 - ISM
Fulvio Paleari – fulvio.paleari@mlib.ism.cnr

---

1. Introduzione a python I

a. Utilizzando `numpy`, generate una matrice 3×3 contentente elementi casuali compresi fra -1 e 1 e trovatene il determinante utilizzando la regola di Sarruss. Fate una funzione per generare la matrice e un'altra per il calcolo del determinante. Controllate il risultato utilizzando il metodo di `numpy` per trovare il determinante di una matrice.

b. Considerando la matrice 3×3 costruita precedentemente, calcolatene autovalori e autovettori con gli opportuni metodi `numpy`, dopodiché visualizzatene lo spettro usando `matplotlib.pyplot` secondo la seguente formula:

$$S(\omega) = \mathrm{Im}\left\{\sum_{i=1}^{3} \frac{|\mathbf{e}_i|^2}{\omega - E_i - \mathrm{i}\eta}\right\},$$

dove $\mathbf{e}_i$ è un autovettore e E_i il corrispondente autovalore della matrice 3×3, mentre $\eta > 0$ rappresenta un parametro di *linewidth* (molto piccolo rispetto alla scala di ω considerata) introdotto *ad hoc*.

c. La sequenza di Fibonacci è definita dalla seguente realazione ricorsiva:

$$F_n = F_{n-1} + F_{n-2} \qquad \text{con } F_1 = 1 \text{ e } F_2 = 1.$$

Quindi i primi 12 termini saranno:

$$F_1 = 1,\ F_2 = 1,\ F_3 = 2,\ F_4 = 3,\ F_5 = 5,\ F_6 = 8,\ F_7 = 13,\ F_8 = 21,\ F_9 = 34,\ F_{10} = 55,\ F_{11} = 89,\ F_{12} = 144.$$

Il dodicesimo termine, F_{12}, è il primo termine formato da tre cifre.
Qual è l'indice del primo termine della sequenza di Fibonacci formato da 10 cifre? Potete usare funzioni, cicli `for`, `if` e `numpy` array.

[Questa è una versione del problema 25 di Project Euler]

d.

3. Visualizzazione dati

4. Analisi dati

5. Programmazione orientata ad oggetti

a. È possibile stimare il valore di π attraverso una simulazione MonteCarlo. Per farlo si campiona l'area di un quadrato di lato uguale a 2, ovvero si estraggono attraverso un generatore di numeri random coppie (x, y) di numeri nell'intervallo $[-1, 1]$. Per ogni coppia estratta si controlla se il punto è interno a un cerchio di raggio 1, ovvero si verifica che $x^2 + y^2 <= 1$. Il rapporto fra il numero di punti interni al cerchio e il numero totale di punti tende al valore $\pi/4$ al crescere del numero di campionamenti.

Realizzate una classe python che gestisca questa simulazione. La classe richiede il numero di campionamenti come parametro da passare al costruttore (metodo `__init__`) e definisce un metodo `run` che esegue la simulazione. Oltre al valore finale il metodo deve produrre il plot di convergenza, ovvero il plot del valore del rapporto durante la simulazione. A questo scopo è necessario definire come membro della classe una lista, o un array `numpy`, che contenga i valori del rapporto punti interni / punti totali durante la simulazione.

[Esercizio proposto da Marco D'Alessandro]