Introduzione al Metodo Monte Carlo per l'ingegneria

Attraverso l'utilizzo del linguaggio Python

Luca Morselli

Laboratori Nazionali di Legnaro, INFN

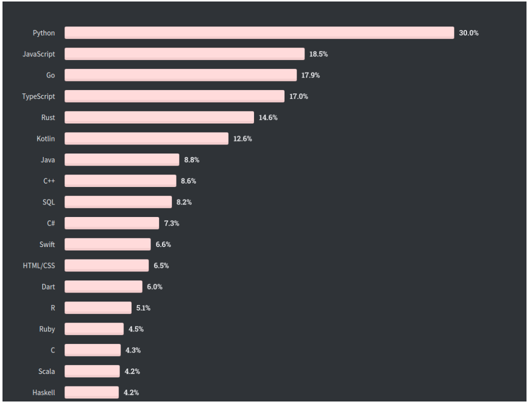
Version: December 5, 2021

Panoramica

Cos'è Python?

python™

- Python è un linguaggio di **scripting** ad alto livello diffuso all'interno di varie comunità scientifiche e industriali.
- Python è un linguaggio **orientato agli oggetti**.
- Python è un linguaggio **interpretato**.



Linguaggi di programmazione più ricercati. Fonte: Stack Overflow.

In Python, a differenza di altri linguaggi quali C,C++,Java non c'è la presenza di un delimitatore di fine istruzione a fine di ogni riga (usualmente il ;)

Dichiarazione delle variabili

```
a = 1
b = 'testo'
c = 2.4
d = 1/2
```

In Python il tipo di variabile non deve essere dichiarato esplicitamente come in altri linguaggi, il tipo di dato verrà riconosciuto automaticamente.

I tipi di dati fondamentali

Per vedere il tipo di dato possiamo usare la funzione `type(<varname>)`, in questo esempio otteniamo i seguenti risultati:

1. type(a) = int
2. type(b) = str
3. type(c) = float
4. type(d) = float

Le **liste** sono una collezione di elementi eterogenei.

Liste: Dichiarazione

```
mylist = [1,2,3,4,5,6]
```

Python: Aspetti Fondamentali
Le Liste

Gli elementi all'interno delle liste sono accessibili attraverso l'operatore [] che permette di accedere al singolo elemento o a determinate parti della lista(operazione denominata **slicing**). Vediamo alcuni esempi:

Liste: Dichiarazione ed Indicizzazione

```
mylist[0] = 1
mylist[:-1] = [1,2,3,4,5]
mylist[::2] = [1,3,5]
mylist[3:5] = [4,5]
```

Importante notare come l'indicizzazione parta da 0 e non da 1 (come ad esempio succede in MATLAB).

Python: Aspetti Fondamentali
Operazioni sulle liste

Le liste supportano diverse operazioni su di esse.
Le più ricorrenti sono le seguenti:

- `insert` : inserisce nella lista un elemento nella posizione desiderata
- `remove` : Rimuove un elemento dalla lista
- `pop` : rimuove l'ultimo elemento della lista
- `append` : aggiunge un elemento in fondo alla lista
- `extend` : aggiunge una lista in fondo ad un'altra lista
- `sort` : ordina gli elementi all'interno della lista

Python mette a disposizione un costrutto sintattico per la creazione di liste in maniera iterativa a partire da altre liste.
Alcuni esempi:

Liste: Dichiarazione ed Indicizzazione

```
# dichiarazione attraverso l'uso della list comphrension
mylist = [ i for i in range(0,10)]
# dichiarazione esplicita
mylist = [0,1,2,3,4,5,6,7,8,9]
```

Python: Le istruzioni condizionali

In Python sono presenti le istruzioni condizionali comuni a diversi linguaggi:

- `if/else` : blocco condizionale semplice
- `for` : iterazione su collezioni
- `while` : iterazione fino al raggiungimento di una condizione particolare

Python: File I/O

In Python risulta molto semplice anche leggere e scrivere su file.
Possiamo ad esempio leggere in input un file di testo (esempio un file csv) eseguire alcune operazioni su di esso e scrivere i risultati su file.

File I/O

```python
with open("inputfile.csv","r") as inputfile:

    # Leggiamo tutte le righe e inseriamole in una lista
    lines = inputfile.readlines()

    # facciamo qualche operazione con le informazioni

with open("outputfile.csv","w") as outputfile:

    # facciamo qualche operazione con le informazioni
```

SESSIONE PRATICA

`Notebooks/01_Python_Fundamentals.ipy`

Python: Le funzioni
Definizione

Le **funzioni** (o **metodi**) sono presenti in tutti i linguaggi di programmazione e servono per incapsulare blocchi di codice in modo da poterli riutilizzare in maniera semplice. Le funzioni prendono in input degli elementi chiamati **parametri** e ritorna un certo tipo di dato. Come con la dichiarazione di variabili non è necessario, in Python, dichiarare il tipo di dato che ritorna la funzione.

Dichiarazione di una funzione

```python
def multiply(a,b):
    return a*b
```

il dato in uscita è fornito attraverso l'uso della keyword `return`.

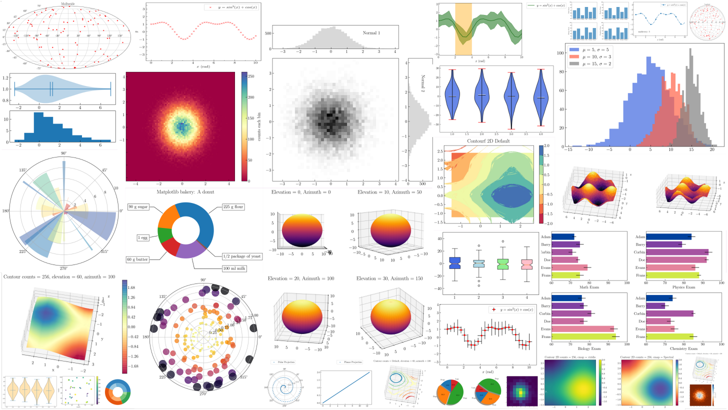
Python: La Libreria Matplotlib
Grafici

matplotlib

Matplotlib è la libreria più diffusa per la visualizzazione dei dati. Permette di creare diverse tipologie di grafico.

Vedremo alcuni esempi ma le potenzialità della libreria sono molto più vaste.

La documentazione ufficiale è disponibile **qui**



Diversi tipi di grafici realizzabili con Matplotlib.

Python: La Libreria Numpy
Calcolo scientifico

La libreria Numpy è una delle più diffuse per il calcolo algebrico e implementa diverse funzioni per le operazioni tra le matrici.

La documentazione ufficiale è disponibile **qui**

NumPy

The fundamental package for scientific computing with Python

Python: La Libreria Scipy

Calcolo scientifico

La libreria SciPy estende la libreria Numpy ed implementa diversi algoritmi utili in diverse aree scientifiche, alcuni esempi:

- Integrazione numerica
- Soluzione di Equazioni Differenziali
- Interpolazioni di serie di dati

SciPy

Fundamental algorithms for scientific computing in Python

La documentazione ufficiale è disponibile **qui**

Esempio: Analisi dati

Misura indiretta del modulo di Young di un filamento di Tungsteno

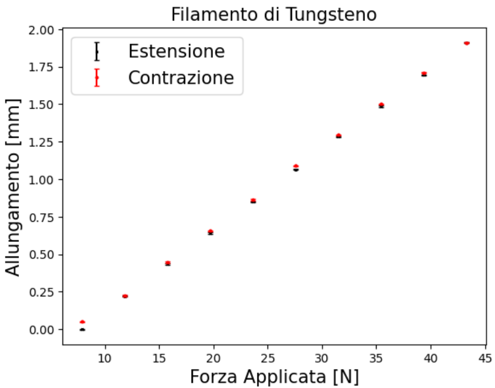
Supponiamo di misurare, utilizzando un estensimetro, l'allungamento di un filamento di tungsteno ($D = 0.0250 \pm 0.005$ mm , $x_0 = 1000 \pm 2$ mm) in funzione della forza applicata.

Dalla legge di Hooke sappiamo che:

$$\Delta x \equiv x_0 - x = K \cdot F \tag{1}$$

La costante di proporzionalità K è chiamata costante elastica, da alcune considerazioni geometriche si ottiene la legge di Hooke per l'allungamento di un filo elastico:

$$\Delta x = \frac{1}{E}\frac{x_0}{S}F \tag{2}$$



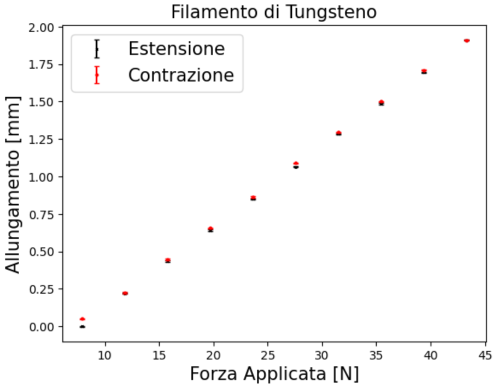
Misure di allungamento di un filamento di Tungsteno in funzione della forza applicata attraverso un'estensimetro.

Esempio: Analisi dati

Misura indiretta del modulo di Young di un filamento di Tungsteno

Possiamo quindi ricavare il modulo di Young dalla stima del coefficiente angolare della della retta interpolante le misure sperimentali:

$$E = \frac{x_0}{K \cdot S} = \frac{4x_0}{K \cdot \pi D^2} \tag{3}$$



Misure di allungamento di un filamento di Tungsteno in funzione della forza applicata attraverso un'estensimetro.

SESSIONE PRATICA

`Notebooks/02_Plotting.ipy`

Timeline

1777 — Calcolo di π attraverso il lancio di un ago.

Conte Buffon

Il Metodo Monte Carlo

Introduzione Storica

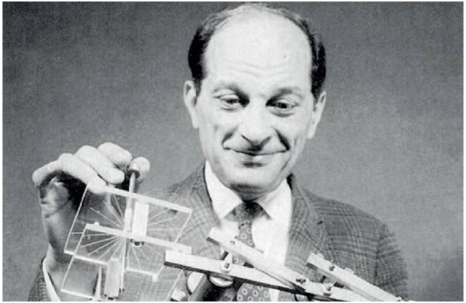
Timeline

1777	Calcolo di π attraverso il lancio di un ago.
1886	Laplace propone il calcolo di π attraverso l'utilizzo di numeri casuali in un rettangolo

Pierre-Simon Laplace

Timeline

1777	Calcolo di π attraverso il lancio di un ago.
1886	Laplace propone il calcolo di π attraverso l'utilizzo di numeri casuali in un rettangolo
1933	Fermi utilizza questo approccio per il calcolo di alcune proprietà dei neutroni appena scoperti



Enrico Fermi con il FERMIAC

Timeline

1777	Calcolo di π attraverso il lancio di un ago.
1886	Laplace propone il calcolo di π attraverso l'utilizzo di numeri casuali in un rettangolo
1933	Fermi utilizza questo approccio per il calcolo di alcune proprietà dei neutroni appena scoperti
1940	Utilizzo di simulazioni durante la fase iniziale del progetto Manhattan

Robert Oppenheimer, diretto del progetto Manhattan.

Il Metodo Monte Carlo

Introduzione Storica

Timeline

1777	Calcolo di π attraverso il lancio di un ago.
1886	Laplace propone il calcolo di π attraverso l'utilizzo di numeri casuali in un rettangolo
1933	Fermi utilizza questo approccio per il calcolo di alcune proprietà dei neutroni appena scoperti
1940	Utilizzo di simulazioni durante la fase iniziale del progetto Manhattan
1949	Von Neumann e Ulam coniano il termine "Monte Carlo"

Stanislaw M. Ulam, member of the Manhattan project.

Possiamo dire che il Metodo Monte Carlo è un **Approccio Matematico** per la risoluzione di problemi attraverso il campionamento di una serie di numeri casuali.

La base di questo approccio è un **algoritmo in grado di generare numeri detti "casuali"** distribuiti secondo la **distribuzione di probabilità** del fenomeno che stiamo osservando.

Generatori di numeri casuali

Vorremmo quindi avere un modo per generare dei numeri casuali attraverso un algoritmo.

Qualche idea?

Generatori di numeri casuali

Vorremmo quindi avere un modo per generare dei numeri casuali attraverso un algoritmo.

Qualche idea?

Purtroppo i computer non sono in grado di generare numeri "casuali" ma solamente sequenze di numeri che sembrano "casuali" per cui ci si riferisce a questi come numeri "pseudo-casuali".

Generatori di numeri Pseudo-Casuali

Questi algoritmi **deterministici** sono in grado di generare un numero casuale a partire dall'ultimo numero casuale generato:

$$x_{n+1} = f(x_n, x_{n-1}, \ldots, x_0)$$

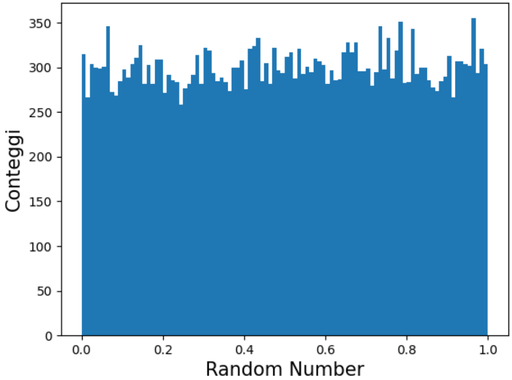
Come si può intuire le sequenze sono completamente definite dal numero di partenza x_0 che viene chiamato **seed**.

RANDU: un esempio da NON seguire

IBM nel 1960 sviluppa il seguente algoritmo per generare dei numeri casuali distribuiti uniformemente tra $[0,2^{31}\text{-}1]$ attraverso l'utilizzo della seguente formula:

$$x_{i+1} = a \cdot x_i + c \mod m$$

con a= 65539, m=2^{31} e c= 0.



Numeri casuali tra [0,1] ottenuti con l'algoritmo RANDU.

Il problema di questo algoritmo sta nel fatto che i numeri casuali generati sono correlati tra loro. Se infatti lo utilizziamo per calcolare triplette di numeri casuali distribuiti uniformemente nel cubo$[0,1]\times[0,1]\times[0,1]$ questi saranno disposti su 15 piani:

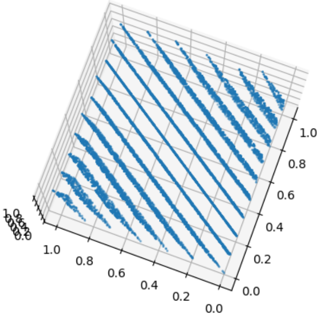
$$x_{k+2} = (2^{16}+3)\cdot x_{k+1} = (2^{16}+3)^2\cdot x_{k+1}$$

Espandendo il termine quadratico:

$$\begin{aligned} x_{k+2} &= (2^{32}+2^{16}\cdot 6+9)_{k+1} \\ &= [6\cdot(2^{16}+3)+2^{32}-9]x_k \end{aligned}$$

Applicando $\mod(2^{31})$:

$$x_{k+2} = 6x_{k+1} - 9x_k$$



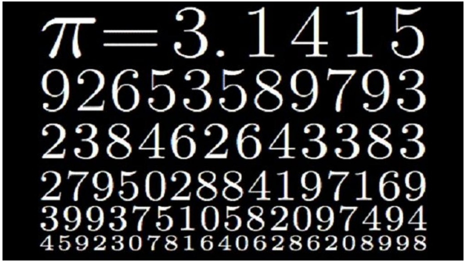
Numeri casuali tra [0,1] ottenuti con l'algoritmo RANDU.

Applicazione del metodo Monte Carlo: Calcolo di π

Abbiamo visto come sia possibile dunque, una volta trovato un buon generatore di numeri casuali, utilizzare le sequenze prodotte per la risoluzione di un problema numerico. Vediamo come sia possibile estrarre π a partire da numeri casuali.
Sappiamo che l'area del cerchio è data da:

$$A = \pi r^2$$

Calcolando l'area del cerchio è allora possibile calcolare una stima di π.



Applicazione del metodo Monte Carlo: Calcolo di π

L'equazione che descrive il cerchio è data da:

$$x^2 + y^2 = r^2$$

Possiamo allora invertirla e tenere la soluzione positiva:

$$y(x) = \pm\sqrt{r^2 - x^2}$$

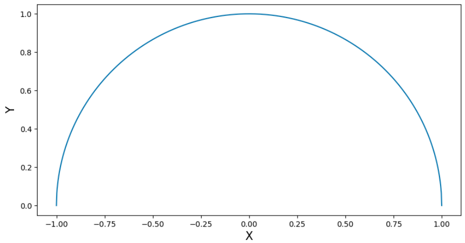
L'area sotto y(x) è dunque A=$\pi \frac{r^2}{2} e quindi$:

$$\pi \approx \frac{2A}{r^2}$$

Generando coppie di numeri casuali tra $[0,1]$ possiamo stimare l'area e dunque π:

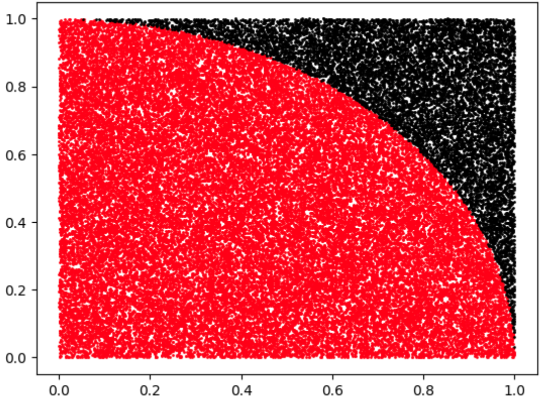
$$A \approx \frac{n}{N}\Delta x \Delta y$$

dove n è il numero di punti al di sotto di f(x) e N il numero totale di eventi generati

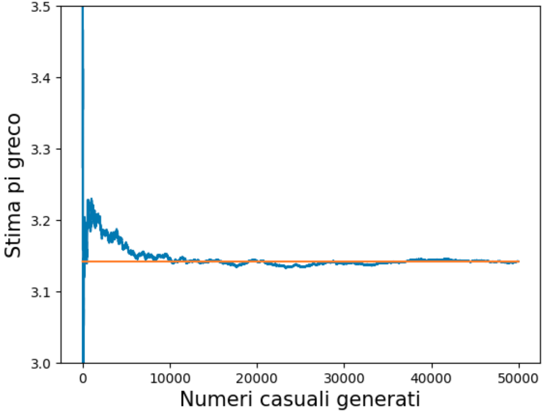


$$y(x) = \sqrt(1 - x^2)$$

Applicazione del metodo Monte Carlo: Calcolo di π



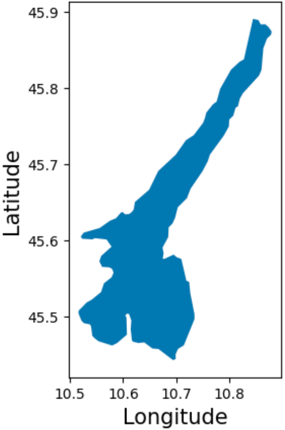
Numeri casuali generati tra [0,1]



Convergenza dell'algoritmo

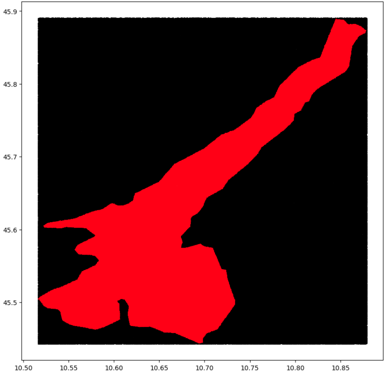
Esercizio: Stimare la superficie del Lago di Garda

Sirmione, Lago di Garda (BS)

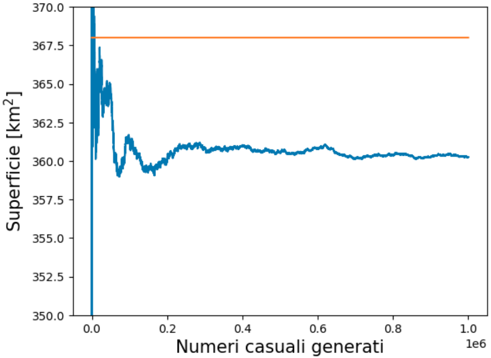


Shape del lago di Garda

Esercizio: Stimare la superficie del Lago di Garda



Lago Di Garda, 10^6 numeri casuali generati



Convergenza

SESSIONE PRATICA

`Notebooks/03_MonteCarlo_introduction.ipy`