

# Laboratorio di Programmazione e Calcolo

## Canale 2

### Appunti del corso

Simone Cacace e Giuseppe Visconti

Dipartimento di Matematica  
Sapienza Università di Roma

Anno Accademico 2025–2026

# Laboratorio – I/O e Gnuplot

## Esercizi

### E0. Matrice, norma, scrittura/lettura su file

Generare matrice  $A$  con  $n$  righe e  $m$  colonne, le cui entrate sono  $A[i][j] = (i+j) / (n*m)$ . Calcolare la norma di Frobenius della matrice, ovvero la quantità

$$\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^m |A_{ij}|^2},$$

che rappresenta una “distanza euclidea” degli elementi della matrice.

Scrivere i dati (numero di righe e colonne, matrice e norma) su file, rileggere i dati da file e confrontarli stampando un messaggio di verifica.

*Suggerimento: nel confrontare i risultati tra double, considerare una tolleranza numerica (ad esempio  $10^{-12}$ ). Se  $a$  e  $b$  sono due double il confronto tra  $a$  e  $b$  può essere fatto in base ad una tolleranza  $tol$ , ovvero  $|a-b| < tol$ .*

### E1. Energia cinetica di un sistema di particelle

Date  $n$  particelle, allocare dinamicamente due array  $m$  e  $v$  di lunghezza  $n$  per contenere le masse e le velocità delle particelle. Calcolare l’energia cinetica totale come

$$E = \frac{1}{2} \sum_{i=1}^n m_i v_i^2.$$

Salvare su file i contenuti degli array e l’energia totale. Rileggere da file i dati e ricontrillare il calcolo dell’energia. Confrontare i due risultati (in base ad una tolleranza) e visualizzare su schermo.

### E2. Moto rettilineo uniformemente accelerato

Scrivere un programma che, dati i valori iniziali della velocità  $v_0$ , dell’accelerazione costante  $a$  e del tempo totale  $T$ , calcoli e salvi su file la posizione  $s(t)$  e la velocità  $v(t)$  del corpo in moto rettilineo uniformemente accelerato.

- Si suddivida l’intervallo temporale  $[0, T]$  in un numero prefissato  $N$  di istanti equispaziati.
- Per ciascun istante  $t_i$ , si calcolino:

$$v(t_i) = v_0 + at_i, \quad s(t_i) = v_0 t_i + \frac{1}{2} a t_i^2$$

- I risultati vengano salvati in un file di testo (ad esempio `mrua.txt`) con tre colonne:

$$t_i \quad v(t_i) \quad s(t_i)$$

- Dopo l’esecuzione del programma, utilizzare `gnuplot` per rappresentare graficamente l’andamento di  $v(t)$  e  $s(t)$  in funzione del tempo.

*Suggerimento: si creino tre array di tipo double, ad esempio  $t$ ,  $v$  e  $s$ , di lunghezza  $N$ , in modo che alla posizione  $i$ -esima corrispondano i valori  $t_i$ ,  $v(t_i)$  e  $s(t_i)$ .*

**Opzionale:** realizzare la rappresentazione grafica direttamente dal programma C, inviando comandi a `gnuplot` tramite una pipe.

### E3. Oscillatore armonico semplice

Scrivere un programma che calcoli e salvi la posizione e la velocità di un oscillatore armonico semplice del tipo

$$x(t) = A \cos(\omega t + \varphi)$$

in  $N$  istanti temporali equispaziati nell'intervallo  $[0, T]$ .

- Il programma chiederà in input i parametri: ampiezza  $A$ , pulsazione  $\omega$ , fase iniziale  $\varphi$ , tempo totale  $T$  e numero di punti  $N$ .
- Per ciascun istante  $t_i$  calcolare

$$x(t_i) = A \cos(\omega t_i + \varphi), \quad v(t_i) = -A\omega \sin(\omega t_i + \varphi)$$

- Salvare i risultati in un file di testo (ad esempio `osc.txt`) con tre colonne:

$$t_i \quad x(t_i) \quad v(t_i)$$

- Dopo l'esecuzione del programma, utilizzare `gnuplot` per rappresentare graficamente  $x(t)$  e  $v(t)$ .

*Suggerimento: memorizzare i dati in tre array double ( $t$ ,  $x$ ,  $v$ ) di lunghezza  $N$  in modo che alla posizione  $i$ -esima corrispondano i valori  $t_i$ ,  $x(t_i)$  e  $v(t_i)$ , per poi scriverli su file.*

**Opzionale:** realizzare la rappresentazione grafica direttamente dal programma C, inviando comandi a `gnuplot` tramite una *pipe*.

### E4. Caduta libera

Scrivere un programma che simuli la caduta libera di una particella soggetta all'accelerazione di gravità  $g = 9.81 \text{ m/s}^2$ .

- Leggere da tastiera l'altezza iniziale  $h_0$ , la velocità iniziale  $v_0$  e il passo temporale  $\Delta t$ .
- Calcolare le altezze della particella a ogni intervallo temporale finché la posizione non diventa minore o uguale a zero (impatto con il suolo).
- Memorizzare le altezze in un array e stampare la tabella tempo-altezza su file.
- Rappresentare graficamente l'andamento dell'altezza nel tempo.

**Suggerimento: usare le equazioni del moto uniformemente accelerato:**

$$h(t) = h_0 + v_0 t - \frac{1}{2} g t^2.$$

**Opzionale:** realizzare la rappresentazione grafica direttamente dal programma C, inviando comandi a `gnuplot` tramite una *pipe*.