

# Relazione di laboratorio: algebra lineare

Samuele Bellini

17 Settembre 2025

## Esercizio 1 - Algebra delle matrici

(a)

Data la matrice:

$$A = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}$$

è possibile ricavarne gli autovalori e i corrispondenti autovettori utilizzando la funzione presente nel pacchetto NumPy di Python: `np.linalg.eig(A)`.

Si ottengono così gli autovalori:

$$\begin{aligned}\lambda_1 &= (-1 + i) \\ \lambda_2 &= (-1 - i)\end{aligned}$$

e i corrispondenti autovettori:

$$\begin{aligned}\mathbf{v}_1 &= (0.70710678, 0.70710678i) \\ \mathbf{v}_2 &= (0.70710678, -0.70710678i).\end{aligned}$$

Si può poi calcolare la matrice esponenziale  $e^{At}$ , in funzione del parametro reale  $t$ , tramite la relazione:

$$e^{At} = Ue^{Dt}U^{-1},$$

dove  $U$  è la matrice degli autovettori di  $A$ ,  $D$  è la diagonalizzazione di  $A$  ed  $e^{Dt} = \begin{bmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{bmatrix}$ . Si riportano i valori degli elementi della matrice esponenziale  $e^{At}$  in funzione di  $t \in [0, 5]$  in Fig. 1. Si può così osservare che i valori della matrice sono sempre reali e che, per  $t = 0$ ,  $e^{At} = I_2$ .

Questi risultati risultano verificati anche analiticamente, in quanto si può ricavare che:

$$e^{At} = e^{-t} \begin{bmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{bmatrix}.$$

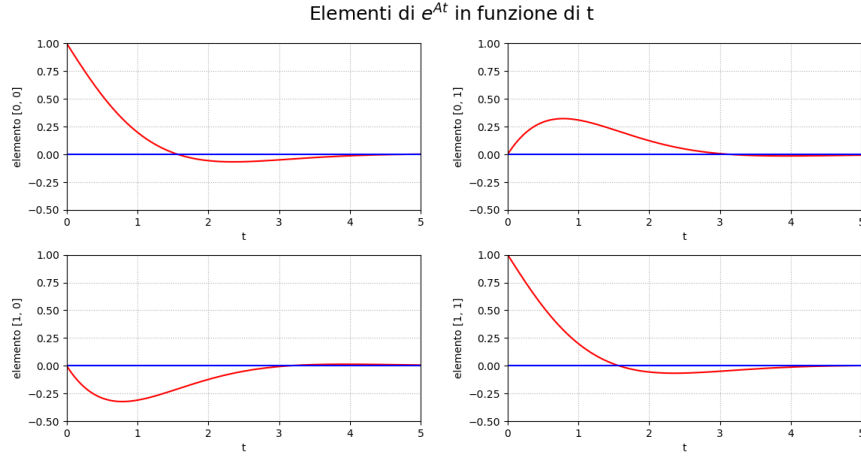


Fig. 1: Grafici dei valori degli elementi di  $e^{At}$  in funzione di  $t$  nell'intervallo  $[0,5]$ . In rosso è riportata la parte reale dell'elemento e in blu la parte immaginaria.

(b)

Dopo aver generato delle matrici  $n \times n$  con gli elementi presi da una distribuzione normale standard, operazione realizzabile in Python tramite la linea di codice `np.random.standard_normal((n,n))`, è possibile calcolarne gli autovalori e graficare questi ultimi (divisi per un fattore  $\sqrt{n}$ ) nel piano complesso per osservare come per  $n$  piccoli i punti sono concentrati principalmente sull'asse reale, mentre, al crescere di  $n$ , la distribuzione tende sempre più ad una distribuzione uniforme all'interno del cerchio di raggio unitario. Si può osservare questo comportamento in Fig. 2, dove sono rappresentati 20 000 autovalori per grafico, con dimensionalità  $n$  delle matrici quadrate crescente.

Per quantificare questo fenomeno si può calcolare il valore della varianza rispetto allo zero sia per la parte reale che per la parte immaginaria degli autovalori divisi per  $\sqrt{n}$ ; come ci si aspetta, per  $n$  piccoli si ottengono valori piccoli della varianza della parte immaginaria e grandi per la parte reale, mentre per  $n$  grandi le due varianze tendono sempre più a  $\frac{1}{4}$ , che è la varianza attesa per una distribuzione uniforme su un cerchio di raggio unitario. I valori di queste varianze sono riportati in Tab. 1.

$n$	$\sigma_{\text{Re}}^2$	$\sigma_{\text{Im}}^2$
4	0.415262	0.168300
10	0.309844	0.213243
100	0.255770	0.245610
1 000	0.250076	0.249655

Tab. 1: Valori delle varianze rispetto allo zero delle parti reali e immaginarie degli autovalori divisi per  $\sqrt{n}$  con i rispettivi valori della dimensionalità  $n$ .

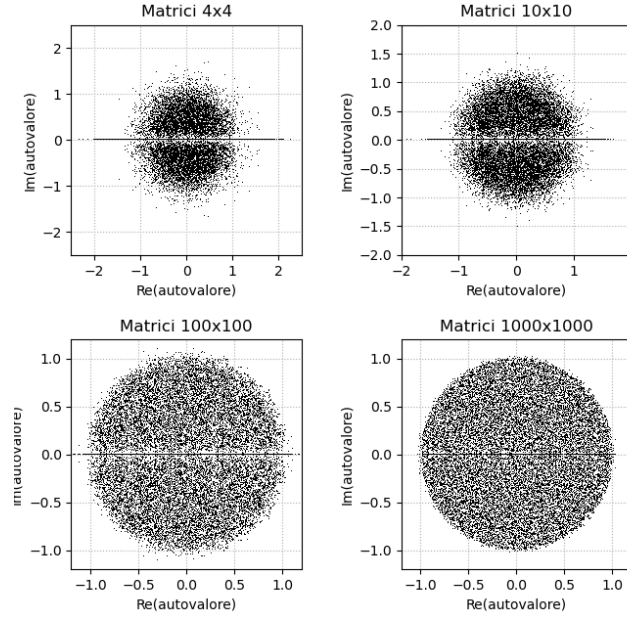


Fig. 2: Grafici degli autovalori di matrici  $n \times n$ , divisi per  $\sqrt{n}$ , nel piano complesso; in ogni figura sono rappresentati 20 000 autovalori, con  $n$  che assume i valori 4, 10, 100, 1 000.

Si può verificare che, al posto di dividere gli autovalori per  $\sqrt{n}$ , sarebbe stato equivalente generare le matrici con elementi presi da una distribuzione normale con deviazione standard pari a  $\sqrt{n}$  e media nulla.

(c)

Dato un dataset di molecole è possibile calcolare i momenti d'inerzia principali delle singole calcolando gli autovalori delle matrici momento d'inerzia:

$$I = \begin{bmatrix} \sum_i m_i (y_i^2 + z_i^2) & -\sum_i m_i x_i y_i & -\sum_i m_i x_i z_i \\ -\sum_i m_i x_i y_i & \sum_i m_i (x_i^2 + z_i^2) & -\sum_i m_i y_i z_i \\ -\sum_i m_i x_i z_i & -\sum_i m_i y_i z_i & \sum_i m_i (x_i^2 + y_i^2) \end{bmatrix},$$

dove  $m_i$  e  $(x_i, y_i, z_i)$  sono massa e posizione, relativa al centro di massa della molecola, dell'atomo  $i$ -esimo che compone la molecola. Per fare ciò, una volta che sono state ricavate le posizioni degli atomi che compongono la molecola rispetto al centro di massa di quest'ultima e poi calcolata la matrice  $I$  della singola molecola, basterà utilizzare la linea di codice `np.linalg.eigvalsh(I)` per ricavare i momenti d'inerzia principali, ordinati in ordine crescente.

Si possono poi classificare le molecole in base alle relazioni tra i momenti d'inerzia  $I_a, I_b, I_c$  (con  $I_a \leq I_b \leq I_c$ ) secondo il seguente criterio:

**Sferica:**  $I_a = I_b = I_c$

**Oblata:**  $I_a = I_b < I_c$

**Prolata:**  $I_a < I_b = I_c$

**Asimmetrica:**  $I_a < I_b < I_c$

Per fare ciò in Python si è preferito, invece di verificare che sussista l'uguaglianza esatta fra valori decimali, che darebbe risultati incorretti a causa delle imprecisioni dei floating point, oltre a possibili imprecisioni nel dataset stesso, andare a verificare che i valori dei momenti d'inerzia principali siano molto simili tramite la funzione `math.isclose()`, nella quale è stato passato 0.01 come valore di tolleranza relativa (si chiede quindi che la distanza fra i due valori sia minore o uguale all'1% del valore massimo).

In questo modo è stato ottenuto il conteggio per categoria dal dataset fornito riportato in Tab. 2.

Sferica	Oblata	Prolata	Asimmetrica
2	18	92	7053

Tab. 2: Numero di molecole per categoria dal dataset fornito.

## Esercizio 2 - Decomposizione SVD

(a)

Data la matrice:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \\ \epsilon & 0 & 0 & 0 \end{bmatrix}$$

è possibile valutare come i suoi autovalori e valori singolari varino per piccole perturbazioni del parametro  $\epsilon$ . Come caso di studio è stato preso  $\epsilon \in [0, 10^{-5}]$ . Per calcolare i valori singolari di  $A$  è stata utilizzata la funzione `np.linalg.svd(A)[1]`, dove con `[1]` si seleziona il secondo oggetto restituito dalla funzione `svd`, che corrisponde al vettore di valori singolari. Per calcolare gli autovalori di  $A$ , invece, è stata utilizzata la funzione `np.linalg.eigvals(A)`, la quale restituisce direttamente il vettore di autovalori. I risultati di queste operazioni sono riportati in Fig. 3. Si osserva così che i primi tre valori singolari rimangono costanti su tutto l'intervallo, mentre il quarto è sempre pari al valore della perturbazione  $\epsilon$ ; gli autovalori, invece, sono sempre reali o immaginari puri, e la relazione fra  $\epsilon$  e la parte non nulla è la stessa per ogni autovalore, a meno di riflessioni rispetto all'asse reale.

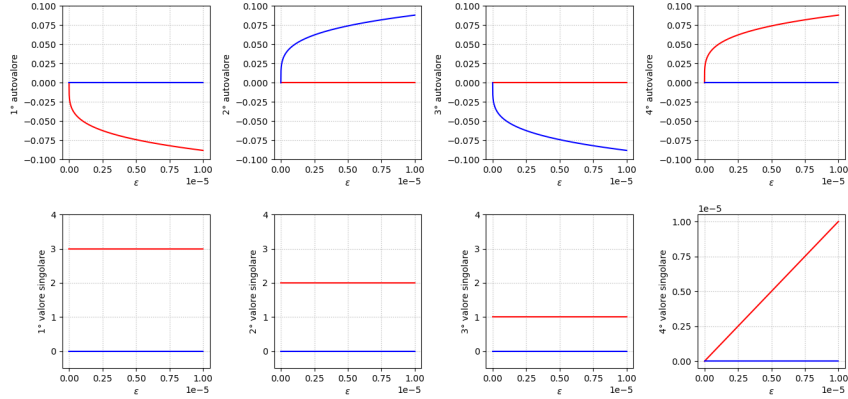
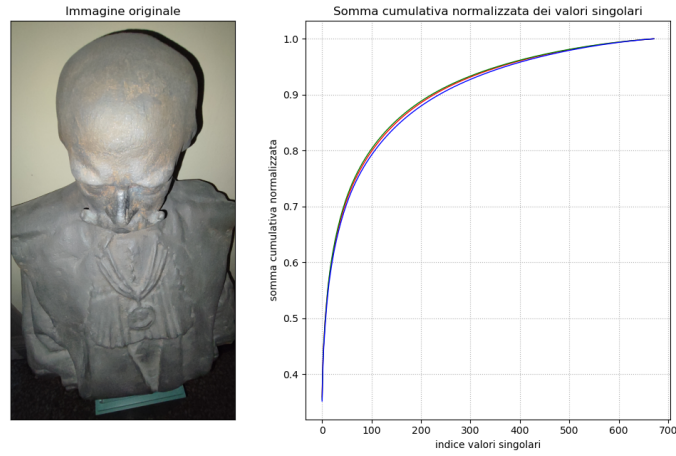


Fig. 3: Grafici degli autovalori e dei valori singolari di  $A$  in funzione della perturbazione  $\epsilon$ ; i primi sono riportati nella prima riga, i secondi nella successiva. In rosso sono riportate le parti reali, in blu le parti immaginarie. Si faccia attenzione al cambio di scala per il quarto valore singolare rispetto ai primi tre.

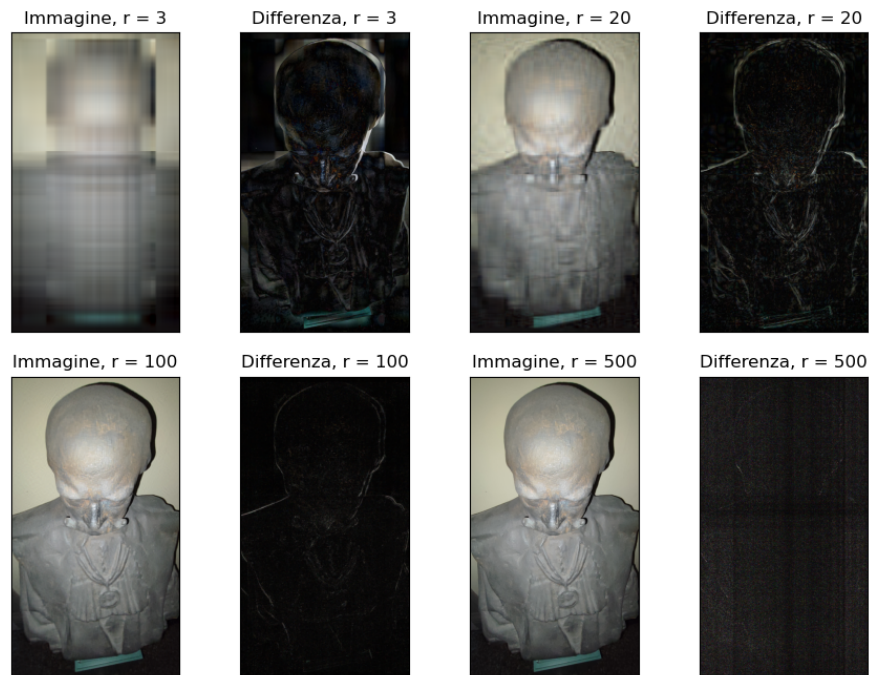
(b)

Data un'immagine è possibile rappresentarla come un insieme di tre matrici reali con elementi  $\in [0, 1]$ , rappresentanti le componenti rosso, verde e blu dei pixel che compongono l'immagine. Per studiare l'andamento dei valori singolari delle matrici e come appare l'immagine a vari livelli di accuratezza si potrebbe andare a lavorare sulla versione in bianco e nero dell'immagine, in modo da avere un'unica matrice, ma così facendo si perderebbero le informazioni cromatiche; per mantenere queste informazioni e ricavarne infine un'immagine a colori si è preferito studiare singolarmente le tre componenti, per poi ricavarne l'immagine complessiva ricombinandole. Innanzitutto è stata calcolata l'SVD per ogni matrice delle componenti che compone l'immagine scelta (riportata in Fig. 4 a sinistra) tramite la linea di codice `np.linalg.svd(img_i, full_matrices=True)`, dove `img_i` rappresenta una delle tre matrici di componenti, ed è stata graficata la somma cumulativa normalizzata dei valori singolari delle singole matrici in Fig. 4 a destra. Si osserva così che l'andamento dei valori singolari è pressoché identico per tutte le componenti, come ci si aspetta da un'immagine in cui non sono presenti preferenze per nessuna componente specifica (in un'immagine completamente rossa, ad esempio, si trovano valori singolari nulli per le componenti blu e verdi e non nulli solo nella componente rossa), e che la stragrande maggioranza dell'informazione è contenuta in pochi valori singolari; questo è evidente in Fig. 4 a destra, dove si può osservare come, con appena un terzo dei valori singolari (circa 250 su 700), la somma cumulativa ha già superato il 90% del suo valore massimo.



*Fig. 4: A sinistra l'immagine originale, a destra il grafico delle somme cumulative normalizzate dei valori singolari delle singole componenti cromatiche, dove il colore della linea indica la rispettiva componente.*

Sapendo come variano i valori singolari si può scegliere un valore di cutoff,  $r \in 1, 2, \dots, 700$  (il massimo valore di  $r$  è 700 perché, per questa immagine, è il valore minimo tra larghezza e altezza, in pixel, dell'immagine), che indica il numero di valori singolari che verranno utilizzati per ricostruire l'immagine dalla decomposizione SVD delle componenti. Si riporta, in Fig. 5, il risultato di questa operazione per  $r = 3, 20, 100, 500$ , assieme alla differenza fra l'immagine originale e la ricostruzione. Dalle figure si osserva bene come per valori di  $r$  molto piccoli la ricostruzione è molto grossolana e mantiene solo le forme essenziali, per valori di  $r$  più grandi si perdono solo piccoli dettagli, e come, per  $r = 500$ , nella differenza rimane quasi esclusivamente rumore. Si noti anche che, nonostante l'utilizzo di molti più dati per ricostruire l'ultima immagine, la differenza fra questa e la penultima sia minimale, evidenziando quindi come sia superfluo andare a utilizzare un numero così elevato di valori singolari, in quanto quasi tutta l'informazione è racchiusa in quelli precedenti.



*Fig. 5: Ricostruzione dell'immagine per  $r = 3, 20, 100, 500$  con la rispettiva differenza (in valore assoluto) dall'immagine originale. Si noti che nelle immagini della differenza è stato massimizzato il contrasto per renderle più visibili.*

## Codice

Il codice utilizzato per risolvere gli esercizi è consultabile su GitHub al link:  
<https://github.com/saemmie4/Lab-Data-Science>