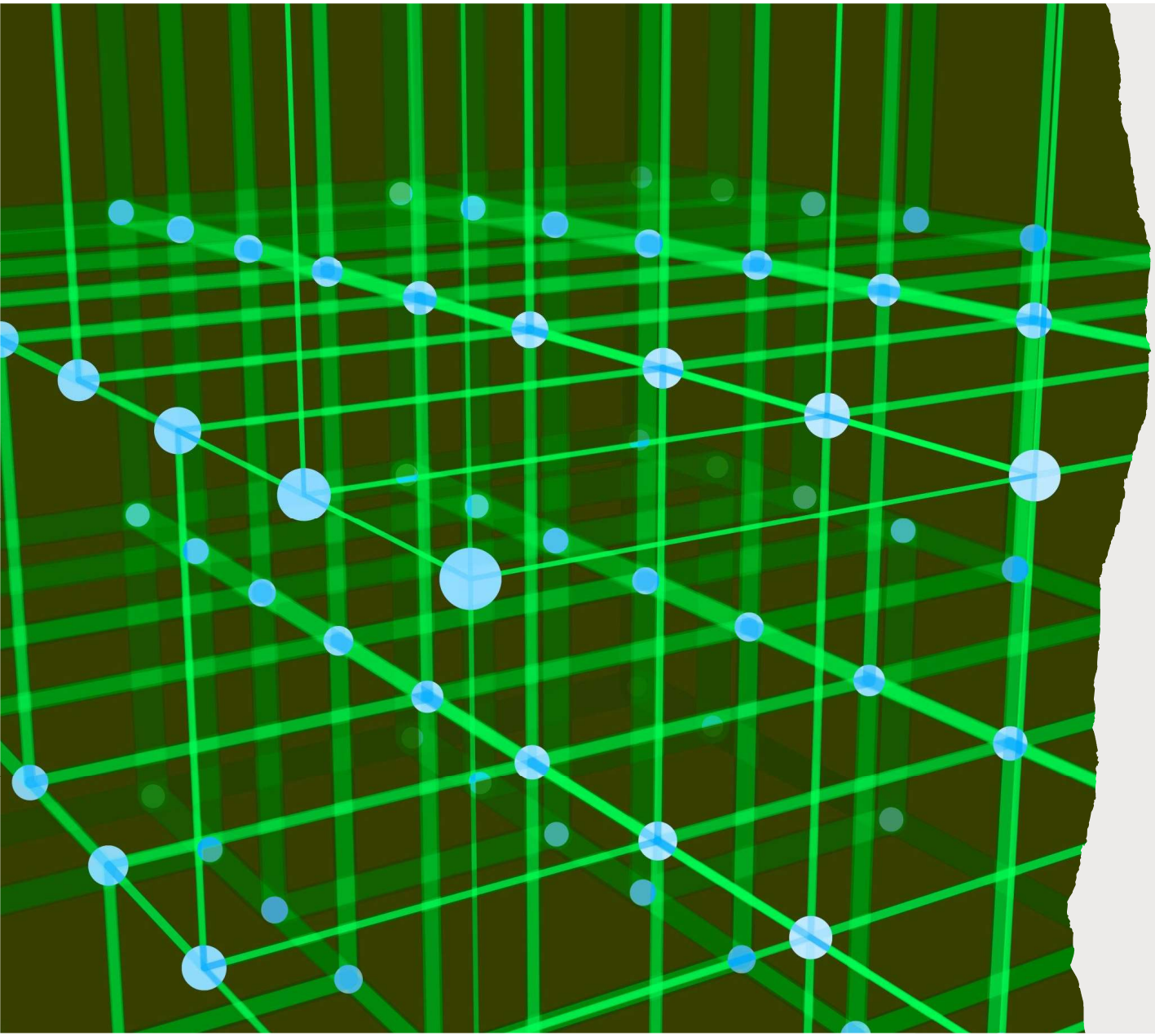
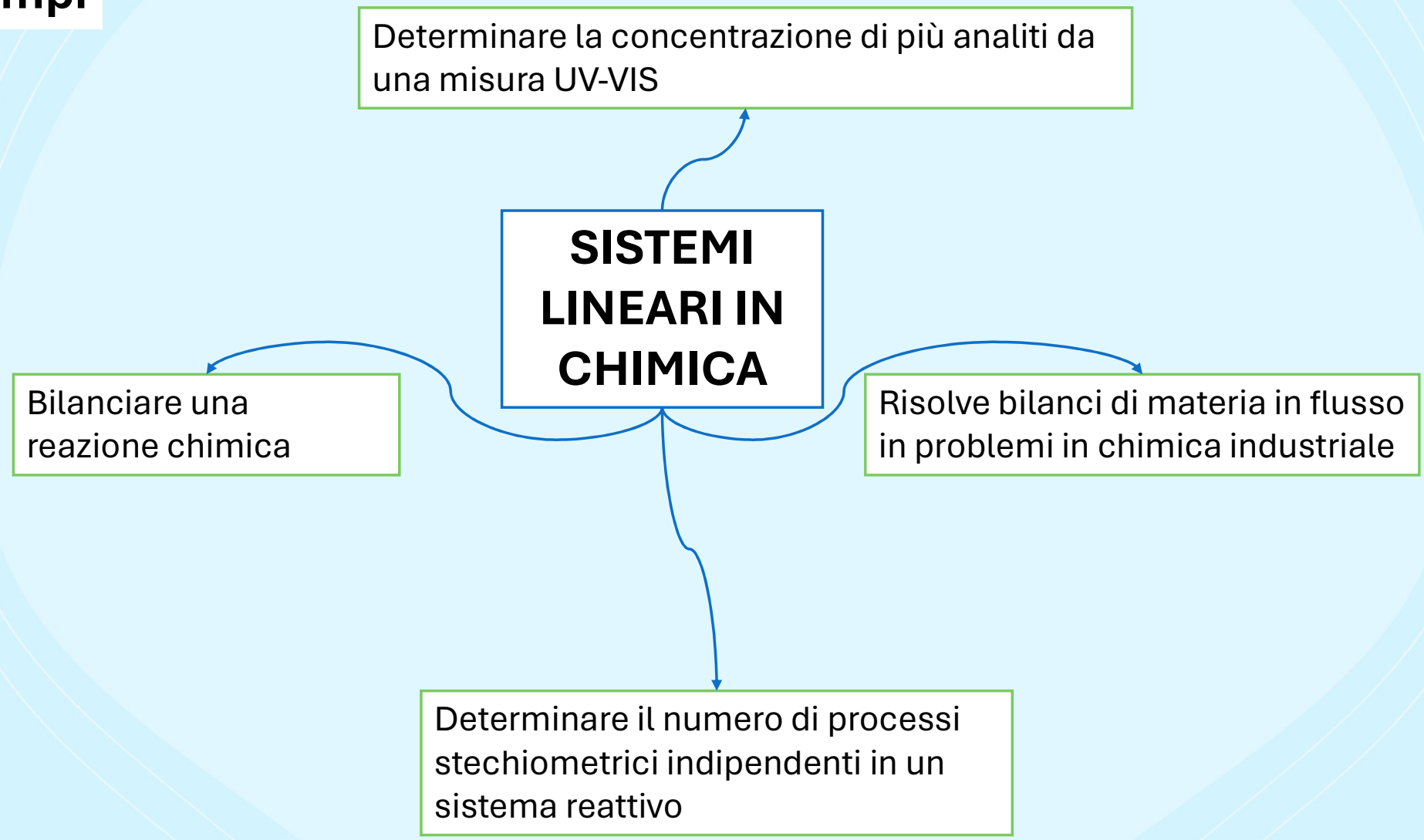


Sistemi Lineari

Obiettivi di apprendimento

- Rappresentazione matriciale di un sistema lineare
- Dedurre il numero di soluzioni attese in funzione delle proprietà della matrice associata
- Metodo di eliminazione per la soluzione
- Decomposizioni
- Problema agli autovalori
- Utilizzo del modulo `numpy.linalg` per la soluzione numerica di sistemi lineari con Python.

Esempi


Determinare la concentrazione di più analiti da una misura UV-VIS
SISTEMI LINEARI IN CHIMICA
Bilanciare una reazione chimica
Risolve bilanci di materia in flusso in problemi in chimica industriale
Determinare il numero di processi stechiometrici indipendenti in un sistema reattivo

Si vuole trovare il modo di risolvere un sistema lineare di m equazioni in n incognite

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,j}x_j + \cdots a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,j}x_j + \cdots a_{2,n}x_n = b_2 \\ \vdots \\ a_{i,1}x_1 + a_{i,2}x_2 + \cdots + a_{i,j}x_j + \cdots a_{i,n}x_n = b_i \\ \vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,j}x_j + \cdots a_{m,n}x_n = b_m \end{cases}$$

Il sistema può essere riscritto in forma matriciale come

$$\boldsymbol{Ax = b}$$

Esempio. Per il sistema:

$$\begin{cases} x_1 + 2x_2 - 3x_3 = 0 \\ x_2 - x_3 = 1 \\ x_1 + 4x_2 = -1 \end{cases}$$

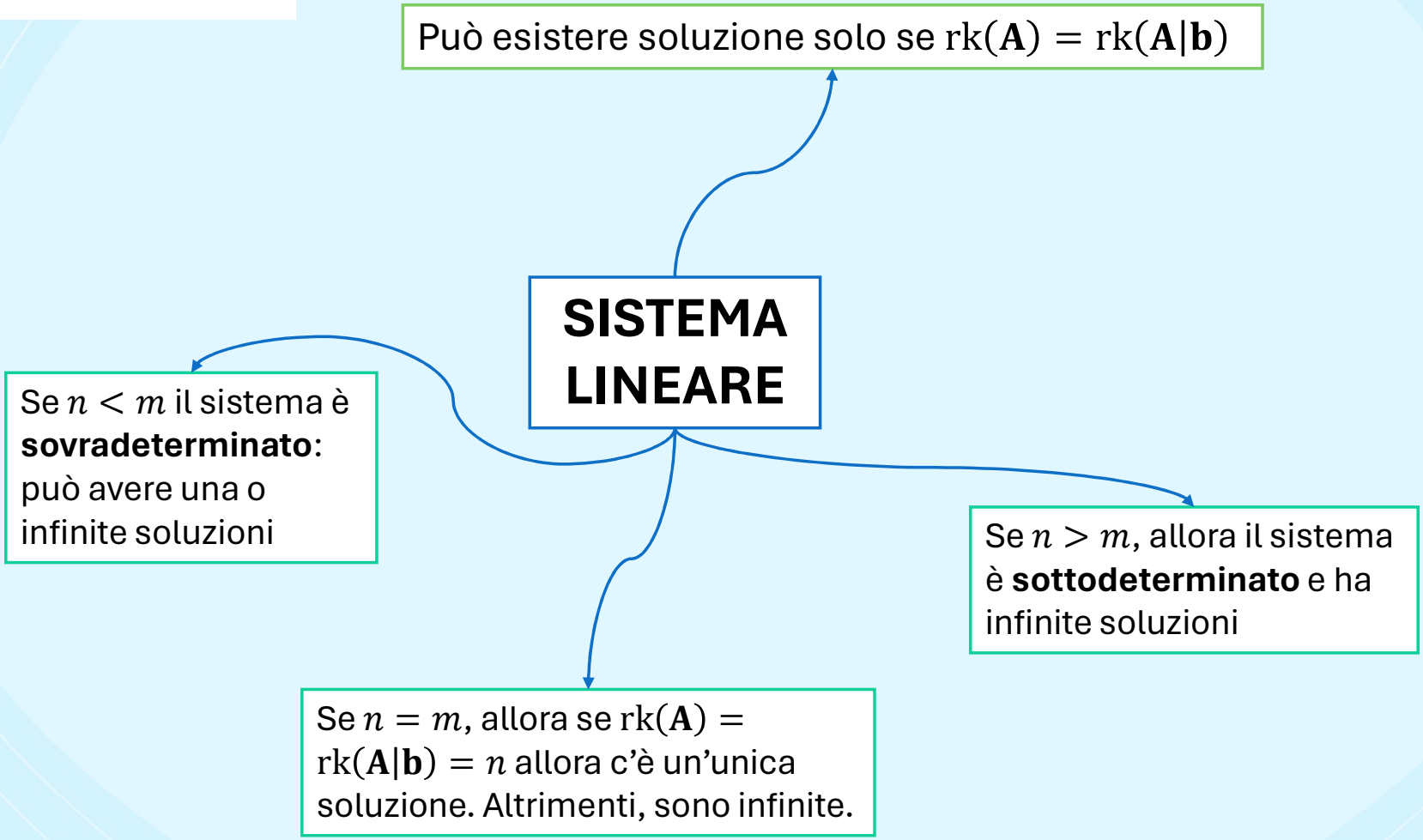
Si possono scrivere:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & -3 \\ 0 & 2 & -1 \\ 1 & 4 & 0 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$$

Si definisce la **matrice associata** $(\mathbf{A}|\mathbf{b})$

$$(\mathbf{A}|\mathbf{b}) = \begin{bmatrix} 1 & 2 & -3 & 0 \\ 0 & 2 & -1 & 1 \\ 1 & 4 & 0 & -1 \end{bmatrix}$$

Si può risolvere?



Generalmente, nei problemi riscontrati in chimica, in cui si debba risolvere un sistema lineare, ci si trova nel caso in cui il numero di incognite e quello di equazioni coincidano. La matrice dei coefficienti, quindi è quadrata.

Esempio. Si vuole preparare una soluzione di acqua e acido isobutirrico dalla massa totale di 5 g, e tale per cui il volume totale sia di 5.1 mL. Si calcolino le masse da pesare, assumendo i volumi additivi e sapendo che le densità di acqua e acido isobutirrico sono, rispettivamente, 0.99 g/mL e 0.97 g/mL.

$$\begin{cases} m_1 + m_2 = 5 \\ \dfrac{m_1}{\rho_1} + \dfrac{m_2}{\rho_2} = 5.1 \end{cases}$$

Come in questo esempio, spesso capita che il rango della matrice corrisponda al numero di righe (la matrice ha rango pieno).

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ \dfrac{1}{\rho_1} & \dfrac{1}{\rho_2} \end{bmatrix}$$

Essendo $\rho_1 \neq \rho_2$, le due righe della matrice sono linearmente indipendenti.

Calcoliamo il rango della matrice associata

$$(\mathbf{A}|\mathbf{b}) = \begin{bmatrix} 1 & 1 & 5 \\ \frac{1}{\rho_1} & \frac{1}{\rho_2} & 5.1 \end{bmatrix}$$

Si procede tramite il metodo di eliminazione: alla seconda riga si può sostituire l'esito della sottrazione della prima riga meno la seconda moltiplicata per ρ_1:

$$(\mathbf{A}|\mathbf{b}) = \begin{bmatrix} 1 & 1 & 5 \\ 0 & 1 - \frac{\rho_1}{\rho_2} & 5 - \rho_1 5.1 \end{bmatrix}$$

Per la coppia (ρ_1, ρ_2) data dal problema, gli elementi della seconda riga non sono tutti nulli, per cui il rango della matrice associata è uguale a quello della matrice $\mathbf{A}$. Non solo il sistema è risolvibile, ma ha una sola soluzione.

Essendo la matrice a rango pieno, il suo determinante è diverso da zero, ed esiste l'inversa $\mathbf{A}^{-1}$ t.c. $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$, per cui la soluzione al sistema può essere calcolata come

$$\begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \mathbf{A}^{-\mathbf{1}} \begin{bmatrix} 5 \\ 5.1 \end{bmatrix}$$

Come si calcola l'inversa di una matrice?

Python time

Il modulo `linalg` di `numpy` ha il metodo `inv` per invertire le matrici (come array numpy):

```
B = numpy.linalg.inv(A)
```

assegna alla variabile `B` l'inversa di `A`.

Il metodo `matmul` di `numpy` permette di moltiplicare le matrici:

```
C = numpy.matmul(A,B)
```

assegna alla variabile `C` l'esito del prodotto tra `A` e `B`.

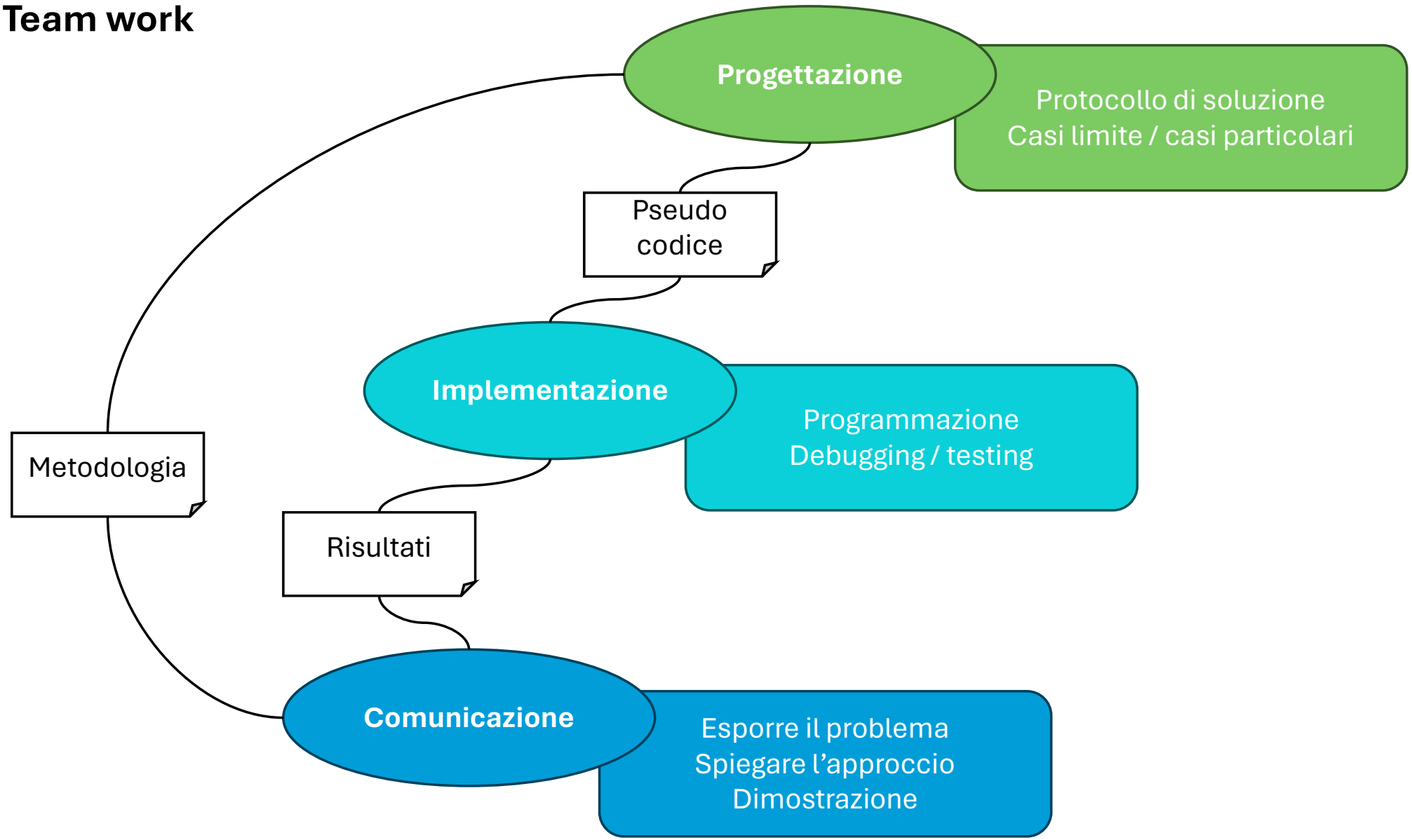
Team work

Costituire gruppi di lavoro di 5 o 6 persone per risolvere gli esercizi che seguono.

Il gruppo sia costituito da:

2 persone che si occupino della progettazione della risoluzione del problema
2 persone che si occupino dell'implementazione in Python della risoluzione del problema
1 o 2 persone che si occupino dell'aspetto comunicativo, per esporre in aula la risoluzione del problema

Team work


Progettazione
Protocollo di soluzione
Casi limite / casi particolari
Pseudo codice
Implementazione
Programmazione
Debugging / testing
Risultati
Comunicazione
Esporre il problema
Spiegare l'approccio
Dimostrazione
Metodologia

Esercizio

Una soluzione contiene nicotinammide (N) e acido nicotinico (A). Si misura lo spettro UV-Vis e si nota che a 261 nm l'assorbanza vale 0.920, mentre a 213 nm l'assorbanza vale 0.487. Per la nicotinammide, il coefficiente di estinzione molare alle due lunghezze d'onda vale: $\epsilon_{\mathrm{N}}\,(261\ \mathrm{nm}) = 4695$ $\mathrm{M^{-1}\ cm^{-1}}$ e $\epsilon_{\mathrm{N}}\,(213\ \mathrm{nm}) = 10$ $\mathrm{M^{-1}\ cm^{-1}}$, mentre per l'acido nicotinico: $\epsilon_{\mathrm{A}}\,(261\ \mathrm{nm}) = 5704$ $\mathrm{M^{-1}\ cm^{-1}}$ e $\epsilon_{\mathrm{A}}\,(213\ \mathrm{nm}) = 4992$ $\mathrm{M^{-1}\ cm^{-1}}$. Se il cammino ottico $b = 1$ cm, qual è la composizione molare della soluzione?

Preparare un Jupiter notebook per risolvere il problema in due modi:

1. usando il metodo di eliminazione

2. trasformando il sistema nel problema di algebra lineare

Esercizio

Bilanciare la reazione

$$x_1 \mathrm{H}_2 + x_2 \mathrm{O}_2 = x_3 \mathrm{H}_2\mathrm{O}$$

trasformando il problema nella risoluzione di un sistema lineare costruito sul bilancio di materia degli elementi che costituiscono le specie chimiche che partecipano alla reazione.

Si risolva il problema usando l'approccio di algebra lineare.

- Si può risolvere il sistema? Rispondere calcolando il rango della matrice dei coefficienti e di quella associata.
- Quante soluzioni si attendono? Discutere la risposta in base al tipo di sistema.
- Che proprietà hanno le soluzioni (sono indipendenti tra loro, o c'è una relazione tra tutte, o solo tra gruppi di soluzioni)?

Eliminazione di Gauss – Jordan

Costituire gruppi di lavoro di 5 o 6 persone. Ogni gruppo dovrà studiare in maniera autonoma la parte del Jupyter notebook in cui viene discusso il metodo di **eliminazione di Gauss-Jordan** e risolvere gli esercizi assegnati (almeno uno per gruppo).

Il primo obiettivo del gruppo di lavoro sia quello di studiare il metodo di eliminazione di Gauss, aiutandovi con questi spunti di discussione:

- riuscite a interpretare il codice della funzione `gauss_el` in funzione della descrizione dell'algoritmo?
- in che modo la matrice a gradini permette di determinare il numero di soluzioni?
- sapete interpretare il codice della funzione `gauss_jordan_el`, per calcolare le soluzioni?
- sapreste dimostrare che (a meno del segno) il valore assoluto del determinante della matrice sia dato dal prodotto degli elementi diagonali della matrice a scalini? Che risultato si otterrebbe per una matrice non invertibile?
- Come risolvereste il calcolo della soluzione di un sistema in cui la matrice ha rango $r = n - 1$, con n il numero di righe della matrice (ossia il numero di equazioni)?

Decomposizioni

Il metodo di Gauss – Jordan, data una matrice di n righe, richiede un numero di operazioni dell'ordine $O(n^3)$. Questo significa che se t_n è il tempo per trattare una matrice di n righe, il tempo richiesto trattare una matrice con $n' = \alpha n$ righe sarà: $t_{\alpha n} = \alpha^3 \, t_n$.

Se per trattare una matrice con 100 righe ci volesse $1 \, \mu s$, quanto occorrerebbe risolvere in sequenza 1000 sistemi lineari a cui sia associata una matrice di 10^6 righe?

Esatto, circa 30 anni!

Essendo comune il problema di dover risolvere ripetutamente un sistema lineare in cui a cambiare sia solo il vettore dei termini noti, e non la matrice, può essere comodo poter suddividere l'operazione di pivoting e quella di risoluzione del sistema lineare. I **metodi di decomposizione** hanno questo obiettivo.

Decomposizioni

Decomposizione LU

Si effettua la decomposizione $\mathbf{A} = \mathbf{LU}$ dove $\mathbf{A}$ è la matrice dei coefficienti, mentre $\mathbf{L}$ e $\mathbf{U}$ sono, rispettivamente, una matrice triangolare inferiore (***L****ower triangular*) e una matrice triangolare superiore (***U****pper triangular*). Il sistema lineare viene risolto in due passaggi:

1. $\mathbf{y} = \mathbf{L}^{-1}\mathbf{b}$
2. $\mathbf{x} = \mathbf{U}^{-1}\mathbf{y}$

La decomposizione ha un costo $O(n^3)$, se n è il numero di righe di $\mathbf{A}$, ma i due passaggi di risoluzione hanno una complessità di $O(n^2)$. Dato l'esempio della pagina precedente, quanto si impiegherebbe a risolvere i 1000 sistemi con il metodo LU?

In genere, per migliorare la stabilità dell'algoritmo, si usa eseguire un *pivoting* parziale della matrice da decomporre in maniera tale che i numeri pivot siano più grandi rispetto agli altri elementi della matrice. Se $\mathbf{P}$ è la matrice (involutoria, ossia $\mathbf{P}^{-1} = \mathbf{P}$) che permuta le righe, allora la decomposizione che si effettua è $\mathbf{PA} = \mathbf{LU}$.

Data la decomposizione LU di una matrice quadrata $n \times n$, si possono facilmente calcolare:
- determinante: $\det(\mathbf{A}) = (-1)^p \prod_{i=1}^{n} L_{i,i} U_{i,i}$, con p la parità di $\mathbf{P}$,
- inversa: $\mathbf{A}^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{P}$.

Decomposizioni

Decomposizione SVD

Nella decomposizione SVD (*singular value decomposition*) una matrice generica di dimensioni $m \times n$ viene decomposta come $\mathbf{A} = \mathbf{U}\mathbf{w}\mathbf{V}^{tr}$, dove $\mathbf{U}$ è una matrice $m \times m$, $\mathbf{V}$ una matrice $n \times n$, e $\mathbf{w}$ una matrice $m \times n$ diagonale. $\mathbf{U}$ e $\mathbf{V}$ sono unitarie; questo significa che $\mathbf{U}^{-1} = \mathbf{U}^{tr}$e, quindi, $\mathbf{U}^{tr}\mathbf{U} = \mathbf{I}$, e relazioni analoghe valgono per $\mathbf{V}$.

Se la matrice $\mathbf{A}$ è quadrata e non singolare:

$$\mathbf{A}^{-1} = (\mathbf{U}\mathbf{w}\mathbf{V}^{tr})^{-} = (\mathbf{V}^{tr})^{-1}\mathbf{w}^{-1}\mathbf{U}^{-1} = \mathbf{V}\mathbf{w}^{-1}\mathbf{U}^{tr}$$

con $(\boldsymbol{w}^{-1})_{i,i} = w_{i,i}^{-1}$.

La soluzione del sistema lineare, quindi, è triviale: $\mathbf{x} = \mathbf{V}\mathbf{w}^{-1}\mathbf{U}^{tr}\mathbf{b}$

Se $\mathbf{A}$ è reale, $\mathbf{U}$ e $\mathbf{V}$ sono ortogonali, ossia $\mathbf{U}\mathbf{V}^{\boldsymbol{tr}} = \mathbf{I}$. Di conseguenza, la decomposizione SVD permette anche di calcolare funzioni di matrici:

$$f(\mathbf{A}) = \mathbf{U}\mathbf{f}(\mathbf{w})\mathbf{V}^{tr}$$

con $f_{i,i}(\mathbf{w}) = f\left(w_{i,i}\right)$.

Decomposizioni

Problema agli autovalori

Si tratta di un caso speciale di decomposizione SVD, in cui la matrice viene decomposta come

$$\mathbf{AV} = \mathbf{V\Lambda}$$

dove $\mathbf{V}$ è la matrice (unitaria) degli autovettori destri della matrice, mentre $\mathbf{\Lambda}$ è la matrice (diagonale) degli autovalori.

Sia $\mathbf{A}$ una matrice quadrata non singolare, la soluzione del sistema lineare è: $\mathbf{x} = \mathbf{V}^{tr}\mathbf{\Lambda}^{-1}\mathbf{Vb}$

Il calcolo degli autovettori di una matrice è esso stesso un problema di risoluzione di un sistema lineare. Data la matrice quadrata $n \times n$, si vuole trovare il set di n vettori $\mathbf{v}_n$ tali per cui

$$\left(\mathbf{A} - \mathbf{I}\Lambda_{n,n}\right)\mathbf{v}_n = \mathbf{B}_n\mathbf{v}_n = \mathbf{0}$$

Si tratta di un sistema lineare omogeneo (ossia i termini noti sono tutti nulli). Per escludere la soluzione banale $\mathbf{v}_n = \mathbf{0}$ occorre assicurare che la matrice $\mathbf{B}_n$ sia singolare. Quindi, gli autovalori sono gli n valori che rendono non singolare $\mathbf{B}_n$ e si ottengono risolvendo

$$\det(\mathbf{B}_n) = \det\left(\mathbf{A} - \mathbf{I}\Lambda_{n,n}\right) = 0$$

Funzione di una matrice: $f(\mathbf{A}) = \mathbf{V}f(\mathbf{\Lambda})\mathbf{V}^{tr}$

Matrici sparse

Metodo del gradiente coniugato

Si può vedere la risoluzione del sistema

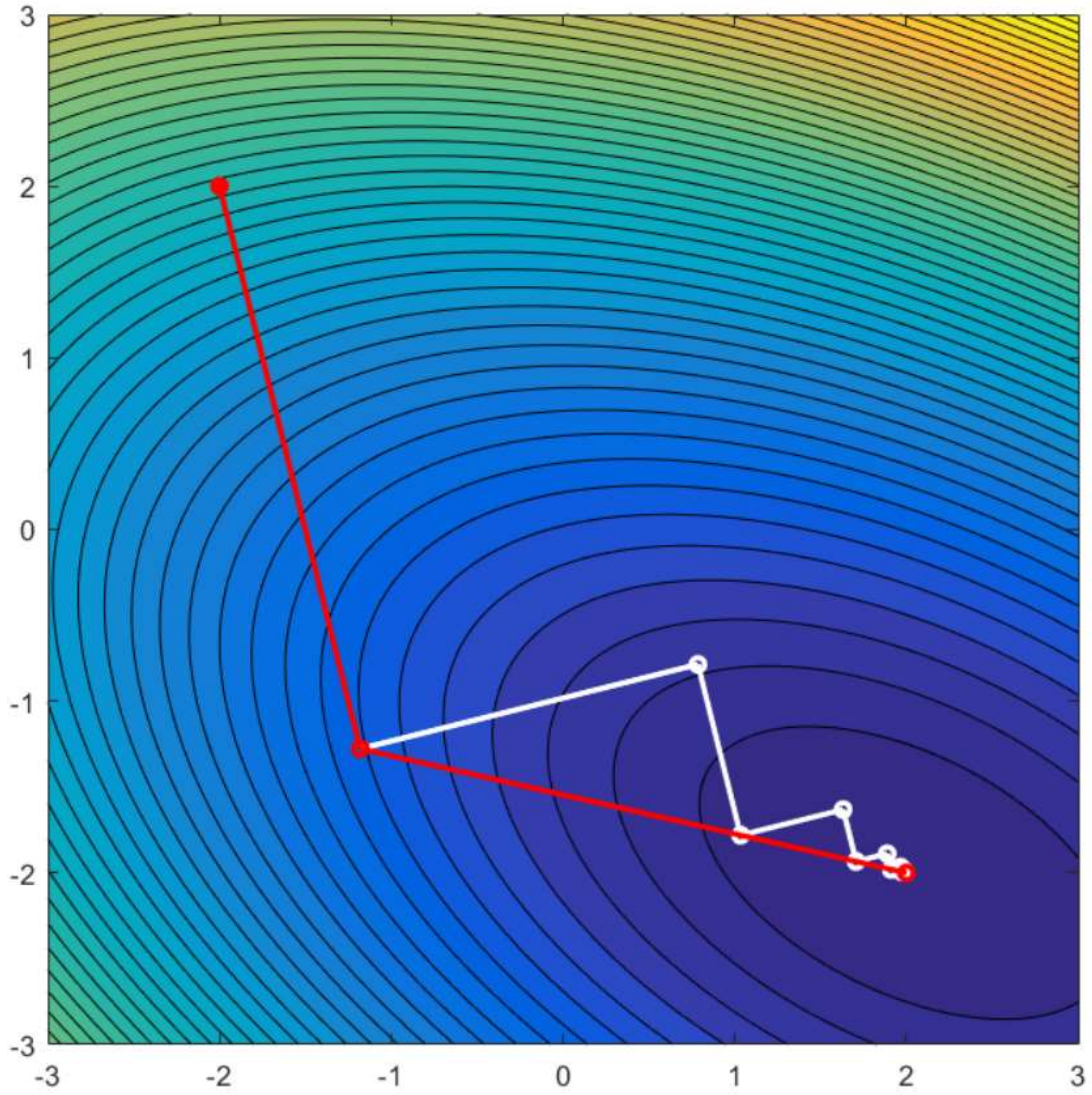
$$\mathbf{Ax} = \mathbf{b}$$

come il problema di minimizzare il gradiente della funzione:

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{tr}\mathbf{Ax} - \mathbf{x}^{tr}\mathbf{b}$$

Difatti, $\nabla f(\boldsymbol{x}) = \mathbf{Ax} - \mathbf{b} = 0$ implica la soluzione del sistema.

Con il metodo del gradiente coniugato si cerca la soluzione in modo iterativo, muovendo dalla posizione iniziale, lungo la direzione che è ortogonale a tutte le direzioni delle iterazioni passate.



Python time

Il modulo `linalg` di `numpy` ha il metodo `eig` per calcolare autovalori e autovettori di una matrice.

Il modulo `linalg` di `numpy` ha il metodo `solve` per risolvere sistemi lineari densi.

Il metodo `sparse` di `scipy` permette di risolvere sistemi lineari sparsi

Team work

Dividersi in gruppi per risolvere i problemi nei capitoli 5.2 e 6 del Jupyter Notebook.

1. Calcolare l'energia π di stato fondamentale dell'1,3-butadiene usando il metodo di Hückel, ponendo l'integrale Coulombiano $\alpha = 0$ ed esprimendo le energie in unità dell'integrale di scambio β. Si usi il metodo `numpy.linalg.eig` per gli autovalori.

2. Risolvere il sistema denso dato dal testo dell'esercizio sfruttando il metodo `numpy.linalg.solve`.

3. Risolvere il sistema lineare sparso dato dal testo dell'esercizio usando il metodo `scipy.sparse.linalg.cg`.