**Corso di Laurea Triennale in Ingegneria e Scienze Informatiche**

---

**Metodi Numerici** **A.A. 2020-2021**

---

# Esercitazione N. 4
# Metodo di eliminazione gaussiana e sua equivalente fattorizzazione - Codici

---

**Obiettivo**
Preparazione dei codici relativi all'applicazione del metodo di eliminazione gaussiana e sua equivalente fattorizzazione con/senza pivoting e sperimentazione.

**Codici**
Si consideri il sistema lineare $Ax = b$, con $A \in \mathbb{R}^{n\times n}$, $b, x \in \mathbb{R}^n$.

1. Si implementi una function che risolve il sistema lineare nel caso di matrice $A$ triangolare superiore con il metodo della risoluzione backward

$$x_i = \left( b_i - \sum_{j=i+1}^{n-1} a_{ij}x_j \right) / a_{ii}, \qquad i = n-1, n-2, ..., 0.$$

2. Si implementi una function che risolve il sistema lineare nel caso di matrice $A$ triangolare inferiore con il metodo della risoluzione forward

$$x_i = \left( b_i - \sum_{j=0}^{i-1} a_{ij}x_j \right) / a_{ii}, \qquad i = 0, 1, ..., n-1.$$

3. Si implementi una function che risolve il sistema lineare nel caso di fattorizzazione $PA = LU$ assegnata, combinando i metodi di risoluzione forward e backward precedentemente implementati.

4. Si implementi una function che utilizzi il metodo di eliminazione gaussiana secondo l'algoritmo base

$$\begin{array}{l}
\text{per } k = 0 : n-1, \\
\quad \text{per } i = k+1 : n-1, \\
\qquad m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \quad \text{t.c.} \quad a_{ik}^{(k+1)} = 0 \\
\qquad \text{per } j = k+1 : n-1, \\
\qquad\quad a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik}a_{kj}^{(k)}
\end{array}$$

   o secondo la sua versione **vettorizzata**

   per ottenere le matrici della fattorizzazione $LU$.

5. Si consideri una variante della function al punto 4, in cui si applica la tecnica del pivot parziale

$$\text{determinare } a_{rpiv,k} \text{ t.c. } |a_{rpiv,k}| = \max_{i=k,\ldots,n} |a_{i,k}|$$

$$\text{e scambiare la riga } rpiv \text{ con la riga } k \text{ se necessario,}$$

e si restituiscono in output le corrispondenti matrici della fattorizzazione $PA = LU$.

**Test da eseguire**

1. Al variare di $n = 100, \ldots, 200$ costruire la matrice

$$A[i,j] = sqrt(2/(n+1)) * sin((i+1) * (j+1) * \pi/(n+1))$$

i,j=0,..,n-1 di dimensione $n \times n$, definire la soluzione `xesatta = (1:n)`$^T$ e calcolare il termine noto come `b=A·xesatta`. Utilizzando le tre diverse function implementate calcolare `xnopivot`, `xparziale`, e confrontarli con `xesatta` usando i grafici in scala semilogaritmica dell'errore relativo al variare di $n$. Confrontare anche `xpython=scipy.linalg.solve(A,b)` con `xesatta`. Inoltre, per verificare che il numero di operazioni è proporzionale a $n^3$, con $n$ dimensione del sistema, riportare in un grafico in scala semilogaritmica il tempo impiegato a risolvere il sistema al variare di $n$. Il grafico dovrebbe essere (asintoticamente) una retta con pendenza 3. Per verificarne la pendenza, disegnare contemporaneamente anche la curva $n^3$ e controllare che siano parallele. La strategia di pivot non dovrebbe influenzare il risultato.

2. Calcolare la soluzione dei seguenti sistemi lineari $Ax = b$ usando il metodo di fattorizzazione LU senza pivoting e con pivoting parziale.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 1 \\ 1 & 3 & 5 \end{bmatrix}, \qquad b = \begin{bmatrix} 6 \\ 1 \\ 9 \end{bmatrix}$$

e

$$A = \begin{bmatrix} 1 & 1 & 0 & 3 \\ 2 & 1 & -1 & 1 \\ -1 & 2 & 3 & -1 \\ 3 & -1 & -1 & 2 \end{bmatrix}, \qquad b = \begin{bmatrix} 5 \\ 3 \\ 3 \\ 3 \end{bmatrix}.$$

In entrambi i casi la soluzione esatta ha componenti $x_i = 1$ per ogni $i = 1, ..., n$. Come è possibile giustificare a priori l'insuccesso del metodo di fattorizzazione LU in assenza di pivot?

3. Costruire una function per il calcolo della soluzione di una generale equazione $AX = B$, con $X, B$ matrici, che usi la fattorizzazione LU con pivoting parziale. Utilizzarla poi per il calcolo dell'inversa delle matrici non singolari

$$A = \begin{bmatrix} 3 & 5 & 7 \\ 2 & 3 & 4 \\ 5 & 9 & 11 \end{bmatrix}, \qquad A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & -4 & 6 & 8 \\ -1 & -2 & -3 & -1 \\ 5 & 7 & 0 & 1 \end{bmatrix},$$

confrontando i risultati ottenuti con l'output della funzione `scipy.linalg.inv(A)`. Ripetere l'esercizio utilizzando nella function per il calcolo di $X$ la fattorizzazione LU senza pivoting. Che cosa si osserva?

4. Sia assegnato il sistema lineare $Ax = b$ con

$$A = \begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 2+\varepsilon \\ 4 \end{bmatrix} \qquad \text{dove} \quad \varepsilon > 0.$$

Dopo aver osservato che quando $\varepsilon = 0$ tale sistema ha soluzione $[2, 2]^T$, risolvere il sistema senza far uso della strategia pivotale per valori di $\varepsilon$ pari a $10^{-k}$ con $k = 2 : 2 : 18$. Confrontare i risultati ottenuti con le soluzioni trovate per i medesimi valori di $\varepsilon$ quando viene applicata la strategia pivotale.

5. Sia assegnato il sistema lineare $Ax = b$ con

$$A = \begin{bmatrix} 3 & 1 & 1 & 1 \\ 2 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} 4 \\ 1 \\ 2 \\ 4 \end{bmatrix},$$

la cui soluzione esatta è $x = [1, -1, 0, 2]^T$. Calcolare la fattorizzazione LU di $A$ e osservare la presenza di fill-in nei fattori $L$ e $U$. Osservare come l'utilizzo di pivot parziale determini moltiplicatori $m_{ik}$ tali che $|m_{ik}| \leq 1$ ed una minor crescita del modulo degli elementi della matrice triangolare superiore $U$. Analizzare inoltre la fattorizzazione LU risultante nel caso del sistema lineare ottenuto permutando nella matrice $A$ la prima riga con l'ultima, la prima colonna con l'ultima e nel termine noto la prima componente con l'ultima:

$$A = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 2 \\ 1 & 1 & 1 & 3 \end{bmatrix}, \qquad b = \begin{bmatrix} 4 \\ 1 \\ 2 \\ 4 \end{bmatrix}.$$

La soluzione esatta di quest'ultimo è $x = [2, -1, 0, 1]^T$ come atteso considerando la permutazione effettuata sulle colonne della matrice $A$.

6. Dato $n = 100$, calcolare $Q = I_n - 2 \cdot vv^T$, dove $I_n$ é la matrice identitá di ordine $n$ e $v$ é un vettore colonna di n componenti formato da numeri casuali ed avente norma 2 unitaria. Quindi, per $k = 1, ..., 20$, porre

$$D = I_n;\ D[n-1, n-1] = 10^k;\ A = Q \cdot D$$

(Tale costruzione produce una matrice $A$ che abbia esattamente numero di condizionamento $10^k$ in norma 2).
Al variare di $k$ e quindi del numero di condizionamento di $A$, studiare l'errore del metodo di eliminazione gaussiana senza pivoting e con pivoting parziale nella risoluzione del sistema lineare $Ax = b$, dove `b=A·xesatta`. e `xesatta = (1,1,...,1)`$^T$

7. Per valori di $n = 4 : 6 : 40$, si consideri il sistema lineare $A_n x = b$ con $A_n$ matrice di Hankel di ordine $n$ di elementi

$$a^{(n)}_{i,n-1+k-i} = \begin{cases} 2^{k+1} & \text{se } k > 0, \\ 2^{1/(2-k-1)} & \text{se } k \leq 0, \end{cases} \qquad i = 0, ..., n-1, \quad k = i+1-n, ..., i,$$

e $b$ scelto in modo che risulti $x = [1, 1, ..., 1]^T$. Si risolva tale sistema con il metodo di fattorizzazione LU con massimo pivot parziale e il metodo di fattorizzazione QR (Q,R=`scipy.linalg.qr(A)`). Calcolare gli errori relativi $\|\delta x\|_2/\|x\|_2$ da cui sono affette le soluzioni calcolate con i due metodi e produrre, al variare di $n$, un grafico in scala logaritmica ( matplotlib.plyplot.loglog) degli errori relativi calcolati. Che cosa si osserva?

8. Ripetere l'esercizio precedente per risolvere il sistema lineare $Ax = b$ con $A$ e $b$ così definiti:

$$a^{(n)}_{i,j} = \begin{cases} 1 & \text{se } i = j \text{ oppure } \text{ se } j = n-1, \\ -1 & \text{se } i > j, \\ 0 & \text{altrimenti} \end{cases} \qquad b = A \cdot [1, ..., 1]^T,$$

per $n = 48 : 2 : 58$ e $b$ scelto in modo che risulti $x = [1, 1, ..., 1]^T$. Che cosa si osserva?