# Specifiche per l'esercitazione sui sistemi lineari.

1) Realizzare una m-function

   **[nor]=function norma(A,flag)**

   che preso in input un vettore o una matrice **A**, ne calcoli, a scelta dell'utente, la norma 1 o la norma infinito, in base al valore di flag. (se flag=0, calcola norma infinito, se flag=1, calcola la norma 1).

2) Realizzare una m-function che presa in input una matrice quadrata ne calcoli l'indice di condizionamento in norma p, p=1, oppure p=infinito (per il calcolo dell'inversa della matrice A fare uso della function di Matlab inv(A), che presa in input una matrice A ne restituisce l'inversa).

   **Condizionamento=function ind_cond(A,p)**

3) Realizzare una m-function che preso in input l'ordine della matrice, restituisca in output la matrice di Wilkinson.

   **A=function wilk(n);**

4) Realizzare una m-function che preso in input l'ordine della matrice, gli estremi dell'intervallo [a,b], costruisca la matrice di Vandermonde di ordine n a partire dal vettore x di n valori equidistanti nell'intervallo [a,b].

   **A=function vander(n,a,b)**

5) Realizzare una m-function che presa in input una matrice triangolare inferiore L ed il termine noto b, restituisca in output

la soluzione del sistema Lx=b, calcolata mediante l'algoritmo di sostituzione in avanti.

**x=function Forward_sostitution (L,b);**

6) Realizzare una m-function che presa in input una matrice triangolare superiore R ed il termine noto b restituisca in output la soluzione del sistema Rx=b, calcolata mediante l'algoritmo di sostituzione all'indietro.

**x=function Backward_sostitution (R,b);**

7) Realizzare una m-function che presa in input una matrice A quadrata di ordine n restituisca in output i fattori L ed R della fattorizzazione di Gauss di A con pivotaggio a perno massimo per colonne e la matrice di permutazione P

**[L,R,P]=function fatt_lu(A)**

8) Costruire una m-function che presa in input una matrice A di ordine n, il termine noto b, restituisca in output la soluzione del sistema Ax=b, facendo uso della fattorizzazione LU con pivotaggio a perno massimo per colonne e della sostituzione in avanti e all'indietro

**x=function risolvi_sistema(A,b);**

Nota importante: **Poiché nel caso di fattorizzazione di Gauss con pivotaggio a perno massimo per colonne si ha che P·A=L·R, il sistema da risolvere diventa P·A x=P·b , cioè LRx=P·b, che equivale a risolvere i due sistemi lineari**

$$\begin{cases} Ly = Pb \\ Rx = y \end{cases}$$

## Alcune note guida

1) **Algoritmo per la costruzione della matrice di Wilkinson di ordine n**

$$A_{ii} = 1 \qquad i = 1,\dots,n-1$$

$$A_{i+1,i} = 1 \qquad i = 1,\dots,n-1$$

$$A_{i,n} = (-1)^{i} \qquad i = 1,\dots,n$$

$$A_{i,j} = (-1)^{i+j-1} \qquad j = 1,\dots,n-1$$
$$i = j+2,\dots,n$$

**Ad esempio, per n=5 si ha**

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ 1 & 1 & 0 & 0 & 1 \\ -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

2) **Matrice di Vandermonde: dato un vettore x, di componenti** $(x_0, x_1, \dots, x_{n-1})$ **la matrice di Vandermonde ad esso associata è una matrice di ordine n il cui generico elemento di posto (i,j) è dato da**

$$a_{i,j} = (x_i)^j, \quad i=0,..,n-1 \text{ e } j=0,1,\dots,n-1$$