# Corso di Laurea in Ingegneria Aerospaziale/ Meccanica
# Laboratorio di Analisi Numerica
# A.A. 2012/2013 – II Ciclo

## Esercitazione 4

Creare una cartella <cognome> in C: dove verranno salvati i file creati nella sessione di lavoro.
Appena entrati in MATLAB posizionarsi in <cognome>.
Risolvere in ambiente MATLAB i seguenti esercizi.

**RISOLUZIONE DI SISTEMI LINEARI:**

**OPERATORE DIVISIONE A SINISTRA**

Per la soluzione di sistemi lineari in MATLAB si può usare l'operatore **** o divisione a sinistra:
**>> x= A \ b**
che calcola la soluzione x del sistema Ax=b con il **metodo di eliminazione di Gauss con pivoting in generale**. Nei casi particolari di sistemi con matrice A triangolare inferiore o superiore risolve con il metodo di sostituzione in avanti o all'indietro rispettivamente.

**FATTORIZZAZIONE LU con pivoting**

La funzione MATLAB che esegue la fattorizzazione LU con pivoting e' **[L,U,P]=lu(A)** , l'output di questa funzione sono le matrici triangolari L ed U ed una matrice di permutazione P tali che

**PA=LU**.

1. Dopo aver esplorato mediante l'*help* di Matlab i seguenti comandi per la creazione di matrici speciali:

| | |
|---|---|
| ▪ **zeros** | matrice nulla |
| ▪ **ones** | matrice con elementi pari a 1 |
| ▪ **eye** | matrice identità |
| ▪ **diag** | matrice diagonale |
| ▪ **tril** | matrice triangolare inferiore |
| ▪ **triu** | matrice triangolare superiore |
| ▪ **hilb** | matrice di Hilbert |
| ▪ **vander** | matrice di Vandermonde |
| ▪ **rand** | matrice di numeri casuali |

Calcolare l'indice di condizionamento delle matrici di Hilbert, di Vandermonde e di numeri casuali al variare dell'ordine n=5,10,15, utilizzando la built-in function **cond** di MATLAB.

2. Dopo aver definito nello script ***solv.m*** i seguenti sistemi lineari nella forma matriciale $Ax=b$

$$\begin{cases} 2x_1+6x_2-2x_4+2=0 \\ -2x_1-7x_2-3x_3+4x_4-13=0 \\ -x_1-2x_2-x_4+7=0 \\ 2x_2+10x_3+5x_4-10=0 \end{cases} \qquad \begin{cases} x_1+4x_2-x_3+x_4=2 \\ 2x_1+7x_2+x_3-2x_4=16 \\ x_1+4x_2-x_3+2x_4=-15 \\ 3x_1-10x_2-2x_3+5x_4=-15 \end{cases} \qquad \begin{cases} 12x_1-15x_2=8 \\ -4x_1+5x_2=10 \end{cases}$$

stabilire se i sistemi hanno soluzione e in caso affermativo, risolverli con il metodo della matrice inversa (utilizzando la funzione **inv** di Matlab) e della divisione a sinistra (utilizzando l'operatore \ che calcola la soluzione con il metodo di eliminazione di Gauss: $x = A \backslash b$). Quale dei due metodi risulterà computazionalmente più efficiente?

3. CONDIZIONAMENTO DEI SISTEMI LINEARI. Nello script file ***ex3_hilb.m***, creare le matrici $A$=hilb(n), per n=5,10,20. Costruire poi un vettore colonna $x_{ex}$ =ones(n,1), che rappresenta la soluzione esatta, e calcolare il vettore dei temini noti $b=Ax_{ex}$.

   a. Per ogni $n$ risolvere i sistemi lineari $Ax=b$ e $Ax=\bar{b}$, dove $\bar{b}$ è una perturbazione del vettore $b$ ($\bar{b}$ =*b+0.01*rand(n,1)*).
   b. Per ogni n stampare l'errore relativo sulla soluzione $\dfrac{\|x-\bar{x}\|_2}{\|x\|_2}$, l'errore sui dati iniziali $\dfrac{\|b-\bar{b}\|_2}{\|b\|_2}$ e il condizionamento della matrice ( con cond($A$)).

Scrivere uno script simile (***ex3_rand.m***) per ripetere l'esercizio per le matrici di numeri casuali A=rand(n), n=5,10,20.

4. Il determinante di una matrice triangolare è uguale al prodotto dei suoi elementi diagonali. Utilizza questo fatto per sviluppare una function MATLAB ( **detA= calcDet(A)**) che calcoli il determinante di una matrice A arbitraria di dimensione n utilizzando la fattorizzazione LU.

5. Realizzare una function MATLAB **UtriSol.m** che risolva il sistema lineare con matrice dei coefficienti triangolare superiore A e vettore dei termini noti b (Ax=b) mediante il metodo di sostituzione all'indietro:

```
function x = UTriSol(A,b)
.....
```

6. Costruire una funzione MATLAB dal nome **LU_solve_gen.m** per il calcolo della soluzione di una generale equazione AX=B, con X,B matrici, che utilizza la fattorizzazione LU. Utilizzarla poi per il calcolo dell'inversa $A^{-1}$ delle seguenti matrici

$$\begin{bmatrix} 3 & 5 & 7 \\ 2 & 3 & 4 \\ 5 & 9 & 11 \end{bmatrix}, \begin{bmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

Confrontare i risultati con la built-in function MATLAB **inv(A)**.