Calcolo Numerico (Ingegneria Energia/Meccanica, Canale A)
Prova di Laboratorio II, del 16 Settembre 2019, fila II

Cognome e nome ______________________________ **Matricola** __________

Postazione ________________

FIRMA PER CONSEGNARE ____

FIRMA PER RITIRARSI ____

1. Si raccomanda agli studenti di **commentare adeguatamente** script e function Matlab.

2. Ogni porzione di codice **deve avere** quale prima riga un commento mediante % con nome, cognome, numero di matricola e postazione.

1. Sia A una matrice con n righe e n colonne, $b \in \mathbb{R}^n$ un vettore colonna e si desideri calcolare la soluzione del sistema lineare $Ax = b$, mediante il metodo di Jacobi. Si implementi tale metodo mediante la routine Matlab `jacobi`, che abbia la seguente intestazione:

```
function [xv,res,flag]=jacobi(A,b,x0,toll,nmax)
% Dati di ingresso:
% A: matrice n x n.
% b: vettore colonna n x 1
% x0: vettore colonna n x 1
% toll: tolleranza del criterio di arresto
% nmax: numero massimo di iterazioni.
% Dati di uscita:
% xv: vettore colonna n x 1 contenente un'approssimazione della soluzione.
% res: vettore contenente i residui.
% flag: 1 il processo non e' terminato correttamente.
%       0 altrimenti.
```

   - La routine controlli che il determinante di A non sia nullo (utilizzare il comando Matlab `det`), altrimenti scriva su monitor `La matrice A non risulta invertibile` ed esca forzatamente dalla routine, dopo aver posto `xv=[]`, `res=[]` e `flag=1`.
   - La routine controlli che il determinante di `M=diag(diag(A))` non sia nullo, altrimenti scriva su monitor `Il metodo di Jacobi non risulta applicabile` ed esca forzatamente dalla routine, dopo aver posto `xv=[]`, `step=[]` e `flag=1`.
   - La routine ponga `xv=x0`, `N=M-A`. Di seguito si assegni `invM=`M^{-1} (come si calcola l'inversa in Matlab?), `B=invM*N`, `c=invM*b`
   - Si esegua un ciclo-for che alla k-sima iterazione (di al piú `nmax`)
     - Effettui l'assegnazione `xv0=xv`.
     - Ponga in `xv` il valore `xv=B*xv0+c`.
     - Il test di arresto sia quello del *residuo pesato* ovvero si concludano le iterazioni se $\text{res}_k := \|\texttt{A*xv-b}\|_2 \leq toll* \|b\|_2$. Se la routine termina correttamente, scriva su monitor `Il metodo di Jacobi termina correttamente` ed esca con `flag` uguale a 0 altrimenti continui a iterare il processo.
   - Se il numero di iterazioni é pari o strettamente maggiore di `nmax`, scriva su monitor `Il metodo di Jacobi non termina correttamente` e ponga `flag=1` (cosa si deve fare esternamente al ciclo for?).

2. Si scriva una function `jacobi_script` che:

   - Inizializzi A come una matrice con venti righe e venti colonne di elementi nulli.
   - Definisca una matrice quadrata $A = (a_{i,j})$ di dimensione 20 che abbia elementi

     $a_{k,k} = 6$ per $k = 1, \ldots, 20$;
     $a_{k,k-1} = -1$ per $k = \mathbf{2}, \ldots, 20$;
     $a_{k-1,k} = -1$ per $k = \mathbf{2}, \ldots, 20$;
     0 altrimenti.
   - Si definisca un vettore colonna b di dimensione compatibile con A ed elementi uguali a 1.
   - Risolva il problema $Ax = b$ mediante il metodo di Jacobi sopra descritto con `x0` il vettore nullo delle dimensioni di b, `toll`$=10^{-6}$ e `nmax`$=10000$.
   - Esegua un grafico in scala semilogaritmica delle coppie (k, res_k) e si salvi il grafico come `figura.jpg`.
   - si scrivano, nella tabella che segue, la prima e l'ultima componente di `res` con 1 cifra prima della virgola e 6 dopo la virgola, in formato esponenziale e la lunghezza del vettore `res` (ovvero il numero di iterazioni eseguite) come numero intero.

`res(1)`	`res(end)`	`length(res)`