Corso di ANALISI NUMERICA
A.A. 2014/15

Foglio di esercizi N.1

N.B. Controllate la sintassi di tutte le funzioni intrinseche di Matlab usate mediante il relativo comando di help.

1. Generare nella command window (CW) una matrice random A 5×5 di numeri reali compresi tra 1 e 10, quindi calcolare le norme matriciali 1, 2, ∞ e di Frobenius di A usando opportunamente i comandi Matlab *max, sum, sqrt, trace, eig* e le usuali operazioni tra matrici. Infine confrontare i valori ottenuti con quelli forniti dal comando *norm(A,p).*

2. Scrivere un codice che letto un numero naturale n e un intervallo (a, b), genera una matrice random A $n \times n$ di numeri reali compresi in quell'intervallo e la stampa insieme ai valori del determinante, della traccia, degli autovalori, del raggio spettrale e delle 4 principali norme matriciali associate.

3. Scrivere un vostro codice che costruisce per ogni $n \in N$ la matrice di Hilbert $n \times n$ H_n, utilizzando per i suoi elementi la definizione:

$$H_n(i, j) = 1/(i + j - 1);$$

   Il comando *hilb(n)* di Matlab che costruisce la stessa matrice si basa invece essenzialmente sui seguenti comandi:

```
J=1:cast(n,'double');
J=J(ones(n,1),:);
I=J';
E=ones(n);
H=E./(I+J-1);
```

   Provate a eseguirli uno alla volta nella CW cercando di capire come funzionano.

4. Scrivere un codice che riempie un vettore con i numeri di condizionamento spettrale $K_2(H_n)$ per n =1:12 usando opportunamente per questo calcolo la funzione *eig* di Matlab. Visualizzare in un grafico in scala semilogaritmica (lineare nelle ascisse e logaritmica nelle ordinate) l'andamento di tale vettore. Confrontare i risultati ottenuti con quelli forniti direttamente dal comando *cond* di Matlab.

5. Verificate direttamente il malcondizionamento della matrice di Hilbert risolvendo il sistema lineare associato con soluzione unitaria al crescere di n e guardando l'andamento dell'errore commesso in funzione del numero di condizionamento. Ecco un possibile listato:

```
A=hilb(n);
solex=ones(n,1);
b=A*solex;
x=A\b
err=norm(x-solex)/norm(solex)
cond(A)
```

   da eseguire per $n = 3, 6, 10, 12$. Provare anche ripetere il tutto con la norma infinito.

6. (Generazione di matrici sparse) Costruire la matrice tridiagonale reale A 7×7 definita da

$$A(i,j) = \begin{cases} 2 & \text{se } i = j \\ -1 & \text{se } |i-j| = 1 \\ 0 & \text{altrimenti} \end{cases}$$

   Verificare tramite comandi Matlab che si tratta di una matrice simmetrica definita positiva e a dominanza diagonale (non stretta). Successivamente usare e interpretare i seguenti comandi:

```
e=ones(7,1);
A=spdiags([-e 2*e -e], -1:1, 7,7);
full(A)
```

7. Costruire una matrice A 5x5 di numeri reali definita da $A(i,j) = i+j$, usando l'idea vista nell'esercizio 3) per la matrice di Hilbert. Quindi costruire la matrice di permutazione $P^{(2,4)}$ e osservare l'effetto del prodotto $P^{(2,4)} * A$.

8. Scrivere i codici **forwardrow** e **backwardrow** come funzioni in grado di implementare i metodi di sostituzione in avanti e indietro (per righe). Si possono anche trascrivere i listati del libro di testo, purché se ne comprendano appieno le istruzioni.

   Costruire quindi una matrice quadrata di ordine 5 a valori complessi e parti reali e immaginarie intere (suggerimento: usare opportunamente i comandi *rand e ceil* per generare 2 matrici casuali A e B di interi, poi costruire $C = A + i * B$); risolvere poi i sistemi lineari $Lx = b$ e $Uy = b$, dove L e U denotano rispettivamente le sottomatrici triangolari inferiore e superiore estratte da C, mentre b è il vettore unitario. Verificare i risultati ottenuti calcolando i vettori residui nei due casi.