ESERCIZIO 1.

1. Costruiamo la matrice di Hilbert di ordine 3

```
>> n=3;
>> for i=1:n
       for j=1:n
              A(i,j)=1/(i+j-1);
       end
    end
>>d=eig(A)
d =
   0.20000000000000
   0.05278331122160
   1.44721668877840
```

   Poiché la matrice A è simmetrica e definita positiva, possiamo calcolarne il numero di condizionamento come rapporto tra il massimo e il minimo autovalore

```
>>KA=max(d)/min(d)
KA =
  27.41807316146669
```

2. Costruiamo il termine noto b e la perturbazione δb:

```
>> b=A*ones(n,1);
b =
   1.83333333333333
   1.25000000000000
   0.78333333333333
>> deltab=1e-10*ones(n,1)
```

```
deltab =
  1.0e-009 *

   0.10000000000000
   0.10000000000000
   0.10000000000000
```

Stimiamo l'errore relativo sulla soluzione:

```
>>ErrRel=KA*norm(deltab,2)/norm(b,2)
ErrRel =
    2.018141354248413e-009
```

Osserviamo quindi che a fronte di una perturbazione sul termine noto dell'ordine di 10^{-10}, la perturbazione sulla soluzione è circa dieci volte più grande.

3. La matrice di Hilbert è un tipico esempio di matrice malcondizionata: osserviamo infatti come al crescere della dimensione, il suo condizionamento diventi rapidamente molto grande. Ciò, per quanto visto prima, causa, anche per perturbazioni piccole sul termine noto (o sulla matrice dei coefficienti), notevoli errori nella soluzione. Rappresentiamo il condizionamento in funzione della dimensione della matrice:

```
for n=1:5
  A=hilb(n);
  KA(n)=cond(A,2);
end
semilogy(1:n,KA)
```

Abbiamo usato nel codice sopra il comando MATLAB `hilb` che costruisce la matrice di Hilbert di dimensione `n`. L'andamento di $K(A)$ è rappresentato in Fig. 1.

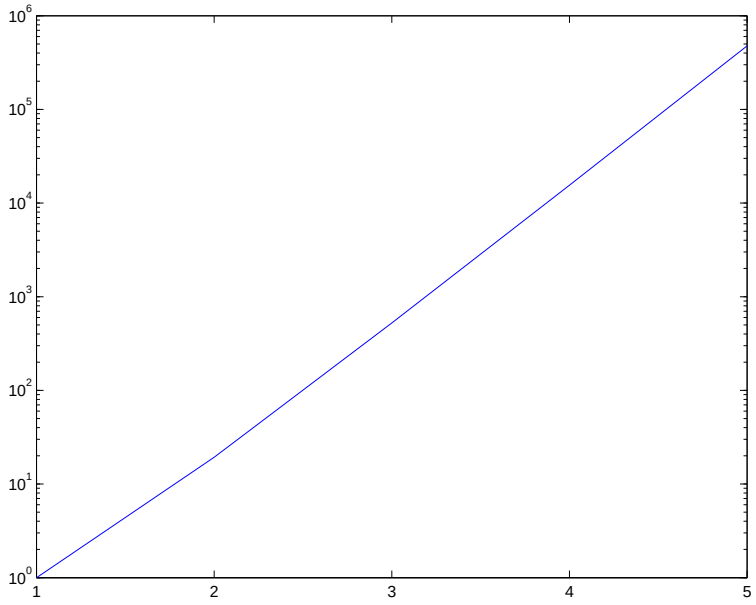

Figura 1: Numero di condizionamento di A per $n = 1, \ldots, 5$

ESERCIZIO 2.

1. La matrice A non soddisfa alcuna delle condizioni sufficienti perché si possa applicare il MEG senza pivotazione, non essendo a dominanza diagonale stretta per linee, né simmetrica definita positiva. Verifichiamo allora se è soddisfatta la condizione necessaria e sufficiente, calcolando i determinanti dei minori principali

```
>> A=[1 1 1 ; 2 2 5; 4 6 8];
>> for i=1:3
     det(A(1:i,1:i))
     if(~det(A(1:i,1:i)))
       break
   end
ans =

     1
ans =

     0
```

   Abbiamo quindi che nemmeno la condizione necessaria e sufficiente è verificata, dunque non sarà possibile portare a termine il MEG senza pivotazione.

2. Calcoliamo i fattori L e U e la matrice di permutazione P in MATLAB

```
[L,U,P]=lu(A)
```

   Abbiamo che

(a) $l_{ii} = 1, i = 1, 2, 3$, dunque il suo determinante deve essere 1; verifichiamo:

```
>> det(L)
    1
```

Inoltre, ricordando che in MATLAB 1 è l'operatore booleano che significa vero, verifichiamo che:

```
prod(diag(U))==det(U)
```

dove `diag(U)` è il vettore costituito dagli elementi diagonali di `U` e `prod` calcola il prodotto di tali elementi

(b) P è una matrice ortogonale, infatti

```
P'==inv(P)
```

3. Le tappe successive della risoluzione del sistema sono

$$Ax = b \quad \rightarrow PAx = Pb \quad \rightarrow LUx = Pb$$

e

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

In MATLAB:

```
>> y=L\(P*b)
>> x=U\y
    0.5000
    0.5000
         0
```

Ricordiamo che questa sequenza di operazioni corrisponde alla semplice applicazione del comando MATLAB

```
>> x=A\b
    0.5000
    0.5000
         0
```

ESERCIZIO 3.

1. La matrice è simmetrica se $A = A^T$. Per fare il confronto tra A e la sua trasposta, possiamo utilizzare la function `isequal` che restituisce 1 se le matrici sono uguali e zero nel caso contrario.

```
>>A=hilb(10);
>>x=ones(10,1);
>>b=A*x;
>>AT=A';
>>tf=isequal(A,AT)

tf =

     1
```

Quindi la matrice A è simmetrica. Per verificare se essa è anche definita positiva, guardiamone gli autovalori

```
>> format long e
>> eig(A)
```

Otteniamo quindi che la matrice A ha tutti autovalori positivi (anche se sarà molto malcondizionata,

infatti il massimo e il minimo autovalore sono molto diversi tra loro!)

2. Poiché A è una matrice simmetrica e definita positiva, possiamo utilizzare la fattorizzazione di Cholesky. I vantaggi sono:

   - richiede circa $n^3/3$ operazioni, cioè circa la metà di quelle richieste dalla fattorizzazione LU
   - per simmetria, verrà memorizzato solo un fattore, ricavando l'altro fattore per semplice trasposizione

```
>> R=chol(A);
>> ychol=R'\b;
>> xchol=R\ychol

xchol =

    1.0000
    1.0000
    1.0000
    1.0000
    1.0001
    0.9998
    1.0004
    0.9997
    1.0002
    1.0000
```

   Se `x=ones(10,1)` è la soluzione esatta, l'errore relativo è $err_{rel} = \frac{||x-xchol||}{||x||}$.

```
>>err=norm(x-xchol)/norm(x)
err =

  1.8053e-004
```