RELAZIONE QUINTA ESERCITAZIONE

1) DATO UN SISTEMA LINEARE Ax=B, DI DIMENSIONE NXN, IMPLEMENTARE L'ALGORITMO DI GAUSS CON PIVOTING PARZIALE PER COLONNE A PERNO MASSIMO, PER LA RISOLUZIONE DEL SISTEMA LINEARE

CODICE MATLAB, ANALISI E RISULTATI:

```
function [ x ] = gauss_pivoting( A,b )
%inizializza L alla matrice identità
L = eye(max(size(A)));
%inizializza R uguale ad A
R = A;
n = size(A);
%implementazione pseudo codice
for k=1:1:n - 1
    for i = k + 1:1:n
        if(R(k,k) == 0)
            %scambia la riga k esima con la riga l esima
            %trovo elemento massimo a partire da k fino ad n
            ind = k + 1;
            mass = R(k+1,k);
            for g = k + 2:1:n
                if(mass < R(g,k))
                    ind = g;
                    mass = R(g,k);
                end
            end
            %ho trovato l'indice della riga da scambiare con la
riga k
            %esima
            tmp = R(k,:);
            R(k,:) = R(ind,:);
            R(ind,:) = tmp;
        end
        L(i,k) = R(i,k) / R(k,k);
        for j=k+1:1:n
            R(i,j) = R(i,j) - L(i,k) * R(k,j);
        end
        %aggiorno termine noto
        b(i,1) = b(i,1) - L(i,k)*b(k,1);
    end
end
R = triu(R);
%calcolo
x = indietro(R,b);
end
```

ESEMPIO DI UTILIZZO:

```
A =
     1     2     3
     4     1     0
     0     1     0


b =
     2
     1
    -1


gauss_pivoting(A,b)


ans =

    0.5000
   -1.0000
    1.1667
```

ANALISI:

La funzione gauss_pivoting prende in input la matrice nxn e il vettore scritto per colonna dei termini noti, in output restituisce il vettore per colonna delle incognite x.
L'algoritmo è come l'algoritmo ad eliminazione di Gauss implementato nella quarta relazione, con la differenza che se trova una riga con il pivot uguale a 0, scambia la riga corrente con la riga con il pivot di valore assoluto massimo.

La funzione indietro è la stessa che è stata usata nella quarta relazione.

2) RISOLUZIONE DEL SISTEMA LINEARE Ax=B CON LA MATRICE DI HILBERT CON N COMPRESO TRA 2 E 15

CODICE MATLAB, ANALISI E RISULTATI:

```
function [z,c] = hilbert()
z = zeros(14,1);
c = zeros(14,1);
cont = 1;
    for n=2:1:15
        H = hilb(n);
        b = zeros(n,1);
        %calcolo termine noto
        for i=1:1:n
            somma = 0;
            for j=1:1:n
                somma = somma + H(i,j);
            end
            b(i,1) =  somma;
        end
        x = gauss_pivoting(H,b);
        reale = ones(n,1);
        z(cont,1) = norm(reale - x) / norm(reale);
        c(cont,1) = cond(H);
        cont = cont + 1;
    end
end
```

ESEMPIO DI UTILIZZO:

```
format long

[z,c] = hilbert()

z = (errore relativo rispetto alla soluzione reale)

   0.000000000000001          n = 2
   0.000000000000010
   0.000000000000378
   0.000000000001553
   0.000000000251678
   0.000000008425197
   0.000000250572556
   0.000008608677940
   0.000223773106799
   0.004640947170787
   0.095457668144563
   3.065467755616769
   4.562223393368791
   3.372079139900268          n = 15


c = (indice di condizionamento della matrice di hilbert)

   1.0e+18 *

   0.000000000000000          n = 2
   0.000000000000001
   0.000000000000016
   0.000000000000477
   0.000000000014951
   0.000000000475367
   0.000000015257576
   0.000000493154110
   0.000016024922771
   0.000522599671075
   0.016775592007686
   1.759036194677610
   0.308209919824903
   0.443327217343445          n = 15
```

ANALISI:

la funzione hilbert genera una matrice di hilbert di ordine n (con n che va da 2 a 15) e, per ogni matrice generata:

1) crea il vettore dei termini noti,
2) calcola la soluzione del sistema lineare,
3) calcola l'errore relativo che si è commesso con l'aritmetica finita rispetto alla soluzione reale (che sarebbe un vettore con tutti gli elementi uguali ad 1)
   (l'errore si calcola con la norma 2),
4) calcola l'indice di condizionamento della matrice di hilbert

Come si vede dall'output, se la matrice di hilbert è di ordine n piccolo (es. 2 <= n <= 5) l'errore che si commette sulla soluzione e l'indice di condizionamento della matrice di hilbert sono piccoli, mentre se n è maggiore l'errore e l'indice di condizionamento incrementano tanto.

Si può concludere quindi che la matrice di Hilbert è una matrice fortemente mal condizionata.

3) RISOLUZIONE DEL SISTEMA LINEARE Ax=B CON LA MATRICE MAGIC CON N COMPRESO TRA 3 E 11 DISPARI

CODICE MATLAB, ANALISI E RISULTATI:

```
function [z,c] = mgc()
z = zeros(5,1);
c = zeros(5,1);
cont = 1;
    for n=3:2:11
        M = magic(n);
        b = zeros(n,1);
        %calcolo termine noto
        for i=1:1:n
            somma = 0;
            for j=1:1:n
                somma = somma + M(i,j);
            end
            b(i,1) =  somma;
        end
        x = gauss_pivoting(M,b);
        reale = ones(n,1);
        z(cont,1) = norm(reale - x) / norm(reale);
        c(cont,1) = cond(M);
        cont = cont + 1;
    end
end
```

ESEMPIO DI UTILIZZO:

```
format long

[z,c] = mgc()
```

z = (errore relativo rispetto alla soluzione reale)

```
                   0          n = 3
   0.000000000000000
   0.000000000000003
   0.000000000000003
  31.893888385530339          n = 11
```

c = (indice di condizionamento della matrice quadrato magico)

```
   4.330127018922192          n = 3
   5.461822491270522
   7.111323446624708
   9.101650755323497
  11.102127391889686          n = 11
```

ANALISI:

la funzione magic genera una matrice quadrata magica (contiene un quadrato magico) di ordine n (con n che va da 3 a 11, solo n displari) e, per ogni matrice generata:

2) crea il vettore dei termini noti,
2) calcola la soluzione del sistema lineare,
3) calcola l'errore relativo che si è commesso con l'aritmetica finita rispetto alla soluzione reale (che sarebbe un vettore con tutti gli elementi uguali ad 1)
   (l'errore si calcola con la norma 2),
4) calcola l'indice di condizionamento della matrice quadrata magica

Come si vede dall'output, se la matrice quadrata magica è di ordine n piccolo (es. n <= 9) l'errore che si commette sulla soluzione e l'indice di condizionamento della matrice quadrata magica sono molto piccoli, mentre se n è maggiore (n >= 11) l'errore e l'indice di condizionamento sono molto alti.

Si può concludere quindi che la matrice quadrata magica è una matrice ben condizionata solo se è di ordine n piccolo (n <= 9), altrimenti è mal condizionata.

4) UTILIZZA IL METODO PIU' ADATTO PER RISOLVERE IL SEGUENTE SISTEMA LINEARE Rx=b

```
R =

     1     2     3     4     5
     0     3     4     5     6
     0     0     5     6     7
     0     0     0     7     8
     0     0     0     0     9


b =

    15
    18
    18
    15
     9


x = indietro(R,b)


x =

     1
     1
     1
     1
     1
```

Aggiungi al vettore termine noto una perturbazione nel seguente modo: bp=b+deltab dove deltab=rand(5,1)*1e-03; Calcola la soluzione del sistema B xp= bp; Calcola l'errore relativo della soluzione xp di questo sistema rispetto a quella del sistema Bx=b.
Cosa puoi concludere sulla matrice B?

```
deltab=rand(5,1)*1e-03

deltab =

   1.0e-03 *

   0.814723686393179
   0.905791937075619
   0.126986816293506
   0.913375856139019
   0.632359246225410
```

```
bp = b + deltab

bp =

  15.000814723686393
  18.000905791937075
  18.000126986816294
  15.000913375856140
   9.000632359246225


x2 = indietro(R,bp)

x2 = (soluzione col vettore di termini noti perturbato)

   1.000151541419181
   1.000255353697661
   0.999866811155393
   1.000050182678341
   1.000070262138469


err = abs((x - x2)./x)

err = (calcolo errore fra la soluzione esatta e la soluzione
       perturbata)

   1.0e-03 *

   0.151541419181278
   0.255353697661276
   0.133188844606891
   0.050182678340560
   0.070262138469390


cond(R)

ans = (indice di condizionamento della matrice R)

  23.870030846699102
```

Come si può vedere, l'errore commesso nel calcolo della soluzione perturbata è molto basso, quindi la perturbazione dei termini noti non ha influenzato negativamente la soluzione reale.

Inoltre si può concludere dicendo che la matrice R è ben condizsionata perchè ha un indice di condizionamento molto basso (circa 23).