ESERCITAZIONE MATLAB 4: Risoluzione di sistemi triangolari

1. Si scriva una function MATLAB che implementi il metodo delle sostituzioni successive in avanti per la risoluzione di un generico sistema triangolare inferiore, ovvero un sistema della seguente forma:

$$\begin{cases} a_{11}\,x_1 & & & & & = & b_1 \\ a_{21}\,x_1 & + & a_{22}\,x_2 & & & = & b_2 \\ \vdots & & \vdots & \ddots & & & \vdots \\ a_{N1}\,x_1 & + & a_{N2}\,x_2 & + \ \dots \ + & a_{NN}\,x_N & = & b_N \end{cases}$$

Riscritto in forma matriciale esso è dato da:

$$A\mathbf{x} = \mathbf{b},$$

dove

$$A = \begin{pmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & \vdots & \ddots & \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix},$$

costituiscono la matrice dei coefficienti, il vettore delle incognite ed il vettore dei termini noti del sistema, rispettivamente.

```
%
% trianginf.m
%
% Metodo delle sostituzione successive in avanti per la
% risoluzione di un sistema triangolare inferiore.
%
% Dati di INPUT:
% A        matrice dei coefficienti
% b        vettore dei termini noti
%
%
% Dati di OUTPUT:
% x        soluzione del sistema.
%
```

Il codice deve anche controllare, <u>senza costo computazionale aggiuntivo</u>, che il sistema specificato in input è ben definito. Qualora questo non sia vero, la risoluzione deve essere interrotta utilizzando il comando `error` di Matlab. Ad esempio, se A è singolare si utilizzi

```
error('Impossibile risolvere il sistema perche'' A e'' singolare')
```

2. Si applichi la function del precedente esercizio per risolvere il seguente sistema

$$\begin{cases} 10x_1 & & & & & & & = & 10 \\ 8x_1 & + & 9x_2 & & & & & = & 17 \\ 5x_1 & + & 6x_2 & + & 7x_3 & & & = & 18 \\ x_1 & + & 2x_2 & + & 3x_3 & + & 4x_4 & = & 10 \end{cases}$$

la cui soluzione esatta è $x_j = 1$ per ogni $j = 1, 2, 3, 4$.

3. Si applichi la function del primo esercizio per risolvere il seguente sistema

$$\begin{cases} 10x_1 & & & & & & & = & 10 \\ 8x_1 & + & 9x_2 & & & & & = & 17 \\ 6x_1 & + & 7x_2 & & & & & = & 18 \\ x_1 & + & 2x_2 & + & 3x_3 & + & 4x_4 & = & 10 \end{cases}$$

4. La seguente matrice è molto mal condizionata

$$A = \begin{pmatrix} 1 & & & & \\ 100 & 1 & & & \\ & 100 & 1 & & \\ & & \ddots & \ddots & \\ & & & 100 & 1 \end{pmatrix} \in \mathbb{R}^{10\times 10}. \tag{1}$$

Per verificarlo, dopo averla costruita, si utilizzi il comando `cond(A,inf)` che ne calcola il numero di condizione in norma infinito.
Si utilizzi quindi la function del primo esercizio per risolvere i seguenti due sistemi lineari

$$A\mathbf{x}_1 = \mathbf{b}_1, \qquad A\mathbf{x}_2 = \mathbf{b}_2$$

dove A è la matrice in (1) mentre

$$\mathbf{b}_1 = (1, 101, \ldots, 101)^T \in \mathbb{R}^{10}, \qquad \mathbf{b}_2 = \frac{1}{10}\mathbf{b}_1.$$

Le soluzioni esatte sono $\mathbf{x}_1 = (1, 1, \ldots, 1)^T \in \mathbb{R}^{10}$ e $\mathbf{x}_2 = \frac{1}{10}\mathbf{x}_1$, rispettivamente. Si spieghino i risultati numerici ottenuti.

5. Si scriva una function MATLAB che implementi il metodo delle sostituzioni successive all'indietro per la risoluzione di un generico sistema triangolare superiore ovvero un sistema la cui matrice dei coefficienti è del seguente tipo

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ & a_{22} & \cdots & a_{2N} \\ & & \ddots & \vdots \\ & & & a_{NN} \end{pmatrix}.$$

6. Sia $\{x_j\}$ una successione che soddisfa

$$x_j + x_{j+1} = 1, \qquad j = 1, 2, 3, \ldots$$

Quale valore deve assumere x_1 affinchè $x_{50} = 16$?
Suggerimento: il problema può essere riformulato come risoluzione di un sistema lineare triangolare superiore.

SOLUZIONI:

1.
```
function x = trianginf(A,b)

%   x = trianginf(A,b)
%
%   Applica il metodo delle sostituzioni successive
%   in avanti per la risoluzione del sistema lineare
%
%   A x = b
%
%   assumendo che la matrice dei coefficienti A sia
%   triangolare inferiore.


[ma,na] = size(A);

if (ma~=na),
    error('La matrice A non e'' quadrata')
end

nb = length(b);

if (nb~=na),
    error('Le dimensioni di A e b non sono compatibili')
end

x = b;

for i=1:nb
    if A(i,i) == 0
       error('Impossibile risolvere il sistema perche'' A e'' singolare')
    end
    for j=1:i-1
        x(i) = x(i) - A(i,j)*x(j);
    end
    x(i) = x(i)/A(i,i);
end
```

La seguente è una versione alternativa che sfrutta meglio le capacità di MATLAB

```
function x = trianginf(A,b)

%   x = trianginf(A,b)
%
%   Applica il metodo delle sostituzioni successive
%   in avanti per la risoluzione del sistema lineare
%
%   A x = b
%
%   assumendo che la matrice dei coefficienti A sia
%   triangolare inferiore.
```

```
[ma,na] = size(A);

if (ma~=na),
    error('La matrice A non e'' quadrata')
end

nb = length(b);

if (nb~=na),
    error('Le dimensioni di A e b non sono compatibili')
end

x = zeros(nb,1);

for i=1:nb
    if A(i,i) == 0
       error('Impossibile risolvere il sistema perche'' A e'' singolare')
    end
    x(i) = (b(i) -A(i,1:i-1)*x(1:i-1))/A(i,i);
end
```

2.
```
>>  A = [10 0 0 0; 8 9 0 0;5 6 7 0; 1 2 3 4]

A =

    10     0     0     0
     8     9     0     0
     5     6     7     0
     1     2     3     4

>> b = [10 17 18 10]'

b =

    10
    17
    18
    10

>> x = trianginf(A,b)

x =

     1
     1
     1
     1

>> A = [10 0 0 0; 8 9 0 0;5 6 0 0; 1 2 3 4]
```

```
A =

    10     0     0     0
     8     9     0     0
     5     6     0     0
     1     2     3     4

>> b = [10 17 18 10]'

b =

    10
    17
    18
    10

>> x = trianginf(A,b)
??? Error using ==> trianginf at 30
Impossibile risolvere il sistema perche' A e' singolare
```

3.
```
>> A=eye(10)+100*diag(ones(9,1),-1);
>> cond(A,inf)
Warning: Matrix is close to singular or badly scaled.
         Results may be inaccurate. RCOND = 9.801980e-21.
> In cond at 48

ans =

   1.0202e+20

>> b1=[1;101*ones(9,1)];
>> b2=0.1*b1;
>> x1=trianginf(A,b1)

x1 =

     1
     1
     1
     1
     1
     1
     1
     1
     1
     1

>> x2=trianginf(A,b2)

x2 =
```

```
    0.1000
    0.1000
    0.1000
    0.1000
    0.1000
    0.1000
    0.1000
    0.1014
   -0.0407
   14.1702
```

Per il primo sistema lineare il risultato calcolato è esatto perchè non sono presenti pertubazioni né sulla matrice dei coefficienti né sul vettore dei termini noti. Per il secondo sistema lineare, invece, sono presenti perturbazioni sul vettore dei termini noti dato che 0.1 non è un numero macchina. Tali perturbazioni sono quindi notevolmente amplificate nel risultato $\mathbf{x}_2$ dato il mal condizionamento di A.

4.
```
function x = triangsup(A,b)

%   x = triangsup(A,b)
%
%   Applica il metodo delle sostituzioni successive
%   all'indietro per la risoluzione del sistema lineare
%
%   A x = b
%
%   assumendo che la matrice dei coefficienti A sia
%   triangolare superiore.


[ma,na] = size(A);

if (ma~=na),
    error('La matrice A non e'' quadrata')
end

nb = length(b);

if (nb~=na),
    error('Le dimensioni di A e b noti non sono compatibili')
end

x = zeros(nb,1);

for i=nb:-1:1
    if A(i,i) == 0
       error('Impossibile risolvere il sistema perche'' A e'' singolare')
    end
    x(i) = (b(i) -A(i,i+1:nb)*x(i+1:nb))/A(i,i);
end
```

5. Per determinare il valore di x_1 è sufficiente risolvere il sistema lineare triangolare

superiore

$$A\mathbf{x} = \mathbf{b}$$

dove

$$A = \begin{pmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \\ & & & & 1 \end{pmatrix} \in \mathbb{R}^{50\times 50}$$

$$\mathbf{x} = (x_1, x_2, \ldots x_{49}, x_{50})^T, \qquad \mathbf{b} = (1, 1, \ldots 1, 16)^T \in \mathbb{R}^{50}$$

e quindi considerare il valore della prima componente del suo vettore soluzione. Per fare questo si possono eseguire le seguenti istruzioni:

```
>> A = eye(50)+diag(ones(49,1),1);
>> b = [ones(49,1); 16];
>> x = triangsup(A,b);
>> x(1)

ans =

   -15
```