ISSN 2239-7035

GIOVANNI DI CECCA
VIRGINIA BELLINO

$$\begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & 0 & a_{33}^{(2)} & \dots & a_{3n}^{(2)} \\ \vdots & \vdots & \dots & \dots & \vdots \\ 0 & 0 & \dots & 0 & a_{nn}^{(n-1)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(1)} \\ b_3^{(2)} \\ \vdots \\ b_n^{(n-1)} \end{pmatrix}$$

METODO DI ELIMINAZIONE DI GAUSS CON PIVOTING PARZIALE

dicecca.net - contributi scientifici

Anno I - № 11 – Supplemento al MONITORE NAPOLETANO - № 153 – Novembre 2020

dicecca.net
web site

www.monitorenapoletano.it/dicecca.net-cs

Direttore Responsabile: Giovanni Di Cecca

Collana dicecca.net – Computer Science

Anno I - № 11 – Supplemento al Numero 153 – Novembre 2020

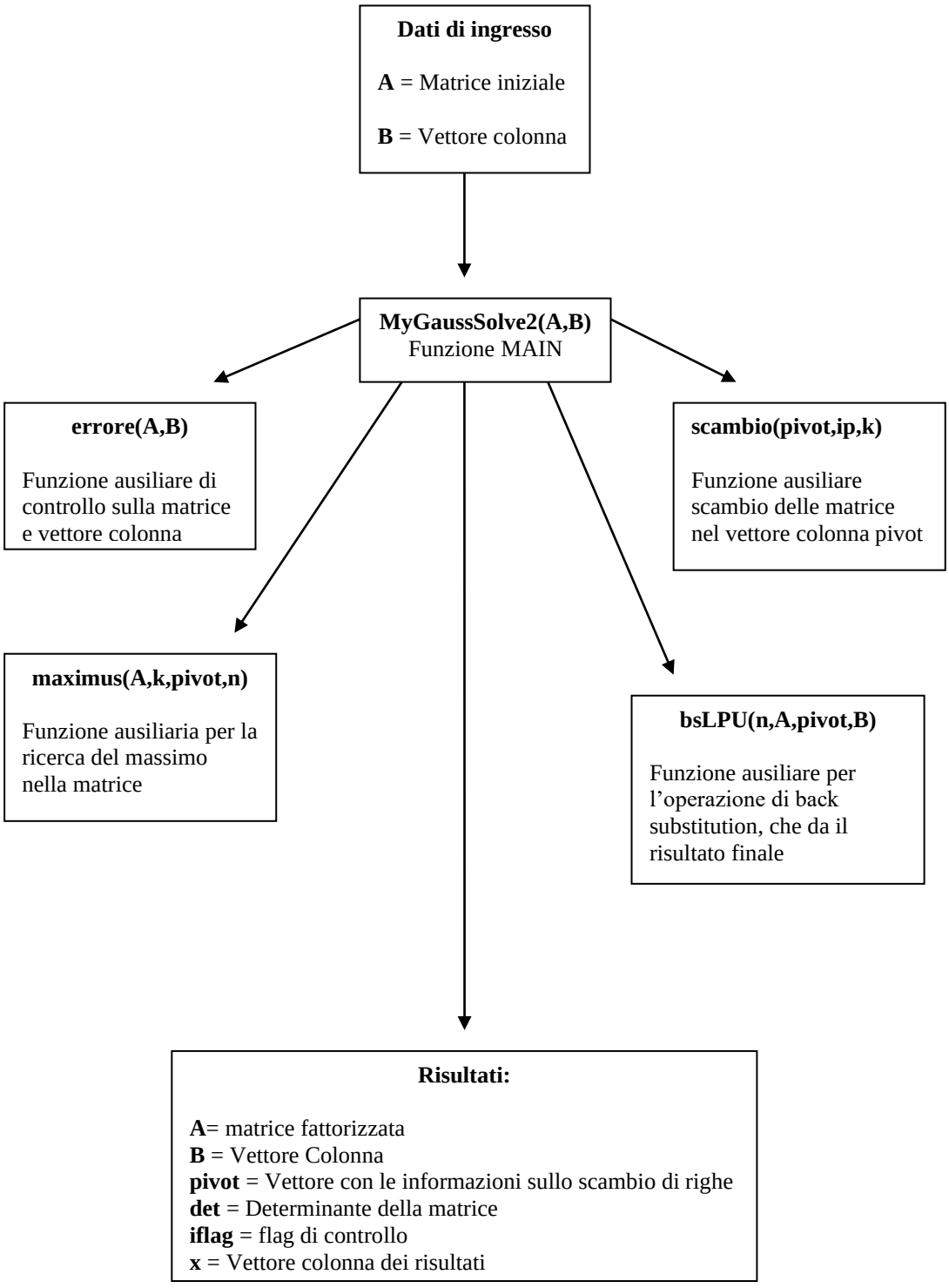
Periodico Mensile Registrato presso il Tribunale di Napoli № 45 dell'8 giugno 2011

ISSN: 2239-7035

Indice

Progetto Metodo di Gauss con Pivoting Parziale

Schema grafico



SEZIONE 1

DOCUMENTAZIONE ESTERNA

- **Scopo:** la funzione esegue la risoluzione di un sistema di equazioni lineari del tipo **Ax = b** utilizzando il metodo di eliminazione di Gauss con pivoting parziale.

- **Specifiche:**

  function [A, B, pivot, det, iflag, x] = myGaussSolve2 (A, B)

- **Descrizione:**

  Dopo aver eseguito un controllo sulla dimensione dei dati di input, l'algoritmo esegue la fattorizzazione LU della matrice dei coefficienti, trasformando la matrice iniziale in una matrice a gradini equivalente, da cui vengono ricavate le soluzioni del sistema, applicando il metodo della back substitution (o sostituzione all'indietro).

- **Riferimenti bibliografici:**

  James F. Epperson
  INTRODUZIONE ALL'ANALISI NUMERICA
  McGraw-Hill

- **Lista dei parametri:**

  Parametri di input:
  A: matrice dei coefficienti del sistema lineare. E' di dimensione n*n.
  B: vettore dei termini noti. E' di dimensione n.

  Parametri di output:
  A: matrice dei coefficienti modificata.
  B: vettore dei termini noti modificati.
  pivot: vettore che registra gli scambi effettuati durante il pivoting.
  det: determinante della matrice.
  iflag: indicatore di errori.
  x: vettore contenente le soluzioni finali del sistema.

- **Indicatori di errore:**

  iflag: indica se durante la procedura si verificano degli errori,bloccando in tal caso l'esecuzione.. Può assumere i seguenti valori:

  0 : significa che la procedura è stata eseguita senza problemi.
  1 : la matrice dei coefficienti non è quadrata

2 : il vettore dei termini noti non ha la stessa dimensione della matrice dei coefficienti
3 : il numero di colonne di termini noti inseriti in input è > 1.
-1: matrice singolare.

- **Funzioni ausiliarie:**

function iflag = errore (A, B)
la routine esegue un controllo sulla dimensione della matrice e del vettore dei termini noti inseriti in input. Se ci sono errori, l'indicatore iflag assume valore 1 e l'esecuzione termina.

function [max, ip] = maximus (A, k, pivot, n)
la routine cerca l'elemento massimo in valore assoluto sulla colonna k e riga da k+l ad n,salvandone il valore e l'indice di riga.

function pivot = scambio (pivot, ip, k)
la routine esegue lo scambio degli indici nel vettore pivot.

function [x] = bsLPU (n, A, pivot, B)
la routine esegue la back-substitution sul sistema a gradini ottenuto applicando il metodo di gauss,consentendo di ricavare le soluzioni finali del sistema Ax = b.

- **Complessità computazionale:**

la complessità totale della funzione è T(n) = 2/3 n^3

- **Accuratezza fornita:** poiché il metodo di gauss appartiene alla categoria dei metodi risolutivi diretti, la soluzione ottenuta risulta esatta a meno di un errore di round off legato alla precisione della macchina.

- **Esempio d'uso:**

Esempio – 1

```
EDU>> a=[1 2 -1
   1 3 1
   2 4 -1]

a =

    1    2   -1
    1    3    1
    2    4   -1

EDU>> b=[1
   5
```

```
    6]

b =

     1
     5
     6

EDU>> [A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)


A =

    0.5000         0   -0.5000
    0.5000    1.0000    1.5000
    2.0000    4.0000   -1.0000


B =

    -2
     2
     6


pivot =

     3
     2
     1

det =

    -2


iflag =

     0


x =

    13
    -4
     4
```

@@@

Esempio – 2

```
EDU>> a=[1,1,0;2,1,3;1,2,-3]

a =

     1     1     0
     2     1     3
     1     2    -3

EDU>> b=[4;3;5]

b =

     4
     3
     5

EDU>> [A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)


Warning: One or more output arguments not assigned during call to 'mygausssolve2'.

A =

    0.5000    0.3333         0
    2.0000    1.0000    3.0000
    0.5000    1.5000   -4.5000


B =

    1.3333
    3.0000
    3.5000


pivot =

     2
     3
     1


det =

     0

iflag =

    -1
```

SEZIONE 2

CODICE MATLAB

```
%  Progetto Metodo di Gauss con Pivoting Parziale
%
%             Programma elaborato da
%
%      Giovanni DI CECCA & Virginia BELLINO
%           50 / 887           408 / 466
%
%             http://www.dicecca.net

% Funzione Main

% SCOPO: la funzione esegue la risoluzione di un sistema
% di equazioni lineari del tipo Ax = b utilizzando
% il metodo di eliminazione di Gauss con pivoting parziale

% Parametri di input:
% A: matrice dei coefficienti del sistema lineare. E' di dimensione n*n.
% B: vettore dei termini noti. E' di dimensione n.

% Parametri di output:
% A: matrice dei coefficienti  modificata.
% B: vettore dei termini noti modificati.
% pivot: vettore che registra gli scambi effettuati durante il pivoting.
% det: determinante della matrice.
% iflag: indicatore di errori.
% x: vettore contenente le soluzioni finali del sistema.


function [A,B,pivot,det,iflag,x]=myGaussSolve2(A,B)


% Inizializza il flag di errore
% Controllo sulla dimensione della matrice A e del vettore B.
% Se non ci sono errori, si può continuare

iflag = errore (A,B);

% Dimensione della matrice
s = size(A);

% Considera il valore delle righe
n=s(1);

% Crea il vettore colonna pivot
for i=1:n

   pivot(i,1)=[i];

end
```

```
% Inizializza il determinante
det = 1;

% k indica il passo dell'algoritmo
for k=1:n-1

    % Cerca il massimo mediante la funzione apposita salvandone valore e i
ndice
    % di riga
    [max,ip]=maximus(A,k,pivot,n);


   if (max == 0) % NOTA1: significa che sulla diagonale c'è uno zero...
      det = 0;   %...per cui il determiniante di una matrice triangolare,
                 % pari al prodotto degli elementi della diagonale, è
                 % necessariamente uguale a zero

      iflag = -1; % segnala il verificarsi di una anomalia durante
                  % l'applicazione dell'algoritmo:il sistema non
                  % è compatibile

      break % Blocca l'esecuzione del programma

   end % Fine dell'if

   % Scambia effettivamente gli indici nel vettore pivot
   % solo se k é diverso da ip
   if (ip ~= k)

      % Esegui la funzione di scambio
      pivot=scambio(pivot,ip,k);

      det = -det; % ad ogni scambio di riga il determinante cambia di segn
o
   end

   % Inizio della decomposizione LU della matrice dei coefficienti
   for i=k+1:n

      % Calcola e memorizza il moltiplicatore nella posizione (i,k)
      A(pivot(i),k) = A(pivot(i),k)/A(pivot(k),k);

      % Modifica gli elementi della riga i-esima in base al valore del mol
tiplicatore
      % memorizzato in A(i,k)
      for j=k+1:n
```

```
            A(pivot(i),j)=A(pivot(i),j)-A(pivot(i),k)*A(pivot(k),j);

        end % Fine del ciclo j

        % Modifica il valore dei termini noti
        B(pivot(i))=B(pivot(i))-A(pivot(i),k)*B(pivot(k));

    end % fine del ciclo i

     % Calcola il determinante moltiplicando gli elementi della diagonale
     % principale della matrice modificata
     det = det * A(pivot(k),k);

end; % del ciclo di k

if (A(pivot(n),n) == 0) % Vedi NOTA1

     det = 0;
     iflag = -1;
     break
end

% Risoluzione del sistema con back Substitution
[x]=bsLPU(n,A,pivot,B);
```

```
%  Progetto Metodo di Gauss con Pivoting Parziale
%
%             Programma elaborato da
%
%      Giovanni DI CECCA & Virginia BELLINO
%           50 / 887           408 / 466
%
%             http://www.dicecca.net


% Routine dei controlli sulla matrice A e vettore colonna B

% SCOPO: La funzione fa il controllo sulla matrice A e vettore colonna B
%
% Parametri di Input: Matrice A, Vettore colonna B
%
% Parametri di Output: iflag = Flag di controllo sugli errori
%                              0 = OK
%                              1 = Matrice non quadrata
%                              2 = Il vettore colonna dei termini noti non è
uguale alla matrice
%                              3 = B è un vettore colonna non una matrice

function iflag=errore(A,B)

% Associa alla variabile matrix la lunghezza e larghezza della matrice A
matrix=size(A);

% Controllo sulla quadratura della matrice
if matrix(1)==matrix(2)

    iflag = 0; % Flag di controllo 0 = OK

else

    iflag = 1; % Matrice non quadrata
    break
end

% calcola la dimensione del vettore colonna
vectorcln=size(B);

% Controllo sul vettore colonna b, il numero delle righe di b deve essere
% uguale a quello della matrice A
if matrix(1)==vectorcln(1)

    iflag = 0; % Flag di controllo 0 = Ok
else
```

```
    iflag = 2; % Il vettore colonna dei termini noti non ha dimensione ugu
ale alla matrice
    break
end

% Controllo sul vettore colonna b, il numero delle colonne deve essere 1
if vectorcln(2)==1
    iflag = 0; % Flag di controllo 0 = Ok

else

    iflag = 3; % B è un vettore colonna non una matrice
    break
end
```

```
%  Progetto Metodo di Gauss con Pivoting Parziale
%
%             Programma elaborato da
%
%      Giovanni DI CECCA & Virginia BELLINO
%           50 / 887           408 / 466
%
%             http://www.dicecca.net

% funzione maximus

% SCOPO: la funzione ha lo scopo di calcolare il massimo della colonna k s
elezionata, salvamdone valore e indice
%
% Parametri di Input: A = matrice
%                     k = Passo dell'algoritmo
%                     pivot = Vettore colonna che contiene gli scambi del
a matrice
%                     n = Dimensione della matrice
%
% Parametri di Output: max = massimo trovato
%                       ip = Indice del massimo trovato

% Funzione che trova l'elemento massimo in val. assoluto sulla colonna k e
 riga da k+1 ad n
function [max,ip]=maximus(A,k,pivot,n)

% Sulla colonna selezionata si ricerca il massimo in valore assoluto,salva
ndone valore e indice
max = abs(A(pivot(k),k));

ip = k;    % ip sta per indice del massimo

for p=k+1:n

    if (abs(A(pivot(p),k))>max)
        % memorizza temporaneamente il valore del massimo
        % della colonna k al di sotto della diagonale principale  e del su
o indice

        max = abs(A(pivot(p),k));

        ip = p;
    end

end
```

```
%  Progetto Metodo di Gauss con Pivoting Parziale
%
%             Programma elaborato da
%
%      Giovanni DI CECCA & Virginia BELLINO
%           50 / 887           408 / 466
%
%             http://www.dicecca.net

% Funzione scambio

% SCOPO: la funzione ha lo scopo di scambiare i valori degli indici nel ve
ttore colonna pivot
%
% Parametri di Input: pivot = Vettore colonna con gli indici della matrice
%                     ip = indice del miglior pivot
%                     k = passo dell'algoritmo
%
% Parametri di Output: pivot = Vettore colonna che contiene gli scambi del
la matrice


% Funzione che esegue lo scambio degli indici,delle righe
% della matrice A,contenuti nel vettore p
function pivot=scambio(pivot,ip,k)

temp=pivot(k);

pivot(k)=pivot(ip);

pivot(ip)=temp;
```

```
%  Progetto Metodo di Gauss con Pivoting Parziale
%
%             Programma elaborato da
%
%      Giovanni DI CECCA & Virginia BELLINO
%           50 / 887           408 / 466
%
%             http://www.dicecca.net

% Funzione di Back Substitution

% SCOPO: la funzione ha lo scopo di calcolare i valori delle incognite med
iante sostituzione all'indietro
%
% Parametri di Input: n = righe della matrice
%                     A = matrice triangolarizzata
%                     pivot = Vettore colonna che contiene gli scambi eseg
uiti sulla matrice
%                     B = Vettore colonna dei termini noti modificati
%
% Parametri di Output: x = Vettore colonna con i risultati

function [x]=bsLPU(n,A,pivot,B)

% Calcolo dell'ultima riga della matrice
x(n,1)=B(pivot(n),1)/A(pivot(n),n);

% Ciclo della funzione di sostituzione all'indietro partendo dalla penultim
a riga fino alla prima
for i=n-1:-1:1

    % Inizializzazione della variabile somma
    somma = 0;

    % Calcolo delle incoglie per sostitutzione
    for k=(i+1):n
        somma = somma + A(pivot(i),k) * x(k,1);
    end

    % Calcolo del valore delle incognite
    x(i,1)=(B(pivot(i),1)-somma)/A(pivot(i),i);

end
```

```
%  Progetto Metodo di Gauss con Pivoting Parziale
%
%              Programma elaborato da
%
%       Giovanni DI CECCA & Virginia BELLINO
%            50 / 887           408 / 466
%
%              http://www.dicecca.net


% Funzione di Test

% Lo scopo di questa funzione è quello di andare ad eseguire
% tutti i test previsti nell'esercizio, in modo automatico
% ed in successione.
%
% Per evitare possibili errori nell'uso delle matrici,
% ad ogni test viene eseguito un reset della memoria,
% sia video (il clc) sia fisica (il clear)

clc % pulisci schermo

clear % Pulisci memoria

disp(' Progetto Metodo di Gauss con Pivoting Parziale')
disp(' ')
disp('              Programma elaborato da')
disp(' ')
disp('       Giovanni DI CECCA & Virginia BELLINO')
disp('             50 / 887          408 / 466')
disp(' ')
disp('              http://www.dicecca.net')
disp(' ')
disp(' ')

disp('test - 1')

a=[2 0 -2 0
    9 10 -4 -1
    3 4 -2 1
    1 -2 -2 3]

b=[0
    -5
    -3
    1]

[A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)
```

```
disp('Premere un tasto per continuare')
pause

%------------------------------------------------------------------------

clc % pulisci schermo

clear % Pulisci memoria

disp('test - 2')

a=[3 10 9
    8 4 -10
    -18 8 48]

b=[1
    0
    -2]

[A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)

disp('Premere un tasto per continuare')
pause

%------------------------------------------------------------------------

clc % pulisci schermo

clear % Pulisci memoria

disp('test - 3')

a=[10 1 2 3
    1 2 0 0
    2 0 6 1
    3 0 1 5]

b=[17
    -6
    12
    31]

[A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)

disp('Premere un tasto per continuare')
pause

%------------------------------------------------------------------------
```

```
clc % pulisci schermo

clear % Pulisci memoria

disp('test - 4')

a=[7 -2 3
    1 11 3
    3 12 15]

b=[8
    35
    42]

[A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)

disp('Premere un tasto per continuare')
pause

%-----------------------------------------------------------------------

clc % pulisci schermo

clear % Pulisci memoria

disp('test - 5.1')

q=5;

% Ciclo di creazione di A
for t=1: q
    for z=1: q
        a(t,z)=(t)^(z-1);
    end
end

% Ciclo di creazione di B
for t=1: q
    for z=1: q
        b(t,1)=a(t,q)/2;
    end
end


[A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)

disp('Premere un tasto per continuare')
pause
```

```
%-----------------------------------------------------------------------

clc % pulisci schermo

clear % Pulisci memoria

disp('test - 5.2')

q=10;

% Ciclo di creazione di A
for t=1: q
    for z=1: q
        a(t,z)=(t)^(z-1);
    end
end

% Ciclo di creazione di B
for t=1: q
    for z=1: q
        b(t,1)=a(t,q)/2;
    end
end


[A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)

disp('Premere un tasto per continuare')
pause

%-----------------------------------------------------------------------

clc % pulisci schermo

clear % Pulisci memoria

disp('test - 5.3')

q=20;

% Ciclo di creazione di A
for t=1: q
    for z=1: q
        a(t,z)=(t)^(z-1);
    end
end

% Ciclo di creazione di B
for t=1: q
```

```
    for z=1: q
        b(t,1)=a(t,q)/2;
    end
end

[A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)

disp('Premere un tasto per continuare')
pause

%------------------------------------------------------------------------

clc % pulisci schermo

clear % Pulisci memoria

disp('test - 5.4')

q=50;

% Ciclo di creazione di A
for t=1: q
    for z=1: q
        a(t,z)=(t)^(z-1);
    end
end

% Ciclo di creazione di B
for t=1: q
    for z=1: q
        b(t,1)=a(t,q)/2;
    end
end

[A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)

disp('Premere un tasto per continuare')
pause

%------------------------------------------------------------------------

clc % pulisci schermo

clear % Pulisci memoria

disp('test - 6')

a=[10E-15 3
    2 3]
```

```
b=[3
    5]


[A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)

disp('Premere un tasto per continuare')
pause

%------------------------------------------------------------------------

clc % pulisci schermo

clear % Pulisci memoria

disp('test - 7')

a=[1 1
    1 1.001]

b=[1
    0]

[A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)

disp('Premere un tasto per continuare')
pause

%------------------------------------------------------------------------

clc

clear

disp('test - 7.1')

a=[1 (1+4/1000)
    1 1.001]

b=[1
    0]

[A,B,pivot,det,iflag,x]=myGaussSolve2(a,b)
```

SEZIONE 3

```
 Progetto Metodo di Gauss con Pivoting Parziale

            Programma elaborato da

      Giovanni DI CECCA & Virginia BELLINO
            50 / 887         408 / 466

            http://www.dicecca.net


test - 1

a =

     2     0    -2     0
     9    10    -4    -1
     3     4    -2     1
     1    -2    -2     3


b =

     0
    -5
    -3
     1


A =

    0.2222    0.7143         0   -2.0000
    9.0000   10.0000   -4.0000   -1.0000
    0.3333   -0.2143   -1.0000    2.0000
    0.1111   -3.1111   -1.5556    3.1111


B =

         0
   -5.0000
   -1.0000
    1.5556


pivot =

     2
     4
     3
     1


det =

   28.0000


iflag =

     0
```

```
x =

    1.0000
   -1.0000
    1.0000
         0

Premere un tasto per continuare
```

```
test - 2

a =

     3    10     9
     8     4   -10
   -18     8    48


b =

     1
     0
    -2

Warning: One or more output arguments not assigned during call to 'mygausssolve2'.
> In C:\Documents and Settings\Giovanni\Desktop\Calcolo progetto - 1\progetto - MATLAB\↙
test.m at line 69

A =

   -0.1667   11.3333   17.0000
   -0.4444    0.6667         0
  -18.0000    8.0000   48.0000


B =

    0.6667
   -1.3333
   -2.0000


pivot =

     3
     1
     2


det =

     0


iflag =

    -1

Premere un tasto per continuare
```

```
test - 3

a =

    10     1     2     3
     1     2     0     0
     2     0     6     1
     3     0     1     5


b =

    17
    -6
    12
    31


A =

   10.0000    1.0000    2.0000    3.0000
    0.1000    1.9000   -0.2000   -0.3000
    0.2000   -0.1053    5.5789    0.3684
    0.3000   -0.1579    0.0660    4.0283


B =

   17.0000
   -7.7000
    7.7895
   24.1698


pivot =

     1
     2
     3
     4


det =

  106.0000


iflag =

     0


x =

         0
   -3.0000
    1.0000
    6.0000

Premere un tasto per continuare
```

```
test - 4

a =

     7    -2     3
     1    11     3
     3    12    15


b =

     8
    35
    42


A =

    7.0000   -2.0000    3.0000
    0.1429    0.8778   -9.4667
    0.4286   12.8571   13.7143


B =

    8.0000
         0
   38.5714


pivot =

     1
     3
     2


det =

   -90


iflag =

     0


x =

    2.0000
    3.0000
         0

Premere un tasto per continuare
```

```
test - 5.1

A =

    1.0000    1.0000    1.0000    1.0000    1.0000
    1.0000    0.2500    0.7500   -1.0000   -6.0000
    1.0000    0.5000   -4.0000  -36.0000 -232.0000
    1.0000    0.7500    0.7500   -3.0000  -39.0000
    1.0000    4.0000   24.0000  124.0000  624.0000


B =

    0.5000
   -3.0000
 -116.0000
  -19.5000
  312.0000


pivot =

     1
     5
     3
     4
     2


det =

   -48


iflag =

     0


x =

         0
         0
         0
         0
    0.5000

Premere un tasto per continuare
```

test – 5.2

pivot =

1
10
5
8
2
9
3
6
4
7

det =

-4.2475e+017

iflag =

0

x =

0
0
0
0
0
0
0
0
0
0.5000

Premere un tasto per continuare

test – 5.3

pivot =

1
20
10
16
4
18
7
2
13
19
3
14
6
17
9
5

12
15
8
11

det =

-4.0411e+125

iflag =

0

x =

0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0
0.5000

Premere un tasto per continuare

test – 5.4

pivot =

1
50
25
39
9
46
16
4
33
49
2
21
43
12
30
48

```
     6
    36
     3
    41
    18
    47
    11
    45
    23
    28
    35
    37
    44
    19
    38
     8
    27
    32
    14
    29
     7
    40
    24
    10
    42
    22
    26
     5
    15
    13
    34
    17
    31
    20


det =

  -Inf


iflag =

     0


x =

         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
```

```
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
0.5000
```

Premere un tasto per continuare

COMMENTO TEST N° 5

Tutti i sistemi implementati dal test hanno come matrice dei coefficienti la matrice di Vandermonde che è soggetta a malcondizionamento. Per questo motivo, tali sistemi risultano essere malcondizionati, poiché dati e soluzione subiscono perturbazioni che non sono dello stesso ordine. Infatti, al crescere della dimensione della matrice dei coefficienti, la soluzione del sistema non subisce variazioni.

```
test - 6

a =

    0.0000    3.0000
    2.0000    3.0000


b =

     3
     5


A =

    0.0000    3.0000
    2.0000    3.0000


B =

    3.0000
    5.0000


pivot =

     2
     1


det =

    -2


iflag =

     0


x =

    1.0000
    1.0000

Premere un tasto per continuare
```

COMMENTO TEST N° 6

L'utilizzo della tecnica di pivoting rende l'algoritmo meno suscettibile agli errori di arrotondamento legati alla precisione della macchina.
Eseguendo infatti un confronto dei risultati, è possibile osservare che, se usiamo la tecnica di pivoting, le soluzioni ottenute sono .....; in caso contrario le soluzioni sono ……..
Come si può notare, nel primo caso otteniamo due valori precisi, nel secondo caso invece no.

Con pivoting

x =

1.0000
1.0000

senza pivoting

x =

0.9770
1.0000

```
test - 7

a =

    1.0000    1.0000
    1.0000    1.0010


b =

     1
     0


A =

    1.0000    1.0000
    1.0000    0.0010


B =

     1
    -1


pivot =

     1
     2


det =

     1


iflag =

     0


x =

  1.0e+003 *

    1.0010
   -1.0000

Premere un tasto per continuare
```

COMMENTO TEST N° 7

Confrontando i due risultati, si evince che la matrice dei coefficienti risulta essere malcondizionata. Infatti, una lieve perturbazione sui dati di input porta alla generazione di una soluzione del sistema molto diversa da quella generata in precedenza.

LIBERTÀ EGUAGLIANZA

MONITORE NAPOLETANO

Fondato nel 1799 da
Carlo Lauberg ed Eleonora de Fonseca Pimentel

Rifondato nel 2010
Direttore: Giovanni Di Cecca

Anno CCXXII

Contatti

C.Ph.: +39 392 842 76 67

www.monitorenapoletano.it

info@monitorenapoletano.it