Calcolo Numerico

A.A. 2019-2020

Ingegneria chimica

Sistemi Lineari: Metodi diretti

D. Vitulano

Metodo di Gauss (1)

```
% Risolve il sistema lineare Ax=b usando il metodo di
% eliminazione di Gauss senza pivoting

clear all % cancella tutte le variabili in Workspace

%lettura della matrice dei coefficienti e del vettore dei termini noti
A = input('inserisci la matrice del sistema A= ');
b = input('inserisci il vettore dei termini noti b = ');
[m,n] = size(A);

% controllo sulla dimensione di A e b
if (m~=n) | ((m==n) & (n ~=length(b)))
    error('Attenzione alla dimensione di A e b!!!');
end

% controllo dell'esistenza di un'unica soluzione
if det(A) == 0
    error('la matrice dei coefficienti e'' singolare!!!')
end
```

Metodo di Gauss (1)

```
for  k =1:n-1 % gestisce il numero di passi del metodo di Gauss
    for  i =k+1:n
          m = A(i,k)/A(k,k); % calcolo dei moltiplicatori
          for j=k:n % modifica  degli elementi sulla i-esima riga di A
                  A(i,j) = A(i,j)-m*A(k,j);
          end
         b(i) = b(i)-m*b(k); % modifica della i-esima riga di b
     end
end
```

Metodo di Gauss (1)

```
% metodo di sostituzione all'indietro.
% la matrice A è adesso triangolare superiore

x(n) = b(n)/A(n,n);
for  i=n-1:-1:1
     x(i) = (b(i)-sum(A(i,i+1:n).*x(i+1:n)))/A(i,i);
end

x = x'; % la soluzione deve essere un vettore colonna

% disp(['la soluzione del sistema e'' [',num2str(x'),' ]^T'])
fprintf('la soluzione del sistema e'' x= \n ')
fprintf('%+6.5f \n',x(:))
```

Metodo di Gauss (2)

```
% meg_with_pivot
% Risolve il sistema lineare Ax=b usando il metodo di
% eliminazione di Gauss con pivoting

clear all % cancella tutte le variabili in Workspace

%lettura della matrice dei coefficienti e del vettore dei termini noti
A = input('inserisci la matrice del sistema A= ');
b = input('inserisci il vettore dei termini noti b = ');
[m,n] = size(A);

% controllo sulla dimensione di A e b
if (m~=n) | ((m==n) & (n ~=length(b)))
    error('Attenzione alla dimensione di A e b!!!');
end

% controllo dell'esistenza di un'unica soluzione
if det(A) == 0
    error('la matrice dei coefficienti e'' singolare!!!')
end
```

Metodo di Gauss (2)

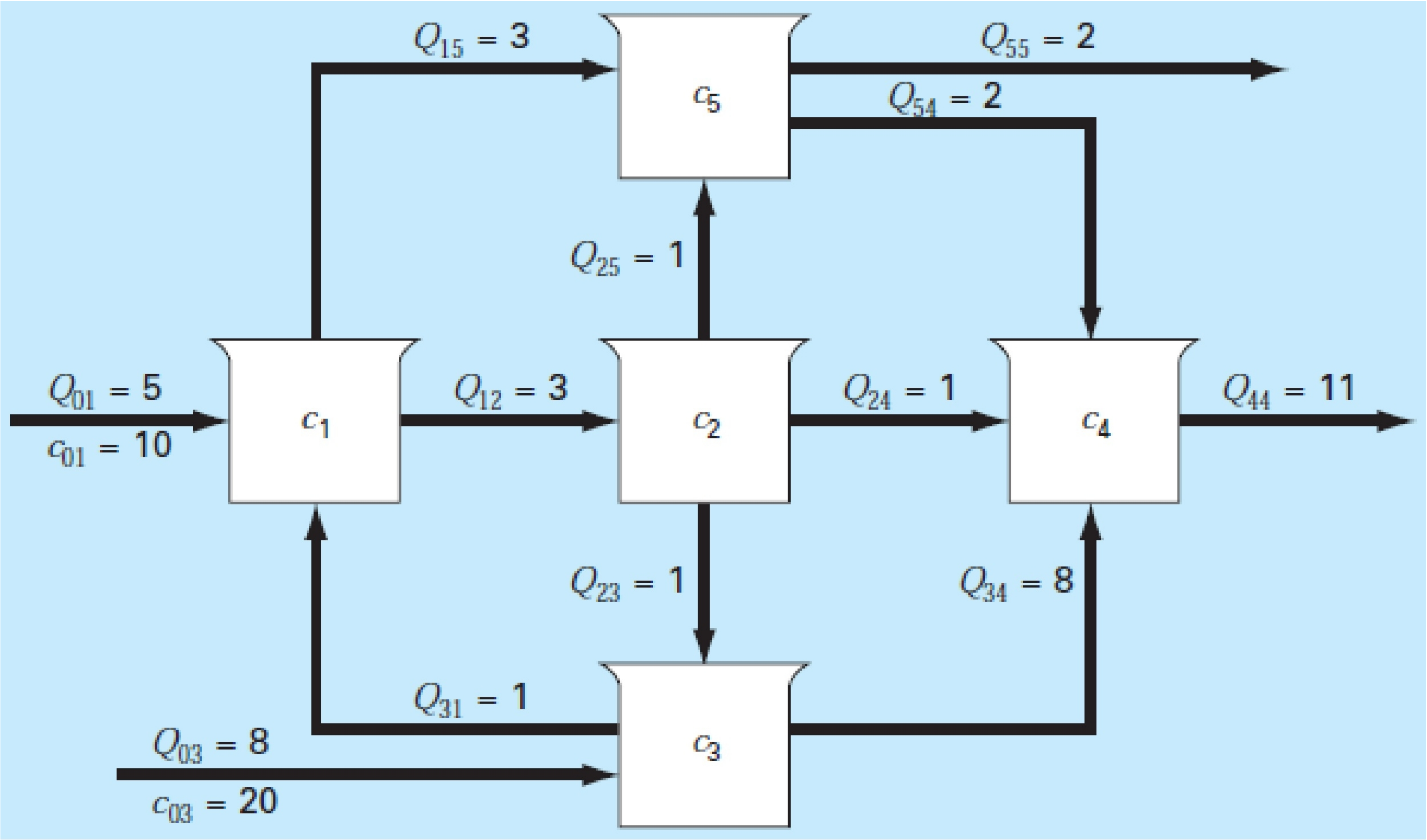
```
for  k =1:n-1 % gestisce il numero di passi del metodo di Gauss
    [tmax,ptmax] = max(abs(A(k:n,k)));
    if tmax ~= abs(A(k,k))
        appo = A(k,:);
        A(k,:) = A(k+ptmax-1,:);
        A(k+ptmax-1,:) = appo;
%           disp(A)   % si può omettere
%           pause
        appo = b(k);
        b(k) = b(k+ptmax-1);
        b(k+ptmax-1) = appo;
    end

      for  i =k+1:n
          m = A(i,k)/A(k,k); % calcolo dei moltiplicatori
             for j=k:n % modifica  degli elementi sulla i-sima riga
                     A(i,j) = A(i,j)-m*A(k,j);
             end
          b(i)=b(i)-m*b(k); % modifica  della i-sima riga componente di b
      end
end
```

Metodo di Gauss (2)

```
% metodo di sostituzione all'indietro.
% All'uscita dei cicli annidati precedenti la matrice A è triangolare
superiore
x(n) = b(n)/A(n,n);
for  i=n-1:-1:1
     x(i) = (b(i)-sum(A(i,i+1:n).*x(i+1:n)))/A(i,i);
   % x(i) = (b(i)-(A(i,i+1:n)*x(i+1:n)'))/A(i,i);
end
x = x'; % la soluzione deve essere un vettore colonna
fprintf('la soluzione del sistema e'' x= \n ')
fprintf('%+6.5f \n',x(:))
```

Condizionamento



Condizionamento

Reattore 1 Input: $5(10) + Q_{31}c_3$ Output: $Q_{12}c_1 + Q_{15}c_1$

$$5(10) + Q_{31}c_3 = Q_{12}c_1 + Q_{15}c_1$$

$$6c_1 - c_3 = 50$$

Facendo lo stesso per gli altri, si ottiene il sistema di equazioni:

$$\begin{cases} 6c_1 - c_3 = 50 \\ -3c_1 + 3c_2 = 0 \\ -c_2 + 9c_3 = 160 \\ -c_2 - 8c_3 + 11c_4 - 2c_5 = 0 \\ -3c_1 - c_2 + 4c_5 = 0 \end{cases}$$

Per la soluzione si puó (<u>si deve, nel caso di molte equazioni</u>) usare **un metodo numerico**.

(**Soluzione:** $C^T = \{11.51 \quad 11.51 \quad 19.06 \quad 17.00 \quad 11.51\}$)

Condizionamento

```
clear

A = [6 0 -1 0 0; -3 +3 0 0 0; 0 -1 9 0 0; 0 -1 -8 11 -2; -3 -1 0 0 4]

B = [50 0 160 0 0]';

K = norm(A,1) * norm(inv(A),1)

X =A\B
```

Condizionamento

A =

6	0	-1	0	0
-3	3	0	0	0
0	-1	9	0	0
0	-1	-8	11	-2
-3	-1	0	0	4

K =

10.5901

X =

11.5094
11.5094
19.0566
16.9983
11.5094

Condizionamento

B1 = [55 0 176 0 0]'

% Q_{01} = 5 C_{01} = 11 (10) => 55 (10%)
% Q_{03} = 8 C_{03} = 22 (20) => 176 (10%)

dB_on_B = norm(B1-B,1)/norm(B,1) %||deltaB|| / ||B||

max_dX_on_X = K * dB_on_B %max||deltaX|| / ||X||

X1 = A\B1

true_dX_on_X = norm(X1-X,1)/norm(X,1)

Condizionamento

```
B1 =

    55
     0
   176
     0
     0

dB_on_B =

    0.1000


max_dX_on_X =

    1.0590
```

Condizionamento

```
X1 =
   12.6604
   12.6604
   20.9623
   18.6981
   12.6604

true_dX_on_X =

    0.1000
```