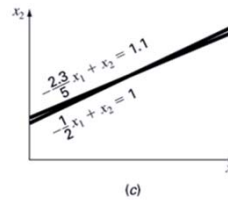
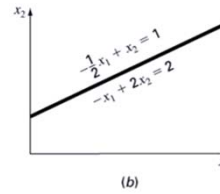
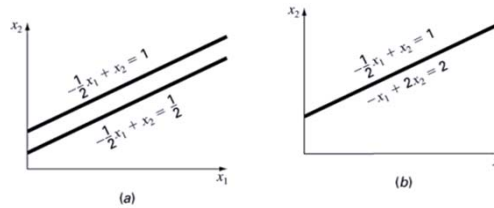
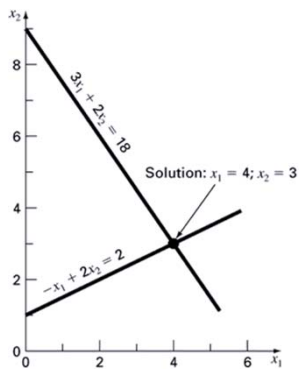


Chap. 9 Gauss 소거법

$$\begin{bmatrix}
 a_{11} & a_{12} & a_{13} & \cdots & a_{1n-1} & a_{1n} & a_{1n+1} \\
 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n-1}^{(1)} & a_{2n}^{(1)} & a_{2n+1}^{(1)} \\
 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n-1}^{(2)} & a_{3n}^{(2)} & a_{3n+1}^{(2)} \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
 0 & 0 & 0 & \cdots & a_{n-1n-1}^{(n-2)} & a_{n-1n}^{(n-2)} & a_{n-1n+1}^{(n-2)} \\
 0 & 0 & 0 & \cdots & 0 & a_{nn}^{(n-1)} & a_{nn+1}^{(n-1)}
 \end{bmatrix}$$

선형 연립방정식



Gauss 소거법 - 기본 원리

원리: 다원 1차 연립방정식에서 방정식의 양변에 같은 수를 곱하거나 나누고,
다른 방정식과 더하거나 빼주어도 연립방정식의 근이 변하지 않는다.

Ex) $2x + 3y = 1$ --- (1) :

$3x - 4y = 1$ --- (2) :

식 (3) - (4) : $17y = 1 \rightarrow y = 1/17$

위의 연립방정식을 다음과 같이 표현할 수 있다.

$$2x + 3y = 1$$

$$0x + 17y = 1$$

$$\Rightarrow y = 1/17, x = (1-3y)/2 = 7/17$$



Gauss 소거법 - 전진 소거

임의의 n원 1차 연립방정식에 대하여

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \quad (1)$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \quad (2)$$

.....

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \quad (n)$$

Gauss 소거법 – 후진 대입

단순 소거 과정을 거쳐 소거된 확대 계수행렬은

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \quad (1)'$$

$$a_{22}x_2 + \dots + a_{2n}x_n = b_2 \quad (2)'$$

.....

$$a_{(n-1)(n-1)}x_{n-1} + a_{(n-1)n}x_n = b_{(n-1)} \quad (n-1)'$$

$$a_{nn}x_n = b_n \quad (n)'$$

Gauss 소거법 – Summary

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3$$



$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \end{Bmatrix}$$

Gauss 소거법 – 예제 9.3 (p.258~259)

$$\begin{aligned}
 3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 & \dots & (1) \\
 0.1x_1 + 7x_2 - 0.3x_3 &= -19.3 & \dots & (2) \\
 0.3x_1 - 0.2x_2 + 10x_3 &= 71.4 & \dots & (3)
 \end{aligned}$$

$$\begin{aligned}
 3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 & \dots & (4) \\
 7.00333x_2 - 0.293333x_3 &= -19.5617 & \dots & (5) \\
 -0.190000x_2 + 10.0200x_3 &= 70.6150 & \dots & (6)
 \end{aligned}$$

$$\begin{aligned}
 3x_1 - 0.1x_2 - 0.2x_3 &= 7.85 & \dots & (7) \\
 7.00333x_2 - 0.293333x_3 &= -19.5617 & \dots & (8) \\
 10.0120x_3 &= 70.0843 & \dots & (9)
 \end{aligned}$$

(9) => $x_3 = 7.0003$, (8) => $x_2 = -2.50000$, (7) => $x_1 = 3.0000$

순수 Gauss 소거법 – M File

```

function x = GaussNaive(A,b)
% x = GaussNaive(A,b):
% Gauss elimination without pivoting.
% input:
% A = coefficient matrix
% b = right hand side vector
% output:
% x = solution vector

[m,n] = size(A);
if m~=n, error('Matrix A must be square'); end
nb = n+1;
Aug = [A b];
    
```

주 1) 가우스 소거법의 연산회수 (p.256 표 9.1)
 주 2) 순수 가우스 소거법에서는 $a(i,i)$ 가 0이
 될 경우 문제 발생

```

% forward elimination

for k = 1:n-1
    for i = k+1:n
        factor = Aug(i,k)/Aug(k,k);
        Aug(i,k:nb) = Aug(i,k:nb)-factor*Aug(k,k:nb);
    end
    disp(Aug);
end

% back substitution

x = zeros(n,1);
x(n) = Aug(n,nb)/Aug(n,n);
for i = n-1:-1:1
    x(i) = (Aug(i,nb)-Aug(i,i+1:n)*x(i+1:n))/Aug(i,i);
end
    
```

Gauss 소거법 – 피벗팅 (Pivoting)

□ Pivoting의 필요성

- 가우스 소거법 연산과정에서 a_{ii} 로 나눠주는 연산 수행
- a_{ii} 가 0인 경우 'Divide by zero' Error 발생
- a_{ii} 가 0이 아닌 경우라도 0에 가까운 경우 반올림오차에 의한 오차 발생 (Ex. p.263 예제 9.4)

$$x_i = \frac{b_i - \sum_{j=i+1}^n a_{ij}x_j}{a_{ii}}$$

□ Pivoting 방법

Gauss 소거법 (Pivoting) – M File

```
function x = GaussPivot(A,b)
% x = GaussPivot(A,b):
% Gauss elimination with pivoting.
% input:
% A = coefficient matrix
% b = right hand side vector
% output:
% x = solution vector

[m,n]=size(A);
if m~=n, error('Matrix A must be square'); end
nb=n+1;
Aug=[A b];

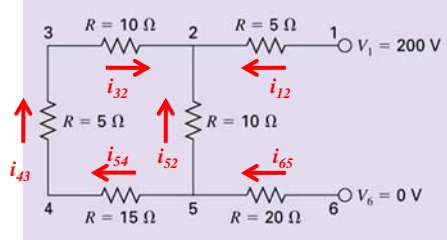
% forward elimination
for k = 1:n-1
    % partial pivoting
    [big,i]=max(abs(Aug(k:n,k)));
```

```
    ipr=i+k-1;
    if ipr~=k
        Aug([k,ipr,:])=Aug([ipr,k,:]);
    end
    for i = k+1:n
        factor=Aug(i,k)/Aug(k,k);
        Aug(i,k:nb)=Aug(i,k:nb)-factor*Aug(k,k:nb);
    end
    disp(Aug);
end

% back substitution
x=zeros(n,1);
x(n)=Aug(n,nb)/Aug(n,n);
for i = n-1:-1:1
    x(i)=(Aug(i,nb)-Aug(i,i+1:n)*x(i+1:n))/Aug(i,i);
end
```

Matrix Equation

$$\begin{bmatrix}
 1 & 1 & 1 & 0 & 0 & 0 \\
 0 & -1 & 0 & 1 & -1 & 0 \\
 0 & 0 & -1 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 1 & -1 \\
 0 & 10 & -10 & 0 & -15 & -5 \\
 5 & -10 & 0 & -20 & 0 & 0
 \end{bmatrix}
 \begin{Bmatrix}
 i_{12} \\
 i_{52} \\
 i_{32} \\
 i_{65} \\
 i_{54} \\
 i_{43}
 \end{Bmatrix}
 =
 \begin{Bmatrix}
 0 \\
 0 \\
 0 \\
 0 \\
 0 \\
 200
 \end{Bmatrix}$$



Matlab Command