9. Gauss Elimination

HoHee Kim

Linear equations (system) 을 풀기 위한 방법:

- 선형방정식의 수가 적은 경우 (n≤3) :

  Graphical method, Cramer's rule, Elimination of unknowns

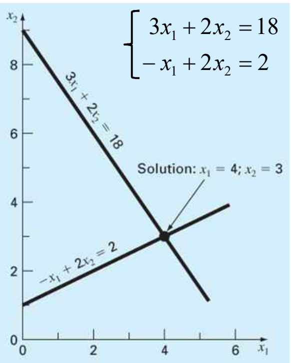
- 선형방정식의 수가 많은 경우: **Gauss elimination, LU factorization**

9장 10장

Graphical method

3개의 선형방정식에 변수가 3개 이면

3개의 평면이 교차하는 점이 해(solution)



Elimination of unknowns

$$\begin{cases} a_{11} x_1 + a_{12} x_2 = b_1 \longrightarrow a_{21}a_{11}x_1 + a_{21}a_{12}x_2 = a_{21}b_1 \\ a_{21} x_1 + a_{22} x_2 = b_2 \longrightarrow a_{11}a_{21}x_1 + a_{11}a_{22}x_2 = a_{11}b_2 \end{cases}$$

미지수 한 개를 소거한 후

$$x_2 = \frac{a_{11}b_2 - a_{21}b_1}{a_{11}a_{22} - a_{21}a_{12}} \qquad x_1 = \frac{a_{22}b_1 - a_{12}b_2}{a_{11}a_{22} - a_{21}a_{12}}$$

Cramer's Rule : 비실용적

$$\begin{cases} 0.3x_1 + 0.52x_2 + x_3 = \boxed{-0.01} \\ 0.5x_1 + x_2 + 1.9x_3 = 0.67 \\ 0.1x_1 + 0.3x_2 + 0.5x_3 = -0.44 \end{cases}$$

$$\begin{vmatrix} + & - & + \\ - & + & - \\ + & - & + \end{vmatrix}$$

$$D = \begin{vmatrix} 0.3 & 0.52 & 1 \\ 0.5 & 1 & 1.9 \\ 0.1 & 0.3 & 0.5 \end{vmatrix} = 0.3\begin{vmatrix} 1 & 1.9 \\ 0.3 & 0.5 \end{vmatrix} - 0.52\begin{vmatrix} 0.5 & 1.9 \\ 0.1 & 0.5 \end{vmatrix} + 1\begin{vmatrix} 0.5 & 1 \\ 0.1 & 0.3 \end{vmatrix} = -0.0022$$

☞ Cofactor Expansion : 하나의 행이나 열을 선택한 후 계산

$$D_2 = \begin{vmatrix} 0.3 & -0.01 & 1 \\ 0.5 & 0.67 & 1.9 \\ 0.1 & -0.44 & 0.5 \end{vmatrix} = 0.3\begin{vmatrix} 0.67 & 1.9 \\ -0.44 & 0.5 \end{vmatrix} + 0.01\begin{vmatrix} 0.5 & 1.9 \\ 0.1 & 0.5 \end{vmatrix} + 1\begin{vmatrix} 0.5 & 0.67 \\ 0.1 & -0.44 \end{vmatrix} = 0.0649$$

$$\rightarrow \quad x_2 = \frac{D_2}{D} = \frac{0.0649}{-0.0022} = -29.5, \qquad x_1 = \frac{D_1}{D}, \qquad x_3 = \frac{D_3}{D}$$

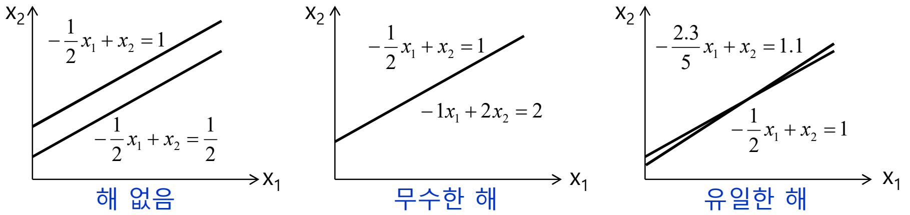

```
>> A =[0.3 0.52 1; 0.5 1 1.9; 0.1 0.3 0.5];
>> D = det(A)
 D =
     -0.0022

>> A(:,1)=[-0.01; 0.67; -0.44]
 A =
   -0.0100    0.5200    1.0000
    0.6700    1.0000    1.9000
   -0.4400    0.3000    0.5000

>> x1 = det(A)/D
 x1 =
     -14.9000
```

행렬식 구하는 built-in 함수

Cramer's rule로 x_1 구하기



해 없음 a) | 무수한 해 b) | 유일한 해 c)

a) 와 b) 는 [A]{x}={b} 에서 det(A)=0 ☞ singular

c) 는 유일한 해 일 때 det(A)≠0 이나, ☞ nonsingular

두 직선이 아주 근접한 경우, det(A) ≒ 0 ☞ ill-conditioned

예)

$$\begin{cases} x_1 + 2x_2 = 10 \\ 1.1x_1 + 2x_2 = 10.4 \end{cases}$$

1.05 로 바꾸면

$$\begin{cases} x_1 = \dfrac{2(10) - 2(10.4)}{1(2) - 2(1.1)} = 4 \\[2ex] x_2 = \dfrac{1(10.4) - 1.1(10)}{1(2) - 2(1.1)} = 3 \end{cases} \qquad \begin{cases} x_1 = \dfrac{2(10) - 2(10.4)}{1(2) - 2(1.05)} = 8 \\[2ex] x_2 = \dfrac{1(10.4) - 1.1(10)}{1(2) - 2(1.05)} = 1 \end{cases}$$

☞ 분모에서 비슷한 두 수의 뺄셈 때문에 이 시스템의 해는 반올림오차에 민감

Naive Gauss elimination : 2 단계로 구성, pivot 0 일 때 문제 발생

- **Forward elimination**

pivot

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 & ① \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 & ② \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

①X $\dfrac{a_{21}}{a_{11}}$:

$$a_{21}x_1 + \frac{a_{21}}{a_{11}}a_{12}x_2 + \frac{a_{21}}{a_{11}}a_{13}x_3 = \frac{a_{21}}{a_{11}}b_1 \quad ①'$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a'_{22}x_2 + a'_{23}x_3 = b'_2 & ②' \\ a'_{32}x_2 + a'_{33}x_3 = b'_3 \end{cases}$$

②-①' :

$$\underbrace{(a_{22} - \frac{a_{21}}{a_{11}}a_{12})}_{a'_{22}}x_2 + \underbrace{(a_{23} - \frac{a_{21}}{a_{11}}a_{13})}_{a'_{23}}x_3 = \underbrace{b_2 - \frac{a_{21}}{a_{11}}b_1}_{b'_2}$$

$$a'_{22}x_2 + a'_{23}x_3 = b'_2 \quad ②'$$

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a'_{22}x_2 + a'_{23}x_3 = b'_2 \\ a''_{33}x_3 = b''_3 \end{cases}$$

- **Back substitution**

$$x_3 = b''_3 / a''_{33} \longrightarrow x_2 = (b'_2 - a'_{23}x_3)/a'_{22} \longrightarrow x_1 = (b_1 - a_{13}x_3 - a_{12}x_2)/a_{11}$$

예)

$$\begin{cases} x_1 - 2x_2 + x_3 = 0 & ① \\ 2x_2 - 8x_3 = 8 & ② \\ 5x_1 - 5x_3 = 10 & ③ \end{cases}$$

$$\begin{array}{r} \\ -5\ \mathrm{X} \\ \\ \\ \end{array}\begin{array}{c} \begin{array}{cccc} -5 & 10 & -5 & 0 \end{array} \\ \left[\begin{array}{ccc:c} 1 & -2 & 1 & 0 \\ 0 & 2 & -8 & 8 \\ 5 & 0 & -5 & 10 \end{array}\right] \end{array}$$

③-(①X 5) :

$$\begin{cases} x_1 - 2x_2 + x_3 = 0 & \\ 2x_2 - 8x_3 = 8 & ② \\ \boxed{10x_2 - 10x_3 = 10} & ③' \end{cases}$$

$$-5\ \mathrm{X}\quad \begin{array}{c} \begin{array}{ccc} -10 & 40 & -40 \end{array} \\ \left[\begin{array}{ccc:c} 1 & -2 & 1 & 0 \\ 0 & 2 & -8 & 8 \\ 0 & 10 & -10 & 10 \end{array}\right] \end{array}$$

③'-(②X 5) :

$$\begin{cases} x_1 - 2x_2 + x_3 = 0 & \\ 2x_2 - 8x_3 = 8 & \\ \boxed{30x_3 = -30} & ③'' \end{cases}$$

$$\left[\begin{array}{ccc:c} 1 & -2 & 1 & 0 \\ 0 & 2 & -8 & 8 \\ 0 & 0 & 30 & -30 \end{array}\right]$$

$\longrightarrow\ x_3 = -1,\quad x_2 = 0,\quad x_1 = 1$

Ex) Use Gauss elimination to solve

$$\begin{cases} 3x_1 - 0.1x_2 - 0.2x_3 = 7.85 \\ 0.1x_1 + 7x_2 - 0.3x_3 = -19.3 \\ 0.3x_1 - 0.2x_2 + 10x_3 = 71.4 \end{cases}$$

$$\begin{cases} 3x_1 \quad - 0.1x_2 \quad - 0.2x_3 = 7.85 \\ 7.00333x_2 - 0.293333x_3 = -19.5617 \\ -0.190000x_2 + 10.0200x_3 = 70.6150 \end{cases}$$

$$\begin{cases} 3x_1 \quad - 0.1x_2 \quad - 0.2x_3 = 7.85 \\ 7.00333x_2 - 0.293333x_3 = -19.5617 \\ 10.0120x_3 = 70.0843 \end{cases}$$

$x_3 = 70.0843/10.0120 = \underline{7.00003}$

반올림오차가 누적 되어 나타남

$x_2 = (-19.5617 + 0.293333(7.00003))/7.00333 = \underline{-2.50000}$

$x_1 = (7.85 + 0.1(-2.50000) + 0.2(7.00003))/3 = \underline{3.00000}$

정확한 해

$x_3 = 7$

$x_2 = -2.5$

$x_1 = 3$

```
function x = GaussNaive( A, b )
    %   Gauss elimination without pivoting.
 [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
  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
 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
end
```


Operation counting of Gauss elimination (for an n x n system)

- **Forward elimination** 에서 연산 횟수 : floating-point operations

Outer loop *k*	*Inner loop* *i*	+,- Flops	*,/ Flops
1	2,n	[n-1][n]	[n-1][n+1]
2	3,n	[n-2][n-1]	[n-2][n]
…	…	…	…
k	k+1,n	[n-k][n+1-k]	[n-k][n+2-k]
…	…	…	…
n-1	n,n	[1][2]	[1][3]

- **Back substitution** 에서 연산 횟수 : $\frac{n(n-1)}{2}$, $\frac{n(n+1)}{2}$

+,- Flops

*,/ Flops

$$\underset{\text{Forward Elimination}}{\frac{2n^3}{3}+O(n^2)} + \underset{\text{Back Substitution}}{n^2+O(n)} \Rightarrow \underset{\text{Total}}{\frac{2n^3}{3}+O(n^2)}$$

☞ n 증가 → Flops 횟수 급격히 증가 → 반올림 오차 증가

Partial Pivoting : pivot=0 일 때 0 으로 나누는 문제 방지하기 위해

pivot 아래의 절대치가 가장 큰 계수 찾아 행 교환시키는 것

→ pivot ≒0 라도 계산상 수반되는 유효숫자에 민감하여 반올림오차 문제발생

Ex) Use Gauss elimination to solve

$$\begin{cases} 0.0003x_1 + 3.0000x_2 = 2.0001 & ① \\ 1.0000x_1 + 1.0000x_2 = 1.0000 & ② \end{cases} \Rightarrow \begin{cases} 0.0003x_1 + 3.0000x_2 = 2.0001 & ① \\ -9999x_2 = -6666 & ②' \end{cases}$$

①X $\frac{1}{0.0003}$: $x_1 + 10000x_2 = 6667$ ③

②-③ : $-9999x_2 = -6666$ ②' → $x_2 = 2/3$ 를 ① 에 대입 :

→ $x_1 = \dfrac{2.0001 - 3(2/3)}{0.0003}$

유효숫자 수	x_2	x_1
3	0.667	-3.33
4	0.6667	0.0000
5	0.66667	0.30000
6	0.666667	0.330000

☞ 뺄셈의 무효화로 유효숫자에 민감

Partial pivoting (행 교환)을 한 후

$$\begin{cases} 1.0000x_1 + 1.0000x_2 = 1.0000 & ① \\ 0.0003x_1 + 3.0000x_2 = 2.0001 & ② \end{cases} \Rightarrow \begin{cases} 1.0000x_1 + 1.0000x_2 = 1.0000 & ① \\ 2.9997x_2 = 1.9998 & ②' \end{cases}$$

①X $\frac{0.0003}{1}$: $0.0003x_1 + 0.0003x_2 = 0.0003$ ③

②-③ : $2.9997x_2 = 1.9998$ ②' → $x_2 = 2/3$ 를 ① 에 대입 :

→ $x_1 = \dfrac{1-(2/3)}{1}$

유효숫자 수	x_2	x_1
3	0.667	0.333
4	0.6667	0.3333
5	0.66667	0.33333
6	0.666667	0.333333

☞ 유효숫자에 덜 민감

참 해 : $x_1 = \frac{1}{3},\ x_2 = \frac{2}{3}$

```
function x = GaussPivot(A,b)
%  Gauss elimination with pivoting.
   …
% 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
end
 % back substitution
   …
end
```


Determinant Evaluation with Gauss Elimination

- triangular 행렬의 determinant 는 대각선 entry 를 모두 곱한 것과 같다

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{vmatrix} = a_{11} \cdot \begin{vmatrix} a_{22} & a_{23} \\ 0 & a_{33} \end{vmatrix} - 0 \cdot \begin{vmatrix} a_{12} & a_{13} \\ 0 & a_{33} \end{vmatrix} + 0 \cdot \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix}$$

$$= a_{11}a_{22}a_{33}$$

- Forward Elimination 을 마친 후

  upper triangular 형태가 되므로

  det(A) = $a_{11}a'_{22}a''_{33}\ldots a_{nn}^{(n-1)}$

- Partial pivoting 으로 P번의 행 교환 있는 경우

  det(A) = $a_{11}a'_{22}a''_{33}\ldots a_{nn}^{(n-1)}(-1)^P$

A

$$\left[\begin{array}{ccc:c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right]$$

$$\Downarrow$$

$$\left[\begin{array}{ccc:c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a'_{22} & a'_{23} & b'_2 \\ 0 & 0 & a''_{33} & b''_3 \end{array}\right]$$