수치해석 - 중간

□ 수치해석 이론

○ 모델화

* 모델화 : 대상물 -> 시스템(구조, 성질) //ex) 펜이 떨어지면 작용하는 큰틀의 성질(중력 등)
  - 방정식을 세우는 등의 작업

* 입력(환경) -> 시스템 -> 출력(대상물)
  - 독립변수 : 자기 자신 이외에 대처할 수 없는 것.
  - 종속변수 : 어떻게 변하는지 보는 것
  - ex) y(종속변수) = ax(독립변수) // x의 값에 따라 y의 값이 바뀜

○ 수치해석

* 수치해석 : 예측을 함 -> 제어함
* 공학적 문제 해석
  - 물리적인 시스템의 모든 중요한 특성을 표현하는 수학적 모델 개발
  - 공학 : 기존에 없는 새로운 것을 만들어 냄 (인공물)
    // 과학 : 기존에 존재하는 것을 쪼개서 파악
  - 지배방정식 : 현상을 가장 근접하게 잘 표현함(지배함)
    - 물리적인 법칙들(평형방적식, 뉴턴의 운동방정식, 질량보존과 에너지 보존의 법칙)을 적용
    - 연립선형, 비선형 대수 방정식, 초월함수 방정식, 상미분, 편미분방정식, 적분, 미분등
    // 부력 : 인간이 가장 먼저 발견한 힘 (아르키메데스)
    // F = ma는 프랑스의 (나이프니치)가 만들어 낸 공식
    // 모멘트 : 각각의 작용하는 힘에 대해서 무게중심이 다르기 때문에 회전력이 생김 (오일러)
    // 구심성 신경계 : 몸안에서 신경을 통하여 신호를 내보내는 것
    // 원심성 신경계 : 외부에서 충격을 받아 몸 안으로 신호를 보내는 것
  - 해의 해석

* 회전과 이동
  - 회전 (오일러) : 위치는 이동하지 않았지만 자세가 바뀌는 것
    // 오일러, 토르크 방정식
    // 모터, 기어
  - 이동 (뉴턴) : 위치가 바뀌는 것 // 변이적 기능 / 병진 운동
  - 프리덤(degree of freedom / DOF) : 3차원(x, y, z)에서의 회전과 이동을 다룬 것
    // 6자유도 : 아무것도 하지 않았을 때 할 수 있는 행동 6가지(이동3, 회전3)

* 강체
  - 강체(고체) : 고체에는 분자원자가 고정되어 있다. // 수치해석은 강체를 다룬다.
  - 유체(액체) : 압축할 수 없다.
  - 유체(기체) : 질량을 계산할 수가 없다. / 압력으로 압축이 가능하다.
    // 응력 : 견뎌내는 것

* 시뮬레이션 : 얼마나 정확하게 표현할 것인지 나타내는 것

* 벡터 : 어떤 관점에서 볼 것인지의 기준이 있어야함
  - 기준 (좌표)이 있어야 방향을 정할 수 있다.
  - 변환(선형변환) : 기준 틀이 틀릴 경우 한 좌표계로 맞추어주는 것(어느 기준에 맞출지를 정하는 것)
    // ex) 10cm 자와 10mm의 자를 잴 때 10cm를 100mm로 바꾼다.
  - 오일러의 회전 좌표계 : 회전의 기준을 정해놓은 좌표계

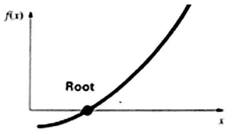
○ 수학적 기초지식

* 수치해석 : 컴퓨터에 방정식을 그대로 쓰면 이해할 수가 없음 => 컴퓨터가 알 수 있게 식을 바꾸어 줌

* 점화식 : 방정식

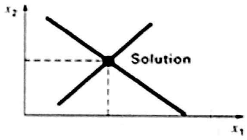
(a) 방정식의 근 : single nonlinear eqn을 만족하는 변수 or 매개변수의 값을 구하는 것

- 방정식이 하나 있을 경우 사용 $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$



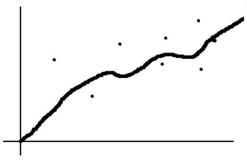
(b) 연립 선형 대수 방정식

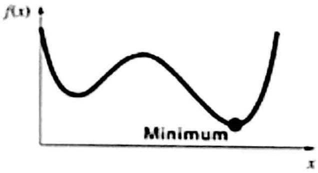
- 여러 방정식들을 동시에 만족하는 변수 값들을 구함 -> (a)와 유사 하지만 방정식이 여러개
- 구조해석, 전기회로, 유체 네트워크 -> 대규모 시스템의 수학적 모델링



(c) 최적화(Optimization) : given fn의 독립변수 값들의 최선 or 최적값 결정

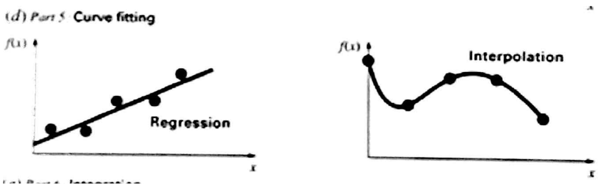
- 최적화 = 최대값, 최소값을 찾는 것 // 오차를 줄이는 과정
- 식은 없고 결과만 있는 것
- 오차를 줄이는 것 (부호의 오차를 줄이기 위해 제곱을 함) =>
- 기울기가 0이다. => 2차방정식에서 E = (y - t $)^2$을 할 경우 E가 0일 때 오차가 제일 낮다.
- ex) $y=ax^3+bx^2+cx+d$ 의 3차방정식이 있다고 쳤을 경우 가장 만족하는 a, b, c, d를 찾아냄





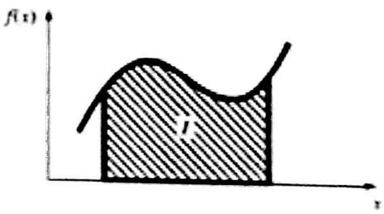
(d) 곡선접합(Curve Fitting)

- 실험 data를 해석하는 경우
- Data의 일반적 경향 -> 하나의 곡선 유도
- (c)최적화와 같은 개념



(e) 적분 (결과 값)

- 곡선 아래 부분의 면적
- 기묘한 형상의 물체 중심, 각각의 측정 값 + 전체 값 -> 응용
- 미분방정식의 해
- 결과 값의 양의 크기가 얼마인지를 나타냄 = 변화량
- 독립변수가 변화해서 나온 결과의 값

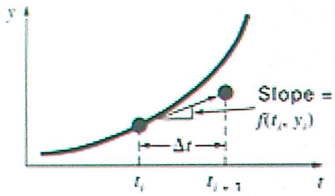


(e) 미분 (예측)

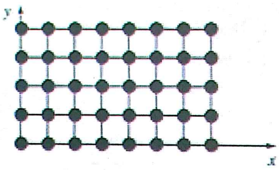
- 미분 : 독립변수에 변화에 따라 종속변수가 변하는 정도 (예측)
  - // 변화율이 0 : 변화가 없다
  - // 얼마나 기울기가 기울여져있는지에 따라 변화량이 달라진다.
- 운동 방정식을 2차 미분 방정식이라 부름

(f) 상미분방정식(ODE) : f(t, y), u=u(x)

- 많은 물리법칙 -> 어떤 양의 크기(X), 양의 변화율(O)
- 인구예측모델, 낙하하는 물체의 가속도
- 초기조건 문제, 경계조건 문제
- 종속변수가 하나
- ex) 시간에 따른 가속도의 변화



(f) 편미분방정식(PDE) : u=u(x, y, t)



- 2개 이상의 독립변수 (종속변수가 하나가 아님)
- 가열된 판의 온도 분포
  - 유한 차분법
  - 유한 요소법
- 속도의 차에 의해 소용돌이가 생김 (한사람이 문에 잡고 있을 경우 다른 사람의 속도에 제한이 생김)
- ex) 온도 조절이 가능한 시계를 들고 있을 경우 눈 산에서 내려갈 때

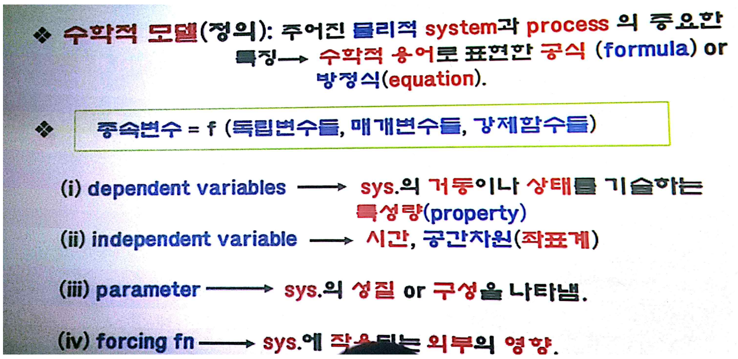
  => 시간에 따른 온도의 변화와 위치의 변화

□ 비판적 사고력과 논리적 사고력

* 비판적 사고
* 논리적 사고

○ 수학적 모델링과 공학문제

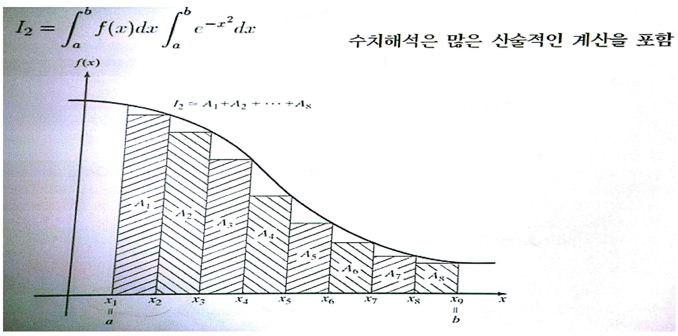
* 종속변수(출력 값) = f(독립변수들, 매개변수들, 강제함수들)
  - 결과 값
  - sys.의 거동이나 상태를 기술하는 특성량
* 독립변수 : 입력 값
  - 시간, 공간차원(좌표계)
* 매개변수 : sys.의 설질 or 구성
* 강제함수 : sys.에 작용되는 외부의 영향



* 해석해 : 손으로 푼 방적식
* 수치해 : 수학적 문제를 재 공식화 하는 것
* 수치해석 : 수학적 문제를 반복적으로 계산
  - 오차를 줄이기 위해 잘게 쪼갠다. => 계산량이 많아진다.
  -> ex) 하트의 면적 구하기 // 몬테카를로 적분

  => 동적을 면적만큼 놓고 잰다 / 오차발생

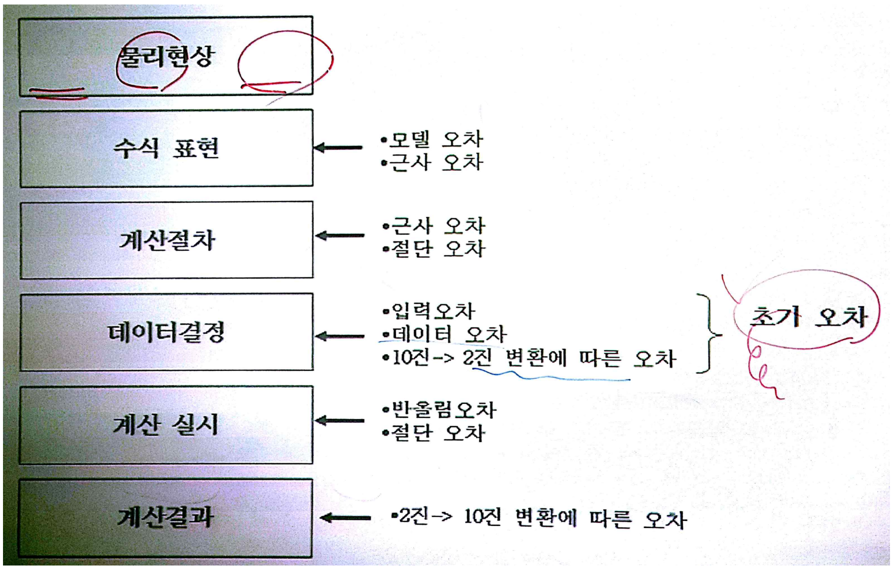
  => 오차를 줄이기 위해 더 작은 동전을 사용한다.
* 수치해석 반복 계산의 종료조건 : 오차 범위 안에 들어왔을 경우 종료



○ 오차

* 허용오차 : 얼마만큼 오차를 허용할 것인지 결정
* 참 값을 모르기 때문에 현재 값을 참 값으로 정의함
  -> 이전 값(근사 값)과 비교를 함 => 오차
* 유효숫자 : 0의 숫자를 포함하지 않는 숫자 // 유효숫자 사이의 0은 포함
  -ex) 0.00001845 일 때, 모두 4개의 유효숫자를 가짐
  -ex) 0. 00180045 일 때, 모두 6개의 유효숫자를 가짐

* 오차의 원인
  * 수학적 모델에서의 오차
  * 실수
  * 입력의 오차
  * 기계 오차
  * 수학적인 처리와 관련된 절단 오차
* 수치 계산에 있어서의 오차



* 전파오차 : 입력데이터의 오차 때문에 과정의 결과에 나타나는 오차
  - 입력한 오차가 다음 결과에도 영향을 미침
  - 반복을 하면 할수록 오차가 커짐

* 테일러 급수 // 극한으로 미분의 미분으로 나열함
  - 어떤 점의 함수 값 -> 함수값 & 그의 도함수 값으로 다른 점의 값을 예측
  - x의 좌표에서 미분한 값에 미분한 값...을 다 더하면 △x 값과 같다
  - f(x+△x) = f(x)+(△x)*∂f/∂x .. 뒤에 것을 다 버리면
    => (f(x+△x) - f(x)) / (△x) = ∂f/∂x(미분) 가 된다.
    * 각각의 미분한 값에 미분 계수 값을 곱하여 다 더해줌
    * 현재 값에서 기울기만 봄

비선형방정식의 해

○ 선형과 비선형방정식의 해

* **선형** : 식과 값이 일대일 대응
* **비선형** : 식과 값이 일대일 대응이 아님 => 값이 두 개 이상일 수 있음
  - 근의 수는 유한, 무한 또는 실수, 복소수
  - 명시적 형태와 그렇지 않은 경우도 있음
  - 명시적 형태의 경우 : 다항식, 초월함수식

* 초월함수 : 정의할 수 없는 함수 ex) 삼각함수(sin, cos, tan), 로그함수, 쌍곡선함수 등
* Newton의 관계
  - 다항방정식의 근과 계수와의 관계
  - 모든 근이 실근일 경우, 근의 최대 절대 값의 상한을 구할 수 있음

  ex) f(x) 가 근이 있다고 할 때, 근이 모두 실근인 경우 최대 크기만 알 수 있고 더 이상 알 수 없다.

* 방정식의 근
  - 근을 쉽게 구할 수 없는 함수들도 많다. -> 수치해법들을 사용, 효과적으로 근을 구함
* 컴퓨터를 사용하지 않을 경우

  (i) 근사해 기법
    - 함수를 그려 x축과 만나는 점을 찾는 것 (f(x) = 0 점)
    - 개략적인 근 -> 정밀성 결여(한계점)

  (ii) 시행 착오법
    - x의 값 추측 -> f(x) = 0인지 여부를 계산하는 것
    - f(x) != 0 -> 새로운 초기 값을 가정
    - f(x) = 0에 가까워질 때까지 반복

○ 구간법 : 영역을 정하고 해를 구하는 방법

* 이분법 : 좌측과 우측의 값을 곱하여 0보다 작으면 두 값을 더해 미분을 한다.
  - y = f(x)라는 함수가 있을 경우, 양 변이 0일 경우로 바꾼다.

  => y = 0, f(x) = 0 => y가 0일 경우는 x축 상에 있는 경우.

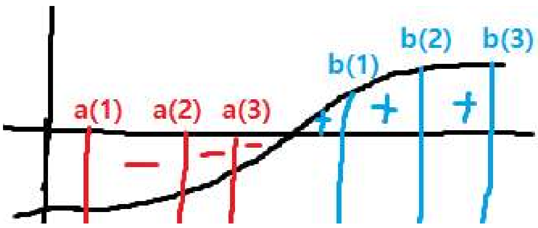
  => 그래프에서 x축에 걸리는 값이 해이다.

  => 두 점 f(x1)*f(x2)를 할 경우, 둘의 부호가 다르면(-) 그 사이에 해가 존재한다.

  => f(x1)*f(x2) = 0 이면 해가 됨

  => f(x1)*f(x2) < 0 이면 될 경우, 왼쪽을 cn으로 택하고 f(x1)+f(x2) / 2를 해준다. => 반복

  => f(x1)*f(x2) > 0 이면 될 경우, 오른쪽을 cn으로 택하고 f(x1)+f(x2) / 2를 해준다. => 반복



정리 : f(x)가 [a, b]에서 연속이고 f(a)f(b)<0이면 이분법 알고리즘에 의한 f(x)=0의 수치해는 정해 α로 수렴하고 허용오차ε의 수치해를 얻기위한 반복시행 횟수 n은

$$n \geq \frac{\ln(b-a) - \ln\varepsilon}{\ln 2}$$

증명) $b_n - a_n = \frac{1}{2}(b_{n-1} - a_{n-1}) = \frac{1}{2^{n-1}}(b-a), \quad n = 1, 2, 3, \cdots$

정해 α는 구간 $[a_n, c_n]$ 또는 $[c_n, b_n]$에 속하므로

$$|\alpha - c_n| \leq c_n - a_n = b_n - c_n = \frac{1}{2}(b_n - a_n) = \frac{1}{2^n}(b-a)$$

또한 $|\alpha - c_n| \leq \frac{1}{2^n}(b-a) \leq \varepsilon$ 으로부터 자연로그를 취하면

$$\therefore n \geq \frac{\ln(b-a) - \ln\varepsilon}{\ln 2}$$

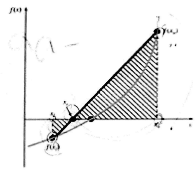
* 가위치법 : 기울기를 사용하여 해를 구하는 방법

- 두 점의 기울기를 사용하여 x축과 만나는 점을 해로 둠

가위치법 (false position method)

❖ 닮은꼴 삼각형으로 부터

$$\frac{f(x_l)}{x_r - x_l} = \frac{f(x_u)}{x_r - x_u} \qquad (5.6)$$



❖ 정리하면

$$\boxed{x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}} \qquad (5.7)$$

가위치 공식

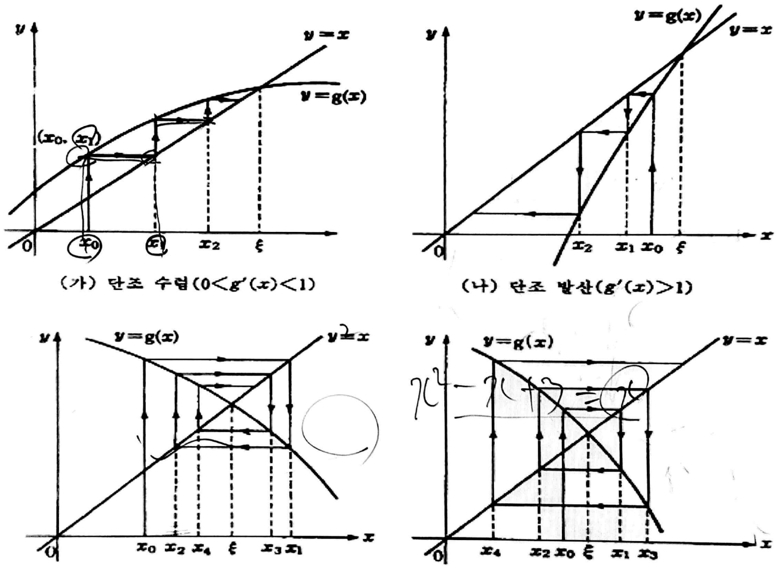
❖ x_r 구하는 방법을 제외하면 **가위치법=이분법**

❖ 이분법의 종료판정 기준을 사용 ⟶ **계산종료**

○ 개구간법 : 영역을 정하지 않고 해를 구하는 방법

- 고정점 반복법 / 뉴턴-랩슨법 / 할선법

* 고정점 반복법



(가) 단조 수렴($0 < g'(x) < 1$)

(나) 단조 발산($g'(x) > 1$)

(다) 진동 수렴($-1 < g'(x) < 0$)

(라) 진동 발산($g'(x) < -1$)

$f(x) = x^2 - 2$

(1) $g(x) = 2 + x - x^2$ (2) $g(x) = \frac{2}{x}$ (3) $g(x) = 1 + x - \frac{1}{2}x^2$ (4) $g(x) = \frac{1}{2}\left(x + \frac{2}{x}\right)$

n	(1)	(2)	(3)	(4)
0	1.0	1.0	1	1
1	2.0	2.0	1.5	1.5
2	0.0	1.0	1.3750	1.4167
3	2.0	2.0	1.4297	1.4142

* |g'(x)| <1을 만족하여야 수렴한다.

* Newton-Raphson법

- 미분을 활용하여 구하는 방법
- * y=e의f(x)승 => y' = e의f(x)승 * f'(x) // '는 미분

* 할선법

- $f'(x_i)$ = f(x+△x)-f(x) / x + △x - x = f(x+△x)-f(x) / △x
  // x : 임의의 점1 , △x : 임의의 점2
- 도함수를 사용함 : 도함수를 알고 있다 : 곡선 위의 임의의 점에서의 '접선의 기울기를 알고 있다'
- 초기에만 값을 두 개를 주고 다음 값을 구함
  => 두 번째 부터는 처음 값을 버리고 그 뒤 두 값으로 반복

- 가위치법 = 두 값 사이에 항상 근이 존재 -> 항상수렴
- 할선법 = 연속대입(x_{i+1} -> x_i, x_i-> x_{i-1}을 대치함), 두 값은 때때로 근의 같은 편에 놓임 or 발산

연립선형대수방정식의 해

□ 연립 선형대수방정식의 해

* 선형화 : 모든 식을 1차식으로 바꾼다.

○ 선형대수 방정식

* 1차선형 결합 : n차원의 공간 안의 직선
* 연립일차방정식과 행렬
  - 계수 / 상수 / 미지수 로 나누어 행렬로 표현
  - [A]x = b   => A역행렬 * [A]x = A역행렬*b   => x = A역행렬 * b // 역행렬을 구하면 해를 안다.

□ 행렬

* View를 만들 때 렌즈로 본 매트릭스와 기존의 매트릭스 2개를 만든다.
* 행렬
  - 특이 행렬 : 해가 없거나 해가 무수히 많을 때 역행렬을 구할 수 없다.
  - 가역 행렬 : 역행렬을 구할 수 있는 행렬 (해가 하나인 행렬)
  - 전치 행렬 : 정방행렬

* 연립방정식 수치해법
  * 직접 소거법 소거
    - Gauss 소거법 : 왼쪽 위에서 밑으로 소거한 것
    - Gauss-Jordan법 : 단점) 계산량이 많음 / 조르단 : 오른쪽 밑에서부터 위로 소거한 것
    - LU 분해법 : 가우스 조르단을 적절히 섞은 것 [시험나옴!!]
      - 반복을 통해서 가우스-조르단 보다 효율적으로 풀 수 있다.

  * 직접 소거법 반복
    - Jacobi법
    - Gauss-Seidel법
    - 이완(SouthWell)법(사루스 공식)

□ 직접(반복) 소거법

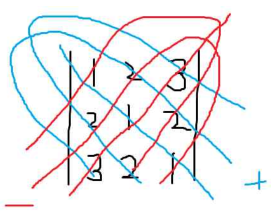
○ 가우스-세이델

* 첫 값을 x1남겨두고 다 넘김 / 둘째 값을 x2 남겨두고 다 넘김 / 셋째값 x3 남기고 다 넘김
  -> 반복하고 , 오차범위 안에 들어가면 반복을 끝냄

○ 사루스의 공식

* 3X3 이하의 행렬에서 대각선으로 (왼쪽위->오른쪽아래)는 플러스 (오른쪽위 ->왼쪽밑)은 마이너스

$$|A| = \begin{vmatrix} 2 & 1 \\ 1 & 4 \end{vmatrix} = 2\times4 - 1\times1 = 8-1 = 7$$



-> 각각을 곱해서 나온 숫자를 다 더해줌.

□ 직접(소거) 소거법

○ 가우스 소거법

* 미지수가 두 개인 방정식 풀이
  (1식) : a11x1 + a12x2 = b1 // (2식) - a21/a11 (1식) => 대입 => 대입하여 대각선행렬만 남김
  (2식) : a21x1 + a22x2 = b2 //
* 축적화 : 불량조건이 있는 경우 바꿔주는 것
* 핏벗화 : 방정식의 위치를 교환하는 것 / 피벗요소가 0에 가까울 경우 해를 구할 수 없다.
  => 순서만 바꿔주면 된다. / 위<->아래 방정식을 바꾼다.

○ 가우스-조르단 소거법 : 정 [3 2 4 |5] = $3x_1 + 2x_2 + 4x_3 = 5$

[2 4 8 |9] = $2x_1 + 4x_2 + 8x_3 = 9$

[q w e |t]

이걸 1 0 0 로 만드는 것

0 1 0

0 0 1

* **역행렬** : 행렬(A) * 역행렬(A-1) = 항등행렬(I) 이 되는 것.
  2a + 5c =x1, 2b + 5d =x2, 1a + 3c = y1, 1b + 3d = y2 가 되는 것
* 역행렬 구하는 방법1)

1-1 소행렬식 : r번째 행과 s번 째 열을 제거해서 얻은 행렬 $\begin{pmatrix} 1 & 2 & 1 \\ 3 & 1 & 2 \\ 2 & 2 & 0 \end{pmatrix} \rightarrow \begin{pmatrix} -4 & -4 & 4 \\ -2 & -2 & -2 \\ 3 & -1 & -5 \end{pmatrix}$	$A = \begin{bmatrix} 1 & 2 & -1 \\ -1 & 3 & 5 \\ 2 & 4 & 3 \end{bmatrix}$ $M_{11} = \begin{bmatrix} 3 & 5 \\ 4 & 3 \end{bmatrix}$ $A_{11} = (-1)^{i+j} \times \begin{bmatrix} 3 & 5 \\ 4 & 3 \end{bmatrix}$ 여인수 $= (+1) \times (3 \times 3 - 5 \times 4)$ $= -11$
1-2 여인자 : 부호를 설정해준다. $\begin{pmatrix} + & - & + \\ - & + & - \\ + & - & + \end{pmatrix} \begin{pmatrix} -4 & -4 & 4 \\ -2 & -2 & -2 \\ 3 & -1 & -5 \end{pmatrix} \rightarrow \begin{pmatrix} 4 & 4 & -4 \\ 2 & 2 & 2 \\ -3 & 1 & 5 \end{pmatrix}$	- 여인수 : I+j가 짝수이면 +, 음수이면 - - det(A) = a11(각각의 겹쳐지는 원소)*A11(남은행렬) // 어떤 행을 택해도 값은 똑같이 나옴
1-3 전치행렬 : 대각선을 기준으로 나머지를 전치해(바꿔)준다 $\begin{pmatrix} 4 & 4 & -4 \\ 2 & 2 & 2 \\ -3 & 1 & 5 \end{pmatrix} \rightarrow \begin{pmatrix} 4 & 2 & -3 \\ 4 & 2 & 1 \\ -4 & 2 & 5 \end{pmatrix}$	$\begin{vmatrix} 3 & 1 & 4 \\ 2 & 0 & 3 \\ 1 & -1 & 2 \end{vmatrix}$ $\det(A) = a_{11}A_{11} + a_{12}A_{12} + a_{13}A_{13}$ $= 3\begin{vmatrix} 0 & 3 \\ -1 & 2 \end{vmatrix} - 1\begin{vmatrix} 2 & 3 \\ 1 & 2 \end{vmatrix} + 4\begin{vmatrix} 2 & 0 \\ 1 & -1 \end{vmatrix}$

* 가우스 조단 역행렬 구하는 알고리즘 : 절차가 존재함 정 [A | I] (역행렬구하는 방법2)

$\begin{bmatrix} 3 & 4 & \vert & 1 & 0 \\ 2 & 3 & \vert & 0 & 1 \end{bmatrix}$ $(\frac{1}{3}) * R1 \rightarrow R1$	1. 첫 번째 줄에 1/3을 전부 곱해줌 -> 첫 번째 줄에 값 입력 -> 1*1이 1이 됨 2. 첫 번째 줄에 -2를 전부 곱하고 R2에 전부 더해줌 -> 두 번째 줄에 값 입력 -> 2*1이 0이 됨 3. 두 번째 줄에 3을 전부 곱해줌 -> 두 번째 줄에 값 입력 -> 2*2가 1이 됨 4. 두 번째 줄에 (-4/3)을 곱하고 첫 번째 줄을 더해줌 -> 첫 번째 줄에 값 입력 -> 1*2가 0이 됨
$\begin{bmatrix} 1 & \frac{4}{3} & \vert & \frac{1}{3} & 0 \\ 2 & 3 & \vert & 0 & 1 \end{bmatrix}$ $(-2) * R1 + R2 \rightarrow R2$	
$\begin{bmatrix} 1 & \frac{4}{3} & \vert & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \vert & -\frac{2}{3} & 1 \end{bmatrix}$ $3 * R2 \rightarrow R2$	
$\begin{bmatrix} 1 & \frac{4}{3} & \vert & \frac{1}{3} & 0 \\ 0 & 1 & \vert & -2 & 3 \end{bmatrix}$ $(-\frac{4}{3}) * R2 + R1 \rightarrow R1$	
$\begin{bmatrix} 1 & 0 & \vert & 3 & -4 \\ 0 & 1 & \vert & -2 & 3 \end{bmatrix}$	

○ LU분해법 [시험 나옴]

[Step 1]
제1행은 변하지 않고 제2-4행은 다음과 같이 나타낼 수 있다.

$$A=\begin{bmatrix}4&12&8&4\\1&7&18&9\\2&9&20&20\\3&11&15&14\end{bmatrix},\quad L=\begin{bmatrix}1&0&0&0\\1/4&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix},\quad U=\begin{bmatrix}4&12&8&4\\0&7-3&18-2&9-1\\2&9&20&20\\3&11&15&14\end{bmatrix}$$

$$A=\begin{bmatrix}4&12&8&4\\1&7&18&9\\2&9&20&20\\3&11&15&14\end{bmatrix},\quad L=\begin{bmatrix}1&0&0&0\\1/4&1&0&0\\1/2&0&1&0\\0&0&0&1\end{bmatrix},\quad U=\begin{bmatrix}4&12&8&4\\0&4&16&8\\0&3&16&18\\3&11&15&14\end{bmatrix}$$

$$A=\begin{bmatrix}4&12&8&4\\1&7&18&9\\2&9&20&20\\3&11&15&14\end{bmatrix},\quad L=\begin{bmatrix}1&0&0&0\\1/4&1&0&0\\1/2&0&1&0\\3/4&0&0&1\end{bmatrix},\quad U=\begin{bmatrix}4&12&8&4\\0&4&16&8\\0&3&16&18\\0&2&9&11\end{bmatrix}$$

[Step 2]
제1행과 2행은 변하지 않고 제3-4행은 다음과 같이 변화한다.

$$A=\begin{bmatrix}4&12&8&4\\1&7&18&9\\2&9&20&20\\3&11&15&14\end{bmatrix},\quad L=\begin{bmatrix}1&0&0&0\\1/4&1&0&0\\1/2&3/4&1&0\\3/4&1/2&0&1\end{bmatrix},\quad U=\begin{bmatrix}4&12&8&4\\0&4&16&8\\0&0&4&12\\0&0&1&7\end{bmatrix}$$

[Step 3]
제4행은 다음과 같이 변화한다.

$$A=\begin{bmatrix}4&12&8&4\\1&7&18&9\\2&9&20&20\\3&11&15&14\end{bmatrix},\quad L=\begin{bmatrix}1&0&0&0\\1/4&1&0&0\\1/2&3/4&1&0\\3/4&1/2&1/4&1\end{bmatrix},\quad U=\begin{bmatrix}4&12&8&4\\0&4&16&8\\0&0&4&12\\0&0&0&4\end{bmatrix}$$

* lower, uppor의 개념
* 가우스 소거법 : 피뻣(대각선)요소들을 우해서 소거를 함
* 가우스 소거법을 하면 대각을 제외한 값들이 0이되는 삼각행렬이 만들어짐
* LU분해법의 가장 큰 특징 : 상이든 하든 삼각형을 만들어야함
* 상삼각형으로 만들 것인지, 하삼각형으로 만들 것인지 임의로 정해줌 => L, U로 나눠짐
* 삼각형이 나오면 대입만 하면 됨 => 반복적으로 해서 풀어냄
* A의 역행렬을 바로 구할 수 없기 때문에 A를 곱하여 I[대각선이 1인 행렬]을 만들어내서 구함

* [A]{X} = {B} // B는 상수 값들이 있는 곳
* {A}{X}-{B} = 0
* [A] = [L][U] // A는 L과 U의 곱으로 나타낼 수 있다 => 미지수로 둔다.
* [U]{X}-{D} = 0 // D라는 행렬이 있다고 가정을 하는 것
* 양 옆에 [L]을 곱해졌을 경우
* [L][U]{X} - [L]{D} = 0
* [A]{X} -[L]{D} = 0 // LD는 [A]{X} = {B}를 통해
* [L]{D} = {B}가 된다.
* [L]과 {B}를 알고 있기 때문에 {D}를 구할 수 있다.
* [U]와 {D}를 알고 있기 때문에 {X}를 구할 수 있다.

* 반드시 순서를 지켜줘야함 => 열->행->열->행

* [L]{D} = {B}
* l11 0 0 | d1 b1
* l21 l22 0 | d2 = b2
* l31 l32 l33 | d3 b3

* 예제

```
1  1   0  3  |  4        // 앞의 계수들이 A
0 -1  -1 -5  |  -7
0  0   3  13 |  13
0  0   0 -13 | -13
```

=>

```
x1 + x2        +3x4 = 4
     -x2  -x3  -5x4 = -7
          3x3 + 13x4 = 13
                13x4 = 13      * 과제 : LU분해법 코드로 풀어오기
```