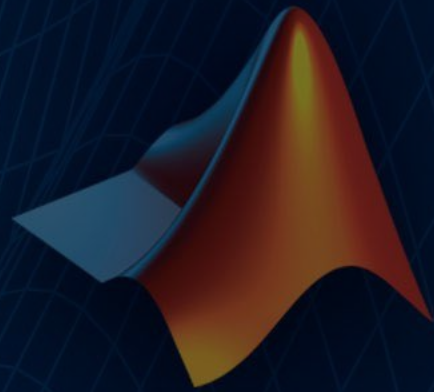


2021-2 MATLAB 5주차



함수, 방정식

Curriculum

1. 소개 및 기본 사용법
2. 벡터, 행렬
3. 그래프
4. 제어문(조건문, 반복문)
5. 함수, 방정식
6. 미분방정식
7. 제어공학 기초

문제(4주차 과제 3번)

반복횟수에 따라서 $f(x) = x^3 + 2x^2 + 3x + 4 = 0$ 의 실수 해를 구하기 위해서 False-position method를 사용한다. 이 때 각 과정마다 나오는 해(x_r)를 plot 하시오.

False-position method

1. $x_l = -5, x_u = 5$ 를 선정한다.
2.
$$x_r = x_u - \frac{f(x_u)}{f(x_l) - f(x_u)} (x_l - x_u)$$
3. 만약 $f(x_l)f(x_r) < 0$ 면 $x_u = x_r$, 그렇지 않으면 $x_l = x_r$ 로 바꿔준다.
4. $\varepsilon_a = \left| \frac{x_r^{new} - x_r^{old}}{x_r^{new}} \right| < 0.005$ 일 때까지 계속 반복하며 x_r 을 plot 한다.
5. plot 할 때 x축은 반복횟수, y축은 x_r 로 한다.

function



```
%함수명.m  
function 반환값 = 함수명(매개변수)  
    <수행할 문장1>  
    <수행할 문장2>  
end
```

```
%f.m  
function y = f(x)  
    y=2*x;  
end
```

- ✓ 함수의 사용 이유 : 반복적인 선언 감소
- ✓ 선언 방법 : 새로운 스크립트 파일 필요
- ✓ MATLAB의 함수 구조 : function으로 정의

함수명 : 함수 호출 시 사용(사용자 지정)

매개변수 : 함수 호출시에 전달되는 값

반환값 : 함수명 앞에 입력

형태

function result = FunctionName(Param1, Param2, ...)

%반환값 있음, 매개변수 있음

function result = FunctionName()

%반환값 있음, 매개변수 없음

function FunctionName(Param1, Param2, ...)

%반환값 없음, 매개변수 있음

function FunctionName()

%반환값 없음, 매개변수 없음

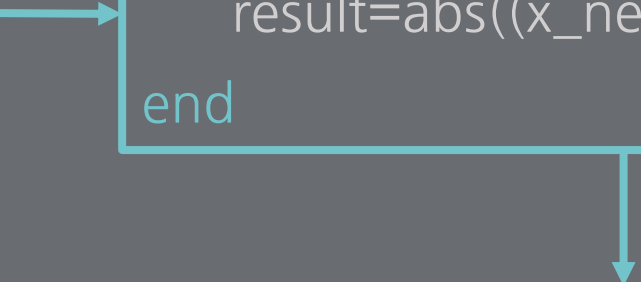
function [result1, result2, ...] = FunctionName(...)

%반환값 있음(여러개)

예시

상대오차 $\varepsilon_a = \left| \frac{x_r^{new} - x_r^{old}}{x_r^{new}} \right|$ 를 함수로 표현하시오.

1. 반환값? Ans: ε_a
2. 매개변수? Ans: x_r^{new}, x_r^{old}



```
function result=e_a(x_new, x_old)
    result=abs((x_new-x_old)/x_new);
end
```

```
clear all; clc;

...
result=e_a(x_new, x_old)

...
```

local function

```
function y=f(a,b,c) %global function
    z=f2(a);
    w=f3(b,c);
    y=z+w;
end

function z=f2(a) %local function
    z=sqrt(a);
end

function w=f3(b,c) %local function
    w=b^2+c^2;
end
```

✓ 함수 내의 함수

메인 함수 : global function

부함수 : local function

- ✓ global function과 같은 m 파일에 작성
- ✓ 함수가 복잡해질 때 쪼개서 해결
- ✓ local function은 외부에서 사용 불가능

anonymous function

✓ 기존 함수는 따로 스크립트를 만드는 번거로움이 존재 → 직접 실행 스크립트에 적어보는 것은 어떨까 → @을 사용!

name = @(arglist) expression

$f(x) = x^3 + 2x^2 + 3x + 4$

f=@(x) x^3+2*x^2+3*x+4

```
>> f=@(x) exp(x)-1
```

```
f =
```

```
다음 값을 갖는 function_handle:
```

```
@(x)exp(x)-1
```

```
>> f(0)
```

```
ans =
```

```
0
```

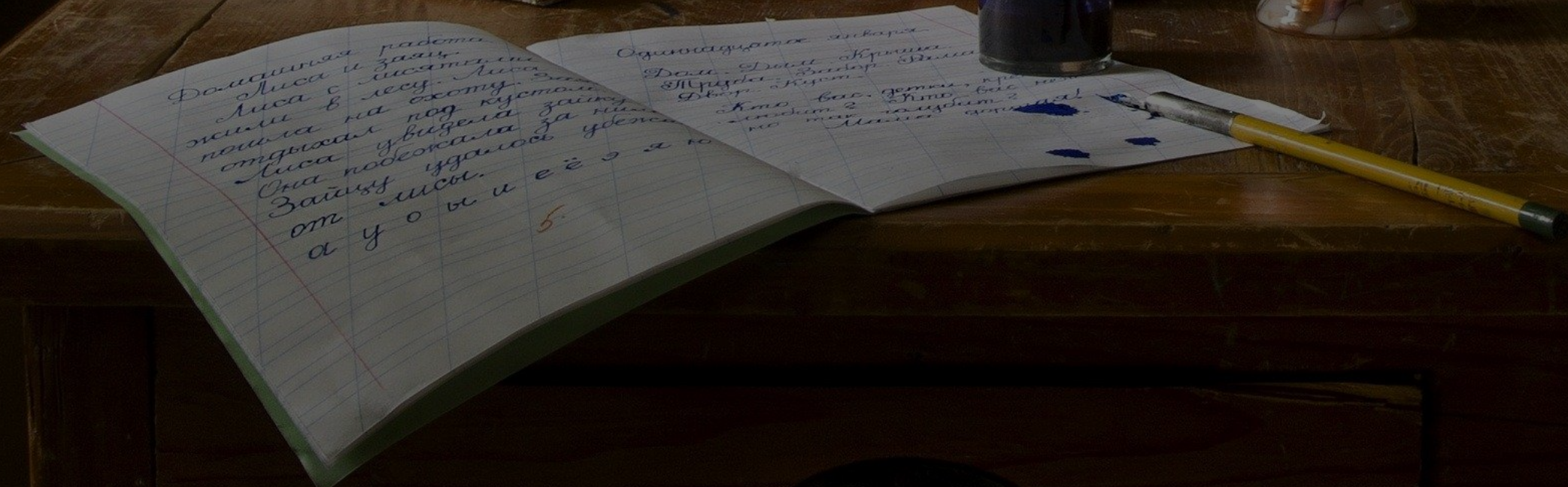
```
>> f(1)
```

```
ans =
```

```
1.7183
```


문제 접근

과제



과제 제출 방법(중요)

1. 메인 스크립트 파일은 week5_(문제번호).m으로 지난 과제와 동일
2. 각 문제 별로 따로 함수 파일을 만들어야 되는 경우가 있는데, 그 경우 함수의 이름 및 파일 이름은 문제의 지시를 따라야 함.
3. 따로 만든 함수가 있는 스크립트 파일까지 제출. 만약 week5_2.m에 plt라는 함수를 만들고 사용하였다 하면 plt.m 파일까지 반드시 제출.

과제1

anonymous function을 사용해 $g(z) = \frac{1}{1+\exp(-z)}$ 를 정의하고,

$g(-1), g(0), g(2)$ 를 fprintf 를 사용해 출력하시오. 단, 스크립트 파일은 week5_1.m이 유일합니다.

```
%week5_1.m
clear all; clc;
z=[-1,0,2];
[REDACTED]
fprintf("%f %f %f\n",g(z(1)),g(z(2)),g(z(3)))
```

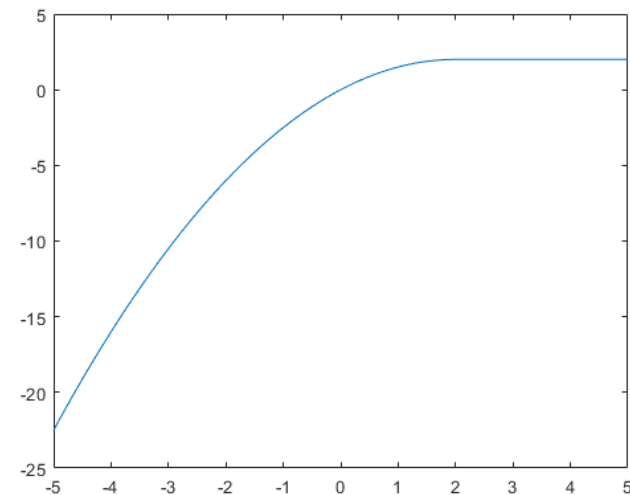
과제2

$f(x) = \begin{cases} -0.5(x-2)^2 + 2 & (x < 2) \\ 2 & (x \geq 2) \end{cases}$ 를 plot 하시오.

1. week5_2.m은 다음과 같은 코드로 고정한다.
2. plt.m에 $f(x)$ 를 $[-5, 5]$ 범위에서 plot 할 수 있는 함수를 만든다.

```
%week5_2.m  
clear all; clc;  
plt();
```

week5_2.m



과제3

ReLU 함수를 plot 하시오.

1. week5_3.m는 오른쪽 상단 코드로 고정한다.
2. ReLU 함수는 result에는 $\text{ReLU}(x)$ 값을 반환하고, 그래프에는 기본적인 $\text{ReLU}(x)$ 함수를 $[-5, 5]$ 범위에서 plot 하며, $\text{ReLU}(x)$ 에 해당하는 점을 찍는다.
3.
$$\text{ReLU}(x) = \begin{cases} 0 & (x < 0) \\ x & (x \geq 0) \end{cases}$$
4. ReLU.m 파일을 만들어 필요한 함수를 만든다.

```
%week5_3.m  
clear all; clc;  
x=2;  
result=ReLU(x)
```

week5_3.m

