

### 3. Programming with MATLAB

HoHee Kim

1

#### M files : 확장자 .m 로 저장

- ① **Script files** : 명령 창에 칠 명령어들의 모음,
- ② **Function files** : 보통 M 파일은 function files 을 가리킴

#### ① Script files

예)

```
g = 9.81; m=68.1; t=12; cd=0.25;
v = sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t)
```

scriptdemo.m

```
>> scriptdemo
```

```
v =
```

```
50.6175
```

```
>> g
```

```
g =
```

```
9.8100
```

파일 실행 후에 메모리에 남아있음

수치해석-3장

경북대 전자공학부 김호희

2

## ② Function files : 다른 언어에서 사용자 정의함수와 유사

```
function 출력변수 = 함수이름 (arguments 목록)
% 주석문
실행문
출력변수 = value;
end
```

freefall.m

예)

```
function v = freefall(t, m, cd)
% v = freefall(t, m, cd) computes the velocity
g = 9.81;
v = sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t);
end
```

```
>> freefall(12, 68.1, 0.25)
ans = 50.6175
>> g
??? Undefined function or variable 'g'
```

파일 실행 후에 메모리에서  
삭제되어서 error

수치해석-3장

경북대 전자공학부 김호희

3

## 출력이 여러 개인 함수 파일

```
function [출력변수1, 출력변수2] = 함수이름 (arguments 목록)
% 주석문
실행문
출력변수1 = value;
출력변수2 = value;
end
```

stats.m

예)

```
function [mean, stdev] = stats(x)
n = length(x);
mean = sum(x)/n;
stdev = sqrt(sum((x-mean).^2/(n-1)));
end
```

```
>> y = [8 5 10 12 6 7.5 4];
>> [m, s] = stats(y)
m = 7.5000
s = 2.8137
```

stats(y) 만 치면 평균만 출력

수치해석-3장

경북대 전자공학부 김호희

4

## 여러 함수 포함하는 함수파일

- Main 함수: 명령 창과 다른 함수들과 script 파일에 접근가능  
명령 창에서 호출가능
- Sub 함수: 메인 함수와 같은 파일 안의 서브함수에 접근가능  
명령 창에서 호출불가

예)

```
function v = freefallsubftn(t, m, cd)
    v = vel(t, m, cd);
end
function v = vel(t, m, cd)
    g = 9.81;
    v = sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t);
end
```

Main function

Sub function

```
>> freefallsubftn(12, 68.1, 0.25)
```

```
ans = 50.6175
```

```
>> vel(12, 68.1, 0.25)
```

error

freefallsubftn.m

수치해석-3장

경북대 전자공학부 김호희

5

## Input 함수

- `n = input('promptstring')`

수치 입력해서 변수에 저장

예) `m = input('weight? :');` 실행되면

weight? : 85

m 에 85 저장

- `n = input('promptstring','s')`

문자열 입력해서 변수에 저장

예) `n = input('Your name? :', 's');` 실행되면

Your name? : kim

n 에 kim 저장

## Output 함수

- `disp(value)`

예) `>> x = 7.5; disp(x), disp('Kim '), disp(3*5)`

7.5

Kim

15

수치해석-3장

경북대 전자공학부 김호희

6

▪ `fprintf('format', x, y,...)`

예) `>> fprintf('%5d %10.3f %8.5e \n', 100, 2*pi, pi);`  
           100          6.283          3.14159e+000

▪ `%d` - 정수 형태, `%e` - 지수 형태, `%f` - 실수 형태,  
`%g` - `%e` 나 `%f` 중에 간단한 표현, `\n` - new line, `\t` - tab

예) `function fprintfdemo`  
       `x = [1 2 3];`  
       `y = [20.4 12.6 17.8];`  
       `z = [x; y];`  
       `fprintf('      x          y \n');`  
       `fprintf('%5d %10.2f \n', z);`  
       `end`

fprintfdemo.m

2개 벡터를 동시  
입력할 때 행렬로

행보다 열이 우선

`>> fprintfdemo`

결과

x	y
1	20.40
2	12.60
3	17.80

수치해석-3장

경북대 전자공학부 김호희

7

## if, if...else, & if...elseif

`if` 조건문  
       실행문  
`end`

grader.m

예) `function grader(grade)`  
       `if grade >= 60, disp('passing grade')`  
       `elseif grade >= 0, disp('not passing grade')`  
       `else error('No!')`  
       `end`  
       `end`

한 줄에 적을 때, 로 구분

error()는 에러발생시 메시지 출력

`>> grader(30)`  
       not passing grade  
`>> grader(-6)`  
       ??? Error using ==> grader at 4  
       No!

수치해석-3장

경북대 전자공학부 김호희

8

**Relational operators**

<code>==</code>	같다
<code>~=</code>	같지 않다
<code>&lt;</code>	보다 작다
<code>&gt;</code>	보다 크다
<code>&lt;=</code>	작거나 같다
<code>&gt;=</code>	크거나 같다

**Logical operators**

- `~x (Not)` : x 가 false(or zero) 이면 true , 그렇지 않으면 false
- `x & y (And)` : x 와 y 가 둘 다 true(or non-zero) 이면 true
- `x || y (Or)` : x 나 y 중 하나라도 true(or non-zero) 이면 true

예) `a = -1, b = 2, x = 1, y = 'b'`

`a*b>0 & b==2 & x>7 || ~(y>'d')`

**F & T & F || ~ F T**

- 실행하다가 pause 만나면 실행중지, 아무 키를 다시 치면 계속실행
- 무한 루프에 빠지면 ctrl+C 또는 ctrl+Break 치면 중단

수치해석-3장

경북대 전자공학부 김호희

9

**switch**

```
switch testexpression
case 값1
    실행문
case 값2
    실행문
...
otherwise
    실행문
end
```

예) `grade = 'B';`

```
switch grade
case 'A'
    disp('Excellent')
case 'B'
    disp('Good')
otherwise
    disp('Huh!')
end
```

Good

```
>> sign(25.6)
ans = 1
>> sign(0)
ans = 0
>> sign(-0.776)
ans = -1
```

부호값을 반환하는 MATLAB 의 built-in 함수

수치해석-3장

경북대 전자공학부 김호희

10

예) `function v = freefall2(t, m, cd)`  
`% v = freefall(t, m, cd) computes the velocity`  
`switch nargin` Number of input argument  
`case 0`  
`error('Must enter time and mass')`  
`case 1`  
`error('Must enter mass')`  
`case 2`  
`cd = 0.27;`  
`end`  
`g = 9.81;`  
`v = sqrt(g*m/cd)*tanh(sqrt(g*cd/m)*t);`  
`end`

`>> freefall2(12, 68.1)`

`ans = 48.8747`

`freefall2.m`

`>> nargin('freefall2')` 함수의 인수 개수 출력

`ans = 3`

수치해석-3장

경북대 전자공학부 김호희

11

**for** : 정확한 횟수만큼 반복할 때

`for` 제어변수 = start : step : finish 1씩 증가할 땐 step 생략  
 실행문  
`end`

예) `for i= 1:5`  
`disp(i)`  
`end`

1  
2  
3  
4  
5

예) `for j= 5:-1:1`  
`disp(j)`  
`end`

5  
4  
3  
2  
1

예) `i = 0;`  
`for t = 0:0.02:50`  
`i = i + 1;`  
`y(i) = cos(t);`  
`end`

벡터화

`t = 0:0.02:50;`  
`y = cos(t);`

수치해석-3장

경북대 전자공학부 김호희

12

예) `t = 0: 0.01: 5;`  
`for i=1 : length(t)`  
`if t(i)>1`  
`y(i) = 1/t(i);`  
`else y(i) = 1;`  
`end`  
`end`

메모리 미리 할당

`t = 0: 0.01: 5;`  
`y = ones(size(t));`  
`for i=1 : length(t)`  
`if t(i)>1`  
`y(i) = 1/t(i);`  
`end`  
`end`

**while** : 조건문에 따라 반복할 때

`while` 조건문  
 실행문  
`end`

예) `x = 8;`  
`while x>0`  
`x = x-3;`  
`disp(x)`  
`end`

5  
2  
-1

예) `while (1)`  
`if x<0 , break, end`  
`x = x - 5 ;`  
`end`

수치해석-3장

경북대 전자공학부 김호희

13

**Anonymous function** : 함수파일 생성 없이 간단한 함수정의 할 때

`f = @(arg1, arg2, ...)` 수식표현

예) `>> f1 = @(x,y) x^2 + y^2;`  
`>> f1(3,4)`  
`ans = 25`  
`>> a = 4 ; b = 2 ;`  
`>> f2 = @(x) a*x^b;`  
`>> f2(3)`  
`ans = 36`  
`>> a = 3 ;`  
`>> f2(3)`  
`ans = 36`  
`>> f2 = @(x) a*x^b;`  
`>> f2(3)`  
`ans = 27`

a 값이 변해도 함수결과가 변하지 않음

parameter 값이 변경되면  
함수를 다시 생성해야 됨

수치해석-3장

경북대 전자공학부 김호희

14

## Function functions : 함수를 argument 로 전달받아 동작시키는 함수

예) `function favg = funcavg(f, a, b, n)`  
`x = linspace(a,b,n);`  
`y = f(x);`  
`favg = mean(y);`  
`end`

funcavg.m

```
>> vel= @(t)...
    sqrt(9.81*68.1/0.25)*tanh(sqrt(9.81*0.25/68.1)*t);
>> funcavg(vel,0,12,60)
ans =    36.0127
>> funcavg(@sin,0,2*pi,180)
ans =   -6.3001e-017
```

Anonymous 함수 정의

built-in 함수 or 파일 함수를  
argument 로 불러들일 때 @

built-in 함수인 fplot 에서  
anonymous 함수 vel 의  
t가 0~12까지 그래프 출력

```
>> fplot(vel,[0 12])
```

수치해석-3장

경북대 전자공학부 김호희

15

## Passing parameters: 다른 함수의 parameter도 argument로 전달받음

예) `function favg = funcavg(f, a, b, n, varargin)`  
`x = linspace(a,b,n);`  
`y = f(x, varargin{:});`  
`favg = mean(y);`  
`end`

```
>> vel= @(t, m, cd)...
    sqrt(9.81*m/cd)*tanh(sqrt(9.81*cd/m)*t);
>> funcavg(vel,0,12,60,68.1,0.25)
ans =
    36.0127
>> funcavg(@(t) vel(t,68.1,0.25),0,12,60)
>> funcavg(vel,0,12,60,100,0.28)
ans =
    38.9345
```

Anonymous 함수 재정의

funcavg.m

파라미터 값을 쉽게 변경 가능

수치해석-3장

경북대 전자공학부 김호희

16



### Local variables

예) 

```
function c = adder(a, b)
    x = 88
    a
    c = a + b
end
```

adder.m

```
>> x = 1; y = 4; c = 8
    c = 8
```

```
>> d = adder(x,y)
    x = 88
    a = 1
    c = 5
    d = 5
```

```
>> c, x, a
    c = 8
    x = 1
```

함수가 종료되어 삭제

Undefined function or variable 'a'. Error in ScopeScript  
수치해석-3장                      경북대 전자공학부 김호희                      17

### Global variables

예) 

```
global SIGMA
SIGMA = 5.670367e-8;

Tair = 30 ; Tw = 15 ; Td = 27.7 ;
Jan = AtmLongWaveRad(Tair, Td),
Jbr = WaterBackRad(Tw),
Jnet = Jan - Jbr
```

test.m

```
function Ja = AtmLongWaveRad(Tair, Td)
    global SIGMA
    eair = 4.596 * exp(17.27 * Td / (237.3 + Td));
    Ja = 0.97 * SIGMA * (Tair + 273.15)^4 * (0.6 + 0.031 * sqrt(eair));
end
```

```
function Jb = WaterBackRad(Twater)
    global SIGMA
    Jb = 0.97 * SIGMA * (Twater + 273.15)^4;
end
```

```
>> test
    Jan = 354.8483
    Jbr = 379.1905
    Jnet = -24.3421
```