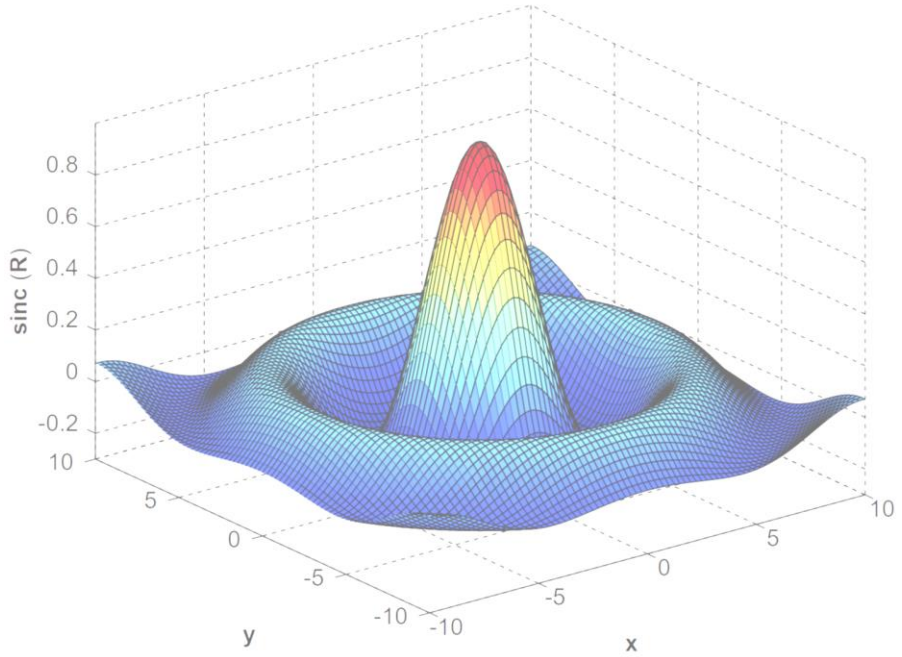
2021년도 2학기

응용전산및실습 II (02) #3



교과목명 : 응용전산 및 실습 II (02)

담당교수 : 이 수 형

E-mail : soohyong@uu.ac.kr

교재명 : 유인물

Selective Statement (switch-case)

- switch 문

```
switch 변수 또는 식
case 값1
    문장;  %변수 또는 식의 값이 값1일 경우 실행
case 값2
    문장;  %변수 또는 식의 값이 값2일 경우 실행
    :
otherwise
    문장;  %변수 또는 식의 값이 어떠한 case 값과도 같지 않을 때
end
```

Selective Statement (switch-case)

- 예제

```
k = input('Enter k value : ');
y = rem(k, 2)
switch(y)
    case 0
        disp('Even number.');
    case 1
        disp('Odd number.');
end
```

mod() vs. rem()

- 나머지 구하는 함수들
  - rem : remainder, mod : modulus
  - `mod(a, b), rem(a, b)` : a를 b로 나누었을 때의 나머지 계산 (7 나누기 -2)
  - `mod(a, b): a-b.*fix(a./b)` ⇒ `7=(-2)*(-3)-1`
  - `rem(a, b): a-b.*floor(a./b)` ⇒ `7=(-2)*(-4)+1`

```
>> rem(5, 3)
ans =  2
>> rem(-5, 3)
ans = -2
>> rem(5, -3)
ans =  2
>> rem(-5, -3)
ans = -2
```

```
>> mod(5, 3)
ans =  2
>> mod(-5, 3)
ans =  1
>> mod(5, -3)
ans = -1
>> mod(-5, -3)
ans = -2
```

mod() vs. rem()

- C 언어의 % 연산자

```c
#include <stdio.h>

int main()
{
    printf("5 % 3 = %d\n", 5 % 3);
    printf("-5 % 3 = %d\n", -5 % 3);
    printf("5 % -3 = %d\n", 5 % -3);
    printf("-5 % -3 = %d\n", -5 % -3);
    return 0;
}
```

```
5 % 3 = 2
-5 % 3 = -2
5 % -3 = 2
-5 % -3 = -2
```

사용자 정의 함수

Review

- 함수(function)의 구조

```
% ------------------------------------------------------------
% Contents of 'help' command
% ------------------------------------------------------------

function [output_variables] = function_name(input_variables)
    % do MATLAB commands
end
```

Item	Description
function	M 파일이 함수를 정의한다는 것을 의미하는 키워드
output_variables	함수의 결과를 저장할 변수명 (생략 가능하며, 두 개 이상인 경우 [] 사용)
function_name	함수 이름 – 파일 이름과 반드시 같게 한다
input_variables	해당 함수의 입력 변수 (단일 혹은 복수 개)

출력변수가 2개 이상인 함수

- 출력변수의 값들을 나열하되, []를 사용.

```
function [A, B, C, ...] = function_name(input_variables)
```

- 예제 : 최소, 최대값 구하기

```
function [min, max] = MinMax(A)
    TotalItem = length(A);
    max = 0;
    min = 100000000;
    for i = 1 : TotalItem
        if max < A(i)
            max = A(i);
        end
        if min > A(i)
            min = A(i);
        end
    end
end
```

출력변수가 2개 이상인 함수

- 예제 : 최소, 최대값 구하기 - 결과

```
>> A = [ 4, 6, 2, 8, 16]
A =

     4     6     2     8    16

>> [mi, ma] = MinMax(A)
mi =   2
ma =   16
>>
```

부함수 (subfunction)

- 부함수
  - 하나의 함수 파일이 두 개 이상의 함수를 가지는 경우
  - 주함수(primary function) : 첫번째 함수 – 함수 파일의 이름과 일치
  - 부함수(subfunction) : 나머지 함수

```
% primary function : same as file name
function a = prifunc(x)
    % ...
end

% subfunctions...
function b = subfunc(x)
    % ...
end
```

부함수 (subfunction)

- 예제 (CalcSub.m) : $1 + (1 \times 2) + (1 \times 2 \times 3) + \cdots + (1 \times \cdots \times n)$

```
% Calculate 1 + (1 x 2) + (1 x 2 x 3) . . . (1 x 2 x 3 x … x n)
% Main Function
function ret = CalcSub(n)
    ret = 0;
    for k=1 : n
        ret = ret + MultToN(k);
    end
end

% Sub Function
function mtot = MultToN(n)
    mtot = 1;
    for k=1 : n
        mtot = mtot * k;
    end
end
```

부함수를 호출하는 것은 가능
각 함수들은 자신만의 workspace을 가짐
→ 서로 다른 함수의 변수 접근 불가능

부함수 (subfunction)

- 예제 (stat.m) : 평균(mean)/표준편차(standard deviation) 계산

```
function [me, stddev] = stat(v)
    n = length(v);
    me = average(v, n);
    stddev = stdandarddev(v, me, n)
end

function av = average(x, num)
    av = sum(x) / num;
end

function sd = stdandarddev(x, ave, num)
    xd = x - ave;
    xd2 = xd.^2;
    sd = sqrt(sum(xd2) / (num - 1));
end
```

$$x_{ave} = (x_1 + x_2 + \cdots + x_n)/n$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x_{ave})^2}{n-1}}$$

부함수 (subfunction)

- 예제 (stat.m) : 결과

```
>> A = [ 4, 6, 2, 8, 16]
A =

    4    6    2    8   16

>> stat(A)
stddev =  5.4037
ans =  7.2000
>>
```

중첩함수 (nested function)

- 중첩함수
  - 사용자 정의함수 내부에 다른 함수를 정의하는 경우
  - 중첩 함수들 사이에는 작업공간(workspace)을 공유
    → 함수 내의 변수들에 접근 가능함

```
% primary function : same as file name
function a = prifunc(x)
    % ...
    function b = nestedfn(y)
        % ...
    end
end
```

중첩함수 (nested function)

- 예제 (stat.m) : 평균(mean)/표준편차(standard deviation) 계산

```
function [me stddev] = stat2(v)
    n = length(v);
    me = average;
    stddev = stdandarddev

    function av = average
      av = sum(v) / n;
    end

    function sd = stdandarddev
      xd = v - me;
      xd2 = xd.^2;
      sd = sqrt(sum(xd2) / (n - 1));
    end
end
```

$$x_{ave} = (x_1 + x_2 + \cdots + x_n)/n$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x_{ave})^2}{n-1}}$$