

수치해석 과제#2

2015111113 김준기

3.4

```
avg_temperature.m  Mannings_equation.m  practice3_5.m  practice3_9.m
1  function avg_temp = avg_temperature(mean_ann_temp, ...
2      peak_temp, t)
3      %특정 지역의 평균 하루 온도를 계산해주는 함수입니다.
4
5      %For initialize some parameters, and variables
6      day_peak_temp = 205;
7      freq_ann_vari = (2*pi)/365;
8      sum_temp = 0;
9
10     %하루 평균 온도이므로 일단 다 더해야함.
11     for i = 1 : numel(t)
12         sum_temp = sum_temp + mean_ann_temp + (peak_temp - mean_ann_temp)* ...
13             cos(freq_ann_vari * (t(i) - day_peak_temp));
14     end
15
16     avg_temp = sum_temp/numel(t);
17
18     end
```

하루 최고 온도, 년 온도 분산 값은 상수값으로 주어졌습니다.

c언어와는 다르게 복합연산자인 +=가 없다는 것을 알게 되었습니다.

따라서 시그마 덧셈을 다음과 같이 표현하였습니다.

마지막으로 t변수의 개수만큼 나누기 위해 행렬 개수를 계산해주는 numel()라는 제공 함수를 사용하였습니다.

```
avg_temperature.m  Mannings_equation.m  practice3_5.m  practice3_9.m
1  AZ = 0:1:59;
2  WA = 180:1:242;
3  result1 = avg_temperature(23.1, 33.6, AZ);
4  result2 = avg_temperature(10.6, 17.6, WA);
5
6  fprintf("(a) answer = %f\n(b) answer = %f\n", result1, result2);
7
8
9
10
```

스크립트 파일명은 제가 잘못 작성했는데, 3_5가 아니라 3_4 입니다.

스크립트 함수에서 특정 지역의 평균 하루 온도를 계산해주는 함수를 호출하도록 했습니다.

명령 창

MATLAB을 처음 사용한다면 [시작하기](#)를 참조하십시오.

```
>> practice3_5
(a) answer = 13.133210
(b) answer = 17.226549
```

fx >>

3.5

```
Mannings_equation.m x practice3_9.m x practice3_actural5.m x taylorforsin.m x
1 function result = taylorforsin(phase, x)
2 %테일러 급수를 이용하여 phase단계까지 sin함수에 대한
3 %여러 계산값들을 계산합니다.
4
5 result_tmp = 0;
6 pivot = 1:2:(2*(phase-1) + 1);
7 true = sin(x);
8
9 fprintf("계산해야 하는 각항의 제곱은 다음과 같습니다.\n");
10 disp(pivot);
11 fprintf("\n");
12
13 for i = 1:phase
14     if(mod(i, 2) == 0)
15         %짝수 승수인 경우, 항의 마이너스를 더해줘야함
16         result_tmp = result_tmp - (x^pivot(i))/factorial(pivot(i));
17     else
18         %그렇지 않을 경우, 항의 플러스를 더해줘야함
19         result_tmp = result_tmp + (x^pivot(i))/factorial(pivot(i));
20     end
21     fprintf("At %d sequence, approximation value = %f\n", i, result_tmp);
22     error = ((true - result_tmp)/true)*100;
23     fprintf("true value = %f\n", true);
24     fprintf("error percentage = %f\n\n", error);
25 end
26 result = result_tmp;
27
28
29 end
30
```

```
Mannings_equation.m x practice3_9.m x practice3_actural5.m x +
1 result = taylorforsin(8, 0.9);
```

phase는 최대 이용할 항의 개수이고, x는 계산하고 싶은 값입니다.

pivot은 항이 하나씩 늘어 때 마다 1, 3, 5, 7, 9, 11, 13, ...순으로 x의 제곱과 분모의 팩토리얼 값이 변하게 되는데, 이 때 사용하기 위한 값들을 이용하고자 하는 항의 개수인 phase를 이용하여 구했습니다.

만약 짝수 번째 항이면 결과 값을 빼야 하므로 마이너스가 붙도록 하고, 홀수 번째 항이면 더하도록 하는것을 나머지 연산인 mod를 이용 하였습니다.

매 반복문 마다 현재 몇번째 항까지 이용하였는지와, error 값 그리고, sin계산 결과 값을 보여줍니다.

스크립트 파일을 이용하여 함수를 실행하였습니다.

명령 창

MATLAB을 처음 사용한다면 [시작하기](#)를 참조하십시오.

계산해야 하는 각점의 제곱은 다음과 같습니다.

1 3 5 7 9 11 13 15

At 1 sequence, approximation value = 0.900000

true value = 0.783327

error percentage = -14.894559

At 2 sequence, approximation value = 0.778500

true value = 0.783327

error percentage = 0.616206

At 3 sequence, approximation value = 0.783421

true value = 0.783327

error percentage = -0.011980

At 4 sequence, approximation value = 0.783326

true value = 0.783327

error percentage = 0.000135

At 5 sequence, approximation value = 0.783327

true value = 0.783327

error percentage = -0.000001

At 6 sequence, approximation value = 0.783327

true value = 0.783327

error percentage = 0.000000

At 7 sequence, approximation value = 0.783327

true value = 0.783327

error percentage = -0.000000

fx

At 8 sequence, approximation value = 0.783327

true value = 0.783327

error percentage = 0.000000

fx >> |

3.5번 문제에 대한 결과 창입
니다.

3.9

```

1 function Mannings_equation(data_mat)
2     tmp = size(data_mat);
3     fprintf("-----|-----|-----|-----|-----|");
4     fprintf("      n      |      s      |      B      |      H      |      U      |");
5     fprintf("-----|-----|-----|-----|-----|");
6     for i = 1:tmp(1)
7
8         for j = 1:tmp(2)
9             fprintf("| %9f ", data_mat(i, j));
10        end
11        tmp_result = sqrt(data_mat(i, 2))/data_mat(i, 1) * ((data_mat(i, 3)...
12            * data_mat(i, 4))/(data_mat(i, 3) + 2* data_mat(i, 4))^(2/3);
13        fprintf("| %9f |%n", tmp_result);
14        fprintf("-----|-----|-----|-----|");
15    end
16 end
17
18 end
19
20

```

상위 label인 n, s, B, H, 그리고 Manning's 방정식의 결과값인 U를 먼저 커맨드창에 표시하도록 하였습니다.

그리고 반복문으로 매 행마다 결과 값을 그려주는 식으로 하였는데, 하나의 행의 각 열의 값은 H를 제외하고 반복문을 이용해 표시했습니다.

H까지 값을 표시하면, 방정식의 결과값인 U는 반복문이 끝난 뒤 표시하도록 하였습니다.

```

1 data_matrix = [0.036 0.0001 10 2; 0.020 0.0002 8 1; 0.015 0.0012 20 1.5;...
2               0.030 0.0007 25 3; 0.022 0.0003 15 2.6];
3
4 Mannings_equation(data_matrix);

```

각 칸의 비율을 맞추기 위해, 모두 9개의 숫자를 표시할 수 있는 공간을 미리 확보하기 위한 형식지정자를 이용하였습니다.

명령 창

MATLAB을 처음 사용한다면 [시작하기](#)를 참조하십시오.

스크립트파일에서 Mannings_equation을 호출하도록 하였습니다.

```
>> practice3_9
```

n	s	B	H	U
0.036000	0.000100	10.000000	2.000000	0.352343
0.020000	0.000200	8.000000	1.000000	0.609366
0.015000	0.001200	20.000000	1.500000	2.756946
0.030000	0.000700	25.000000	3.000000	1.589379
0.022000	0.000300	15.000000	2.600000	1.220713

fx >>

3.11

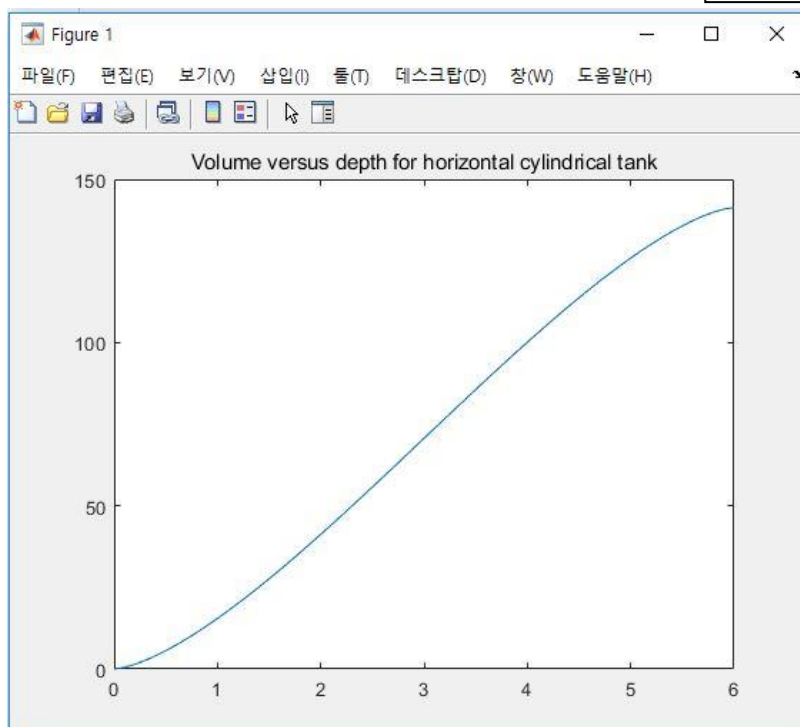
```
practice3_11.m x Fnorm.m x cylinder.m x practice3_27.m x +
1 function cylinder(r, L, plot_title)
2     % volume of horizontal cylinder
3     % inputs:
4     % r = radius
5     % L = length
6     % plot_title = string holding plot title
7     h = 0:0.1:2*r;
8     % hollow horizontal cylinder 모양이 원기둥 얹어놓은 상태에서 liquid를
9     % 채우는 형식임. 따라서 liquid의 최대 깊이는 원의 지름의 길이인 2r임.
10    V = (r^2 * acos((r-h)/r) - (r-h).*sqrt(2*r*h - h.^2)) * L;
11
12    plot(h, V);
13    title(plot_title);
14    %순서 주의 -> plot하고 title해야함!
15
16
17 end
```

명령 창

MATLAB을 처음 사용한다면 [시작하기](#)를 참조하십시오.

```
>> cylinder(3, 5, 'Volume versus depth for horizontal cylindrical tank')
fx >> |
```

커맨드 창을 이용하여, 함수를 실행하였습니다.



3.27

```

1  function Norm = Fnorm(x)
2      tmp = size(x);
3      m = tmp(1);
4      n = tmp(2);
5      sum_result = 0;
6
7      for i = 1:m
8          for j = 1:n
9              sum_result = sum_result + x(i, j)*x(i, j);
10         end
11     end
12     Norm = sqrt(sum_result);
13
14     % 다른 방법으로는 반복문 사용 안하고, api함수랑 제공되는 연산만 이용
15
16     fprintf("api sum함수와 행렬요소 제공 .^ 연산 이용 = %f\n", sqrt(sum(x.^2, 'all')));
17
18     % 주의 matlab에서는 변수이름과 함수이름이 동일하면 변수를 먼저 참조함.
19     % 이 문제 때문에, sum함수 호출했는데, 변수 sum이 호출되어 index를 1초과
20     % 한다는 에러가 발생하였음. 따라서 변수 sum을 sum_result로 변경.
21
22 end
23
24

```

위에 방법은 단순히 중첩 반복문을 이용해 계산하였고, 밑의 방법은 매트랩에서 제공하는 sum() 함수와 각 행렬 요소를 더해해주는 .^ 연산을 이용하였습니다.

sum() 함수 이용 시 vector의 모든 요소 값을 더하려면 두번째 인수로 'all' 값을 주면 된다는 것을 알게 되었습니다.

```

1  A = [5 7 9; 1 8 4; 7 6 2];
2  Fn = Fnorm(A);
3  fprintf("3.27문제 answer = %f\n", Fn);
4

```

명령 창

MATLAB을 처음 사용한다면 [시작하기](#)를 참조하십시오.

A =

5	7	9
1	8	4
7	6	2

api sum함수와 행렬요소 제공 .^ 연산 이용 = 18.027756

3.27문제 answer = 18.027756

fx >>