시뮬레이션 기초 및 실습

HW 2

제출일 2019년 4월 20일
과목명 시뮬레이션 기초 및 실습
담당교수 김지범 교수님
학과 산업경영공학과
학번 201401210
이름 강형원

1. 아래와 같은 각각의 x범위에 대해서 세 개의 서로 다른 함수 값을 갖는 그래프를 Matlab으로 그리고 싶다.

$$\begin{cases} y = (x-1)^2, & x \le 0 \\ y = -2(x-0.5)^2 + 1.5, & 0 < x \le 1 \\ y = -x+2, & x > 1 \end{cases}$$

x축에서 [-2, 4] 범위에서 균등하게 떨어진 총 200개의 점을 sampling해서 'plot' 함수를 이용해 그래프를 그리고 원형태의 마커를 이용하여 sampling한 점의 위치를 표시하자. 코드 작성시 반드시 for 반복문과 조건문을 사용하자. 세 개의 그래프를 하나의 figure 창에 그리고 빨간색 실선을 사용하자.

- Algorithm

```
% x축에서 [-2, 4] 범위 균등하게 200개 점

% x에 따른 fx값 구해서 plot 생성
```

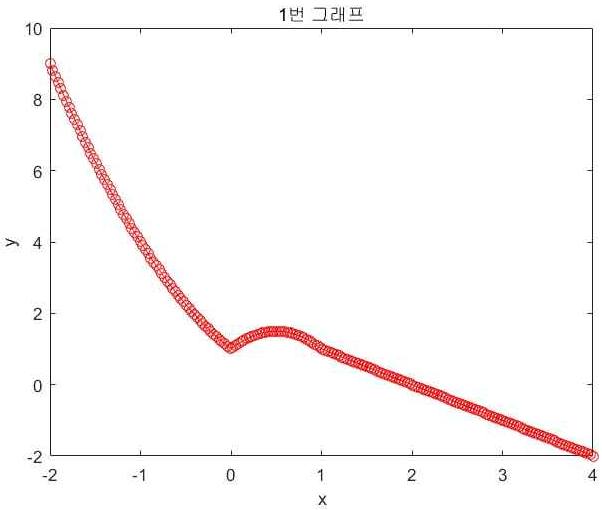
- code (func.m)

```
function y = func(x)

for k=1:length(x)

    if x(k) <= 0    % x<=0인경우
        y(k) = (x(k)-1)^2;
    elseif (x(k) > 0) && (x(k) <= 1)    % 0<x<=1인경우
        y(k) = -2*(x(k)-0.5)^2+1.5;
    else    % x>1인경우
        y(k) = -x(k)+2;
    end
end
```

- code (HW2_1.m)

```
x = linspace(-2, 4, 200);   % x축에서 [-2, 4] 범위 균등하게 200개 점

plot(x, func(x), '-or') % x에 따른 fx값 구해서 plot 생성
title('1번 그래프');
xlabel('x');
ylabel('y')
```

- 결과



2. 아래 왼쪽과 같이 정의된 함수 cusum.m을 작성하라. 이 함수는 입력을 행렬 M으로 받고 출력은 행렬 A이다. 행렬 M과 행렬 A의 행렬 크기는 항상 같다. 행렬 A의 각 요소 (원소)는 같은 열에서 그 행보다 같거나 작은 행에 있는 모든 요소의 합으로 이루어져 있다. 아래 가운데 예를 보자. 함수 작성 시 다른 함수는 사용하지 말고 함수 'size'를 사용하여 입력 행렬 M의 행수와 열수를 출력으로 받자. 아래 오른쪽에는 여러분이 작성한 함수가 제대로 동작하는지 확인하기 위한 main.m 함수가 있다. 이를 3번 실행한 결과를 리포트에 추가하자

- Algorithm

```
function A = cusum(M)
% M: 임의의 크기를 갖는 행렬
% A: M과 행수와 열수가 같은 행렬
% 같은 열에서 그 행보다 같거나 작은 행에 있는 요소들의 합 구하기
```

- code(cusum.m)

```
function A = cusum(M)

col = size(M, 1);   % M의 행수
row = size(M, 2);   % M의 열수

for k=1:row
    for j=1:col
        A(j,k) = sum(M(1:j, k));     % 같은 열에 있고 그 행보다 같거나 작은 행에 있는 요소들의 합을 A 행렬에 입력
    end
end
```

- code(HW2_2.m)

```
row = randi([2, 5], 1, 1);
col = randi([2, 5], 1, 1);
M = randi([3, 10], row, col);
A = cusum(M)
```

- 결과

M:

8	8
6	10
4	5
9	3
9	5

A:

8	8
14	18
18	23
27	26
36	31

M:

3	4	6	7	9
10	5	9	3	7
9	6	8	7	3
10	10	5	10	10

A:

3	4	6	7	9
13	9	15	10	16
22	15	23	17	19
32	25	28	27	29

M:

3	3	6	4
8	3	9	8
10	6	10	5
10	10	5	5
8	6	5	5

A:

3	3	6	4
11	6	15	12
21	12	25	17
31	22	30	22
39	28	35	27

3. 1) 극 좌표계란 아래 왼쪽 그림과 같이 평면 위의 위치를 각도 (θ)와 거리 (r)를 써서 나타내는 2차원 좌표계이다. xy 좌표계에서 (x, y)가 극 좌표계, (r, θ), 에서는 $r = \sqrt{x^2 + y^2}$, $\theta = \tan^{-1}(\frac{y}{x})$ 로 표시된다. r값은 원점에서 이점까지의 거리를 나타내고 $0 \leq \theta \leq 2\pi$ 이다. 예로 xy 좌표계에서 (3, 4)는 극 좌표계 (r, θ)에서 (5, $\tan^{-1}(\frac{4}{3})$)이다. xy좌표계를 입력으로 받아서 출력으로 극 좌표계를 출력하는 함수 xypolar.m 함수를 작성하라. 이 함수의 원형은 아래 가운데에 있다. (x, y)=(3,4)와 (x, y)=(-4, 4)에서의 극 좌표, (r, θ), 를 각각 구해보고 출력해보자

- Algorithm

```
function [r, theta] = xypolar(x, y)
% x: xy 좌표계의 x 좌표
% x: xy 좌표계의 y 좌표
% r: 극 좌표계의 r
% theta: 극 좌표계의 theta (0부터 2pi)
```

- code (xypolar.m)

```
function [r, theta] = xypolar(x, y)

r = sqrt(x^2 + y^2);
theta = atan(y/x);
```

- code (HW2_3_1.m)

```
[r, theta] = xypolar(3, 4)
[r, theta] = xypolar(-4, 4)
```

- 결과

(3, 4) 일 때

```
r =

     5


theta =

    0.9273
```

(-4, 4) 일 때

```
r =

    5.6569


theta =

   -0.7854
```

2) x와 y모두 (-2, 2)사이에 정의된 다트판이 있다. 이 다트판에서 다트를 한번 던지면 아래 오른쪽 그림과 같이 점수판에 표시된 점수를 얻는다. 먼저, 이 다트판에서 반지름이 0.5인 원안에 다트가 들어오면 점수를 10점 얻고 hits를 1 얻는다. 만일, 던진 다트가 반지름이 1.5인 원밖에 있으면 y의 위치에 따라서 y>0 이면1점을 잃거나 y≤0이면 2점을 잃는다. 반지름이 0.5에서 1.5사이에 있으면 각도에 따라 1점에서 8점 사이의 점수를 얻는다. 이 다트 실험은 점수≥25이거나 hits≥2 (즉, 점수가 25점이상이거나 hits가 2회 이상이면) 종료한다. 이 다트 시뮬레이션을 100번 반복해보자. 평균적으로 다트를 몇 번 던져야 실험이 종료되는가? 실험이 종료되었을 때의 점수의 평균은 얼마인가? 실험 시 1)에서 사용한 극 좌표계를 이용하고 다트판을 그릴 필요는 없다.

- Algorithm

```
counts = [];    % 총 횟수 기록할 배열 선언
scores = [];    % 총 score 기록할 배열 선언

for % 100번 반복

score = 0;  % score 선언
hits = 0;   % hit 수
count = 0;  % 조건을 만족 할 때까지 총 던진 횟수

    while 점수가 25점 이상이거나 hits가 2회 이상일 경우 종료
        다트가 맞은 점 랜덤값
        극 좌표계로 변환

       if 반지름이 0.5인 원 안일 경우
       elseif 반지름이 1.5인 원 밖일 경우
          if y 좌표가 양수인 경우
          else    y 좌표가 음수인 경우
```

```
        else    반지름 0.5에서 1.5 사이 일 경우
            if theta가 음수인 경우 양수로 변환

            if 0도에서 45도 사이 인 경우
            elseif 45도에서 90도 사이 인 경우
            elseif 90도에서 135도 사이 인 경우
            elseif 135도에서 180도 사이 인 경우
            elseif 180도에서 225도 사이 인 경우
            elseif 225도에서 270도 사이 인 경우
            elseif 270도에서 315도 사이 인 경우
            else   315도에서 360도 사이 인 경우
```

- code

```
counts = [];    % 총 횟수 기록할 배열 선언
scores = [];    % 총 score 기록할 배열 선언

for i=1:100     % 100번 반복

score = 0;  % score 선언
hits = 0;   % hit 수
count = 0;  % 조건을 만족 할 때까지 총 던진 횟수

    while score < 25 && hits < 2 %점수가 25점 이상이거나 hits가 2회 이상일 경우 종료
        count = count + 1;
        xy = -2 + rand(1, 2)*4;     % 다트가 맞은 점 랜덤값
        [r, theta] = xypolar(xy(1), xy(2)); % 극 좌표계로 변환

        if r <= 0.5     % 반지름이 0.5인 원 안일 경우
            score = score + 10;
            hits = hits + 1;
        elseif r > 1.5  % 반지름이 1.5인 원 밖일 경우
            if xy(2) > 0    % y 좌표가 양수인 경우
                score = score - 1;
            else    % y 좌표가 음수인 경우
                score = score - 2;
            end
        else    % 반지름 0.5에서 1.5 사이 일 경우
            if theta < 0    % theta가 음수인 경우 양수로 변환
                theta = 2*pi + theta;
            end

            if theta >= 0 && theta < (2*pi/8)   % 0도에서 45도 사이 인 경우
                score = score + 1;
```

```
            elseif theta >= (2*pi/8) && theta < (2*pi/8)*2  % 45도에서 90도 사이
                score = score + 2;
            elseif theta >= (2*pi/8)*2 && theta < (2*pi/8)*3    % 90도에서 135도
                score = score + 3;
            elseif theta >= (2*pi/8)*3 && theta < (2*pi/8)*4   % 135도에서 180도
                score = score + 4;
            elseif theta >= (2*pi/8)*4 && theta < (2*pi/8)*5   % 180도에서 225도
                score = score + 5;
            elseif theta >= (2*pi/8)*5 && theta < (2*pi/8)*6   % 225도에서 270도
                score = score + 6;
            elseif theta >= (2*pi/8)*6 && theta < (2*pi/8)*7   % 270도에서 315도
                score = score + 7;
            else    % 315도에서 360도
                score = score + 8;
            end
        end
    end
    counts = [counts count];    % 총 횟수 기록
    scores = [scores score];    % 총 score 기록
end
disp('평균 던진 횟수:');
disp(mean(counts));
disp('평균 score:');
disp(mean(scores));
```

- 결과

```
평균 던진 횟수:
   18.8100

평균 score:
   25.9100
```