Experiment 22 – Monte Carlo Simulation

Abstract

This report simulates the probabilities by using a Monte Carlo model. In fact, this report applies the Monte Carlo model to a geometric model to calculate the probability of scoring a penalty kick in various situations. This report also answers all the questions in the script in detail. This experiment provides an understanding of the general principles of the Monte Carlo model.

Declaration

Contents

1 Introduction

1.1 Objectives

The objective of this experiment is to explore and learn Monte Carlo techniques to simulate real-life stochastic processes and find solutions. Also this experiment used MATLAB as a simulation tool, although this was not a learning objective it did allow us to become more familiar with MATLAB.

1.2 Theoretical Background

The Monte Carlo method is also known as the statistical simulation method and the statistical experiment method. It is a numerical simulation method that takes probabilistic phenomena as the object of study. It is a method of calculating the statistical values to be obtained by the sampling method to presume unknown characteristic quantities. Monte Carlo is the famous gambling town of Monaco and the method is named to indicate its random sampling nature. It is therefore suitable for conducting computational simulation tests on discrete systems. In computational simulation, the stochastic properties of a system can be simulated by constructing a probabilistic model that is similar to the performance of the system and performing random tests on a digital computer [1].

2 Materials and Methods/Procedure

2.1 Materials

- MATLAB (R2022a – academic use)

2.2 Methods/Procedure

- First, write the code according to the script requirements. The code is in two parts because there are two different requirements.

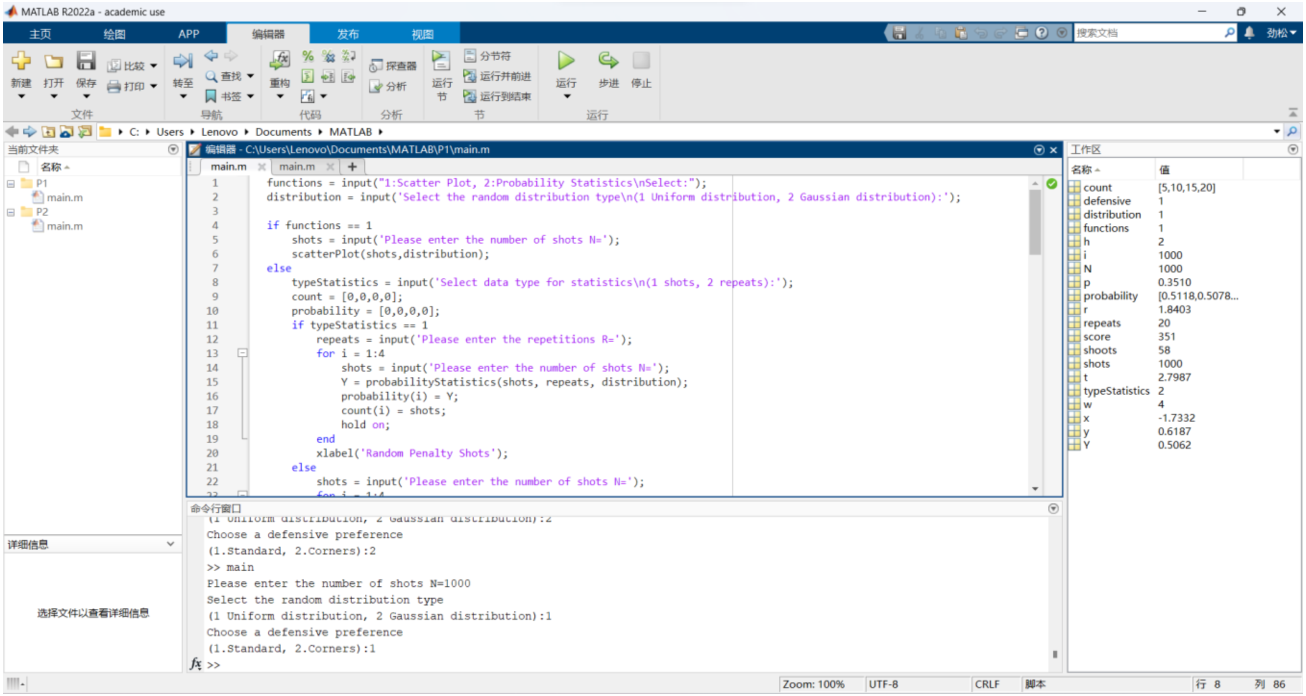

Fig.1. Part 1

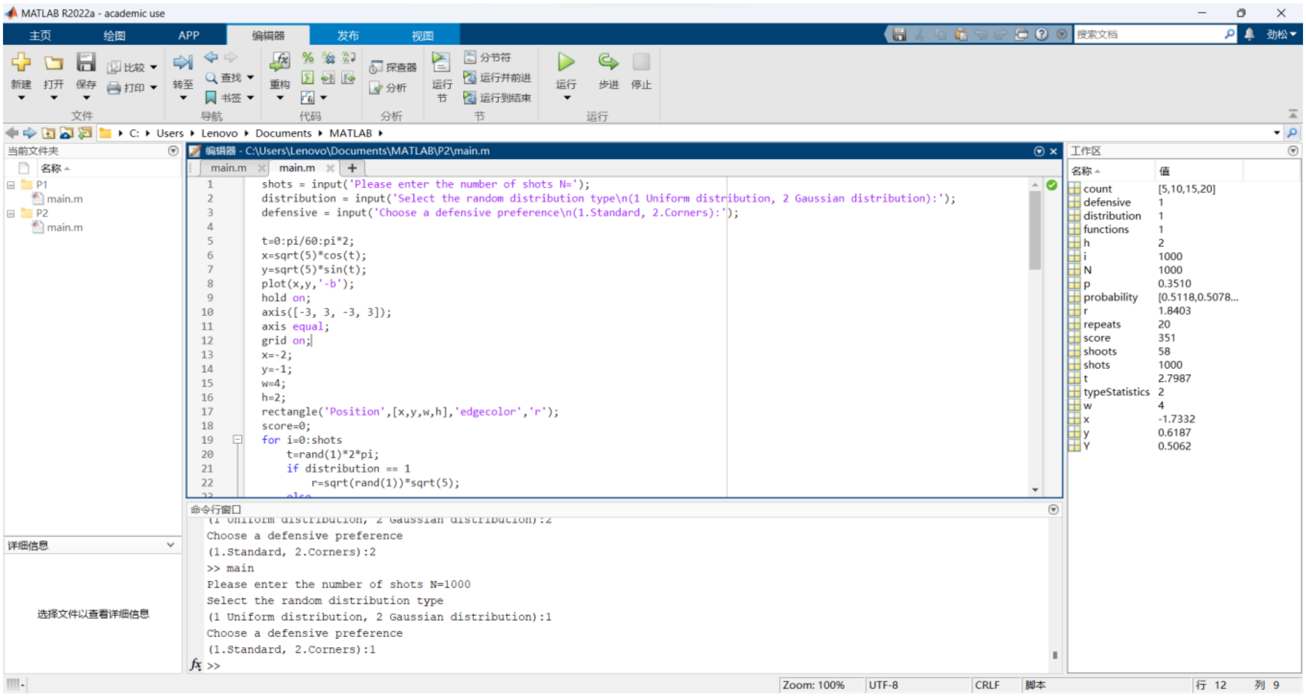

Fig.2. Part 2

- Perform simulations according to the script requirements and obtain the simulated probabilities.

3 Results and Analysis

3.1 Part 1: No Goalkeeper Tests

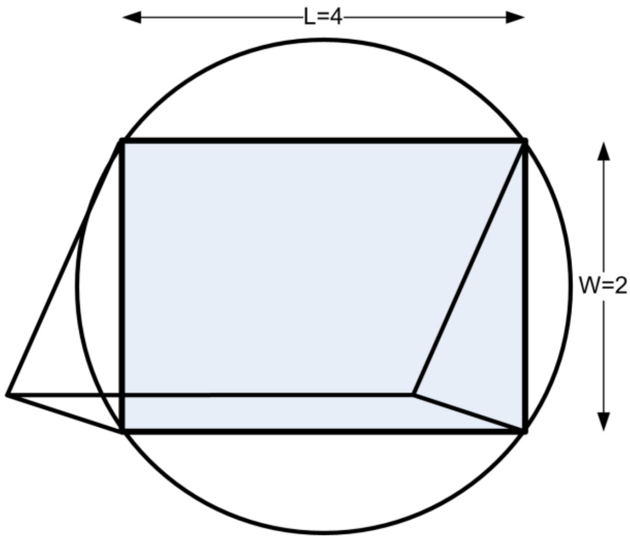

Fig.3. The goal arrangement

- **Task 1:** If a large number of shots is attempted, derive a numerical value for the fraction of balls entering the goal to the total number of balls in the circular area. Assume the penalty taker is blindfolded (i.e., the shots are uniformly distributed within the circle).

  Ans: Ideally, the probability of scoring a goal is the area of the rectangle divided by the area of the circle. The diameter of the circle is then the diagonal of the rectangle. The area of the rectangle is calculated to be 8 and the area of the circle is 5π. The equation is as follows: $4 \times 2 = 8$, $2^2 \times 4^2 = 20$, $\sqrt{20} = 2\sqrt{5}$, $\pi \times \sqrt{5}^2 = 5\pi$, $\frac{8}{5\pi} \approx \mathbf{0.5092958}$

- **Task 2:** Design and write a computer programme to find the probability of scoring by simulating N random penalty shots and repeating this experiment R times and taking the mean of the attempts. Let N and R be inputs to your code. Use a uniform random number generator in the simulation.

Ans: The code has been put together for convenience. This single string of code solves all the tasks of the first part, such as plotting scatter plots, finding probabilities, plotting line graphs, etc. The uniformly distributed function rand () was used for the plotting, and the parametric equation was used for the implementation of the circle, but the randomization of the parameter ‘r’ would lead to a more central tendency. The figure below intuitively explains why the uniform distribution of parameters directly leads to denser at the center of the circle, but taking $\sqrt{r}$ will counteract this effect. See the appendix for details.

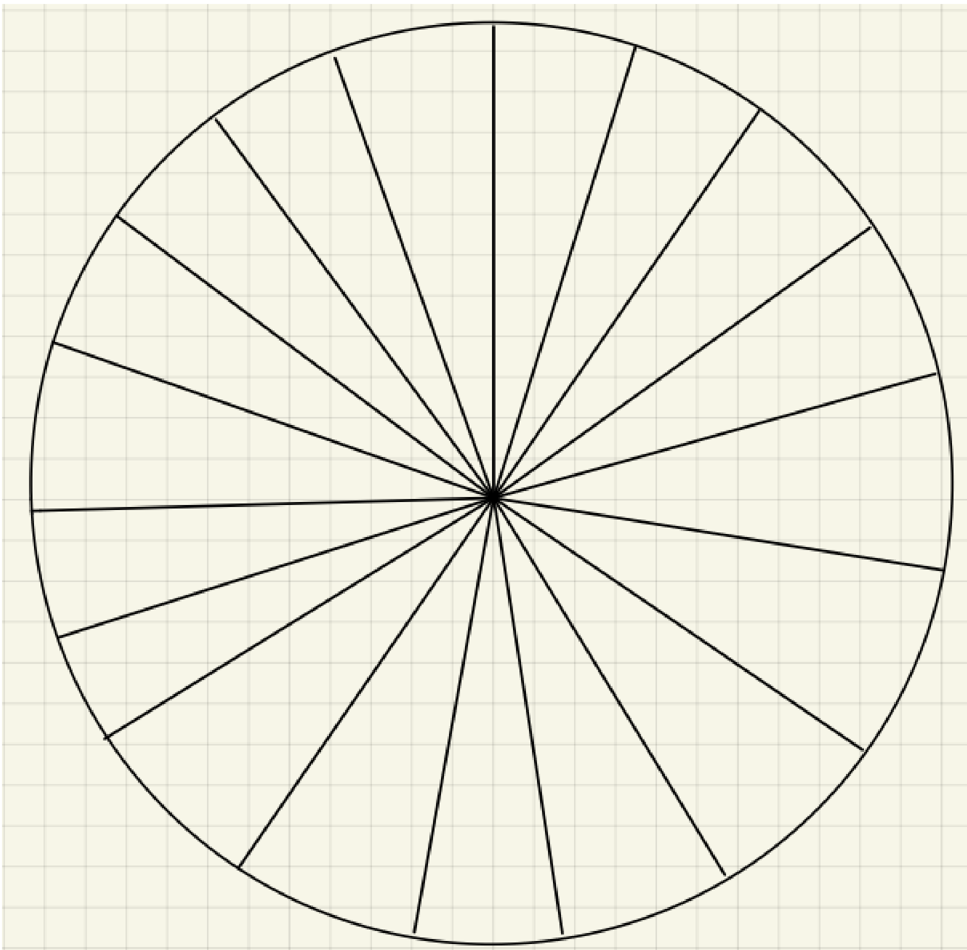

Fig.4. For uniformly distributed circles of parametric equations

- **Task 3:** Produce an appropriate scatter plot illustrating your experiment for N = 1,000 and R = 1, using red crosses to indicate score (i.e., balls on target) and blue circles to indicate miss (i.e., balls off target). Insert an appropriate legend.

Ans:

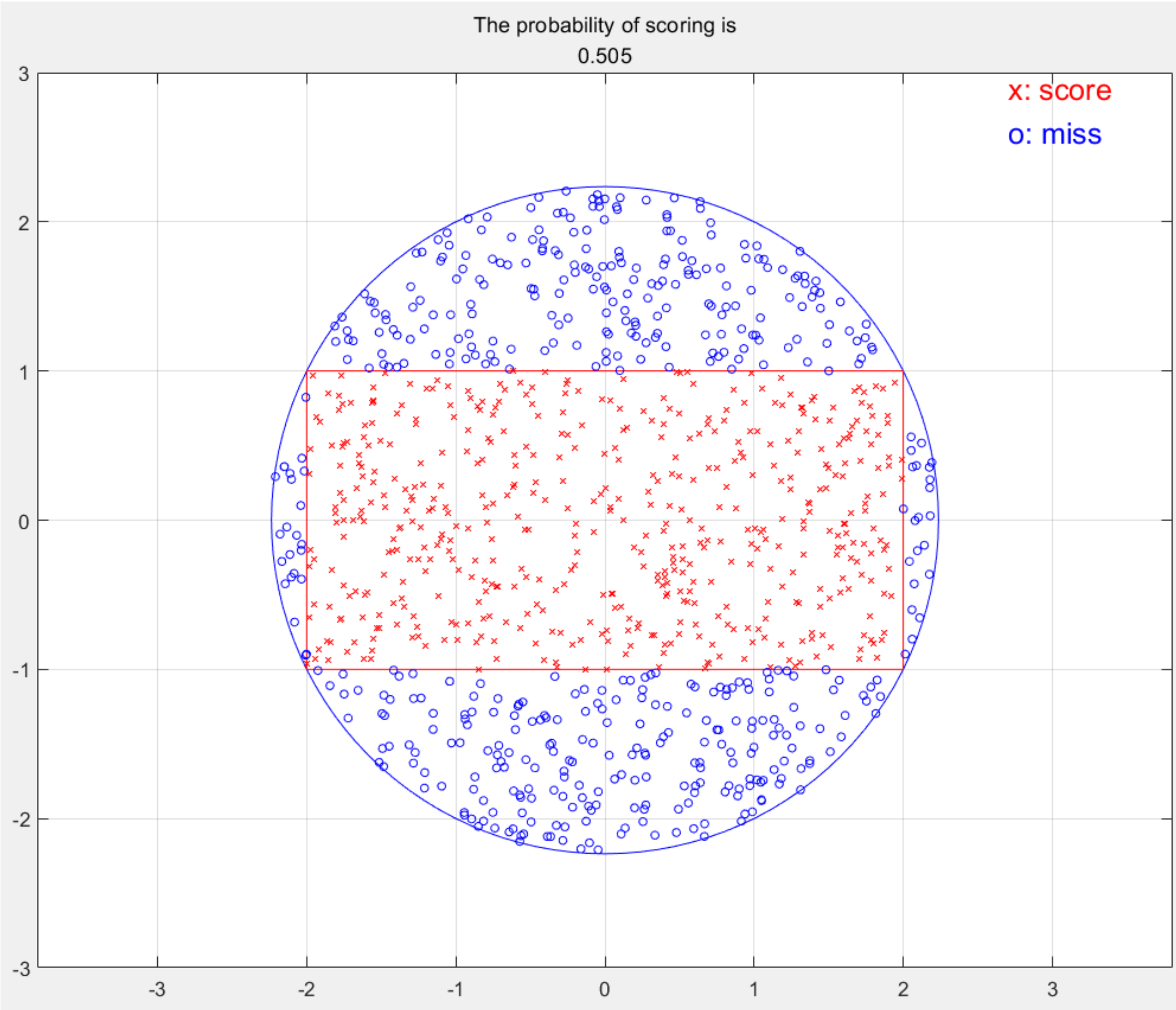

Fig.5. Scatter plot (Uniformly distribution)

- **Task 4:** For R = 5, find the probability of scoring for N = 100, N = 1,000, N = 10,000 and N = 100,000. Plot the probability against the value of N. Comment on the shape of the plot, making reference to the theoretical probability calculated in Task-1. Remember to label the axes and to insert an appropriate caption in your report.

Ans: The graph below shows that as the number of random penalties N gets larger, the probability of scoring a goal gets closer to the theoretical probability value. The theoretical value is around 0.50929 and it is already very close to this value when the number of penalties reaches 100,000.

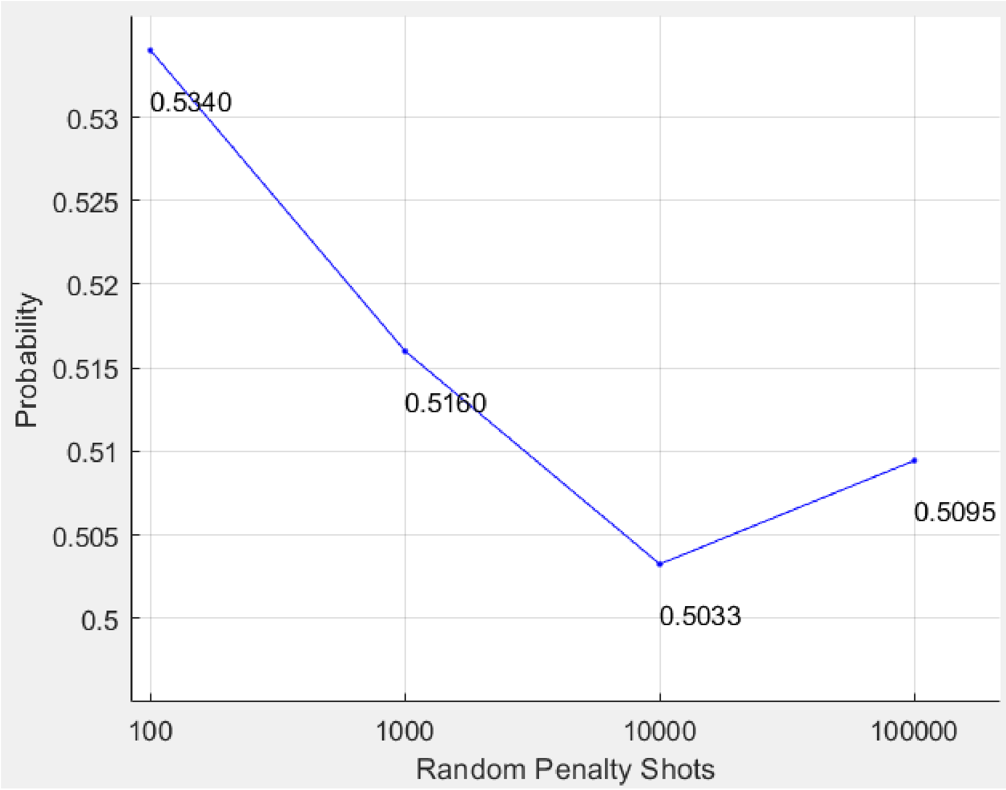

Fig.6. Probability against the value of N

- **Task 5:** For N = 1,000, find the probability of scoring for R = 5 times, R = 10 times, R = 15 times and R = 20 times. Plot this probability against the value of R. Comment on the shape of the plot.

  Ans: As with task 4 the simulated probabilities get closer to the theoretical values as the number of rounds increases. This means that it approaches around 0.5092.

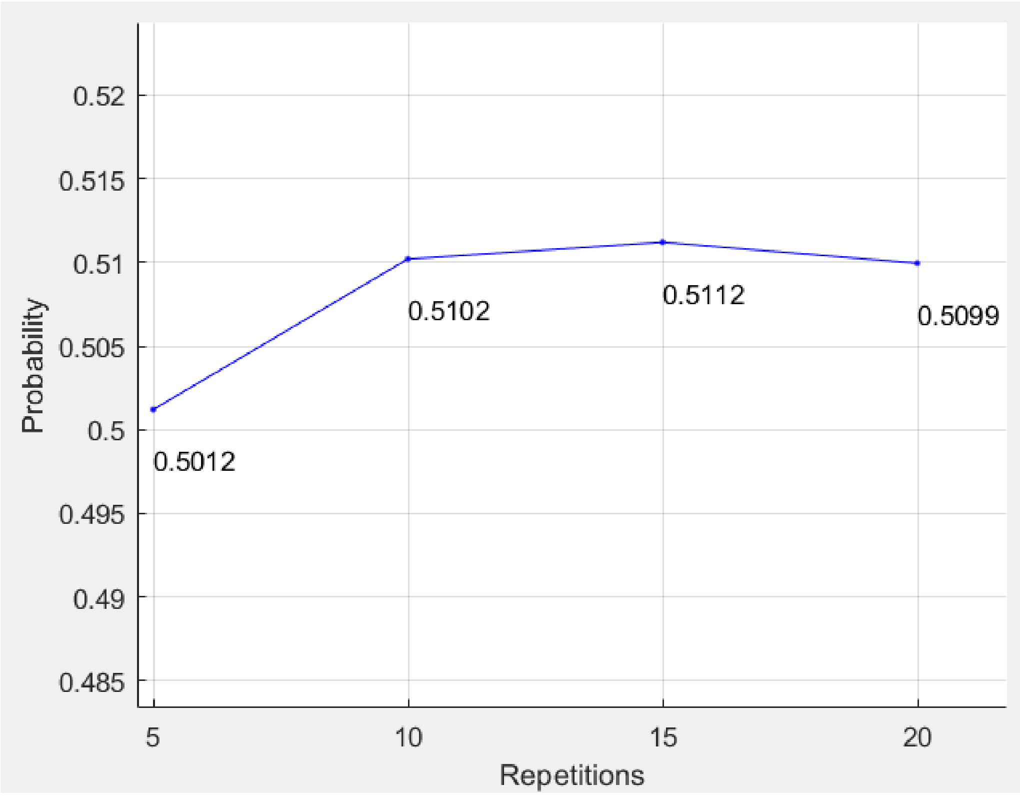

Fig.7. Probability against the value of R

- **Task 6:** The graph shows that either a gradual increase in the number of repetitions or an increase in the number of penalties will bring the probability gradually closer to the theoretical value, essentially making no difference whichever way is used. This is because the computer uses pseudo-random numbers. For example, repeating the experiment five times, each time simulating 1000 penalty kicks, is essentially the same as simulating 5000 penalty kicks directly, because the pseudo-random numbers are used and averaged in any case.

- **Task 7: Gaussian random**

  1) The program uses the randn () function to obtain Gaussian distributed random numbers with radius as the standard deviation. The consequence of this is that some points will fall outside the circle, i.e., only about 68% of the points will fall

inside the circle. The Gaussian distribution is inherently more probable the closer you get to the centre, so you can just take the value. See the code in the appendix for details.

2)

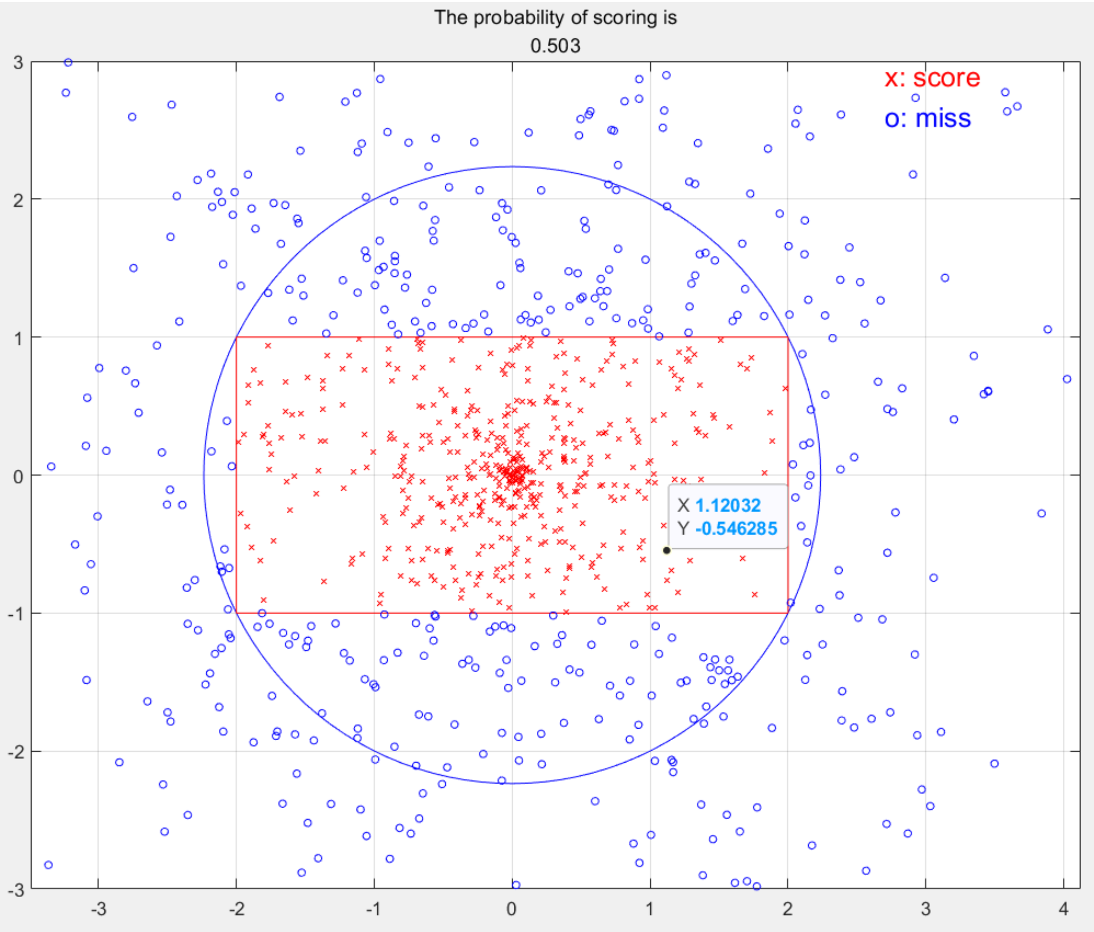

Fig.8. Scatter plot (Gaussian distribution)

3) It can be seen that as the number of simulated penalties increases, the probability gets closer to 0.499.

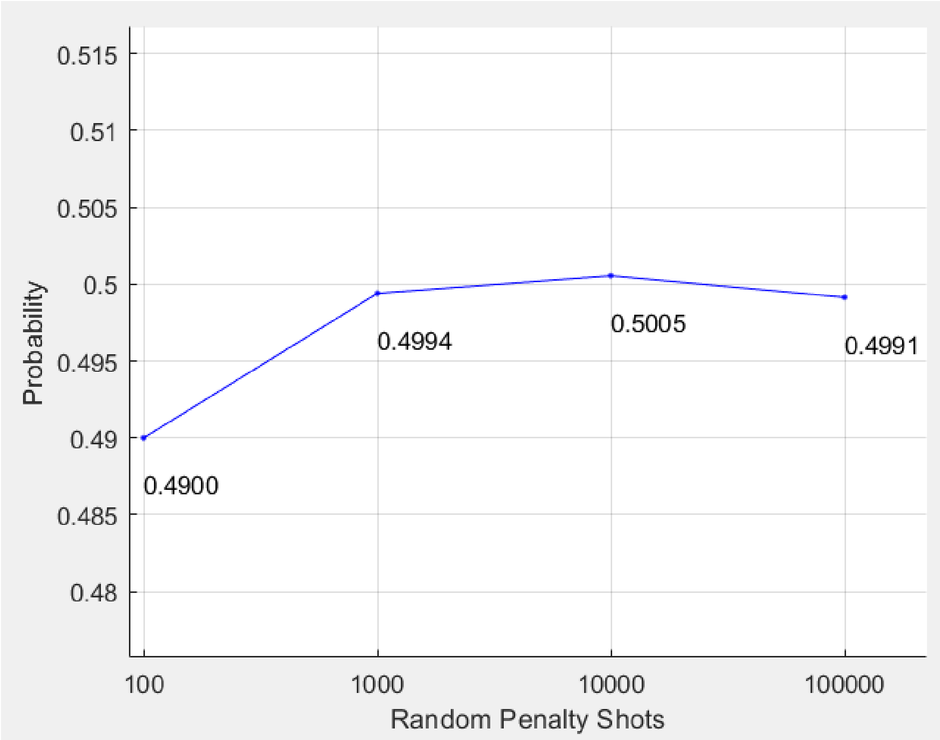

Fig.9. Probability against the value of N (Gaussian)

4) From this time, it can be seen that as the number of repetitions increases, the probability of scoring gets closer to 0.5.

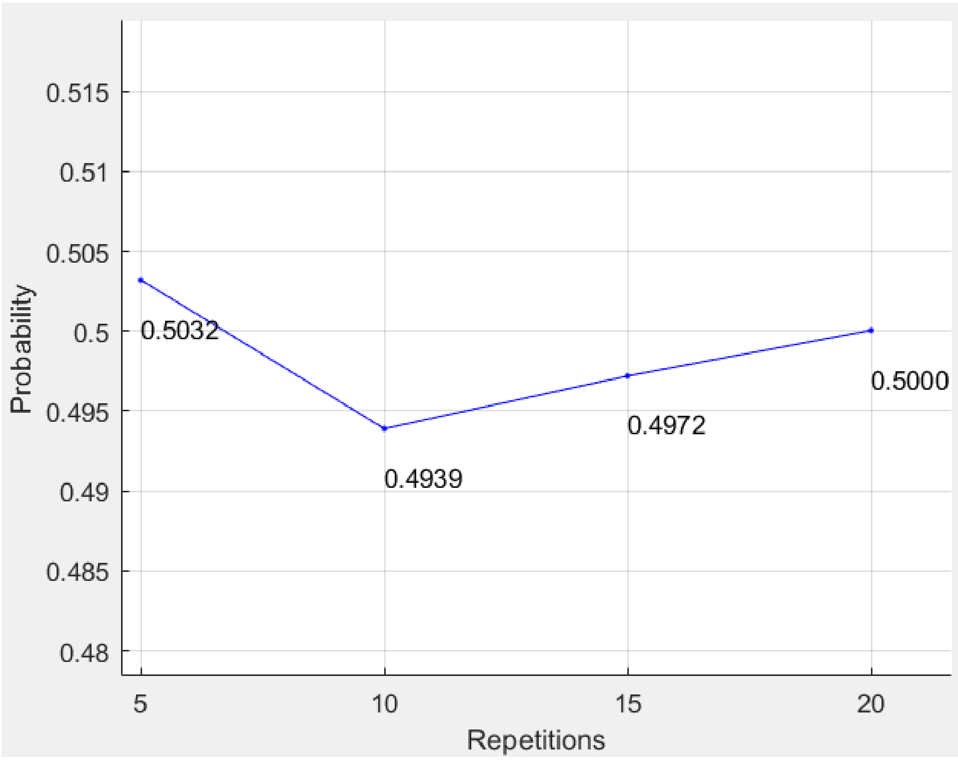

Fig.10. Probability against the value of R (Gaussian)

5) These two simulations show that the penalty is closer to 0.499 when the number of penalties increases, but closer to 0.5 when the number of repetitions increases.

3.2 Part 2: With Goalkeeper Tests

- **Task 8:** Assuming that the goalkeeper action is modelled as a uniform random process, what is the probability of scoring a goal if the penalty taker kicks 100 balls with uniform random distribution within the circle, as before. Increase the kicks to 1000. Compare the probability values with the case where no goalkeeper was in the goal (Task-1 and Task-3 above).

  Ans: First of all, we know that the 5 ways of keeping the goal are even, so we can find the goal probability based on the theory: $\frac{\left(4\times\frac{1}{5}+6\times\frac{4}{5}\right)}{5\times\pi} \approx \mathbf{0.3565}$. This is also confirmed by the scatter plot obtained from the simulation. In contrast to the absence of a goalkeeper, the presence of a goalkeeper reduces the probability of scoring by a fraction: from about 0.509 to about 0.3565. The probability of scoring from 100 and 1000 penalties was obtained by simulation: 0.41 and 0.339 respectively as shown in the graph below.

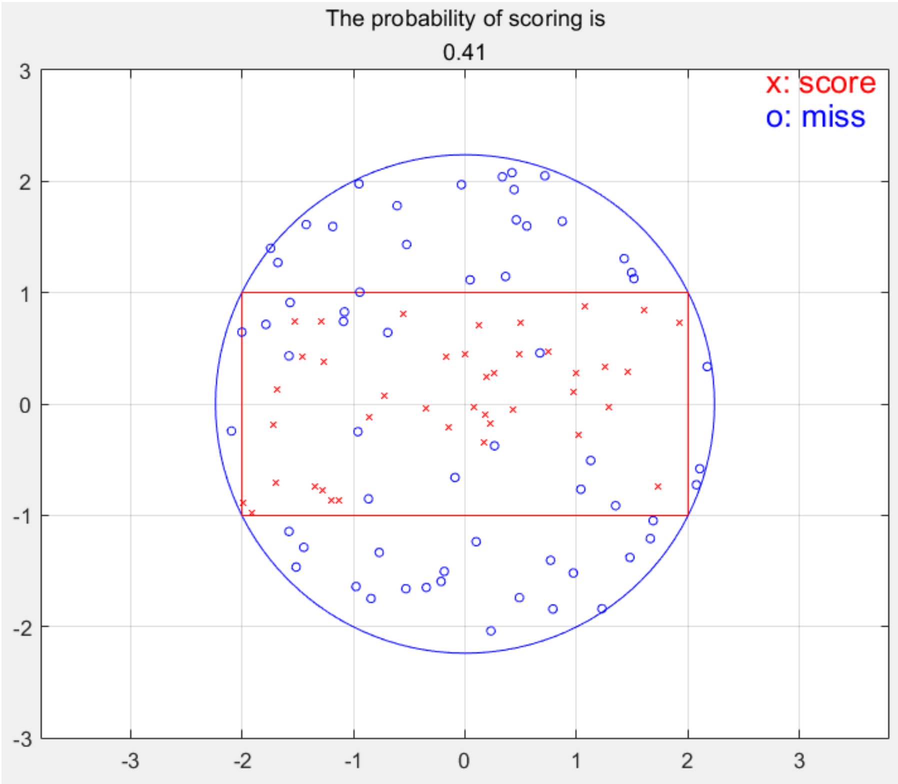

Fig.11. With goalkeeper (100 uniform distributions)

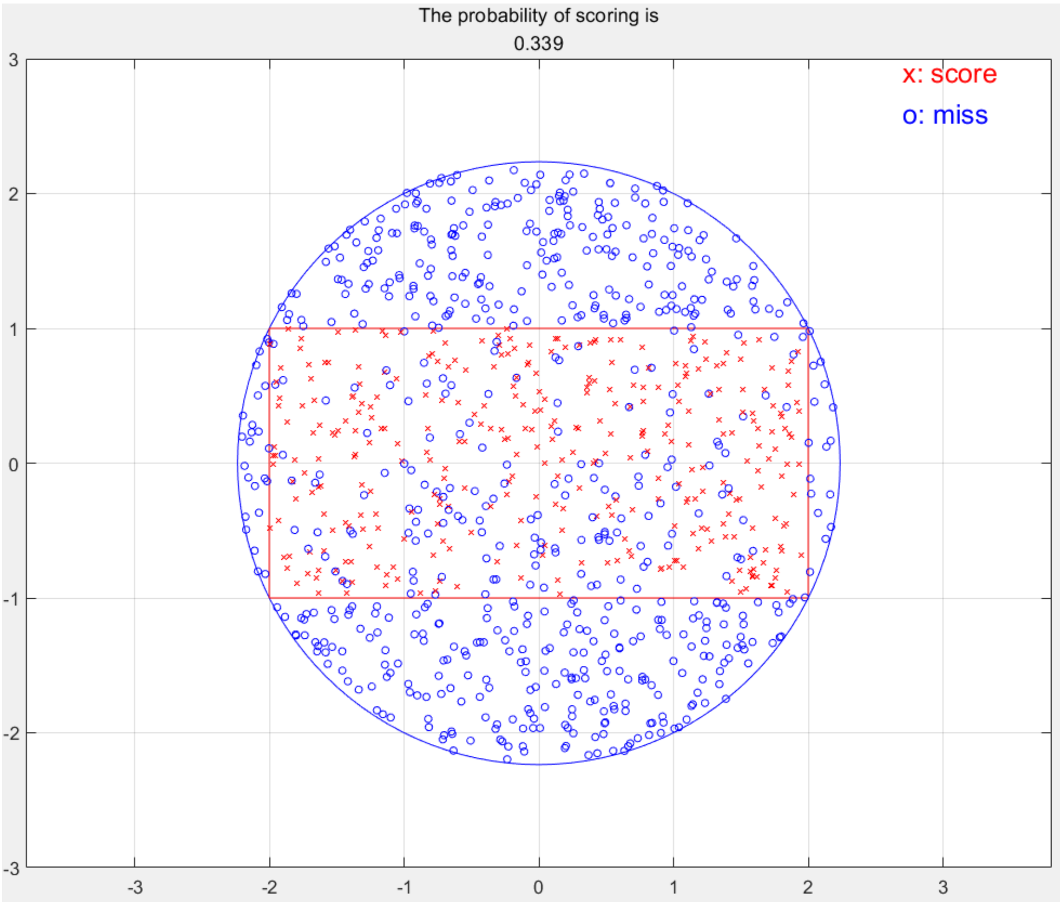

Fig.12. With goalkeeper (1000 uniform distributions)

- **Task 9:** Repeat Task-8 if the balls are kicked with a Gaussian random distribution (as in Task-7). Compare your results with those obtained in Task-7 and Task-8.

  Ans: From the graph below, the probability of scoring a goal from 100 and 1000 penalties is 0.36 and 0.341 respectively. Due to the nature of the Gaussian distribution, shots are more biased towards the centre and there are also more keepers near the centre than at the corners, so this results in a lower probability of scoring a penalty shot with a keeper than with a uniform distribution.

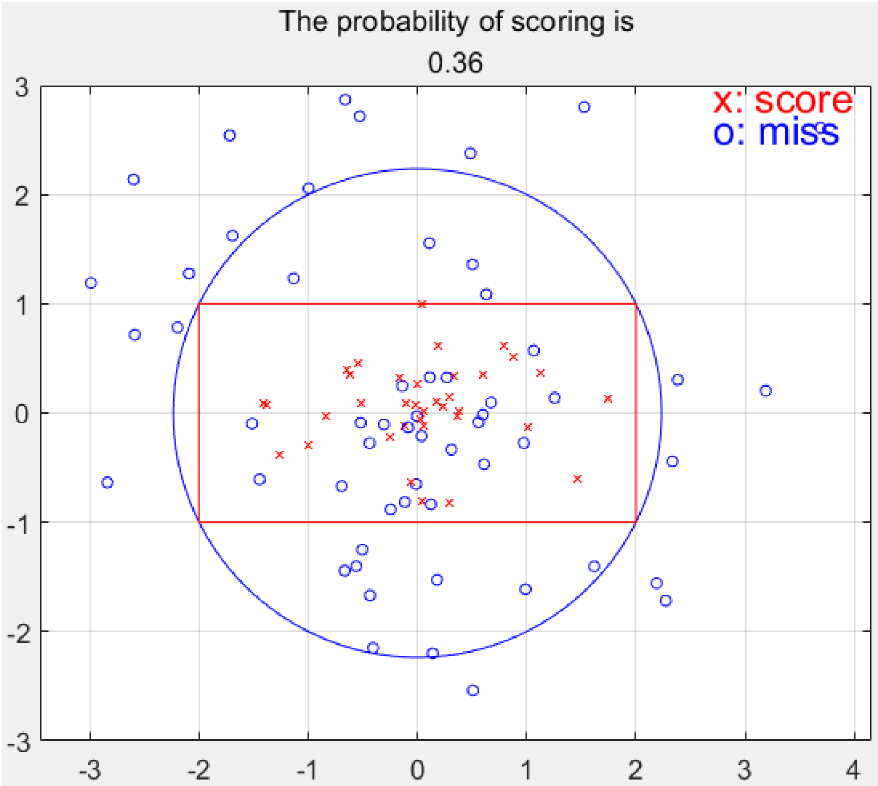

Fig.13. With goalkeeper (100 Gaussian distributions)

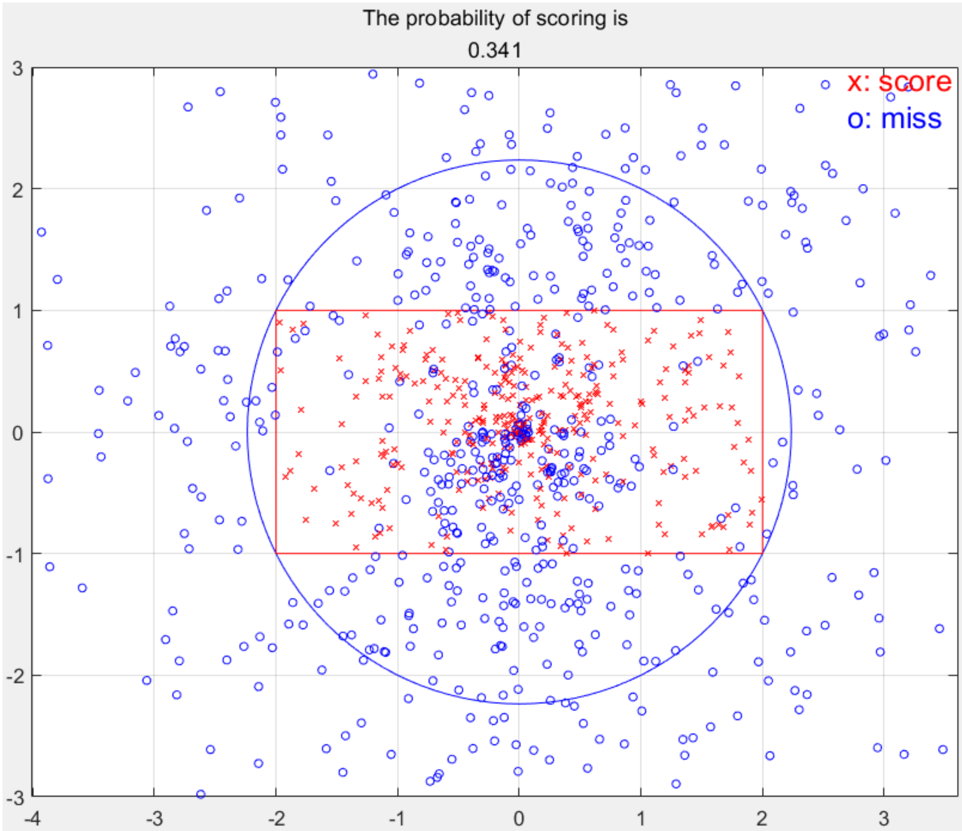

Fig.14. With goalkeeper (1000 Gaussian distributions)

- **Task 10:** If the goalkeeper pounces on the two low corners in 90% of the cases, the probability of scoring from 100 and 1000 penalty kicks is 0.43 and 0.389 respectively according to the simulation. When replacing the random distribution with a Gaussian distribution, the probabilities of scoring 100 and 1000 penalties are: 0.35 and 0.351. Details are shown in the diagram below. The Gaussian distribution has a further reduced probability of scoring in this case, and many penalty shots will be held due to the nature of the Gaussian distribution being concentrated near the centre. However, an even distribution of penalty kicks will further increase the probability of scoring. This is because in most cases the top half of the goal will be scored.

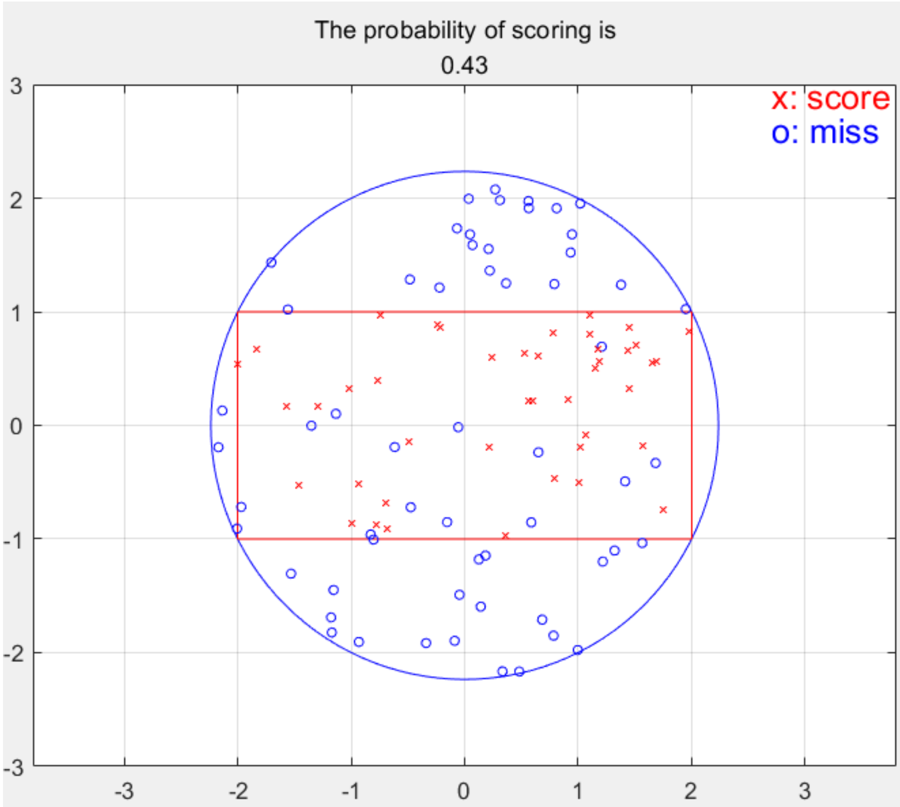

Fig.15. With goalkeeper and preferences (100 uniform distributions)

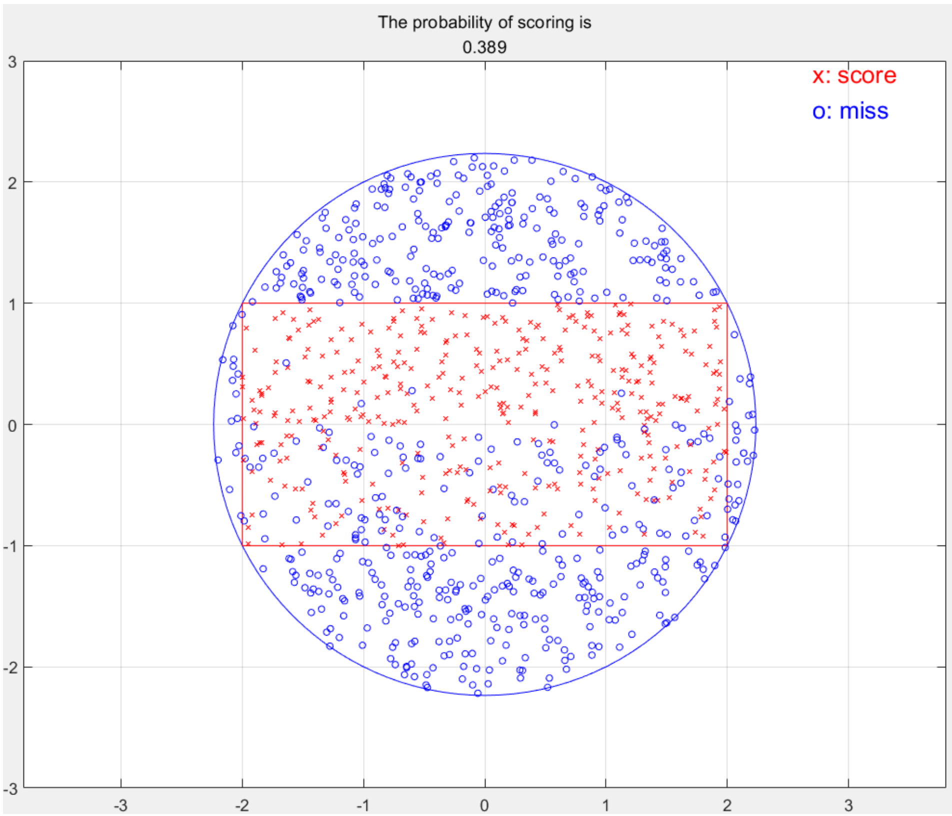

Fig.16. With goalkeeper and preferences (1000 uniform distributions)

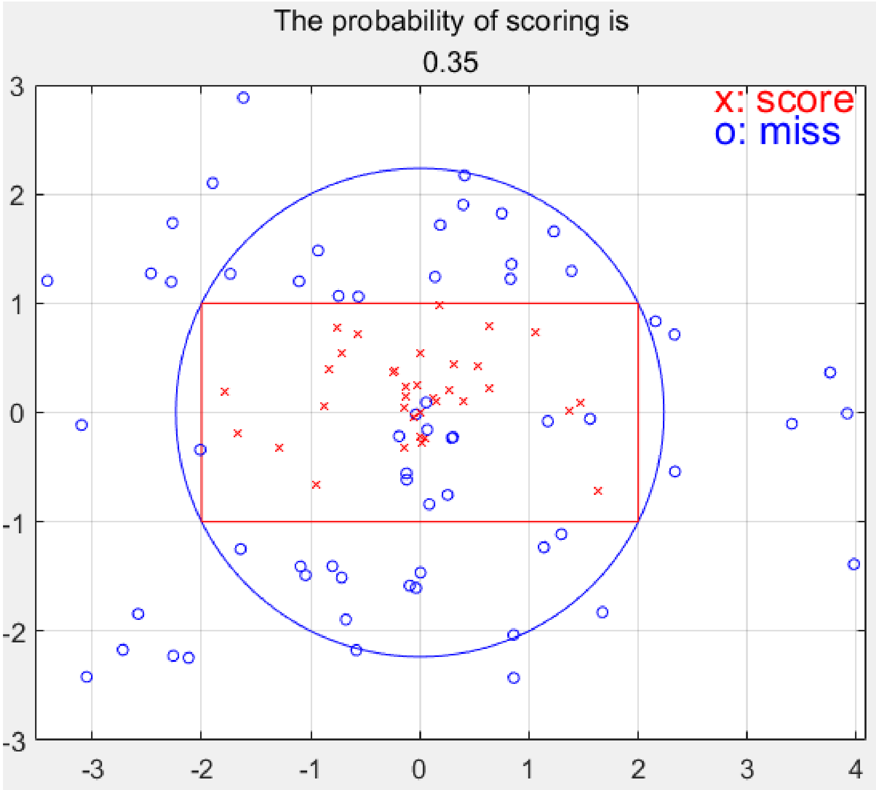

Fig.17. With goalkeeper and preferences (100 Gaussian distributions)

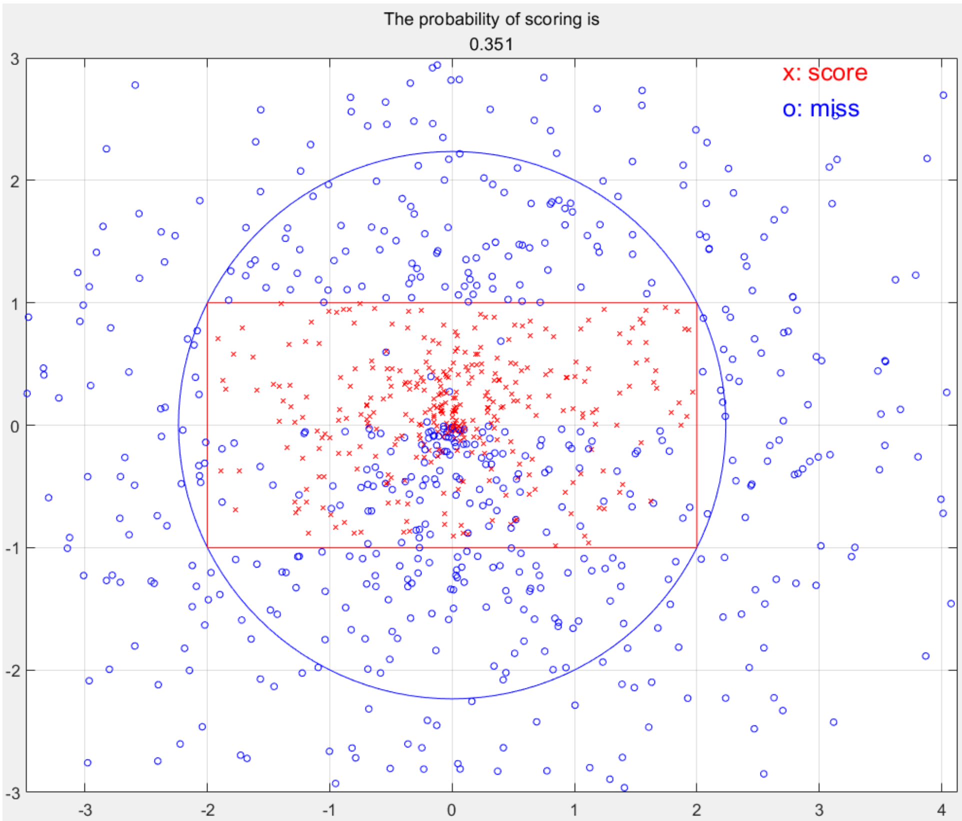

Fig.18. With goalkeeper and preferences (1000 Gaussian distributions)

4 Review Questions

- **Q1:** In terms of what you've done in this experiment, comment on the advantages and disadvantages (or drawbacks) of the Monte Carlo experiment.

  Ans: This experiment is very vivid and provides a quick understanding of the basic principles of the Monte Carlo model. However, this experiment also has many shortcomings, for example, the model is very crude and not realistic. The number of simulations is not large enough, resulting in a large error in each simulation. There is also no practical point in designing for the number of repetitions and the number of penalties, as the computer's pseudo-random numbers use the same seed, whether the experiment is repeated or the number of penalties is increased.

- **Q2:** Discuss the ways in which the above model could be made more accurate and realistic.

  Ans: Increase the number of penalties simulated. However, scatter plots do not need to simulate too many penalty kicks, as they are only used to aid understanding, but increasing the number of penalty kicks simulated for statistical probability can greatly improve accuracy. Changing the model might improve the realism of the experiment. This model is more different from the real situation because the area under the goal is ground level, so there is no situation where a penalty kick would go that way.

- **Q3:** With reference to Task-7 and Task-9, discuss the effect of changing the standard deviation of the Gaussian distribution on both the accuracy and precision of the penalty shots.

Ans: Due to the properties of the Gaussian distribution, there is about a 68% probability of falling within a circle with a standard deviation of the radius, and the probability increases the further you go to the centre and decreases the further you go to the edge. Due to this property, if the standard deviation is smaller, the probability of scoring a goal increase somewhat. However, because of the presence of a goalkeeper and the fact that there is more goalkeeping close to the centre, the accuracy will not increase much, but the precision will increase considerably.

- **Q4:** If a large number of balls are kicked on the goal (i.e., if N is sufficiently large), the value of π can be estimated using (some function of) the ratio of the number of scores to the total number of the shots. Hence, find the relation that estimates the value of π. Verify this using your results for both uniform and Gaussian distributions.

  Ans: To estimate the value of π, it is easier to use a uniform distribution. You can design a square with an inner tangent circle inside it and use the uniform distribution to model the probability of falling in the circle and multiply this probability by the area of the square to get the area of the circle. Since then, it is easy to derive the probability that the value of pi is four times p. After MATLAB simulation the value of pi can be estimated as $4 \times p = 4 \times 0.78409 = \mathbf{3.13636}$. The accuracy of the results obtained is not high enough because only 100,000 simulations have been carried out; when the number of simulations is increased to a large order of magnitude, the accuracy will be much higher.

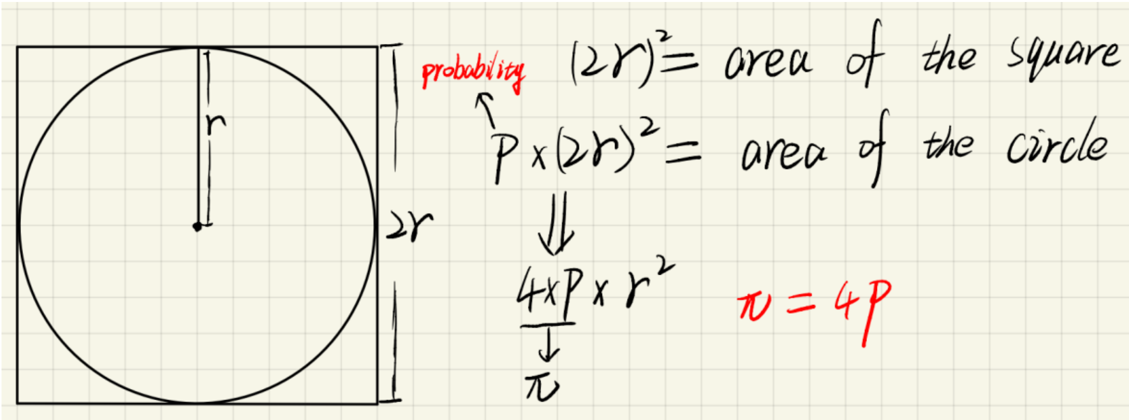

Fig.19. Derivation of π

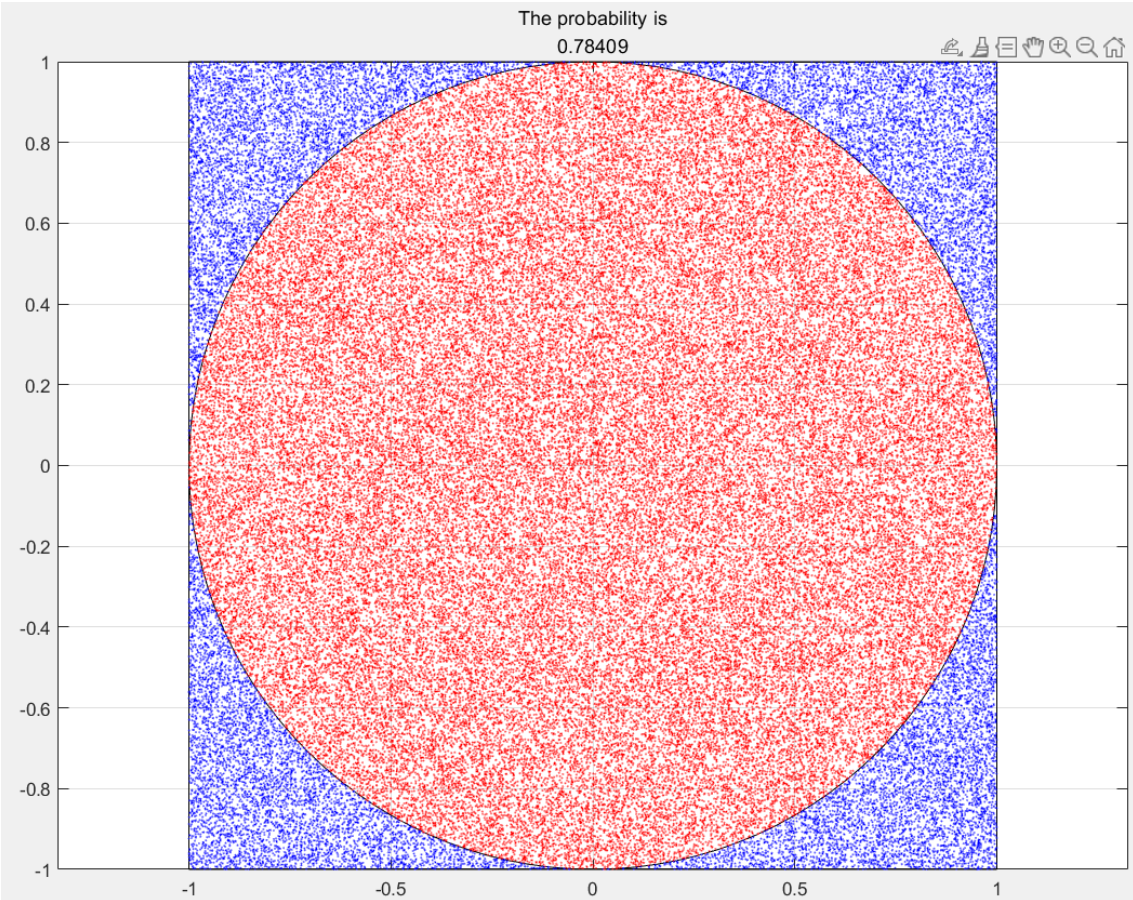

Fig.20. Uniform random distribution simulation

- **Q5:** From your observation and results of Part II, what is the best strategy that should be adopted by the penalty taker? What is the best strategy that should be adopted by the goalkeeper?

Ans: Based on the five goalkeeping situations in Part 2 for goalkeepers and their goalkeeping preferences, the easiest way to score a penalty kick is to shoot towards the two high corners. Therefore, the penalty taker should shoot towards the two high corners. Based on the evenly distributed shots and the Gaussian distributed shots, the goalkeeper should not favour the two low corners. The goalkeeper should adopt a preference for the first way of guarding the middle, which will significantly reduce the probability of scoring a goal.

6 Conclusions

This experiment uses MATLAB as a tool to simulate a solution to a real problem using Monte Carlo methods. In the report the probability of scoring a penalty kick is simulated through a uniform random distribution as well as a Gaussian random distribution and a probability close to the theoretical value is obtained. The value of π is also estimated by Monte Carlo methods after answering a number of questions. This experiment gave me a deep understanding of the core ideas of Monte Carlo techniques. It also improved my proficiency in using MATLAB.

7 References

[1] Garc´ıa-Fern´andez, A., 2022. *Experiment 22 – Monte Carlo Simulation.* [online] Canvas.liverpool.ac.uk. Available at: https://liverpool.instructure.com/courses/59021/pages/experiment-22?module_item_id=1543467

8 Appendix

Part 1 MATLAB code:

```
functions = input("1:Scatter Plot, 2:Probability Statistics\nSelect:");
distribution = input('Select the random distribution type\n(1 Uniform distribution, 2
Gaussian distribution):');
%This is a highly automated program, so there will be more input
if functions == 1
    shots = input('Please enter the number of shots N=');
    scatterPlot(shots,distribution);
    %If you want to draw a scatter plot, you only need the number of spot shots and call
the drawing function
else
    %The following sections are used to generate probability line graphs
    typeStatistics = input('Select data type for statistics\n(1 shots, 2 repeats):');
    count = [0,0,0,0];
    probability = [0,0,0,0];
    if typeStatistics == 1
        %Take the number of spot shots as the x label
        repeats = input('Please enter the repetitions R=');
        for i = 1:4
            shots = input('Please enter the number of shots N=');
            Y = probabilityStatistics(shots, repeats, distribution);
            probability(i) = Y;
            count(i) = shots;
            %Record the probability and input to facilitate the production coordinate
axis
            hold on;
        end
        xlabel('Random Penalty Shots');
    else
        %Take the number of repeatitions as the x label
        shots = input('Please enter the number of shots N=');
        for i = 1:4
            repeats = input('Please enter the repetitions R=');
            Y = probabilityStatistics(shots, repeats, distribution);
            probability(i) = Y;
            count(i) = repeats;
            hold on;
        end
        xlabel('Repetitions');
    end
    plot(1:4,probability,'.-b');
```

```
    axis([0,5,0.47,0.53]);
    grid on;
    set(gca,'XTick',0:1:5);
    set(gca,'YTick',0.47:0.005:0.53);
    set(gca,'XTicklabel',{0,count(1),count(2),count(3),count(4),' '});
    ylabel('Probability');
    text(1:4,probability-0.003,num2str(probability.','%.4f'))
    hold off;
    %Some operations for coordinates
end

function [] = scatterPlot(shots,distribution)
%This function is used to plot a scatter plot
t=0:pi/60:pi*2;
x=sqrt(5)*cos(t);
y=sqrt(5)*sin(t);
plot(x,y,'-b');
%Draw a circle using parametric equations
hold on;
axis([-3, 3, -3, 3]);
axis equal;
grid on;
x=-2;
y=-1;
w=4;
h=2;
rectangle('Position',[x,y,w,h],'edgecolor','r');
%Draw a rectangle
score=0;
for i=0:shots
    t=rand(1)*2*pi;
    if distribution==1
        r=sqrt(rand(1))*sqrt(5);
    else
        r=randn(1)*sqrt(5);
    end
    %Select different distribution types
    x=r*cos(t);
    y=r*sin(t);
    if x<2&&x>-2&&y<1&&y>-1
        score=score+1;
        plot(x,y,'rx','MarkerSize',4);
    else
        plot(x,y,'bo','MarkerSize',4);
```

```
    end
    %Select different distribution types
end
p=score/shots;
title('The probability of scoring is', p);
text(2.7, 2.9, 'x: score', Color='r', FontSize=15);
text(2.7, 2.6, 'o: miss', Color='b', FontSize=15);
hold off;
end

function Y = probabilityStatistics(shots, repeats, distribution)
%This function is used to plot a statistical graph
P=0;
for i=1:repeats
    score=0;
    for j=0:shots
        t=rand(1)*2*pi;
        if distribution==1
            r=sqrt(rand(1))*sqrt(5);
        else
            r=randn(1)*sqrt(5);
        end
        %Select Distribution Type
        x=r*cos(t);
        y=r*sin(t);
        %Parametric equation
        if x<2&&x>-2&&y<1&&y>-1
            score=score+1;
        end
    end
    p=score/shots;
    P=P+p;
    %Obtaining probability
end
Y=P/repeats;
end
```

Part 2 MATLAB code:

```
shots = input('Please enter the number of shots N=');
distribution = input('Select the random distribution type\n(1 Uniform distribution, 2
Gaussian distribution):');
defensive = input('Choose a defensive preference\n(1.Standard, 2.Corners):');
%Highly intelligent program with multiple input ports
t=0:pi/60:pi*2;
x=sqrt(5)*cos(t);
y=sqrt(5)*sin(t);
plot(x,y,'-b');
%Draw Circle
hold on;
axis([-3, 3, -3, 3]);
axis equal;
grid on;
x=-2;
y=-1;
w=4;
h=2;
rectangle('Position',[x,y,w,h],'edgecolor','r');
%draw rectangle
score=0;
for i=0:shots
    t=rand(1)*2*pi;
    if distribution == 1
        r=sqrt(rand(1))*sqrt(5);
    else
        r=randn(1)*sqrt(5);
    end
    %Select Distribution Type
    x=r*cos(t);
    y=r*sin(t);
    if defensive == 1
        p=randi(5);
    else
        shoots=randi(100);
        if shoots<11
            p=randi(3);
        else
            p=randi(4:5);
        end
        %Use this method to approximate 90% probability
    end
```

```
    %Choose a gatekeeper preference
    if x > 2 || x < -2 || y > 1 || y < -1
        plot(x,y,'bo','MarkerSize',4);
    else
        switch p
            case 1
                if x<1&&x>-1&&y<1&&y>-1
                    plot(x,y,'bo','MarkerSize',4);
                else
                    score=score+1;
                    plot(x,y,'rx','MarkerSize',4);
                end
            case 2
                if x<-1&&x>-2&&y<1&&y>0||x<0&&x>-1&&y<0&&y>-1
                    plot(x,y,'bo','MarkerSize',4);
                else
                    score=score+1;
                    plot(x,y,'rx','MarkerSize',4);
                end
            case 3
                if x<2&&x>1&&y<1&&y>0||x<1&&x>0&&y<0&&y>-1
                    plot(x,y,'bo','MarkerSize',4);
                else
                    score=score+1;
                    plot(x,y,'rx','MarkerSize',4);
                end
            case 4
                if x < 2 && x > 0 && y < 0 && y > -1
                    plot(x,y,'bo','MarkerSize',4);
                else
                    score=score+1;
                    plot(x,y,'rx','MarkerSize',4);
                end
            case 5
                if x < 0 && x > -2 && y < 0 && y > -1
                    plot(x,y,'bo','MarkerSize',4);
                else
                    score=score+1;
                    plot(x,y,'rx','MarkerSize',4);
                end
                %Conditions of each type of gatekeeper
        end
    end
end
```

```
p=score/shots;
%Obtaining probability
title('The probability of scoring is', p);
text(2.7, 2.9, 'x: score', Color='r', FontSize=15);
text(2.7, 2.6, 'o: miss', Color='b', FontSize=15);
hold off;
```

MATLAB code for estimating π:

```
n=input('N=');
t=0:pi/60:pi*2;
x=cos(t);
y=sin(t);
plot(x,y,'-k');
%Draw Circle
hold on;
x=-1;
y=-1;
w=2;
h=2;
rectangle('Position',[x,y,w,h],'edgecolor','k');
%draw rectangle
count=0;
for i=0:n
    x=rand(1);
    y=rand(1);
    %Use uniform distribution
    if x*x+y*y<1
        count=count+1;
        plot(x,y,'.r','MarkerSize',1);
    else
        plot(x,y,'.b','MarkerSize',1);
    end
    %Draw scatter points and get record probability
end
p=count/n;
title('The probability is', p);
axis([-1, 1, -1, 1]);
axis equal;
grid on;
hold off;
```