Experiment 22 - Monte Carlo Simulation

November 5, 2021

Abstract

Monte Carlo method is widely used to simulate and solve statistical problems in reality. This report will use six functions to solve the problem of penalty kick probability. Each part of functions were explained in detail. Among them, uniform and normal random generators are used to evaluate the situation with and without goalkeeper. The 10 tasks and all the review questions were well completed, and the results of different tasks were also compared and analyzed. Through the results of the experiment, some important conclusions have also been summarized, including the differences between uniform and normal distribution; the comparison between no goalkeeper and with goalkeeper; the influence of R and N on the simulated probability. All functions used in the report and simulated figures obtained from experiments will be provided in the appendix.

Contents

1 Introduction

In this experiment, Monte Carlo techniques will be used to find solutions to random processes. Matlab is the simulation software in this experiment, However, Matlab is not the learning goal of this experiment. The purpose of this experiment is to test Monte Carlo techniques in simulating and finding specific results to random processes [1]. In this experiment, uniform and normal (Gaussian) random number generator are both used. This experiment is divided into two situations: with goalkeeper and without goalkeeper. Next, theoretical background, theoretical analysis and experimental objectives will be explained in the introduction part.

1.1 Theoretical Background

In this part, the theoretical background of Monte Carlo and Matlab will be introduced including the application of Monte Carlo in the field of mathematics. Also, in this section, the lab requirements will be restated.

1.1.1 The background of Monte Carlo

1. **The introduction of Monte Carlo**

   Monte Carlo (Monte Carlo method), also known as statistical simulation method, which is also a kind of numerical calculation method. Monte Carlo method is proposed in the mid-1940s under the guidance of probability and statistics theory [2]. Monte Carlo is a method of using random numbers to solve many computational problems. Moreover, Monte Carlo method is widely used in financial engineering, macroeconomics, computational physics (such as particle transport calculation, quantum thermodynamics calculation, aerodynamics calculation) and other fields [1]. A correct Monte Carlo experiment involves the following steps [1]:

   (a) Draw (pseudo) random samples with size N from the probability distribution functions of the random model.
   (b) Assume the values of each part of the model, or get the values from their respective distribution functions.
   (c) Calculate each part of the statistical model.
   (d) Calculate the interesting value (e.g. valuation).
   (e) Repeat Step (1) to Step (4) for R times.
   (f) Test the empirical distribution of R value.

   When the Monte Carlo method is used to estimate the distribution of an estimator, the certain size N and the number of repeated experiments R need to be known. Based on these values, the sampling distribution can be approximately estimated.

2. **Specific applications about the Monte Carlo**

   The Monte Carlo method can be used in different scientific area including the Physical, Design, Finance, Telecommunications, Games. The followings are the specific applications of the Monte Carlo method [1]:

(a) **Physical Science**

Monte Carlo method can be used to estimate the exact values in the computational physics, physical chemistry fields.

(b) **Design the Computer graphics**

Monte Carlo can help us create virtual 3D models, and can also be applied to the field of special effects in movies.

(c) **Finance and business**

Monte Carlo can help people evaluate the correctness of financial models, and can help to improve the certainty and accuracy of financial models

(d) **Telecommunications**

Monte Carlo method can help to deal with the problems of the wireless network and test the probability of a network [3].

(e) **Games**

Monte Carlo can make games better and playable.

3. **The Practical Work**

In this experiment, making a simulation of Penalty kicks is required. Penalty kicks are a critical time, which may decide the winner, often need to let goalkeeper choose to jump to the left or the right. In the first part of the experiment, we will simulate the situation without the goalkeeper and analyze the situation of the goal. In the first part, the more realistic situation with goalkeepers will be considered. In both cases, the Uniform and Normal (Gaussian) random number generator will be considered. In the case of with goalkeeper and without goalkeeper, the schematic diagram is as follows:

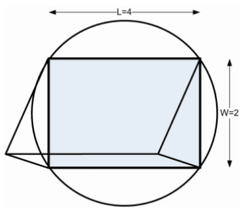


Figure 1: The goal arrangement(taken from [1])

In this experiment, it is considered that all shots will be within the area of circle with the radius: $\sqrt{5}$ and a rectangle which is with a length of 4 and a width of W=2. The rectangular area means goal is shot successfully. The circular area will represent the area where the whole range that soccer can be. Finally, the number of times that shot is in the rectangular area and the number of times shot is successfully in the circular area will be both counted and the specific ratio will also be calculated. In this experiment, the uniform and normal (Gaussian) random number generator will be used in different tasks.

In addition, considering goalkeepers and the tendency of goalkeepers to jump in different directions will be also considered. According to the requirements, the goalkeeper has the following 5 possible actions which can be seen in following diagram:

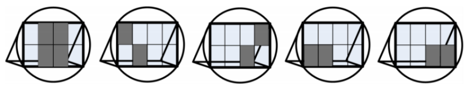


Figure 2: Five possible actions (taken from [1])

As can be seen from the picture, goalkeepers may have the following jump actions:

(a) stay in the middle
(b) jumps to the upper left corner
(c) jumps to the upper right corner
(d) jumps to the lower left corner
(e) jumps to the lower right corner

The uniform and normal (Gaussian) random number Generator are also used in different tasks. In particular, in Task 10, the fact that seconds has a 90% probability to jump to the lower two corners of the goal will be considered specially. In each case, many experiments will be done to reduce the errors. Because once the times is too small, it will lead to large errors in the experiment.

1.1.2 The background of Matlab and simulation

MATLAB is a business mathematics software produced by MathWorks. It is used in data analysis, wireless communication, deep learning, image processing and computer vision, signal processing, quantitative finance and risk management, robotics, control systems and other fields [4]. Due to the superior simulation function of Matlab, Matlab is chosen as the software using Monte Carlo method in this experiment. For each different task the corresponding code will be provided in the appendix.

1.2 Probability (Theoretical) analysis

Although this experiment is a simulation-based experiment, the theoretical probability of this experiment can be calculated according to mathematical methods. This part will conduct theoretical analysis, and the specific calculation results will be placed in the **result** part. In the result section, the calculated theoretical results and the results obtained from Matlab will be also compared and the reasons will be analyzed .

1. **Uniform distribution**

   In the case of uniform distribution of random numbers, the corresponding analytic method should be based on geometric method. In the process of shooting, random numbers are uniformly distributed, so the calculation method of experimental probability is as follows [5]:

$$P = \frac{S_{Rectangle}}{S_{Circle}} \tag{1}$$

2. **Normal distribution (radius obeys normal distribution)**

In this case, if it is assumed that only radius obeys the normal distribution while the angle obeys the uniform distribution in the polar system. The probability in this case will be:

$$P_x = \frac{1}{\sigma * \sqrt{2\pi}} * e^{\frac{-(x-\mu)}{2\sigma^2}} \tag{2}$$

3. **Normal distribution (The horizontal and vertical coordinates satisfy normal distribution, not required in this experiment)**

   In this case, the horizontal and vertical coordinates of the data satisfied the normal distribution. The corresponding analytical method is the integration of probability density based on normal distribution [6]:

$$P(x) = \frac{1}{2\pi * \sigma_x * \sigma_y * \sqrt{1-r^2}} e^{\frac{1}{2(1-r^2)} * [\frac{(x-\mu_x)^2}{\sigma_x^2} - 2r\frac{(x-\mu_x)(y-\mu_y)}{\sigma_x \sigma_y} + \frac{(y-\mu_y)^2}{\sigma_y^2}]} \tag{3}$$

1.3 Objectives

The main purpose of this experimental is to familiarize with the Monte Carlo method. Moreover, this simulation method can be put into practice by solving problem of penalty kick analysis, and the advantages and disadvantages of this method can be comprehensively understood by comparing the possibilities produced by two types of random number generators (uniform distribution and normal distribution). In addition, experimental simulation values and the theoretical values will be compared and the reasons will be analyzed to find the correlation between them. In the last, the differences produced by different values of N and R will be evaluated in detail.

2 Materials and methods

2.1 Apparatus

In this experiment, Matlab was the simulation software by evaluating the Monte Carlo method. In this experiment, the Matlab 2020 version will be used. All the functions will be saved in the ".m" file format. When changing the value of N and R, it is necessary to enter a specific value in the command window.

2.2 No Goalkeeper Tests

In this experiment, No Goalkeeper was considered. The following functions were designed to solve Task1-7. Among them, the function 1 is designed for task2-5. And Function 2 was designed for Task 7 . Moreover, the results of each task will be displayed and explained in detail in the result part,.

2.2.1 Methods and procedure

1. Function 1: The uniform generator

   The function 1 used a uniform random generator.

First, according to the requirements of the task, the following experimental variables were defined:

- P: The final possibilities of shooting in the rectangle area
- R: is the times that repeat this experiment
- N: N means N random penalty shots
- T: T means the number of shooting in the rectangle area in one experiment
- x: The horizontal x-coordinate
- y: The vertical y-coordinate
- radius: The radius coordinate
- angle: The angle coordinate

Next, a circular area and a rectangular area were drawn. The following experiment would be carried out in the circular area. The following was the arrangement for drawing the figure:

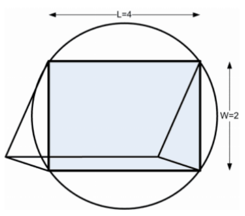
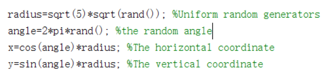


Figure 3: The area of rectangle and circle

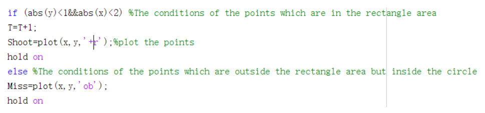
The following code is the specific algorithm code. In this function, the polar coordinate system was converted to the x-y coordinate system. The specific code was as follows:



```
radius=sqrt(5)*sqrt(rand()); %Uniform random generators
angle=2*pi*rand(); %the random angle
x=cos(angle)*radius; %The horizontal coordinate
y=sin(angle)*radius; %The vertical coordinate
```

Figure 4: The specific algorithm code

In the first function, a uniform random generator was used. In the judgment of the position of the points, the following codes would be used:

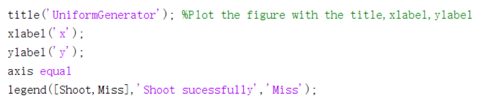


```
if (abs(y)<1&&abs(x)<2) %The conditions of the points which are in the rectangle area
T=T+1;
Shoot=plot(x,y,'+r');%plot the points
hold on
else %The conditions of the points which are outside the rectangle area but inside the circle
Miss=plot(x,y,'ob');
hold on
```

Figure 5: The condition that determines whether a point was in the rectangle

The x and y coordinates were used to judge whether the point is in the rectangle. If this point was in the rectangle, the value of S would plus 1. Finally, by calculating the value of S/N, the probability of a single experiment could be known.

The last part was the drawing the figure. In the first function, the red crosses was used to indicate score and blue circles to indicate miss. At the same time, the horizontal and vertical coordinates in the figure would be also indicated.



```
title('UniformGenerator'); %Plot the figure with the title,xlabel,ylabel
xlabel('x');
ylabel('y');
axis equal
legend([Shoot,Miss],'Shoot sucessfully','Miss');
```

Figure 6: The code of drawing the figure

2. Function 2: normal (Gaussian) random number generator

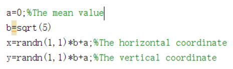
   The function was used for scenarios where there was no goalkeeper in the (a normal random number generator). This function would be used in Task 7. This function was established by the following parts:

   First, the variables used in programming will be declared, each of the different variables would be introduced below:

   - P is the final possibilities of shooting in the rectangle area
   - R is the times that repeat this experiment
   - N means N random penalty shots
   - T means the number of shooting in the rectangle area in one experiment
   - x is the horizontal x-coordinate
   - y is the vertical y-coordinate
   - COUT means the points which set outside the circle
   - Total means total shots in one experiment
   - a means the mean value (In this experiment, a is 0)

   Next, the area of circular and rectangular needed to be drawn, which was same as the uniform random generator.

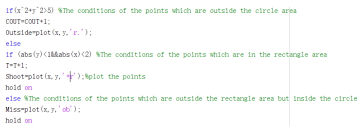
   Moreover, the algorithm code would be shown. In this part, the function 'randn()' was used to achieve normal (Gaussian) random number generator [7].



```
a=0;%The mean value
b=sqrt(5)
x=randn(1,1)*b+a;%The horizontal coordinate
y=randn(1,1)*b+a;%The vertical coordinate
```

Figure 7: The code of normal generator

Finally, it is necessary to determine whether the point was in the **rectangular area, in the circular area outside the rectangular area, or outside the circular area**. Because this algorithm used a normal (Gaussian) random number generator, which meant some points would appear outside the circle. After Dr. Uney's explanation, the points in this part also needed to be considered [7]. The specific judgment conditions would be listed, including the above three situations:



```
if(x^2+y^2>5) %The conditions of the points which are outside the circle area
COUT=COUT+1;
Outside=plot(x,y,'r.');
else
if (abs(y)<1&&abs(x)<2) %The conditions of the points which are in the rectangle area
T=T+1;
Shoot=plot(x,y,'+r');%plot the points
hold on
else %The conditions of the points which are outside the rectangle area but inside the circle
Miss=plot(x,y,'ob');
hold on
```

Figure 8: The condition that determines whether a point was in the rectangle

2.3 With Goalkeeper Tests

2.3.1 Methods and procedure

1. **Function 3: The uniform generator for part I**

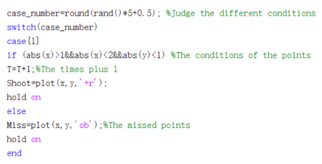
   This function was used for the situation where the goalkeeper protected the area in different positions. This situation was based on a uniformly distribution(uniform random number generator). This function would be used for task 8. Similarly, some variables needed to be defined, too:

   (a) P: the final possibilities of shooting in the rectangle area
   (b) R is the times that repeat this experiment
   (c) N means N random penalty shots
   (d) T means the number of shooting in the rectangle area in one experiment
   (e) x: the horizontal x-coordinate
   (f) y: the vertical y-coordinate
   (g) radius: the radius coordinate
   (h) angle: the angle coordinate
   (i) Pe: each possibility in every experiment

   As with no goalkeeper, the area of rectangle and circle would be needed,too.

   Similarly, for the uniform random generator, the codes used were the same as in the case of no goalkeeper.

   Next, the goalkeeper situations needed to be considered. Due to the goalkeeper action was modelled as a uniform random process [1], a uniform random generator also needed to implement the goalkeeper condition. Therefore, the function 'switch() would be used to select situations. The specific code was as follows (here only took the first situation as an example), other examples could be viewed in the appendix:



```
case_number=round(rand()*5+0.5); %Judge the different conditions
switch(case_number)
case{1}
if (abs(x)>1&&abs(x)<2&&abs(y)<1) %The conditions of the points
T=T+1;%The times plus 1
Shoot=plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');%The missed points
hold on
end
```

   Figure 9: The condition that determines whether a point is in the rectangle

   Next, the corresponding figure would be drawn,too.

2. **Function 4: Normal Generator for part II**

   First, the following variables were declared:

   (a) P is the final possibilities of shooting in the rectangle area
   (b) R is the times that repeat this experiment
   (c) N means N random penalty shots
   (d) T means the number of shooting in the rectangle area in one experiment
   (e) x is the horizontal x-coordinate

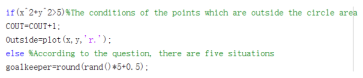
(f) y is the vertical y-coordinate

(g) COUT means the points which set outside the circle

(h) Total means total shots in one experiment

(i) a means the mean value (In this experiment, a is 0)

Next, the rectangular and circular would also be needed. Unlike the uniform generator, the Normal generator was used this time. Since the normal generator would cause the points to appear outside the circle, these points still needed to be considered with **red dot** in this experiment. The specific judgment conditions were as follows:



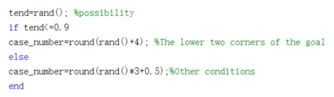
```
if(x^2+y^2>5)%The conditions of the points which are outside the circle area
COUT=COUT+1;
Outside=plot(x,y,'r.');
else %According to the question, there are five situations
goalkeeper=round(rand()*5+0.5);
```

Figure 10: The condition that determined whether a point was outside the circle

Like the uniform random generator, five cases were considered separately. The specific judgment conditions remained unchanged. When drawing the picture, the points outside the circle were indicated by red dots.

3. **Function 5: Uniform Generator(90%)**

   The difference from other uniform generators was that the probability of the goalkeeper's action was also be considered. The goalkeeper has a 90% probability to jump to the lower two corners of the goal. Therefore, the following code was for this situation:



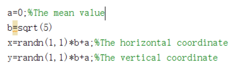
```
tend=rand(); %possibility
if tend<=0.9
case_number=round(rand()+4); %The lower two corners of the goal
else
case_number=round(rand()*3+0.5);%Other conditions
end
```

Figure 11: The designed tend codes

The code of the five conditions remained unchanged. If the random number obtained was between 0-0.9, the fourth and fifth cases would be selected, and the fourth and fifth cases would be selected. **(The lower two corners)**

4. **Function 6: Normal Generator(90%)**

   In this part, it is necessary to design a normal generator that considers probability(90%). In this part, the code of the Normal generator would be used, too:



```
a=0;%The mean value
b=sqrt(5)
x=randn(1,1)*b+a;%The horizontal coordinate
y=randn(1,1)*b+a;%The vertical coordinate
```

Figure 12: The code of normal generator

3 Results

3.1 No Goalkeeper Tests

In this part, various situations are considered without a goalkeeper, including the design of a uniform/Normal generator to achieve various mission requirements. The code in this part includes Function 1 and 2. The detailed code can be viewed in the appendix.

3.1.1 Task One

Assuming that the penalty kicker is blindfolded, it is needed to calculate the probability that the penalty kicker will score a goal (the range within the rectangular area where the ball appears). According to task requirements, the shots are distributed uniformly within the area of the circle[1].

First, it is necessary to calculate the area of the circle. If it is assumed that the diameter of the circle is D, then D:

$$D = \sqrt{4^2 + 2^2} = 2\sqrt{5} \tag{4}$$

Through the above formula, it is right to consider that the diameter of the circle is $2\sqrt{5}$. Therefore, the radius of the circle is $\sqrt{5}$. The area of the circle is:

$$S_{cir} = \pi * r^2 = \pi * \sqrt{5}^2 = 5 * \pi \tag{5}$$

The area of the rectangle is:

$$S_{rect} = W * L = 8 \tag{6}$$

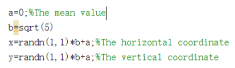
Therefore, the possibility can be calculated by the following formula:

$$P = \frac{S_{rect}}{S_{cir}} = \frac{8}{5\pi} \approx 0.509 \tag{7}$$

In the process of calculating the probability, the area of the rectangle and the area of the circle are used. Since each point is randomly distributed, the theoretical probability can be calculated in the case of uniform distribution through their ratio.

3.1.2 Task Two

In this task, it is necessary to design a simulation program. In the design of the program, the algorithm is:



```
a=0;%The mean value
b=sqrt(5)
x=randn(1,1)*b+a;%The horizontal coordinate
y=randn(1,1)*b+a;%The vertical coordinate
```

Figure 13: The uniform random generator

Since the range is in a circular area, the coordinates of the points need to be got firstly in the polar system, then converting the points which are in the polar system into x-y coordinates. The specific code will be in the appendix. For detailed information, please refer to Function 2.

3.1.3 Task Three

In this task, it is necessary to make N=1000 and R=1. And, if the point is in the rectangular area, red crosses will be used to mark, and use blue circles as the sign of miss for other points.

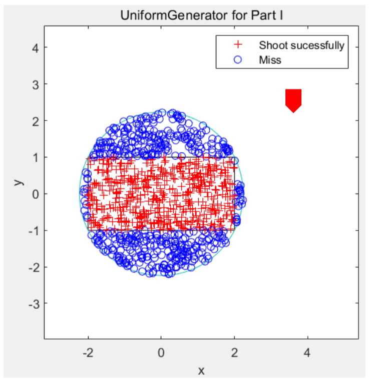


Figure 14: The uniform random generator for N=1000 R=1

Similarly, after calculation, the probability simulated by Matlab is 0.4950 which is very close to the theoretically calculated value(0.509).The reason for the error may be because the sample size is too small, increase the sample size (N) can reduce the error. This proves that the simulation is as expected.

3.1.4 Task Four

In this task, five situations are simulated. They are: R=5, N=100; R=5, N=1000; R=5,N=10000; R=5, N=100000. Due to the limited space of this part, the specific figures will be placed in the appendix. Only the final results are listed here, and the specific probabilities are as follows:

Table 1: The table for Task 4 when R=5

N	100	1000	10000	100000
P	0.4780	0.5128	0.5099	0.5111

With the increase of N, the probability tends to close to the calculated theoretical value(0.509). When the value of N is **large**, the error between the simulated result and the theoretical value is small. This proves that when increasing the value of N, the error of the simulation result will also decrease. After analysis, the line chart drawn is as follows (the abscissa is N, the ordinate is P when R=5):

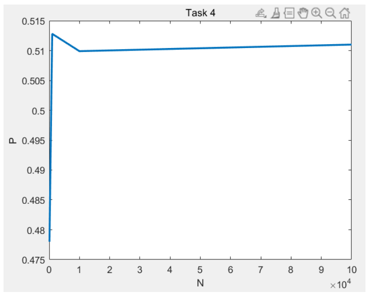


Figure 15: The figure for task 4 results when R=5

As N increases, P first increases and then decreases by analyzing the line chart and finally stays near P=0.509 (the theoretical value calculated is 0.509). Therefore, it can be found that as N increases, the probability will get closer and closer to the theoretical value.

3.1.5 Task Fifth

In this task, the value of R need to be changed and keep N=1000 unchanged. Therefore, the following situations need to be simulated: R=5, N=1000; R=10, N=1000; R=15, N=1000; R=20, N=1000. Only the final results are listed here. The specific probability is as follows:

Table 2: The table for Task 5 when N=1000

R	5	10	15	20
P	0.5110	0.5085	0.5081	0.5082

Through the table, it can be founded that as R increases, the value does not change much.

The line graph drawn is as follows (the abscissa is N, the ordinate is P):

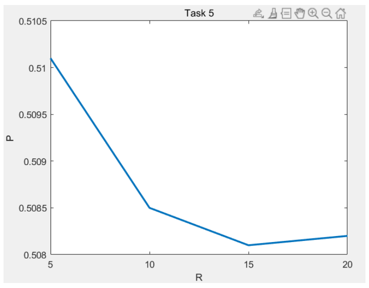


Figure 16: The figure for task 5 results when N=1000.PNG

By analyzing the line chart, it can be founded that as R increases, P first decreases and then increases but the range of changing is small (the theoretical value we calculated is 0.509). Therefore, as R increases, the precision of the experimental results will increase. **However, the accuracy of the results will not increase accordingly.**

3.1.6 Task Six

In this task, the results of Task 4 and Task 5 will be compared. By comparison, it can be founded that when the value of size : N increases, the accuracy of the probability will also increase. However, when the value of R increases, the accuracy does not improve significantly. Increasing R will only improve the precision of the data but not the accuracy. The following is a specific explanation:

When the experimental time remains constant and the sample size (number of shots) increases, the simulation results will be more **accurate**. Since this is a random process, the results can be more accuracy, which is closer to the theoretical possibility.

However, when the sample size (number of shots:N) remains constant, only the experimental time R increases. The results are not very close to the real situation. Increasing the time: R only improves the precision of the results. Therefore, it can be found that the result in Task 5 is closer to the condition of R=1 and N=1000, rather than the theoretical value.

Therefore, the summary is that the accuracy of the results can be improved by increasing the value of N. If the simulation times are infinite, the possibility can be considered equal to the real results. However, if only R increases, the precision can be improved without affecting the accuracy.

3.1.7 Task Seven

In this part, it is required to repeat the experiment above with a normal random generator. The results of the experiment are as follows:

1. **Repeat Task 2**

   This part needs to use the normal random generator. The introduction of the algorithm part can be viewed in the previous part (the entire code will be placed in the appendix: function 2)

2. **Repeat Task 3**

   In this task, it is needed to make N=1000 and R=1. And, if the point is in the rectangular area, the red crosses will be marked and blue circles will be the sign of miss for other points. The specific result is as follows:

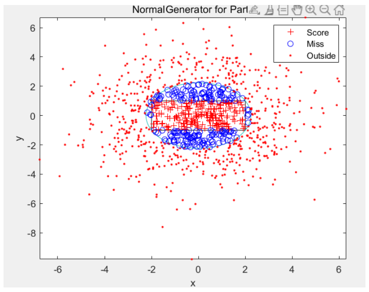
   

   Figure 17: The normal random generator for N=1000 R=1.PNG

   The simulation calculation probability: 0.2210. It can be summarized that possibility 0.2210 is not very close to the real possibility which was calculated in task 1 (0.509). The reason is that normal distributions tend to be dense in the center. In addition, since normally distributed points will appear outside the circle, the overall probability will be lower than that of a uniform generator.

3. **Repeat Task 4**

   In this task, five situations are simulated including: R=5, N=100; R=5, N=1000; R=5, N=10000; R=5, N=100000. Only final results are listed, and the specific probabilities are as follows:

   Table 3: The table for repeated Task 4

N	100	1000	10000	100000
P	0.2341	0.2098	0.2171	0.2175

   Through the table, it can be founded that with the increase of N, the probability tends to the calculated theoretical value. When N is large, the error between the simulated result and the theoretical value is small. This proves that when increasing the size of N, the error of the simulation result will also decrease.

   The line graph drawn is as follows (the abscissa is N, the ordinate is P):

   By analyzing the line chart, it can be found that as N increases, P first drops and then rises, and finally stays near P=0.2175.

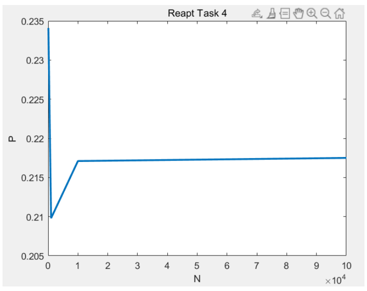


Figure 18: The possibility of repeated Task 4

4. **Repeat Task 5**

   In this task, only five situations are used, which are: R=5, N=1000; R=10, N=1000; R=15,N=1000; R=20, N=1000. The specific probabilities are as follows:

   Table 4: The table for repeated Task 5

R	5	10	15	20
P	0.2160	0.2185	0.2214	0.2223

   Through the table, it can be founded that as R increases, the result does not change much and the value is basically stable around 0.22.

   The line graph drawn is as follows (the abscissa is N, the ordinate is R):

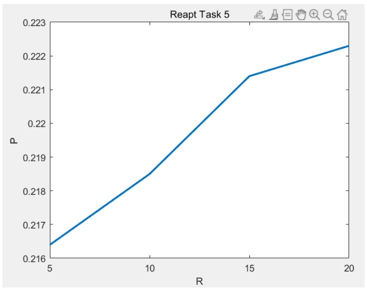


Figure 19: The possibility of repeated Task 5

   As R increases, P will also increase, but the range is not large, and finally stays near P=0.22 by analyzing the line graph. Therefore, it can be founded that as R increases, the precision of the result will increase. However, the accuracy of the experimental results will not increase accordingly.

5. **Repeat Task 6**

   Through the results of repeated task 4, it can be founded that when N is very large, the simulated value is about 0.2175. Since the value of N is very large, it can be assumed as the real result. Compared with a uniform random generator, this probability is relatively small. This is because the normal distribution will make the points appear outside the circle, which may reduce the probability.

   Comparing the results in task 5 and repeated task 5, the gap between the probabilities is not very large, which is similar to the conclusion in uniform generator. As R increases, the precision of the experimental results will increase. However, the accuracy of the experimental results will not increase accordingly. The following are the reasons for the analysis:

- When the sample size (number of shots) remains constant, only the experimental times R increases. The results are not very close to the real situation. Increasing the times R only improves the precision of the results.
- Compared with the uniform random generator, it can be founded that when using the normal generator, the points are more distributed in the **center** because the normal generator will cause the points tend to gather in the center instead of random uniform distribution. In addition, using the normal generator will definitely cause not all points to be distributed in the circle. The outside points cause the probability in this case to be lower than that of the uniform generator.

3.2 With Goalkeeper Tests

In this part, the situation where there is a goalkeeper will be discussed. And the possible actions of the goalkeeper are divided into five, they are the goalkeeper stays in the middle, jumps to the upper left corner, jumps to the upper right corner, jumps to the lower left corner or jumps to the lower right corner. The probability of goalkeeper action will also be considered. This part will list the results required by each task. The specific code can be viewed in the appendix.

3.2.1 Task Eight

In this part, the case of a goalkeeper and experiment with a uniform random generator will be considered. The results of this experiment will be compared with the situation without a goalkeeper. The specific results are as follows:

Table 5: The simulated possibilities for Task 8

N	100	1000
P	0.360	0.376

The following is the result of a simulation experiment when R=1 and N=100.

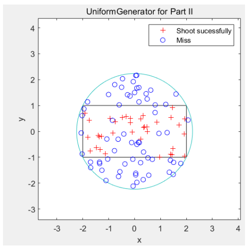


Figure 20: The possibility for Task 8 when N=100

After simulation, the probability in this case is 0.360.

The following is the result of a simulation experiment when R=1 and N=1000.

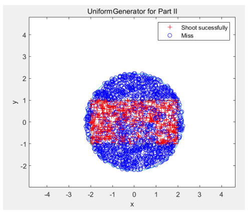


Figure 21: The possibility for Task 8 when N=1000

After simulation, the probability in this case is 0.376. After the above analysis, when the value of size : N is relatively large, the simulation possibility will be close to the theoretical value.

1. **Compare the results in Task 1**

   Since the position of the goalkeeper is not fixed, it is necessary to consider the possible actions of the goalkeeper. And in Task 8, it can be considered that the probability of every possible action of the goalkeeper is uniform distributed, which means the probability of every action is equal:

   $$P_1 = P_2 = P_3 = P_4 = P_5 = 0.2 \tag{8}$$

   Although the goalkeeper has five possible actions, this will only produce two probabilities p=0.5 and P=0.75. Therefore, the total probability can be calculated as follows:

   $$P = \frac{P_1 * S_{Rect} * \frac{1}{2} + P_2 * \frac{3}{4} * S_{Rect} * 4}{S_{Circle}} = \frac{(0.2 * 0.5 * 8 + 0.2 * 0.75 * 8 * 4)}{5 * \pi} \approx 0.357 \tag{9}$$

   Although there are five situations to consider, but the probability of four possible actions is equal. Therefore, the theoretical value can be calculated by considering only two probabilities.The theoretical value of this case is approximately 0.357 which is very close to the experiment value.

2. **Compare the results in Task 3**

   In this part, the specific results will be compared with the results of Task 3. The probability of Task 3 is 0.4950, and the probability of Task 8 is 0.357 under the same conditions: R=1,N=1000. It can be found that the successfully possibility of shooting with a goalkeeper is significantly less than the situation without the goalkeeper ($0.360 < 0.495$). Therefore, it can be summarized that the possibility of shooting successfully are reduced by the existence of goalkeepers. Most shots which are in the rectangular areas will score, as is similar to case in real life.

3.2.2 Task Nine

In this part, the normal random generator will be used to re-do the Task 8. The function can be viewed in Appendix Function 4. This section will only list specific results.

Table 6: The simulated possibility for Task 9

N	100	1000
P	0.120	0.137

When N=1000, the result is larger than when N=100. By comparing with the result of task 8 (uniform random generator), it can be seen that the probability of the normal distribution is less than the probability of the uniform. The specific reason is: some points of the normal distribution appear outside the circle, which reduces the probability.

The following is the result when N=100:

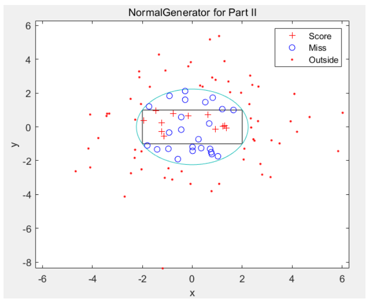


Figure 22: The possibility for Task 9 when N=100

The following is the result when N=1000:

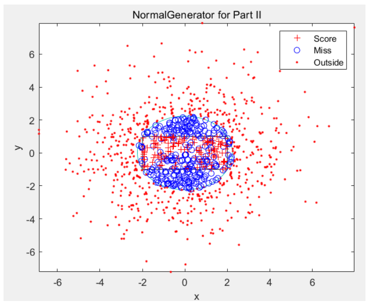


Figure 23: The possibility for Task 9 when N=1000

After simulation, the probability in this case is about 0.137. After the above analysis, when the size of N is very large, the simulation result will be very close to the theoretical value. Therefore, the possibility when N=1000 is more convincing. In the following part, the normal random generator will be used to simulate the situation with a goalkeeper. The repeated simulation experiment will be introduced below:

1. **Compare the results in Task 2(Normal)**

   Unlike Task 2, goalkeeper is considered in Task 9. Additionally, five different situations are considered separately. Codes for five different situations are provided. The specific code can be viewed in Appendix Function 4.

2. **Compare the results in Task 3(Normal)**

When using the normal random generator to complete Task 3, the probability is 0.2210. However, under the same conditions (R=1, N=1000), in Task 9, the probability is 0.137. The probability of having a goalkeeper is less than that of without a goalkeeper. This is in line with expectations. In addition, most points which are in the rectangular area are scored, which also conforms to the actual situations.

3. **Compare the results in Task 4(Normal)**

   In this part, the function was used to conduct four experiments. They are: R=5 N=100 and R=5 N=1000 and R=5 N=10000 and R=5 N=100000.

   The specific results are as follows:

   Table 7: The simulated possibilities for repeated Task 4

N	100	1000	10000	100000
P	0.1480	0.1534	0.1504	0.1502

   Compared with the situation without a goalkeeper, this probability is generally small.

   The line chart drawn is as follows:

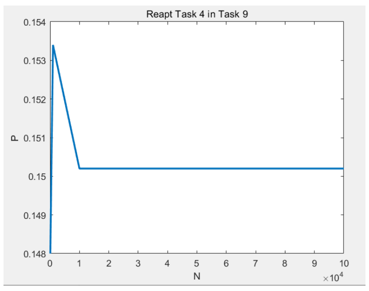


   Figure 24: The figure for Task 4 in Task 9 when R=5

   By comparison, the probability of considering the goalkeeper is less than the case of not considering the goalkeeper in the case of the normal random generator. When N is very large, the probability of having a goalkeeper is about 0.15. This possibility can also be approximately equal to the actual value.

4. Compare the results in Task 5(Normal)

   In this part, four different experiments were performed using Function 4 including: R=5 N=1000 and R=10 N=1000 and R=15 N=1000 and R=20 N=1000.

   The specific results are as follows:

   Table 8: The simulated possibilities for repeated Task 5

R	5	10	15	20
P	0.1572	0.1467	0.1478	0.1526

   Compared with the situation without a goalkeeper, the probability is generally small.

   The line chart drawn is as follows:

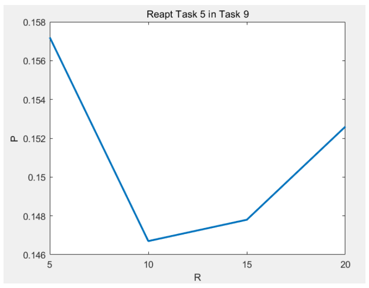


Figure 25: The figure for Task 5 in Task 9 when N=1000

By comparison, the probability of considering the goalkeeper is generally less than the case of not considering the goalkeeper in the case of the normal random generator. In addition, although the value of R is changing, the changes in probability is relatively small. This is in line with the conclusion.

5. **Compare the results in Task 6(Normal)**

   It's the same as without a goalkeeper. When increasing the value of N, the possibility is closer to the theoretical value. But if only increasing the value of R, the precision of the simulation results will increase, but the accuracy of the data will not be improved. Although the goalkeeper is considered, the probability is still very accurate when N is large. Therefore, the situation which considers the goalkeeper will only reduce the probability of a goal, and will not affect other aspects.

6. **Compare the results in Task 7(Normal)**

   When only changing the value of N, the situations with and without goalkeeper are as follows(Normal random):

Table 9: The results of task 9 and task 7

N	100	1000	10000	100000
P(Task 4)	0.2341	0.2098	0.2171	0.2175
P(Task 9)	0.1480	0.1534	0.1504	0.1502

When only changing the value of R, the situations with and without goalkeeper are as follows(Normal random):

Table 10: The results of task 9 and task 7

R	5	10	15	20
P(Task 5)	0.2160	0.2185	0.2214	0.2223
P(Task 9)	0.1572	0.1467	0.1478	0.1526

Because both experiments are using normal random generators, the results of Task 9 are only related to whether or not a goalkeeper is used. Task 9 only changed the condition of the presence or absence of a goalkeeper. It can be seen from the results that when a goalkeeper is set, all the probabilities are reduced compared to the situation without a goalkeeper. This situation is in line with expectations.

7. **Compare the results in Task 8**

   When only considering the situation when N=1000 and R=5, the results of the two tasks are as follows:

   Table 11: The results for task 9 and task 8

N	100	1000
P(Task 9)	0.12	0.137
P(Task 8)	0.36	0.376

   By comparison, the probability of the normal random generator will be less than the probability of the uniform random generator. This is because the normal distribution causes the points to cluster in the center. In addition, many points will appear **outside** the circle for the normal random generator. This has led to the probability of normal generators being lower than that of uniform generators, even though they all have goalkeeper.

3.2.3 Task Ten

In this task, the probability of goalkeeper's possible actions needs to be considered. There is a 90% probability that the goalkeeper will tend to jump to the lower two corners of the goal. In response to this situation, Function 5 and 6 are used to simulate the situation in uniform and normal random generator, respectively. Here, for each situation, two experiments were conducted. They are R=1, N=100 and R=1, N=1000. The specific results are as follows:

The following is the result of the uniform random generator:

Table 12: The results for task 10 in uniform random generator

N	100	1000
P	0.35	0.388

The specific experimental simulation figures are as follows:

When N=100:

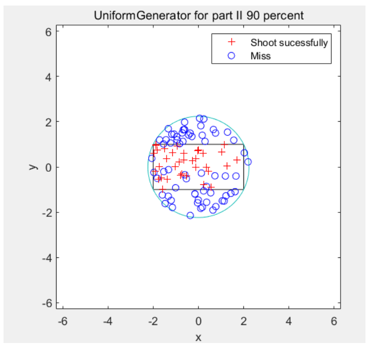


Figure 26: The figure of simulated experiment when R=5,N=100

When N=1000:

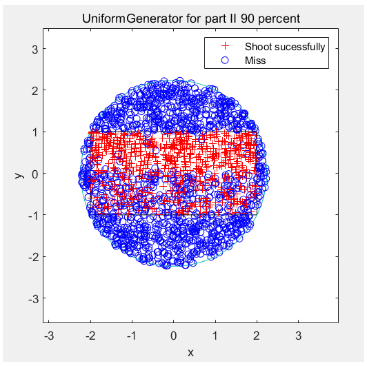


Figure 27: The figure of simulated experiment when R=5,N=1000

The following is the result of the normal random generator:

Table 13: The results for task 10 in normal random generator

N	100	1000
P	0.35	0.147

The specific experimental simulation figures are as follows:

When N=100:

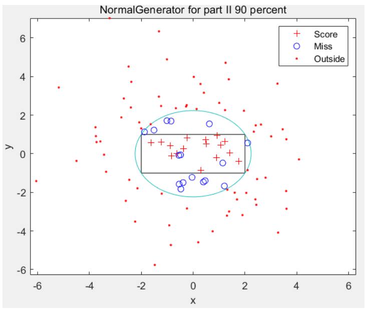


Figure 28: The figure of simulated experiment when R=5,N=100

When N=1000:

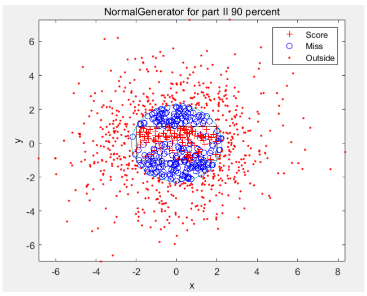


Figure 29: The figure of simulated experiment when R=5,N=1000

Through the experimental results, it can be seen that the goalkeeper has a 90% probability to jump to the lower two corners of the goal. Therefore, the results can be seen from any figure that the blue circle is mostly in the **lower corner of the rectangle**. This is because many goals below are protected by the goalkeeper. Simulation experiments show this phenomenon, which is in line with actual situation. The following is the summary of Task 10.

1. **For uniform random generator**

   When N=100, the probability of considering the goalkeeper's action probability is 0.388, and the probability of not considering the goalkeeper's possible actions is 0.376. It can be seen that the two results are still different, but the difference is not big. The reason for this is that when the goalkeeper has a greater chance of jumping to the lower two corners of the goal, it reduces the probability that the goalkeeper stays in the middle. Staying in the middle can provide a larger protection area. And reducing the occurrence of this situation, the probability of scoring will naturally increase.

2. **For normal random generator**

   Due to the normal random generator, the points will appear outside the circle area. This will reduce the probability of scoring a goal, which can also be verified from the results. The goal probability of the normal distribution is significantly lower than that of the uniform distribution.

4 Conclusions/Discussion

In this part, the experiment will be summarized and every review question will be answered. Moreover, the limitations and suggestions of the experiment will be written in this section.

4.1 Experiment objectives

The main goal of the project is completed. First, the Monte Carlo method is introduced in detail in the Introduction section. Since this simulation method has a wide range of applications in science, it is necessary to be familiar with this method. In addition, Monte Carlo method is also understood by searching for information, and 6 Functions are used to complete all the task objectives.

1. **For situations where there is no goalkeeper**

   Two functions are designed, which are respectively aimed at uniform random generator and normal random generator. In Task 7, the probability of these two distributions is compared. The influence between the values of N and R and the probability obtained from the experimental is also discussed. If only the size of R increases, the accuracy of the experimental probability will not increase, but only the precision. And increasing the size of N will increase the accuracy of the experiment. When N is large, it can be regarded that the probability obtained from the experiment as that under the real situation. This conclusion has also been verified in the simulation experiment. The specific explanation method can be viewed in the results section.

2. **For situations where there is goalkeeper**

   The probability of the goalkeeper's possibile actions is also taken into account in both uniform and normal random generators experiments. And the final results are compared with the situation of without goalkeeper. Finally, it can be found that with a goalkeeper, the probability of scoring is reduced, which is also in line with real conditions.

4.2 Review Questions

In this part, the review questions will be answered.

4.2.1 Question 1

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

1. **Advantages**

   (a) **Simple and convenient**

   The Monte Carlo method can be performed on simulation software. Compared to real experiments, it is obviously easier and more convenient to conduct experiments on a computer. Moreover, when the value of N is very large, Monte Carlo can also accurately simulate the probability under real conditions, which is very convenient and convincing.

   (b) **Monte Carlo method can solve complex problems that are inconvenient to calculate**

   Because mathematics often need to calculate some difficult problems which are hard to calculate. Using Monte Carlo method can be more convenient and accurate the process of calculation. For example, when calculating the area of an irregular figure, the irregular figure can be put in the figure where the area can be calculated like this experiment. And then through the probability, the ratio and the area of the irregular figure can be calculated. Compared with calculating by mathematical methods, this method is obviously more accurate and efficient, and the problem which cannot be calculated in math can be solved.

   (c) **Practicality**

   The Monte Carlo method has a wide range of applications. This method can be applied to both mathematical calculations and computer graphics and telecommunications. This shows that Monte Carlo can be applied in various fields. Therefore, it is necessary to learn Monte Carlo method.

2. **Disadvantages**

   (a) **Experimental error**

   When the sample size is small, the error of the Monte Carlo method is relatively large. For example, when N=100, the general error of the probability obtained by the experiment is large. This conclusion has also been verified in simulation experiments. In other words, if the sample size is too small, the Monte Carlo method is not suitable. The Monte Carlo method is only suitable for large sample sizes. The larger the sample size, the more accurate the results obtained. For this simulation experiment, a relatively accurate probability can be obtained only when the sample size is at least 1,000.

(b) **Monte Carlo is not very suitable for large areas**

In this experiment, a relatively small area was used, so N=1000 is enough to get accurate results. But when the area of the experiment expands, it is necessary to increase the size of N to ensure the accuracy of the experiment. Due to different computer configurations, when R and N are very large, the calculation of the computer will be slower.

(c) **Limitations of simulation**

In this experiment, Matlab was used for simulation experiments with the assumed mean and standard deviation. For experiments with higher accuracy, the more accurate the input data, the more accurate the results can be obtained. However, the input data is determined by the user. In real experiments, these accurate input data are difficult to obtain.

4.2.2 Question 2

Discuss the ways in which the above model could be made more accurate and realistic.

First of all, for the model provided by the lab script, some changes need to be proposed. The improved model provided is as follows:

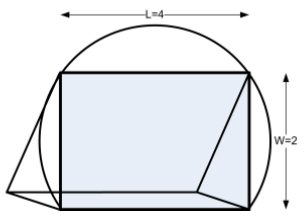


Figure 30: More realistic model(taken from [1])

Because of the real situation, it is impossible to kick the bottom half of the circle, because this area is on the grass, it is impossible for the player to kick the ball into the ground. Therefore, this improved model is more convincing.

In addition, the player's shooting range is a circle which is **relatively rough**. According to the statistics, the player's accustomed shooting position can be determined

The soccer itself is an object. In this simulation experiment, it was assumed that the ball is a point, which is a rough method. A more accurate way is to regard the soccer **as a circle**. The result obtained in this way is obviously more accurate.

In addition, the goal frame in the experiment was not taken into consideration. In a real game, it is very common to hit the goal frame and bounce back. Therefore, if the result need to be more accurate, it was necessary to **provide the width of the frame** and the soccer which hits the frame is not counted as a score. This is more in line with the real situation.

4.2.3 Question 3

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.

From a mathematical point of view, when the standard deviation increases, the points will be more concentrated in the middle(center). The example diagram is as follows:

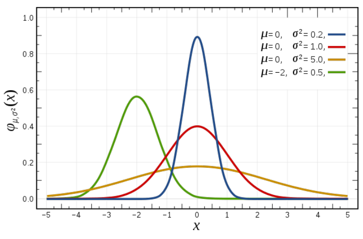


Figure 31: Normal Distribution(taken from [8])

From a mathematical point of view, increasing the standard deviation will only increase the precision of the data. However, through simulation experiments, it can be found that when the standard deviation is increased, the accuracy of this experiment will also increase.

This is because the standard deviation σ used in the simulation experiment is smaller than the radius of the circle r=$\sqrt{5}$, and increasing the standard deviation will make more points cluster in the center, which also leads to an increase in the standard deviation. At the same time, it will also increase the accuracy of the result.

4.2.4 Question 4

If a large number of balls are kicked on the goal (i.e. if N is sufficiently large), the value of 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 . Verify this using your results for both uniform and Gaussian distributions.

In Task 1, the formula for calculating the probability is as follows:

$$P = \frac{8}{5 * \pi} \tag{10}$$

Therefore, the value of π can be calculated through the following formula:

$$\pi = \frac{8}{5 * p} \tag{11}$$

1. **For the uniform random generator**

   When the value of N is relatively large (N=10000), the probability obtained is 0.5099. Therefore, the value of π is:

   $$\pi = \frac{8}{5 * 0.5099} \approx 3.14 \tag{12}$$

   It can be seen from the result that the π calculated by uniform is very close to the real value. The reason for that is the uniform random generator makes the points evenly distributed in the circle. When N is large, the sample size can be considered infinite, and all points are randomly and uniformly distributed in the circle. The probability is the ratio between the area of the rectangle and the area of the circle. Under the condition of accurate probability, the value of π will be very close to the true value.

2. **For normal random generators**

   When the value of N is relatively large (N=10000), the probability obtained is 0.2171. Therefore, the value of π is:

$$\pi = \frac{8}{5 * 0.2171} \approx 7.37 \tag{13}$$

   Through calculation, it can be found that the value of π is about 7.37, which is very different from the real value. This is because normally distributed points are not evenly distributed. Most of the points are gathered in the central part, and there are many points distributed outside the circular area, which will cause errors in the calculation of π. That is to say, when calculating the value of π, the formula should be changed rather than simply the area ratio.

4.2.5 Question 5

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?

1. **For goalkeepers**

   The probability of jumping to the lower corners should be reduced. Because according to the situation of the lab script, when the goalkeeper is standing in the middle, goalkeeper can block half of the goal, which can prevent the goal better, and if the goalkeeper jumps to the lower corners, goalkeeper can only prevent 1/4 of the goal. But in real situations, players rarely kick in the middle of the goal. The above analysis is only an ideal method. A better way is to increase the area where that goalkeeper can block goals in other positions, rather than only when the goalkeeper is in the middle can block 1/2 of the goals.

2. **For the players**

   The better suggestion is to increase the possibility of kicking to the upper left and right corner. This is because according to the requirements of the lab script, the goalkeeper has a greater possibility of going to the lower two corners. Shooting to the upper left And right corner can avoid the goalkeeper's action, which obviously can increase the probability of scoring. In addition, it is not recommended for players to kick the ball in the middle position. This is because the goalkeeper's resistance range is very large in the middle, and the soccer has a greater chance of being stopped by the goalkeeper. The above analysis is aimed at the ideal situation, the real situation should analyze the accurate probability of the goalkeeper's possible actions, and choose a position where the goalkeeper rarely jumps.

4.3 Limitations and suggestions

4.3.1 Limitation

1. **Experimental error**

   When the value of N is small, the experimental error will be relatively large. In this experiment, although N=100, there is still a relatively large error compared to the theoretical value.

2. **Computer limitations**

   Due to the limited operating speed of the computer, when the experiment with R=5 and N=100000 is carried out, the loading speed of the computer is very slow. It takes a long time to get the results.

3. **Limitations of simulation experiments**

   Since the Monte Carlo method is mostly realized by simulation, there will be errors in the simulation. Since the sample size cannot be infinite, the probability will definitely produce an error, although this error is basically negligible.

4.3.2 Suggestions and Future improvement

1. **Reduce the value of N=100000**

   In this experiment, it is possible to reduce the value of N(100000). During the simulation experiment, when doing the experiment with R=5 and N=100000, the computer's calculation time will be very long. It takes more than ten minutes to get the result. For this experiment, N=10000 is enough to get relatively accurate results, and there is no need to conduct an experiment with N=100000. The value of N is too large to cause a great burden on the calculation of the computer.

2. **Change the design of the model**

   The experimental hypothesis provided to us is unreasonable. There should be no circular area under the rectangle. The reality is that the bottom of the circle is the ground, and the soccer cannot appear in this range. There are some other problems with the model, which can be viewed in review question 2.

3. **Set the width of the frame**

   Because in real experiments, it is very common that the soccer hits the frame and bounces. This is not reflected in this simulation experiment, which is a part that can be improved.

4.4 Summary

In this experiment, 6 functions were applied to solve all tasks. The specific code could be viewed in the appendix. Every review question was explained in detail. In addition, for all situations, the following rules were summarized:

- If only changing the value of R will not improve the accuracy of the experiment, it will only improve the precision.
- The scoring probability of experiments using the normal distribution is generally lower than that of the uniform distribution. This is because there are many points distributed outside the circle. This reduces the probability of scoring a goal.
- The goalkeeper's role is to reduce the probability of a goal, rather than affecting other results. For the normal and uniform random generator, the goalkeeper will reduce the chance of scoring.

- If considering the probability of the goalkeeper's possible actions, it can indeed be seen through experiments that many goals in the lower half of the rectangle have been stopped. However, this situation increases the probability of a goal because the probability of the goalkeeper staying in the middle becomes smaller, and this situation can prevent half of the goals. The situation in the lower half corners can only prevent a quarter of a goal.

References

[1] M. L´opez-Ben´ıtez, "Monte carlo simulation," https://liverpool.instructure.com/courses/46000/pages/experiment-22?module_item_id=1258991, Department of Electrical Engineering Electronics, 2019.

[2] IBM Cloud Education, "Monte carlo simulation," https://www.ibm.com/cloud/learn/monte-carlo-simulation, 2020.

[3] G.Fishman, "Monte carlo concepts, algorithms and applications," Springer, 1996.

[4] MathWorks, "Math, graphics, programming," https://ww2.mathworks.cn/products/matlab.html, 2021.

[5] FrontlineSolvers, "Monte carlo simulation," https://www.solver.com/monte-carlo-simulation-tutorial, 2021.

[6] J.woller, "The basics of monte carlo simulations," https://mathworld.wolfram.com/NormalDistribution.html, 1996.

[7] M.Uney, "Exp 22 primer on probability statistics matlab-1," https://liverpool.instructure.com/courses/46000/pages/experiment-22?module_item_id=1258991, University of Liverpool, 2021.

[8] Commons Wikimedia, "Normal distribution pdf," https://commons.wikimedia.org/wiki/File:Normal_Distribution_PDF.svg, 2008.

Appendices

A Six functions used in the simulation

Listing 1: Function 1

```
%R R is the times that repeat this experiment
%N N means N random penalty shots
%T T means the number of shooting in the rectangle area in one experiment
%x The horizontal x-coordinate
%y The vertical y-coordinate
%radius The radius coordinate
%angle The angle coordinate
%P The final possibilities of shooting in the rectangle area

function [Final_Result]=UniformGenerator_for_Part_I(N,R)
P=0; %Initialize value of P
rect=[-2,-2,2,2,-2;-1,1,1,-1,-1]; %Draw the area of rectangle and circle
plot(rect(1,:),rect(2,:),'k') %The rectangle area
hold on
ezplot('x^2+y^2=5')%The circle area
hold on

for R=1:R
T=0; %Initialize the value of T
for N=1:N
radius=sqrt(5)*sqrt(rand()); %Uniform random generators
angle=2*pi*rand(); %The random angle
x=cos(angle)*radius; %The horizontal x-coordinate
y=sin(angle)*radius; %The vertical y-coordinate

if (abs(y)<1&&abs(x)<2) %The conditions of the points which are in the rectangle
T=T+1; %The times plus 1
Shoot=plot(x,y,'+r'); %The inside points
hold on
else
Miss=plot(x,y,'ob'); %The outside points
hold on
end
end
Pe=T/N; %Get each possibility in every simulation
P=P+Pe; %Sum up to get the final possibility

%Plot the figure with the title,xlabel,ylabel
title('UniformGenerator_for_Part_I');
xlabel('x');
ylabel('y');
legend([Shoot,Miss],'Shoot_sucessfully','Miss');
axis equal
end
Final_Result=P/R; %Calculate the final possibilty
```

Listing 2: Function 2

```
%P The final possibilities of shooting in the rectangle area
%R R is the times that repeat this experiment
%N N means N random penalty shots
%T T means the number of shooting in the rectangle area in one experiment
%x The horizontal x-coordinate
%y The vertical y-coordinate
%COUT COUT means the points which set outside the circle
%Total Total shots
%a The mean value

function [Final_Result]=NormalGenerator_for_part_I(N,R)
P=0; %Initialize the value of P
%Draw the area of rectangle and circle
rect=[-2,-2,2,2,-2;-1,1,1,-1,-1]; %The rectangle area
plot(rect(1,:),rect(2,:),'k')
hold on
ezplot('x^2+y^2=5')%The circle area
hold on

for R=1:R
Total=0; %Total shots
COUT=0; %points which set outside the circle
T=0; %Initialize the T
while(Total<N)
a=0;%The mean value
b=sqrt(5)
x=randn(1,1)*b+a;%The horizontal x-coordinate
y=randn(1,1)*b+a;%The vertical y-coordinate

if(x^2+y^2>5) %The conditions of the points which are outside the circle
COUT=COUT+1;
Outside=plot(x,y,'r.');
else
if (abs(y<1)&&abs(x<2)) %The conditions of the points which are in the rectangle
T=T+1;
Shoot=plot(x,y,'+r');%plot the points
hold on
else
%The conditions of the points which are outside the rectangle but inside circle
Miss=plot(x,y,'ob');
hold on
end
end
Total=Total+1;
end
Pe=T/N; %Get each possibility in every simulation
P=P+Pe; %Sum up to get the final possibility

%Plot the figure with the title,xlabel,ylabel
title('NormalGenerator_for_Part_I');
xlabel('x');
ylabel('y');
legend([Shoot,Miss,Outside],'Score','Miss','Outside');
end
Final_Result=P/R; %The final possibilty
```

Listing 3: Function 3

```
%P The final possibilities of shooting in the rectangle area
%R R is the times that repeat this experiment
%N N means N random penalty shots
%T T means the number of shooting in the rectangle area in one experiment
%x The horizontal x-coordinate
%y The vertical y-coordinate
%radius The radius coordinate
%angle The angle coordinate
%Pe Each possibility in every experiment

function [Final_Result]=UniformGenerator_for_Part_II(N,R)
P=0; %Initialize the value of possibility P
rect=[-2,-2,2,2,-2;-1,1,1,-1,-1]; %Draw the area of the rectangle and the circle
plot(rect(1,:),rect(2,:),'k') %The rectangle area
hold on
ezplot('x^2+y^2=5')%The circle area
hold on

for R=1:R
T=0; %Initialize the value of times T
for N=1:N
radius=sqrt(5)*sqrt(rand()); %Uniform random generators
angle=2*pi*rand(); %The random angle
x=cos(angle)*radius; %The horizontal x-coordinate
y=sin(angle)*radius; %The vertical y-coordinate

case_number=round(rand()*5+0.5); %Judge the different conditions

switch(case_number)
case{1}
%The conditions of the points
if (abs(x)>1 && abs(x)<2 && abs(y)<1)
T=T+1;%The times plus 1
Shoot=plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');%The missed points
hold on
end

case {2}
%The conditions of the points
if ((abs(y)<1 && x<=0 && x>-2)||(x>=0 && x<2 && y>=0 && y<1))
T=T+1;%The times plus 1
plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');%The missed points
hold on
end

case{3}
%The conditions of the points
if ((x<0 && x>-1 && y<1 && y>0)||(x>0 && x<2 && abs(y)<1)||(y<0 && y>-1 && x>-2 && x<-1))
T=T+1;%The times plus 1
plot(x,y,'+r');
hold on
else
```

```
Miss=plot(x,y,'ob');%The missed points
end

case {4}
%The conditions of the points
if ((abs(y)<1 && x>0 && x<2)||(x>-2 && x<0 && y<1 && y>0))
T=T+1;%The times plus 1
plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');%The missed points
hold on
end

case{5}
%The conditions of the points
if ((y<0 && y>-1 && x>1 && x<2)||(x>-2 && x<0 && abs(y)<1)||(x>0 && x<1 && y>0 && y<1))
T=T+1;%The times plus 1
plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');%The missed points
hold on
end
end
end
Pe=T/N; %Get each possibility in every simulation
P=P+Pe; %Sum up to get the final possibility

%Plot the figure with the title,xlabel,ylabel
title('UniformGenerator for Part II');
xlabel('x');
ylabel('y');
axis equal
legend([Shoot,Miss],'Shoot sucessfully','Miss');
end
Final_Result=P/R; %The final possibilty
```

Listing 4: Function 4

```
%P The final possibilities of shooting in the rectangle area
%R R is the times that repeat this experiment
%N N means N random penalty shots
%T T means the number of shooting in the rectangle area in one experiment
%x The horizontal x-coordinate
%y The vertical y-coordinate
%COUT COUT means the points which set outside the circle
%T Total shots
%a The mean value

function [Final_Result]=NormalGenerator_for_part_II(N,R)
P=0; %Initialize the value of possibility P
rect=[-2,-2,2,2,-2;-1,1,1,-1,-1];%Draw the area of rectangle and circle
plot(rect(1,:),rect(2,:),'k')%The rectangle area
hold on
ezplot('x^2+y^2=5')%The circle area
hold on

for R=1:R
Total=0; %Total shots
T=0; %Initialize the T for each estimation
COUT=0; %Points which set outside the circle
while (Total<N)
a=0;%The mean value
b=sqrt(5)
x=randn(1,1)*b+a;%The horizontal x-coordinate
y=randn(1,1)*b+a;%The vertical y-coordinate

%The conditions of the points which are outside the circle area
if(x^2+y^2>5)
COUT=COUT+1;
Outside=plot(x,y,'r.');
else %According to the question, there are five situations
goalkeeper=round(rand()*5+0.5);

switch (goalkeeper)
case {1}
%The conditions of the points
if ((x<2 && x>0 && abs(y)<1)) || (x>-2 && x<0 && y>0 && y<1)
T=T+1;
plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');
hold on
end

case {2}
%The conditions of the points
if (abs(x)>1 && abs(x)<2 && abs(y)<1)
T=T+1;
Shoot=plot(x,y,'+r');
hold on
else
plot(x,y,'ob');
hold on
end
```

```
case{3}
%The conditions of the points
if  (x>0 && x<1 && y>0 && y<1)|| ((x>-2 && x<0 && abs(y)<1) || (y>-1 && y<0 && x<2 && x>1 ))
T=T+1;
plot(x,y,'+r');
hold on
else
plot(x,y,'ob');
end

case {4}
%The conditions of the points
if ((x<-1 && x>-2 && y<0 && y>-1) || ( x<2 && x>0 && abs(y)<1) || (x>-1 && x<0 && y<1 && y>0))
T=T+1;
plot(x,y,'+r');
hold on
else
plot(x,y,'ob');
hold on
end

case{5}
%The conditions of the points
if ((x>=0 && x<2 && y>=0 && y<1)|| (abs(y)<1 && x>-2 && x<=0) )
T=T+1;
plot(x,y,'+r');
hold on
else
plot(x,y,'ob');
hold on
end
end
end
Total=Total+1;
end
PO=T/N; %Get each possibility in every simulation
P=P+PO; %Sum up to get the final possibility
end

%Plot the figure with the title,xlabel,ylabel
title('NormalGenerator_for_Part_II');
xlabel('x');
ylabel('y');
legend([Shoot,Miss,Outside],'Score','Miss','Outside');
Final_Result=P/R; %The final possibilty
```

Listing 5: Function 5

```
%P The final possibilities of shooting in the rectangle area
%R R is the times that repeat this experiment
%N N means N random penalty shots
%T T means the number of shooting in the rectangle area in one experiment
%x The horizontal x-coordinate
%y The vertical y-coordinate
%tend The possibility

function [Final_Result]=UniformGenerator_for_part_II_90_percent(N,R)
P=0; %Initialize the value of possibility P
sect=[-2,-2,2,2,-2;-1,1,1,-1,-1];%Draw the area of rectangle and circle
plot(sect(1,:),sect(2,:),'k')%The rectangle area
hold on
ezplot('x^2+y^2=5')%The circle area
hold on

for R=1:R
T=0; %Initialize the Times T
for N=1:N
r=sqrt(5)*sqrt(rand());%Uniform random generators
a=2*pi*rand(); %The random angle
x=cos(a)*r; %The horizontal x-coordinate
y=sin(a)*r; %The vertical y-coordinate

tend=rand(); %possibility
if tend<=0.9
case_number=round(rand()+4); %The lower two corners of the goal
else
case_number=round(rand()*3+0.5);%Other conditions
end

switch (case_number) %According to the question,there are five conditions

case {1}
%The conditions of the points
if ((y>-1 && y<1) && (x>0 && x<2)|| (x>-1 && x<0 && y<1 && y>0) || (x<-1 && x>-2 && y>-1 && y<
T=T+1;
Shoot=plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');
hold on
end

case {2}
%The conditions of the points
if ( abs(x)>1 && abs(x)<2 && abs(y)<1)
T=T+1;
Shoot=plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');
end

case{3}
%The conditions of the points
if ((abs(y)<1 && x>-2 && x<0) || (x<2 && x>1 && y<0 && y>-1) || (y>0 && y<1 && x>0 && x<1))
T=T+1;
```

```
Shoot=plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');
hold on
end

case {4}
%The conditions of the points
if ((abs(y)<1 && x>0 && x<2) || (x>-2 && x<0 && y<1 && y>0) )
T=T+1;
Shoot=plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');
hold on
end

case{5}
%The conditions of the points
if ((x>=0 && x<2 && y>=0 &&y<1) || (x<=0 && x>-2 && abs(y)<1))
T=T+1;
Shoot=plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');
hold on
end
end
end
Pe=T/N; %Calculate the shooting possibility
P=P+Pe; %Sum up the possibilities

%Plot the figure with the title,xlabel,ylabel
title('Uniform Generator for part II 90 percent');
xlabel('x');
ylabel('y');
axis equal
legend([Shoot,Miss],'Shoot sucessfully','Miss');
end
Final_Result=P/R; %The final possibilty
```

Listing 6: Function 6

```
%P The final possibilities of shooting in the rectangle area
%R R is the times that repeat this experiment
%N N means N random penalty shots
%S S means the number of shooting in the rectangle area in one experiment
%x The horizontal x-coordinate
%y The vertical y-coordinate
%COUT COUT means the points which set outside the circle
%T Total shots
%a The mean value

function [Final_Result]=NormalGenerator_for_part_II_90_percent(N,R)
P=0; %Initialize the P
sect=[-2,-2,2,2,-2;-1,1,1,-1,-1];%Draw the area of rectangle and circle
plot(sect(1,:),sect(2,:),'k')%The rectangle area
hold on
ezplot('x^2+y^2=5')%The circle area
hold on

for R=1:R
Total=0; %Total shots
T=0; %Initialize the Times T
COUT=0; %points which set outside the circle

while (Total<N)
a=0;%The mean value
b=sqrt(5)
x=randn(1,1)*b+a;%The horizontal x-coordinate
y=randn(1,1)*b+a;%The vertical y-coordinate

if(x^2+y^2>5) %The conditions of the points which are outside the circle area
COUT=COUT+1;
Outside=plot(x,y,'r.');
else
jump=rand(); %possibility
if jump<=0.9
case_number=round(rand()+4); %The lower two corners of the goal
else
case_number=round(rand()*3+0.5);
end

switch (case_number) %According to the question, there are five situations

case {1}
%The conditions of the points
if ((abs(y)<1 && x>0 && x<2) || (x>-2 && x<-1 && y<0 && y>-1) || (y<1 && y>0 && x>-1 && x<0))
T=T+1;
Shoot=plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');
end

case {2}
%The conditions of the points
if (abs(x)>1 && abs(y)<1 && abs(x)<2)
T=T+1;
Shoot=plot(x,y,'+r');
hold on
```

```
else
Miss=plot(x,y,'ob');
hold on
end

case{3}
%The conditions of the points
if ( (x>0 && x<1 && y>0 && y<1) || (abs(y)<1 && x>-2 && x<0) || ( y<0 && y>-1 && x>1 && x<2))
T=T+1;
Shoot=plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');
hold on
end

case {4}
%The conditions of the points
if ((abs(y)<1 && x>0 && x<2) || (x>-2 && x<0 && y<1 && y>0))
T=T+1;
Shoot=plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');
hold on
end

case{5}
%The conditions of the points
if ((x>=0 && x<2 && y>=0 &&y<1) || (abs(y)<1 && x>-2 && x<=0))
T=T+1;
Shoot=plot(x,y,'+r');
hold on
else
Miss=plot(x,y,'ob');
hold on
end
end
end
Total=Total+1;
end
Pe=T/N; %Get each possibility in every simulation
P=P+Pe; %Sum up to get the final possibility
end

%Plot the figure with the title,xlabel,ylabel
title('NormalGenerator for part II 90 percent');
xlabel('x');
ylabel('y');
legend([Shoot,Miss,Outside],'Score','Miss','Outside');
Final_Result=P/R; %The final possibilty
```

B The final simulated figures for each Task

B.1 Task 3

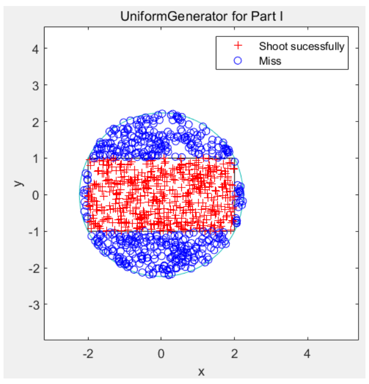
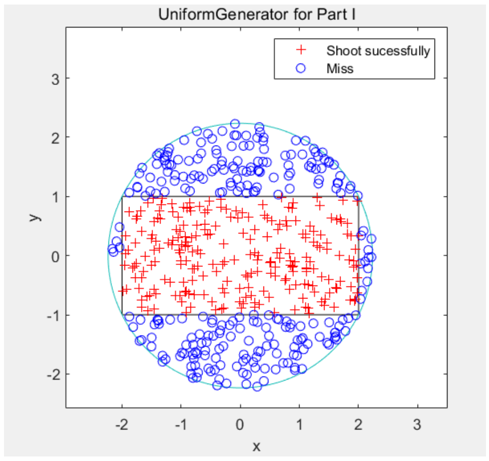
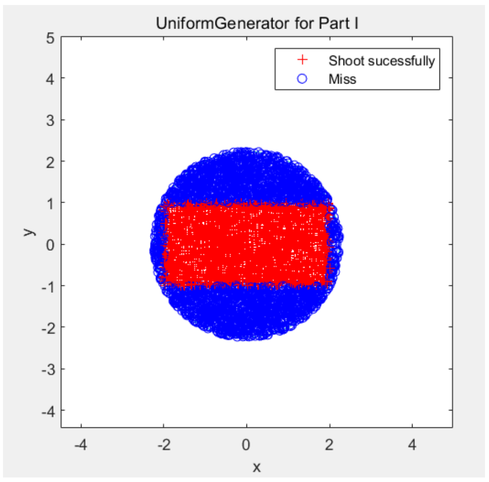


Figure 32: The uniform random generator for N=1000 R=1

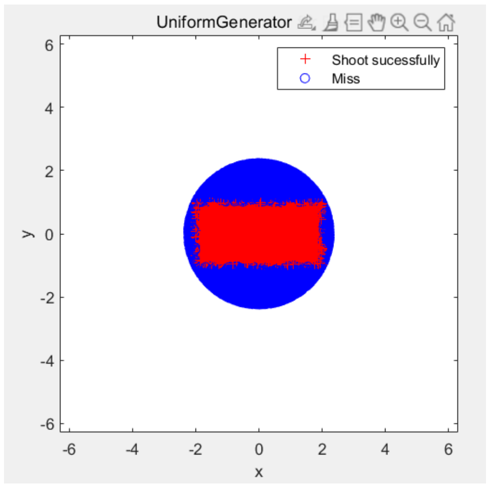
B.2 Task 4



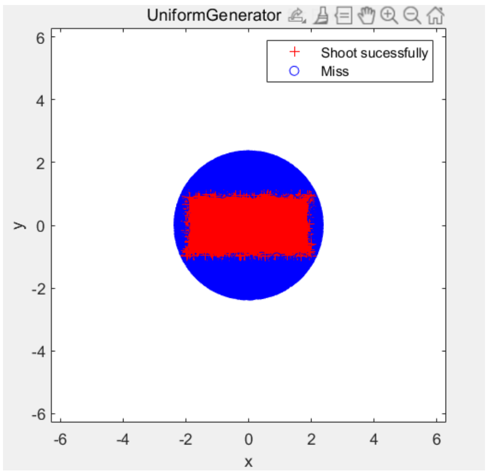
(a) The uniform random generator for N=100 R=5



(b) The uniform random generator for N=1000 R=5



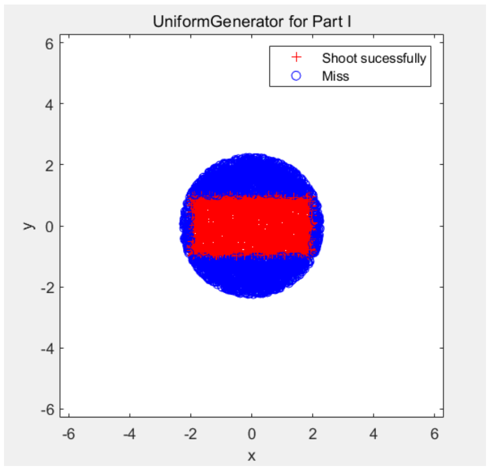
(c) The uniform random generator for N=10000 R=5



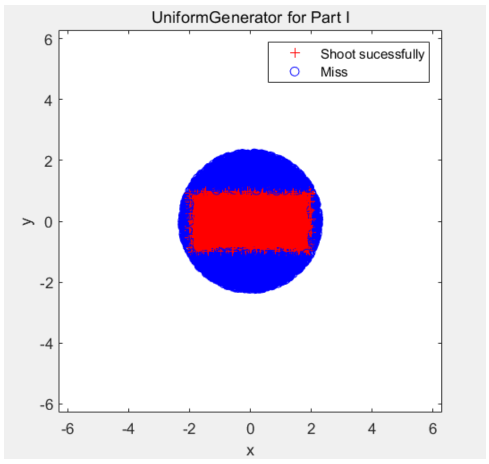
(d) The uniform random generator for N=100000 R=5

Figure 33: The final results for Task 4

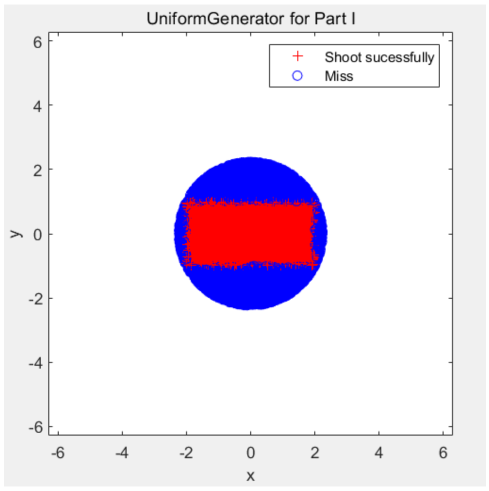
B.3 Task 5



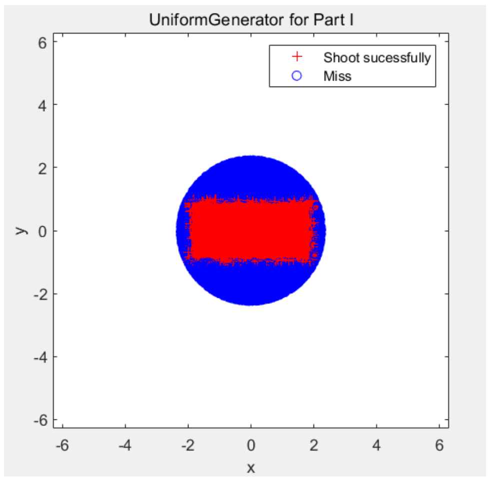
(a) The uniform random generator for N=1000 R=5



(b) The uniform random generator for N=1000 R=10



(c) The uniform random generator for N=1000 R=15



(d) The uniform random generator for N=1000 R=20

Figure 34: The final results for Task 5

B.4 Task 7:Repeat experiments

B.4.1 Repeat Task 3

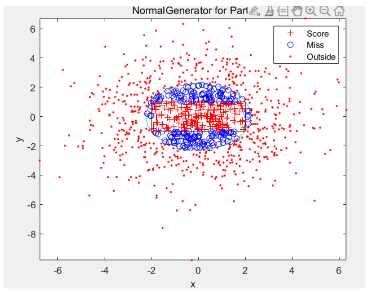
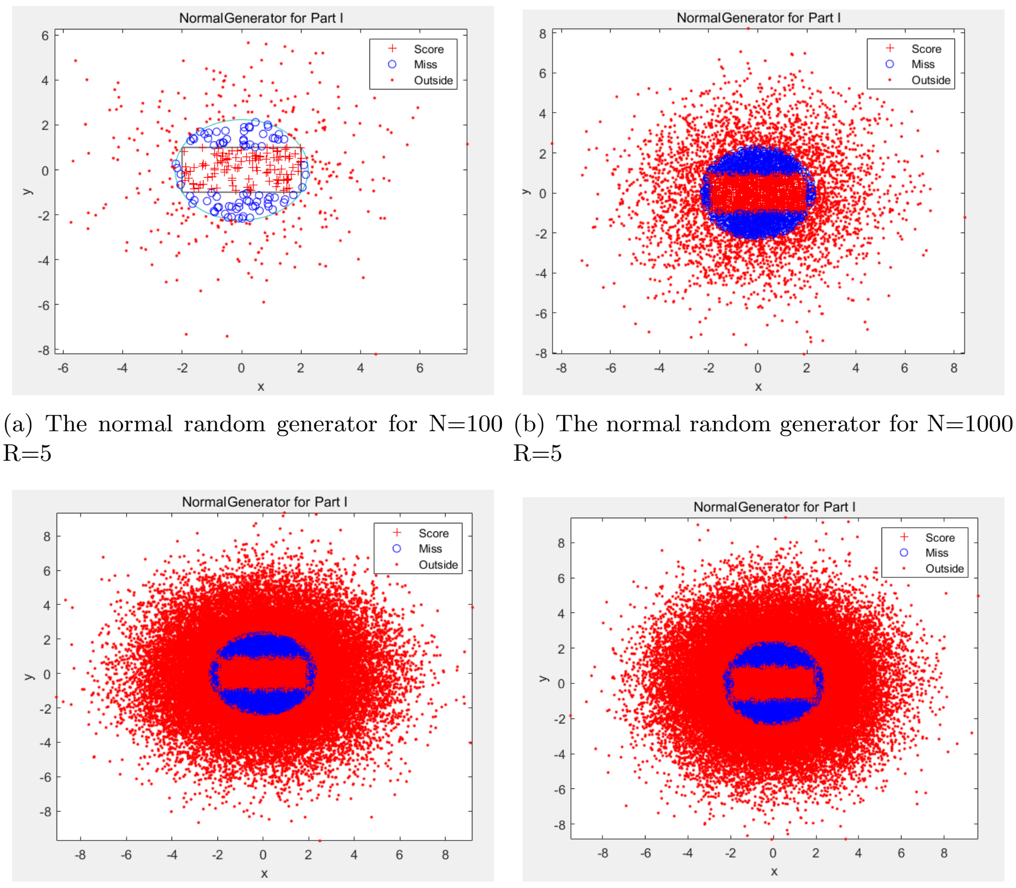


Figure 35: The normal random generator for N=1000 R=1.PNG

B.4.2 Repeat Task 4

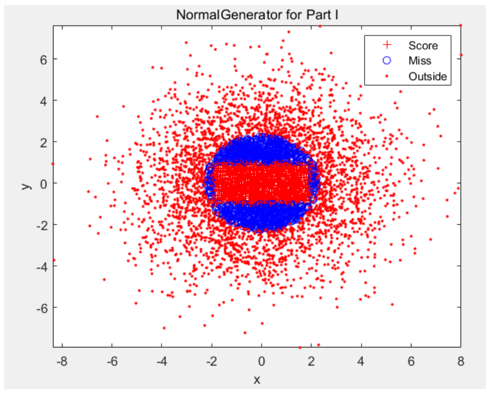


(a) The normal random generator for N=100 R=5 (b) The normal random generator for N=1000 R=5

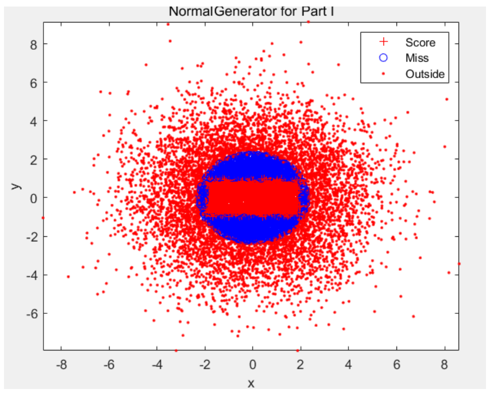
(c) The normal random generator for N=10000 R=5 (d) The normal random generator for N=100000 R=5

Figure 36: The final results for repeated Task 4

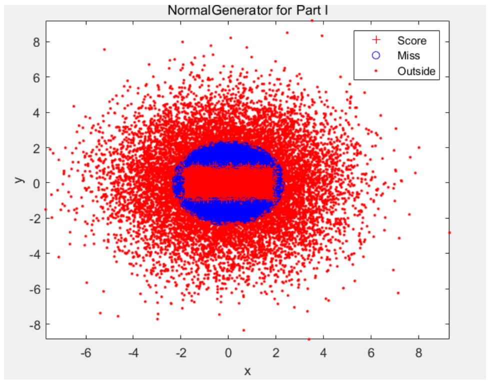
B.4.3 Repeat Task 5



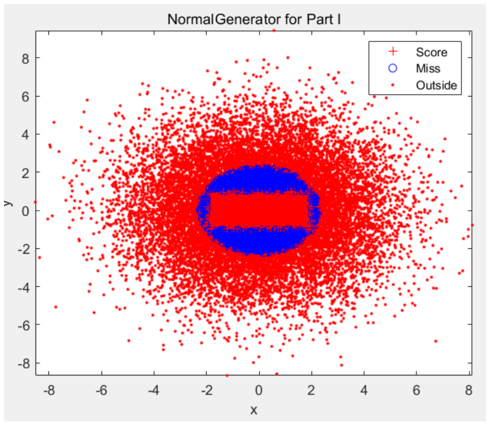
(a) The normal random generator for N=1000 R=5



(b) The normal random generator for N=1000 R=10



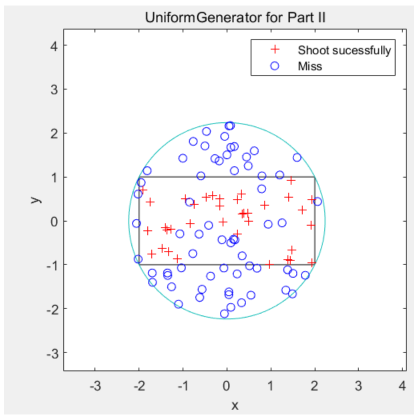
(c) The normal random generator for N=1000 R=15



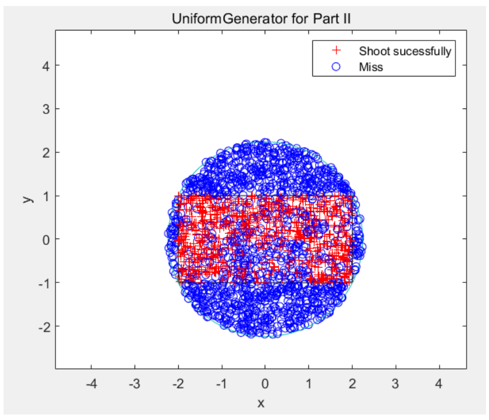
(d) The normal random generator for N=1000 R=20

Figure 37: The final results for repeated Task 5

B.5 Task 8



(a) The result for Task 8 when N=100

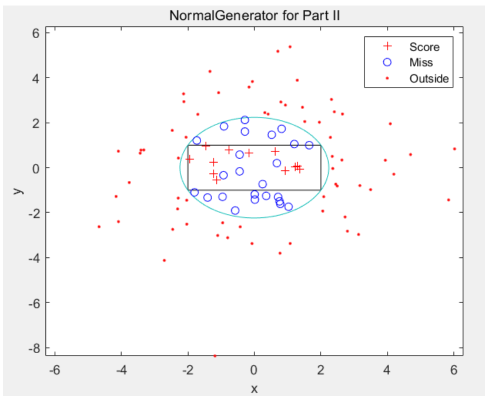


(b) The result for Task 8 when N=1000

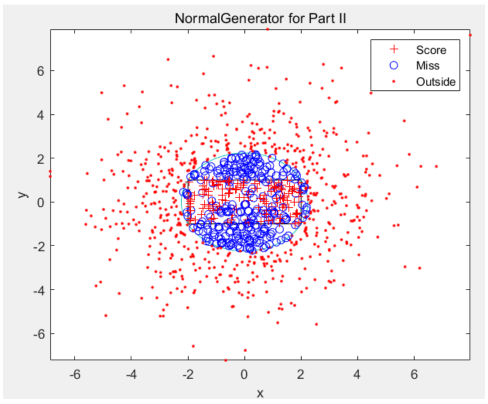
Figure 38: The final results for Task 8

B.6 Task 9

B.6.1 Repeat Task 8



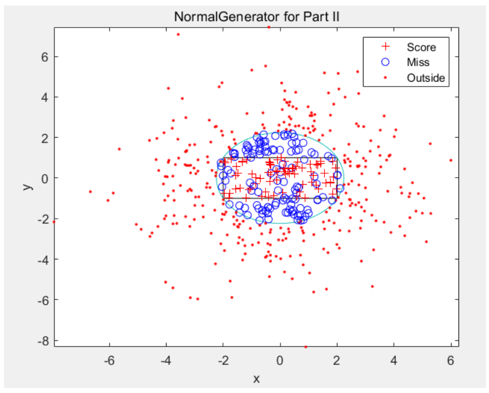
(a) The result for repeated Task 8 when N=100



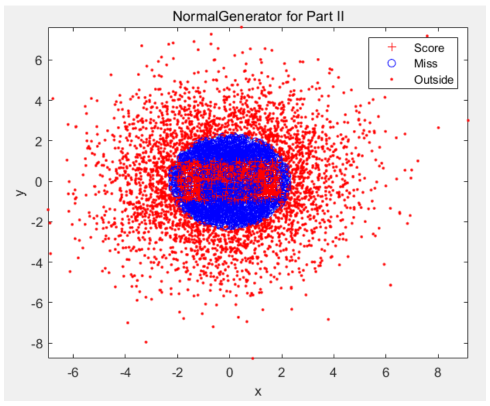
(b) The result for repeated Task 8 when N=1000

Figure 39: The final results for repeated Task 8

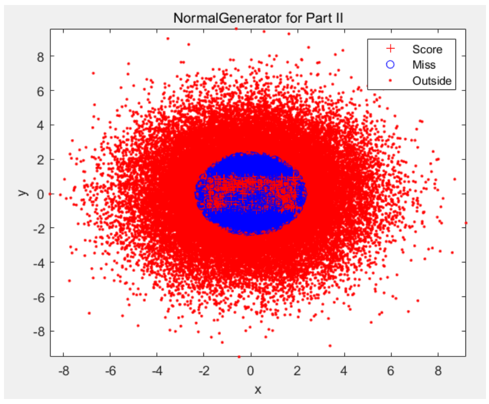
B.6.2 Repeat Task 4



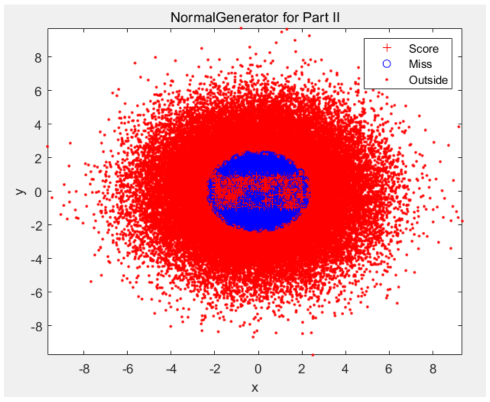
(a) The normal random generator for N=100 R=5



(b) The normal random generator for N=1000 R=5



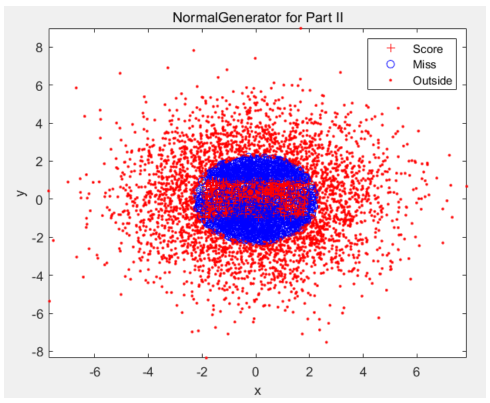
(c) The normal random generator for N=10000 R=5



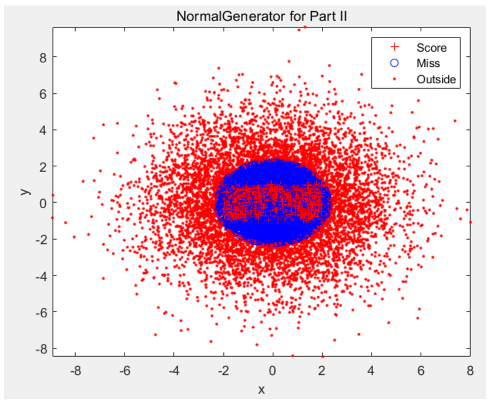
(d) The normal random generator for N=100000 R=5

Figure 40: The final results for repeated Task 4

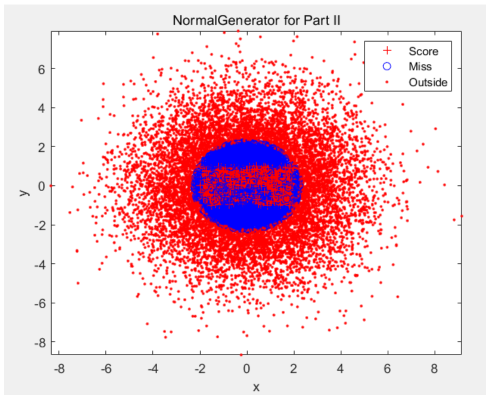
B.6.3 Repeat Task 5



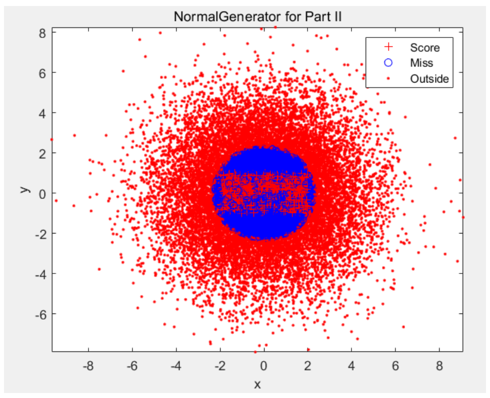
(a) The normal random generator for N=1000 R=5



(b) The normal random generator for N=1000 R=10



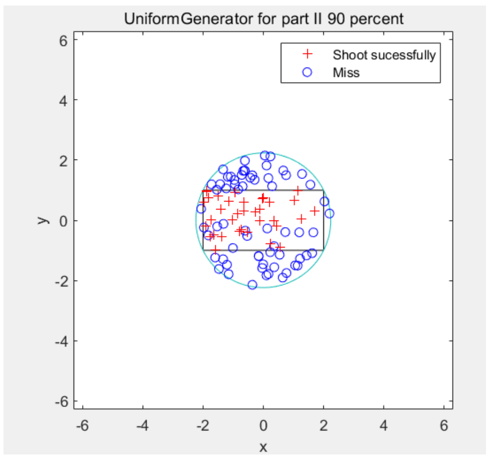
(c) The normal random generator for N=1000 R=15



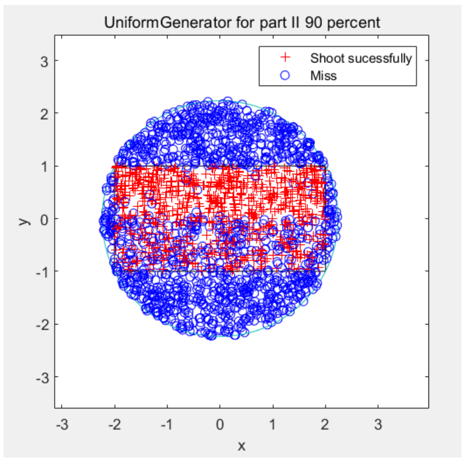
(d) The normal random generator for N=1000 R=20

Figure 41: The final results for repeated Task 5

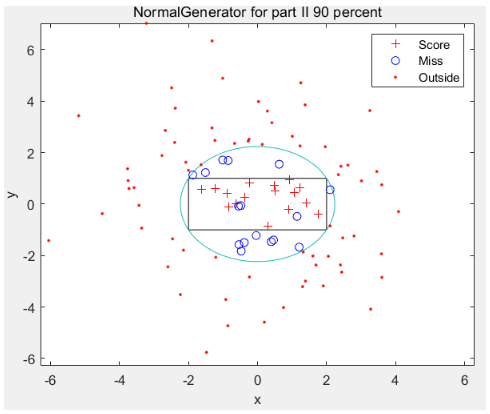
B.7 Task 10



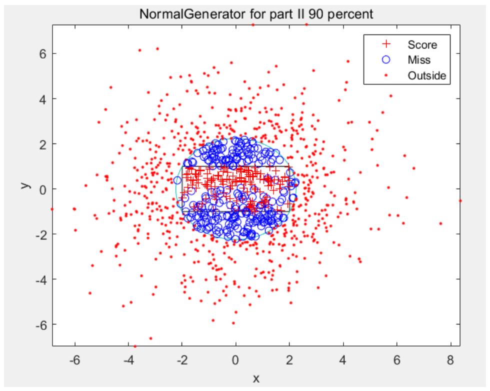
(a) The uniform random generator for N=100 R=5



(b) The uniform random generator for N=1000 R=5



(c) The normal random generator for N=100 R=5



(d) The normal random generator for N=1000 R=5

Figure 42: The final results for Task 10