

Modeling and Simulation II

2.24.2024
Sohun

1 Random Walk

(1) Experiment overview

A drunk person wanders backwards, forwards, left and right indiscriminately. A drunk person tends to move around and keep coming back to the same place in an attempt to reach his destination. Suppose now that a drunk person is at the origin on the xy coordinate plane.

If a drunk person starts from the origin and staggers 30 times, how far from the origin is the drunk person's position on the 30th staggers? However, suppose that a drunk person staggers in four directions: up, down, left, and right, and the distance traveled by each stagger is 1. Also, if a drunk person staggers only in two directions: left and right, and starts from the origin and moves on the x-axis, how far from the origin will the 30th stagger position be?

(2) Experimental result (VB version simulation)

【Experiment day】

February 24 . 2024

【PC used】

Lavie NX850/N

【Software used】

Self-made software

『random walk 6』

【Method of operation】

■ For staggering in four directions, click the check box [Plane movement].

■ For staggering in two directions, click the check box [Linear movement].

(If you want to change from 4-way staggering to 2-way staggering, or vice versa, first click the [Init] button.)

Click the [Experiment Start] button and the drunk person's staggers will begin.

(However, it will take a while before it starts to stagger.)

Click the [Experiment Stop] button to stop the drunk person from staggering.

(However, it will take a while before it starts to stagger.)

Click the [Graph display] button to display the graph.

(However, in the case of staggering in four directions, the horizontal axis is the distance between the position of the drunk person and the origin when he staggers 30 times. In the case of staggering in two directions, the horizontal axis is the x-coordinate of the position of the drunk person when he staggers 30 times.)

【Consideration】

In experiments ① to ③, when a drunk person starts from the origin on the x-axis and staggers only left and right, the x-coordinate of the 30th stagger is counted. The horizontal axis of the graph is the x-coordinate of the 30th staggering position, and the vertical axis is the frequency.

From the graphs ① to ③, the drunk person often return close to the starting point for the 30th time. Therefore, we can say, "A drunk person keeps returning to the same place as he walks around trying to research his destination".

In experiments ④ to ⑥, a drunk person starts from the origin on the xy coordinate plane. When he staggers up and down, left or right, we count the distance between the coordinates of the 30th time he staggered and the origin.

The horizontal axis of the graph is the distance between the coordinates of the 30th staggering position and the origin, and the vertical axis is the frequency.

From the graphs ④ to ⑥, you can see that he often returns to the vicinity of the origin, which is the starting point, for the 30th time.

Therefore, we can say "A drunk person keeps returning to the same place as he walks around trying to research his destination".

Modeling and Simulation II

2.24.2024
Sohun

1 Random Walk

(2) Experimental result (VB version simulation)

① 1st experiment (When moving in a straight line)

Random walk Copyright (C) Sohun 5.2022

Exit	Random walk																	Help	Init
Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
x coordinate	-4	-2	-4	8	-8	4	8	4	6	-6	2	0	4	-4	8	6	4		
y coordinate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Distance	4.0	2.0	4.0	8.0	8.0	4.0	8.0	4.0	6.0	6.0	2.0	.0	4.0	4.0	8.0	6.0	4.0		
Number	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34		
x coordinate	10	0	12	14	-4	-2	4	2	10	4	-2	8	-4	-4	4	-2	0		
y coordinate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Distance	10.0	.0	12.0	14.0	4.0	2.0	4.0	2.0	10.0	4.0	2.0	8.0	4.0	4.0	4.0	2.0	0		
Number	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51		
x coordinate	4	0	-6	-2	-6	0	-2	-4	-2	2	-4	-2	-4	4	2	6	2		
y coordinate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Distance	4.0	.0	6.0	2.0	6.0	.0	2.0	4.0	2.0	2.0	4.0	2.0	4.0	4.0	2.0	6.0	2.0		

Number of stagers: **0**

Average distance: **4.3**

Graph display:

51 th

It takes a little time ...

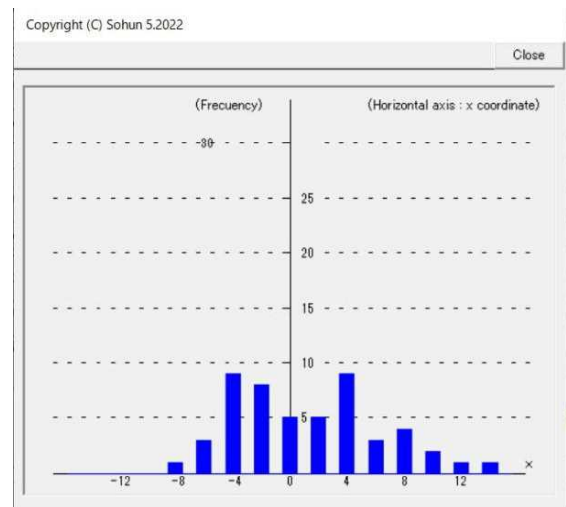
Experiment Start

Experiment Stop

Plane movement:

Linear movement:

x = 0 y = 0



② 2nd experiment (When moving in a straight line)

Random walk Copyright (C) Sohun 5.2022

Exit	Random walk																	Help	Init
Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
x coordinate	6	-6	-8	4	0	0	4	-2	-6	4	-8	6	-8	4	4	-8	6		
y coordinate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Distance	6.0	6.0	8.0	4.0	.0	.0	4.0	2.0	6.0	4.0	8.0	6.0	8.0	4.0	4.0	8.0	6.0		
Number	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34		
x coordinate	0	-13	6	-12	2	2	6	2	6	-4	8	2	-2	-6	-6	12	4		
y coordinate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Distance	.0	13.0	6.0	12.0	2.0	2.0	6.0	2.0	6.0	4.0	8.0	2.0	2.0	6.0	6.0	12.0	4.0		
Number	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51		
x coordinate	-2	6	2	4	0	-4	-2	6	6	8	4	0	6	-4	-14	-4	-8		
y coordinate	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Distance	2.0	6.0	2.0	4.0	.0	4.0	2.0	6.0	6.0	8.0	4.0	.0	6.0	4.0	14.0	4.0	8.0		

Number of stagers: **0**

Average distance: **5.0**

Graph display:

51 th

It takes a little time ...

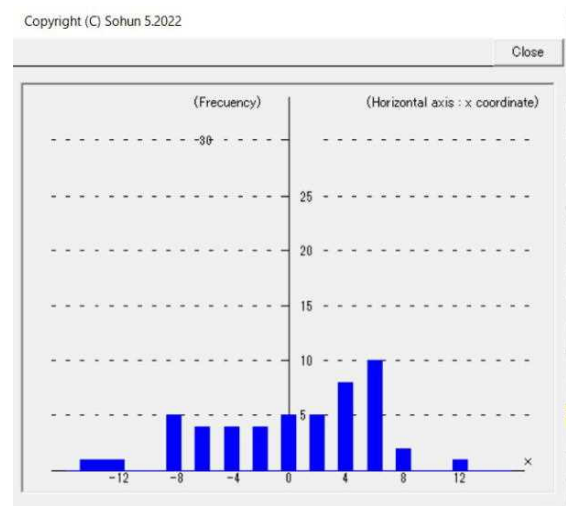
Experiment Start

Experiment Stop

Plane movement:

Linear movement:

x = 0 y = 0



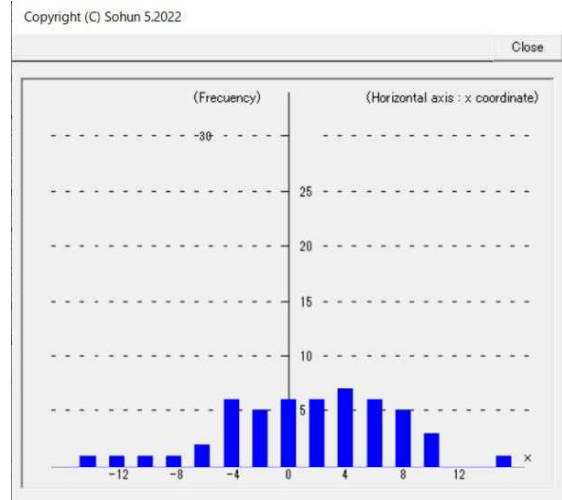
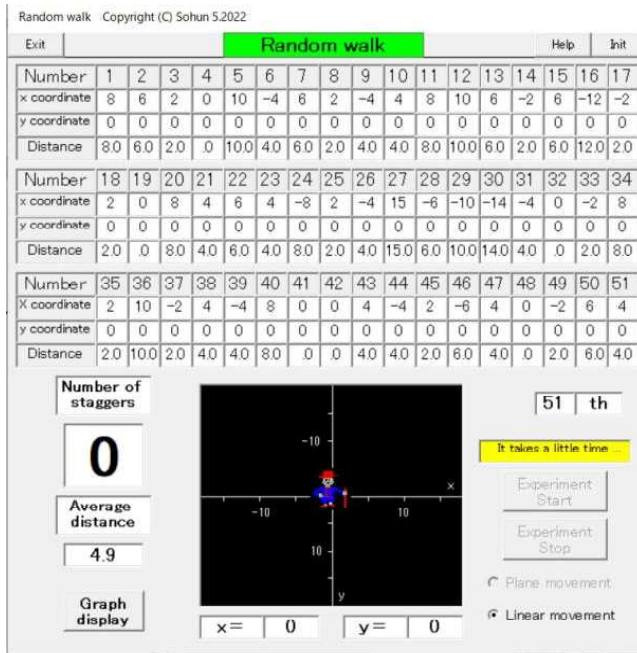
Modeling and Simulation II

2.24.2024
Sohun

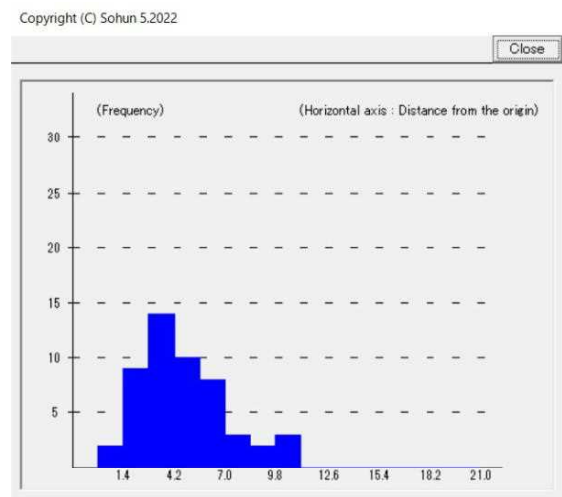
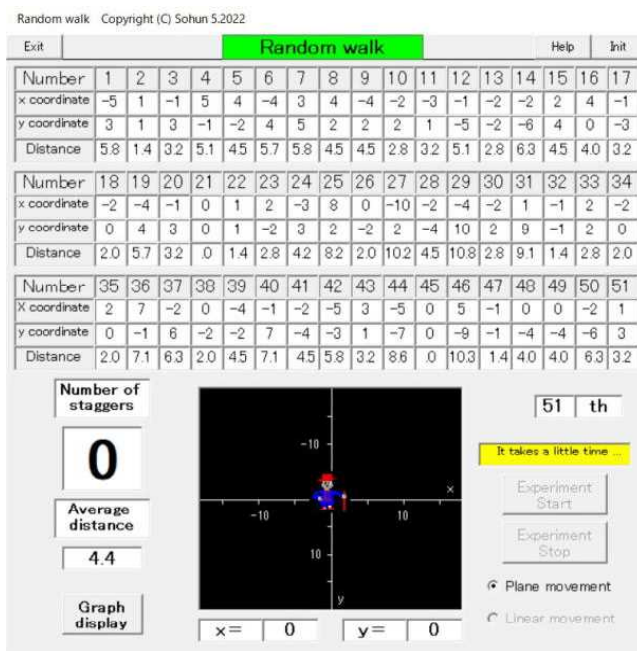
1 Random Walk

(2) Experimental result (VB version simulation)

③ 3rd experiment (When moving in a straight line)



④ 4th experiment (When moving on a plane)



Modeling and Simulation II

2.24.2024
Sohun

1 Random Walk

(2) Experimental result (VB version simulation)

⑤ 5th experiment (When moving on a plane)

Random walk Copyright (C) Sohun 5.2022

Exit	Random walk																	Help	Init
Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
x coordinate	1	2	5	5	-2	3	0	-2	-2	-3	0	4	1	4	-1	3	-3		
y coordinate	1	4	-1	-1	2	-1	-4	-2	-4	-3	-2	8	-5	2	7	1	-3		
Distance	1.4	4.5	5.1	5.1	2.8	3.2	4.0	2.8	4.5	4.2	2.0	8.9	5.1	4.5	7.1	3.2	4.2		
Number	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34		
x coordinate	8	2	2	0	0	1	-4	-2	-2	-4	1	3	4	2	0	-1	6		
y coordinate	-10	0	0	-4	-2	5	0	6	-4	-4	3	-5	-4	-4	6	3	-2		
Distance	12.8	2.0	2.0	4.0	2.0	5.1	4.0	6.3	4.5	5.7	3.2	5.8	5.7	4.5	6.0	3.2	6.3		
Number	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51		
x coordinate	-1	-1	0	2	4	5	-2	0	9	-2	-3	-1	-1	-6	3	4	2		
y coordinate	-7	-3	0	-6	-2	-1	2	8	1	-2	-1	3	-1	6	-5	2	0		
Distance	7.1	3.2	0	6.3	4.5	5.1	2.8	8.0	9.1	2.8	3.2	3.2	1.4	8.5	5.8	4.5	2.0		

Number of stagers: **0**

Average distance: **4.6**

Graph display:

x = 0, y = 0

51th

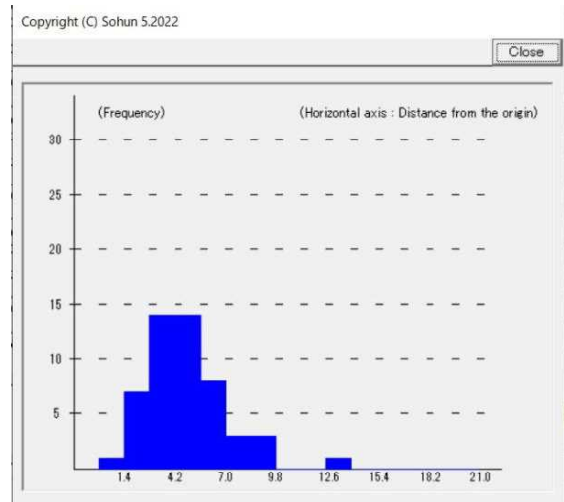
It takes a little time ...

Experiment Start

Experiment Stop

Plane movement

Linear movement



⑥ 6th experiment (When moving on a plane)

Random walk Copyright (C) Sohun 5.2022

Exit	Random walk																	Help	Init
Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
x coordinate	0	-2	-9	-3	-7	2	-4	-1	6	-2	7	-6	-8	-2	3	2	5		
y coordinate	2	-4	1	5	-5	-4	-2	7	10	2	-1	4	4	0	3	-2	3		
Distance	2.0	4.5	9.1	5.8	8.6	4.5	4.5	7.1	11.7	2.8	7.1	7.2	8.9	2.0	4.2	2.8	5.8		
Number	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34		
x coordinate	2	-1	2	-7	-13	-1	1	-3	8	8	6	-2	-2	-5	2	2	-5		
y coordinate	6	1	6	-3	1	1	-7	-1	-4	2	-4	0	2	-1	0	-4	-1		
Distance	6.3	1.4	6.3	7.8	13.0	1.4	7.1	3.2	8.9	8.2	7.2	2.0	2.8	5.1	2.0	4.5	5.1		
Number	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51		
x coordinate	1	-4	-1	2	-9	1	0	6	4	0	-4	-3	3	7	3	-4	-3		
y coordinate	1	0	11	0	3	1	-4	4	0	4	-8	-7	-3	-1	-1	4	-5		
Distance	1.4	4.0	11.0	2.0	9.5	1.4	4.0	7.2	4.0	4.0	7.2	7.8	4.2	7.1	3.2	5.7	5.8		

Number of stagers: **0**

Average distance: **5.5**

Graph display:

x = 0, y = 0

51th

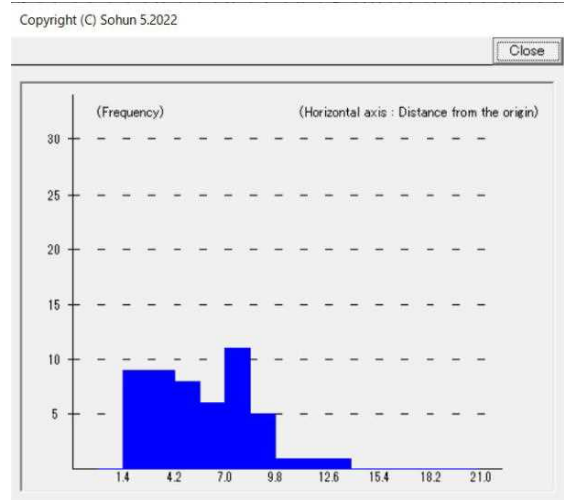
It takes a little time ...

Experiment Start

Experiment Stop

Plane movement

Linear movement



Modeling and Simulation II

2.27.2024
Sohun

2 Pachinko Ball Falling Experiment

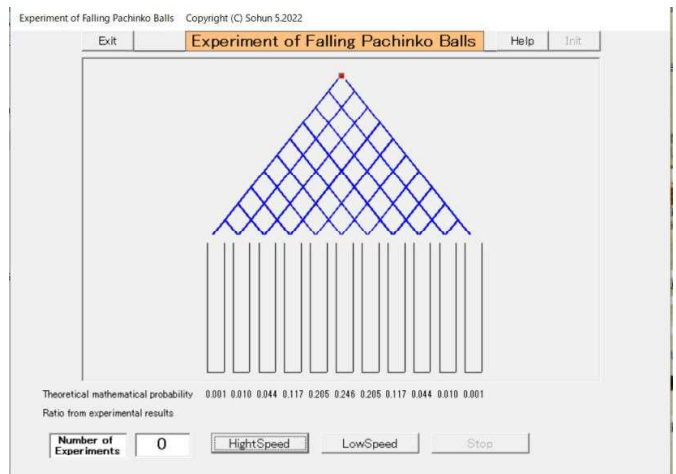
(1) Experiment overview

When a pachinko ball hits nails and falls , splitting into left and right sides , where does it fall ?

A pachinko ball (red dot) falls from the top along a blue line along the path shown in the image on the right.

However , when a pachinko ball splits into left and right at a fork , it is assumed that the split is 50-50 and 50-50.

Are there any places where pachinko balls are likely to fall ? Or is the ease of falling the same everywhere ?



(2) Experimental result (VB version simulation)

【Experiment day】

February 27 . 2024

【PC used】

Lavie NX850/N

【Software used】

Self-made software

『pachinko ball 6』

【Method of operation】

- Compare the theoretical numbers calculated using mathematical probability and the percentages from experiment results.

Click the [HighSpeed] or [LowSpeed] button and the pachinko balls will start falling.

Click the [Stop] button and the pachinko balls will stop falling.

【Consideration】

From the left , the positions where the pachinko balls fall are ①,②,③, ⋯ ,⑨,⑩,⑪.

The probability of falling to the kth position from the left is calculated by ${}_{10}C_{k-1} \left(\frac{1}{2}\right)^{10}$ ※

Calculated using the formula ※ ,

The mathematical probability of falling on ① is 0.001.

The mathematical probability of falling on ② is 0.010.

The mathematical probability of falling on ③ is 0.044.

The mathematical probability of falling on ④ is 0.117.

The mathematical probability of falling on ⑤ is 0.205.

The mathematical probability of falling on ⑥ is 0.246.

The mathematical probability of falling on ⑦ is 0.205.

The mathematical probability of falling on ⑧ is 0.117.

The mathematical probability of falling on ⑨ is 0.044.

The mathematical probability of falling on ⑩ is 0.010.

The mathematical probability of falling on ⑪ is 0.001.

From the above calculation results , position ⑥ is the place where it is most likely to fall. Position ⑤ and ⑦ are the second most likely to fall. Position ④ and ⑧ are the third most likely to fall. Position ③ and ⑨ are the fourth most likely to fall. Position ② and ⑩ are the fifth most likely to fall. Position ① and ⑪ are the least likely to fall.

Therefore , it can be seen that the three experimental results (a) , (b) and (c) are consistent with the mathematical probabilities mentioned above.

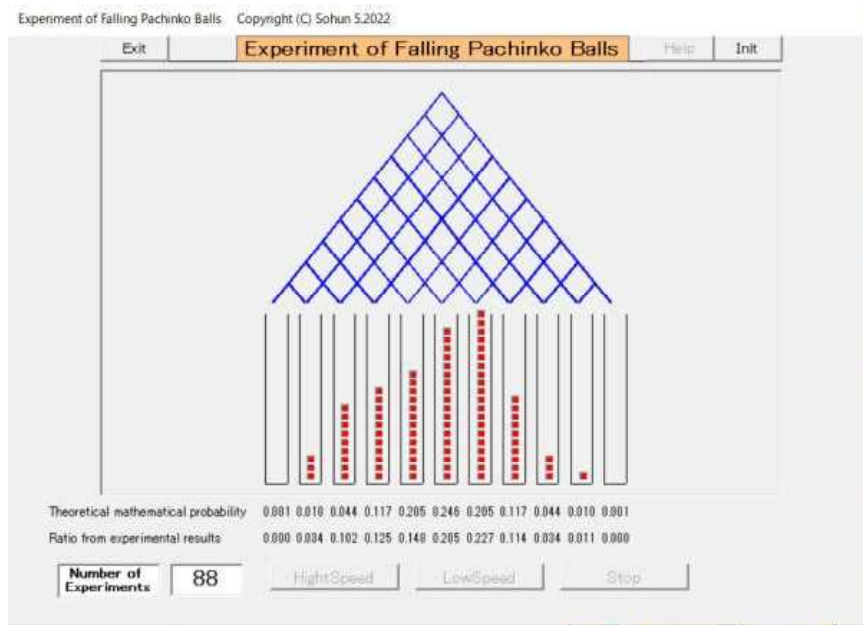
Modeling and Simulation II

2.27.2024
Sohun

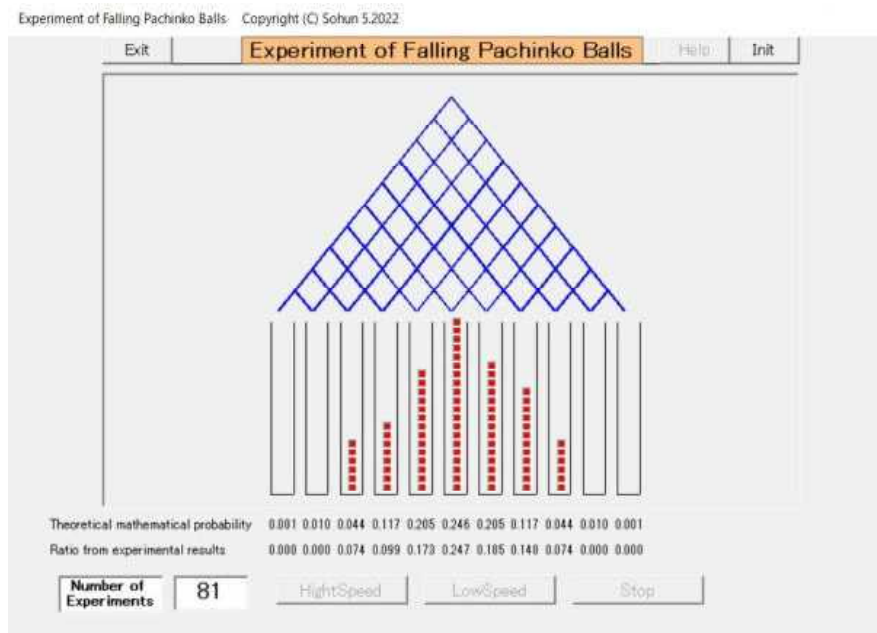
2 Pachinko Ball Falling Experiment

(2) Experimental result (VB version simulation)

(a) 1st experiment



(b) 2nd experiment



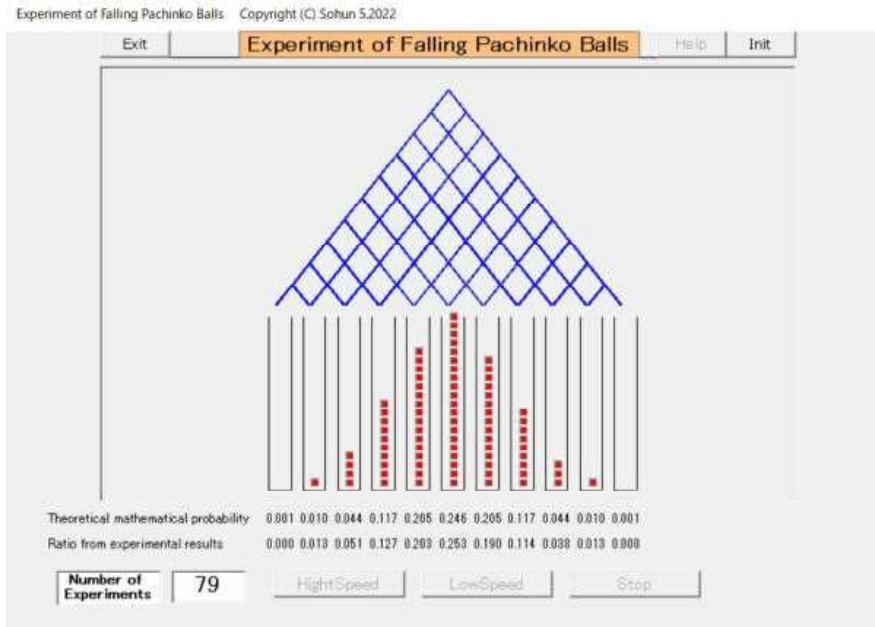
Modeling and Simulation II

2.27.2024
Sohun

2 Pachinko Ball Falling Experiment

(2) Experimental result (VB version simulation)

(c) 3rd experiment



Modeling and Simulation II

2.29.2024
Sohun

3 Even If You Are Not Good at Shooting Guns , If You Shoot a Lot , You Will Hit the Target.

(1) Experiment overview

Have you ever heard the word "Even if you are not good at shooting guns , if you shoot a lot , you will hit the target." ? Suppose that a person fires a gun that is capable of hitting once out of every 10 shots. Mathematically , to find the probability of hitting at least once by shooting a gun several times , you can calculate it by subtracting the probability of not hitting at all from 1. Please try experiencing "Even if you are not good at shooting guns , if you shoot a lot , you will hit the target." Also , compare the probability of hitting at least once between the theoretical probability calculated mathematically and the experimental result.

(2) Experimental result (VB version simulation)

【Experiment day】

February 29 . 2024

【PC used】

Lavie NX850/N

【Software used】

Self-made software

『bad gun 6』

【Concept of experiment】

The person who has the skill to hit once out of 10 shots will shoot the gun. For example , consider the case where 20 shots are fired in on experiment. If at least one of the 20 shots hits , it is considered a "success" , otherwise it is considered a "failure". Perform the experiment several times and find the probability of hitting at least once "(number of successes) ÷ (number of experiments)".

【Method of operation】

■ Compare the theoretical values and the experimental results.

[the setting windows for the number of times the gun is fired] Enter the number of times the gun is fired in one experiment as an integer , and click the [OK] button.

[Firing] Click the button to perform the experiment manually.

[AutoStart] Click the button to perform the experiment automatically.

[AutoStop] Click the button to stop the gun from automatically firing.

[Init] Click the button to restart the experiment from the beginning.

【Consideration】

The person who has the skill to hit once out of 10 shots will shoot the gun. The probability of hitting for each shot is $\frac{1}{10}$. The probability of missing for each shot is $\frac{9}{10}$. The probability that all 20 shots miss is $\left(\frac{9}{10}\right)^{20}$. Therefore , the probability that at least one shot out of 20 will

hit is $1 - \left(\frac{9}{10}\right)^{20} = 0.8784233$.

In experiment ① , (number of successes)/(number of experiments) = 0.8750000.

In experiment ② , (number of successes)/(number of experiments) = 0.8727273.

When firing 20 shots , the theoretical value and experimental results were close to each other (0.87).

By the way , the probability that at least one shot out of 50 will hit is $1 - \left(\frac{9}{10}\right)^{50} = 0.9948462$.

I think it says "Even if you are not good at shooting guns , if you shoot a lot , you will hit the target."

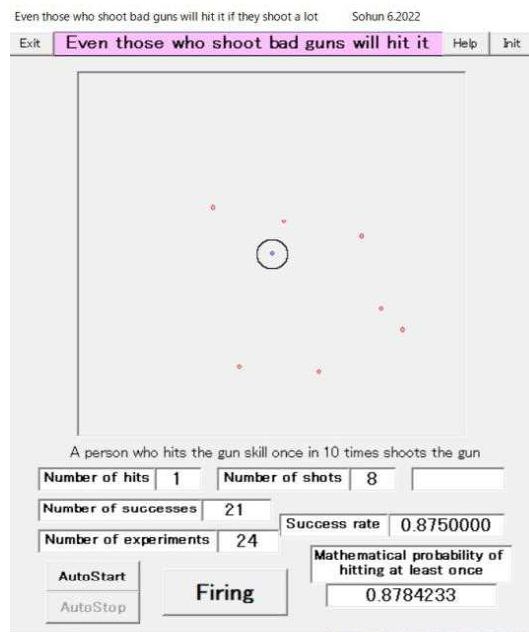
Modeling and Simulation II

2.29.2024
Sohun

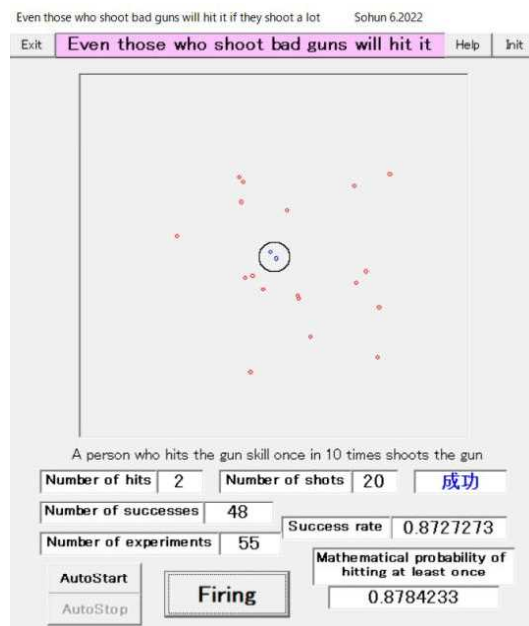
3 Even If You Are Not Good at Shooting Guns , If You Shoot a Lot , You Will Hit the Target.

(2) Experimental result (VB version simulation)

① 1st experiment



② 2nd experiment



Modeling and Simulation II

3.1.2024
Sohun

4 A Dog Chasing a Mouse

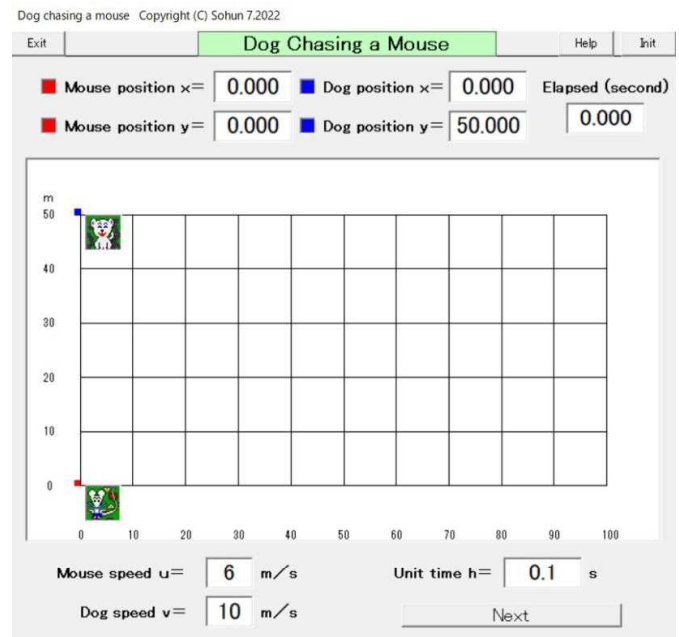
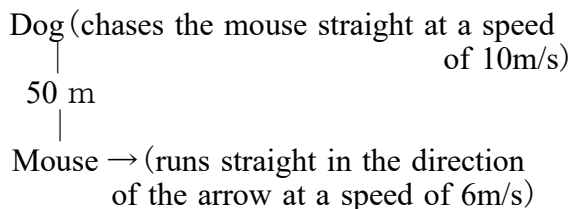
(1) Experiment overview

As shown in the figure below, the mouse and dog are initially 50 meters apart.

The mouse runs straight in the direction of the arrow at a speed of 6m/s. The dog chases the mouse straight at a speed of 10m/s.

What path does the dog follow when chasing the mouse?

Also, how many seconds does it take for the dog to catch up with the mouse?



(2) Experimental result (VB version simulation)

【Experiment day】

March 1 . 2024

【PC used】

Lavie NX850/N

【Software used】

Self-made software

『dog chasing mouse 6』

【Method of operation】

The dog and mouse move each time you click the [Next] button.

Click the [Init] button to return the dog and mouse to their original positions.

You can change the number in the text box for Unit time h=[] s.

【Consideration】

The path of the dog chasing the mouse is now a convex downward curve instead of a straight line.

In experiments ① to ④, observations were made every 0.1 seconds until 7.7 seconds had elapsed. After 7.7 seconds had elapsed, in experiment ④, the x-coordinate of the dog's position was 45.873, and the x-coordinate of the mouse's position was 46.200. The difference in distance between the dog and the mouse is 0.327, and the dog has not yet caught up with the mouse.

After 7.7 seconds, observations were made every 0.001 seconds. In experiments ⑤ after 7.781 seconds have passed, the x-coordinate of the dog's position is 46.683, and the x-coordinate of the mouse's position is 46.686. The difference in distance between the dog and the mouse is 0.003, and the dog has not yet caught up with the mouse.

In experiment ⑥ after 7.782 seconds have passed, the x-coordinate of the dog's position is 46.693, and the x-coordinate of the mouse's position is 46.692. The dog's x-coordinate exceeded the mouse's x-coordinate by 0.001, and the dog overtook the mouse.

Therefore, the dog will catch up with the mouse after 7.782 seconds.

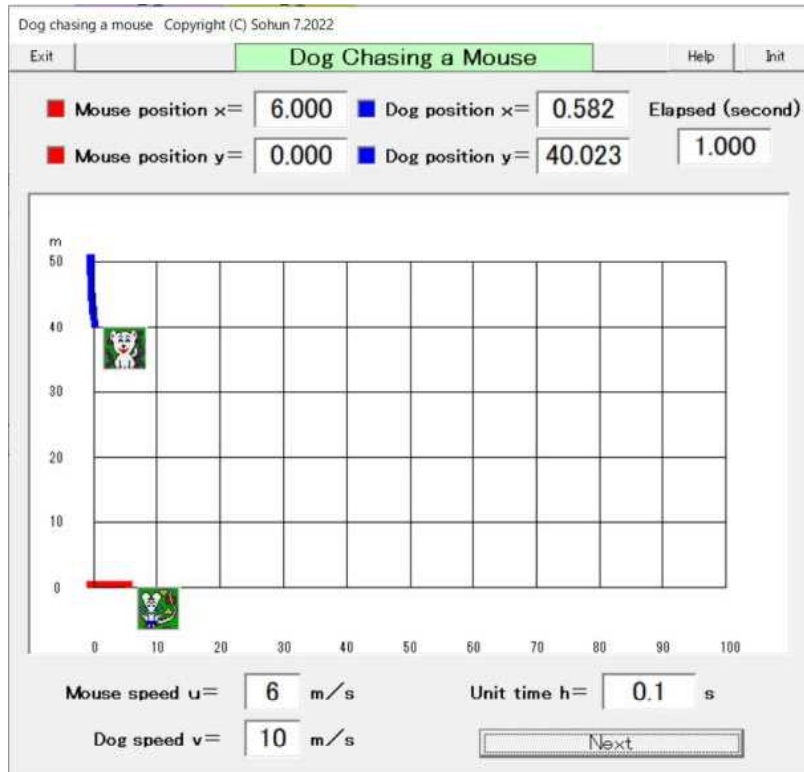
Modeling and Simulation II

3.1.2024
Sohun

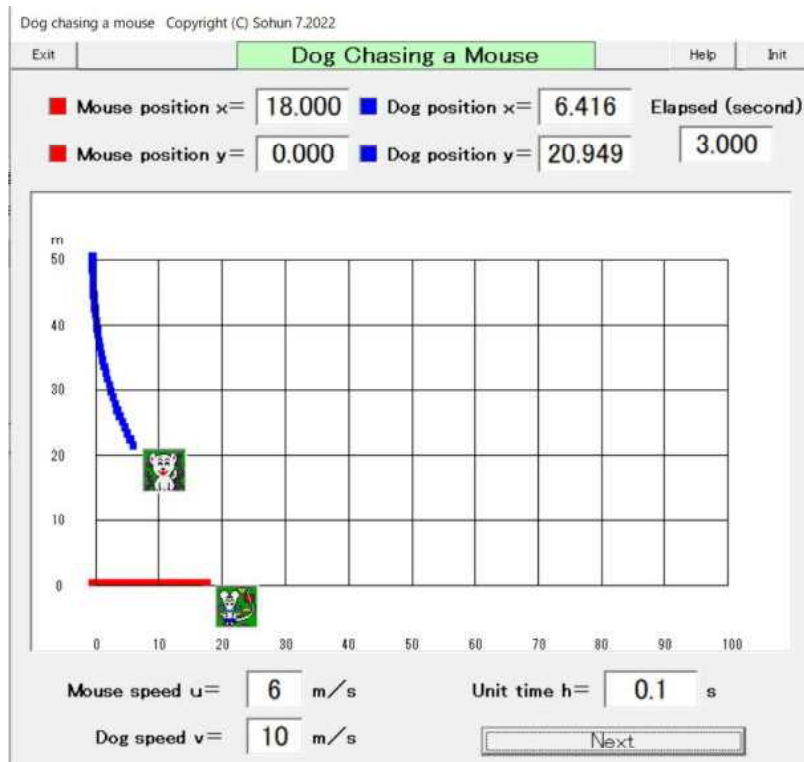
4 A Dog Chasing a Mouse

(2) Experimental result (VB version simulation)

① After 1 second



② After 3 second



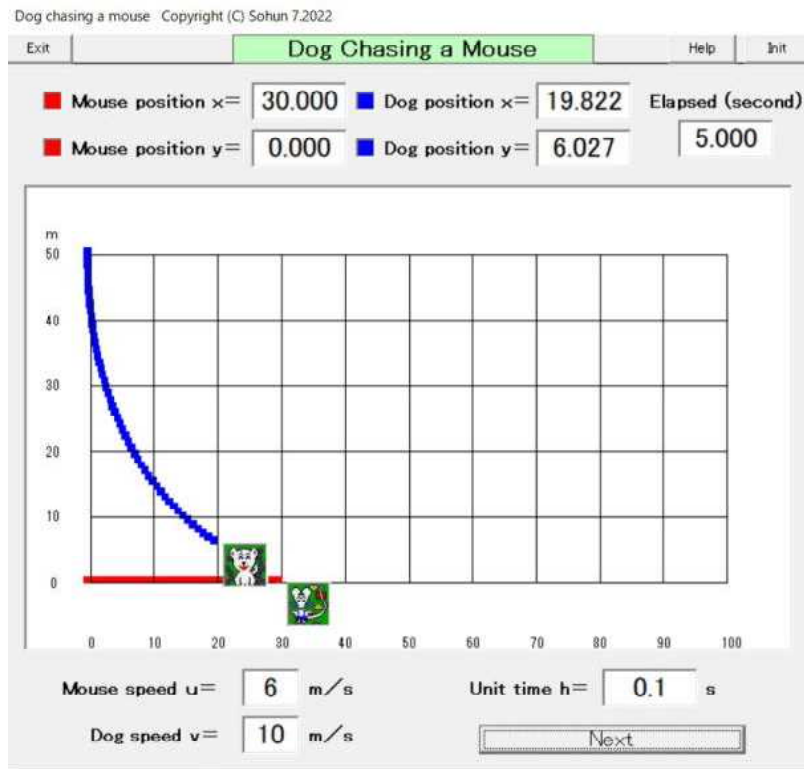
Modeling and Simulation II

3.1.2024
Sohun

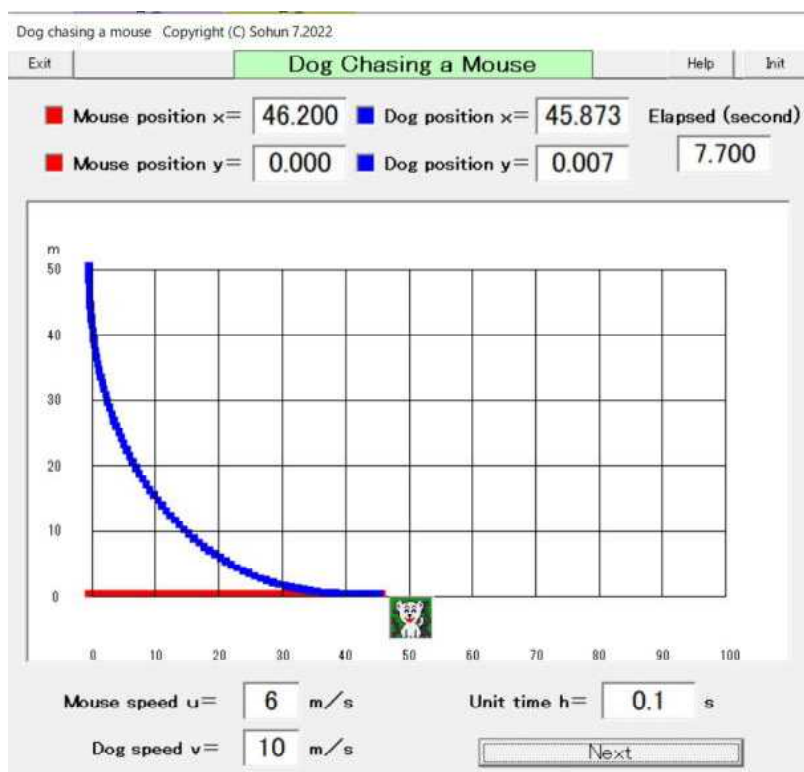
4 A Dog Chasing a Mouse

(2) Experimental result (VB version simulation)

③ After 5 second



④ After 7.7 second



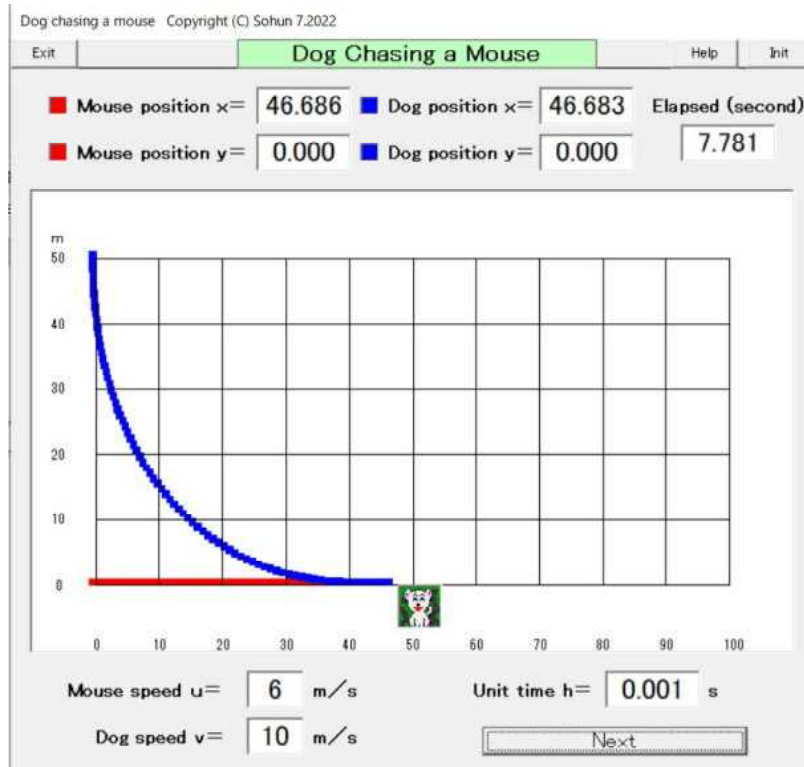
Modeling and Simulation II

3.1.2024
Sohun

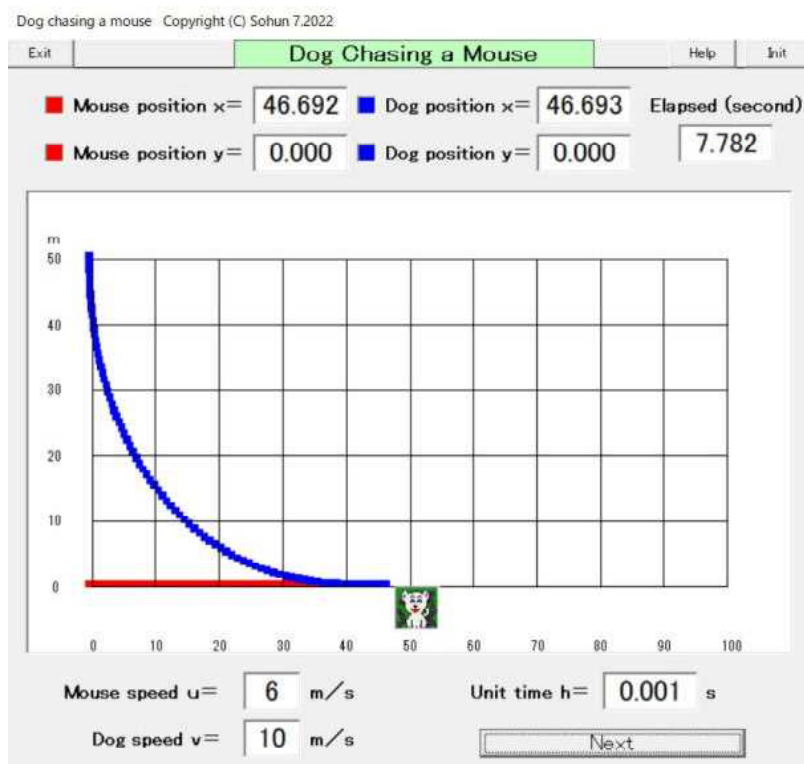
4 A Dog Chasing a Mouse

(2) Experimental result (VB version simulation)

⑤ After 7.781 second



⑥ After 7.782 second (caught up !!)



Modeling and Simulation II

3.2.2024
Sohun

5 Is Amidakuji Equal ?

(1) Experiment overview

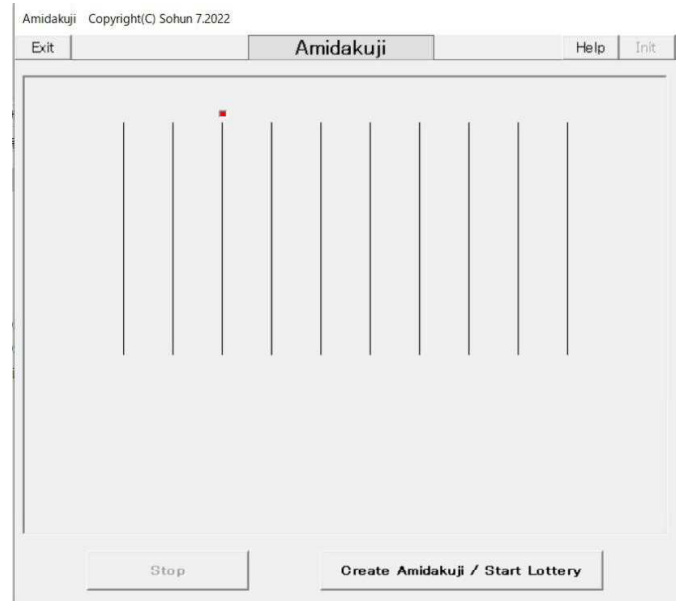
Do you ever use "Amidakuji" for lottery ?
By the way , is "Amidakuji" equal in all respects ?

If there is no equality , what can we say ?
Create an Amidakuji by randomly drawing 10 vertical lines and 50 horizontal lines as shown in the image on the right.

Assume that the red dot at the top of the third vertical line from the left is the winning position.

Select one end of the vertical line and draw the Amidakuji.

When you win the Amidakuji , a red dot will be displayed below the vertical line.



(2) Experimental result (VB version simulation)

【Experiment day】

March 2 . 2024

【PC used】

Lavie NX850/N

【Software used】

Self-made software

『amidakuji 6』

【Method of operation】

In the setting window for the number of horizontal lines in Amidakuji , enter the number of horizontal lines in Amidakuji in half-width characters , and click [OK] button.

Click the [Create Amidakuji / Start Lottery] button to start the experiment.

Click the [Stop] button to stop the creation and lottery of Amidakuji.

Click the [Init] button to restart the experiment from the beginning.

【Consideration】

The experiment was conducted 10 times.

Each experiment was run until the red dot (representing a hit) could no longer be displayed.

Set the positions of the bottomend of the vertical lines ①,②,③,⋯,⑩ from the left.

In each of 10 experiments , the position with the most hits was given +3 points , the position with the second most hits was given +2 points , and the position with the third most hits was given +1 point. The total for 10 times was 15 points for ① , 20 points for ② , 12 points for ③ , 10 points for ④ , 3 points for ⑤ , 2 point for ⑥ , 1 point for ⑦ , and 0 points for ⑧-⑩.

In addition , in each of the 10 experiments , only the position where the number of hits was the highest was given +1 point.

The total for 10 times was 2 points for ① , 5 points for ② , 1 points for ③ , 2 point for ④ , and 0 points for ⑤-⑩.

From these results , you can see that positions ① , ② , ③ , and ④ are easy to hit.

Based on the winning position (the top edge of the third vertical line from the left) , it is predicted that the side with fewer vritical lines will be more likely to win.

However , this is only true if you know the location of the hit.

In reality , since we don't know the location of the hit , we can say that "Amidakuji is equal".

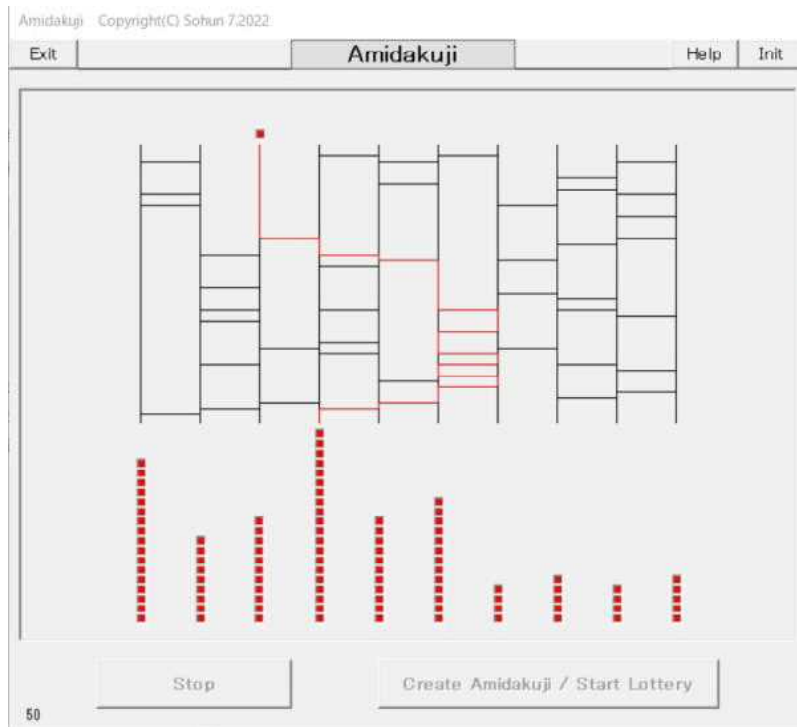
Modeling and Simulation II

3.2.2024
Sohun

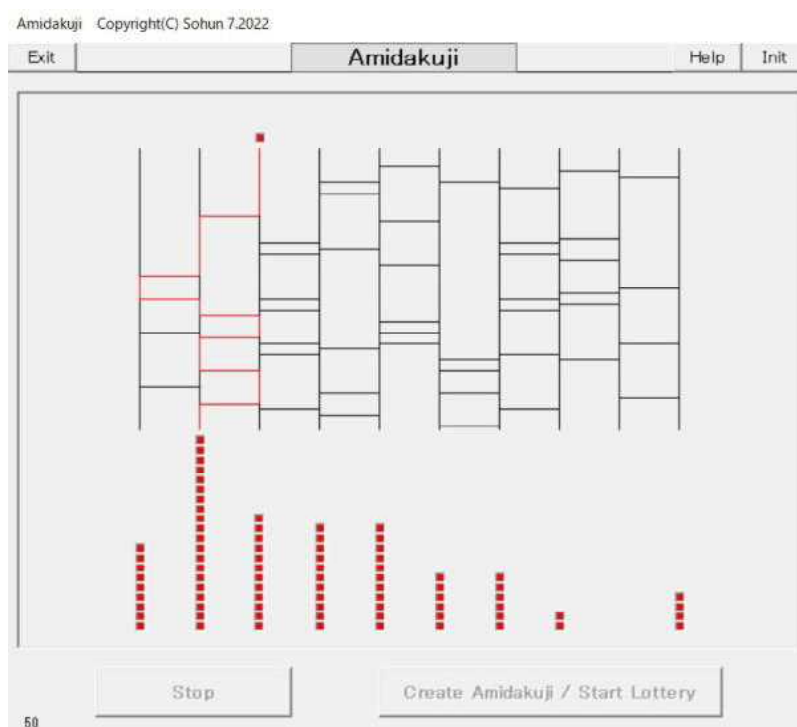
5 Is Amidakuji Equal ?

(2) Experimental result (VB version simulation)

(a) 1st experiment



(b) 2nd experiment



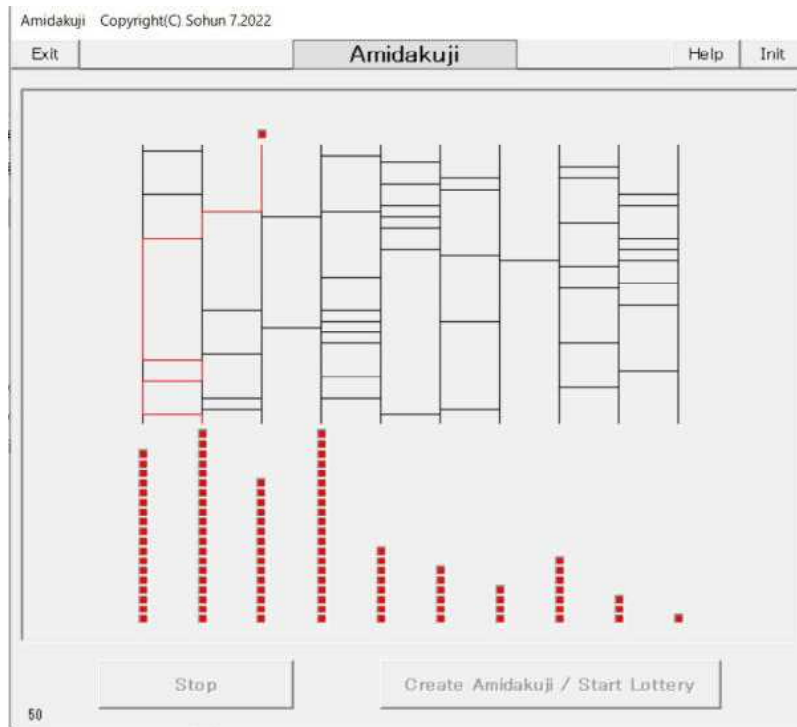
Modeling and Simulation II

3.2.2024
Sohun

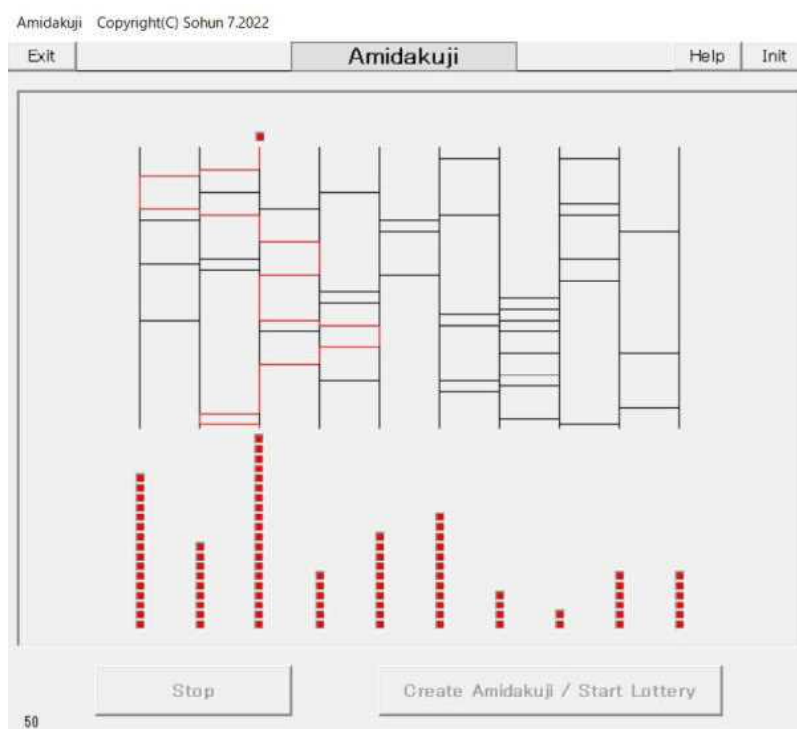
5 Is Amidakuji Equal ?

(2) Experimental result (VB version simulation)

(c) 3rd experiment



(d) 4th experiment



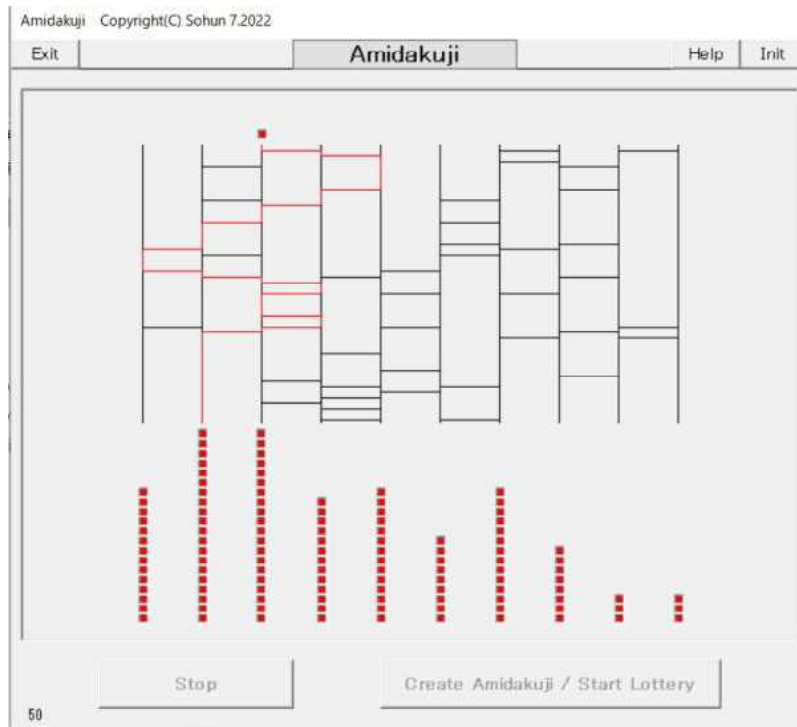
Modeling and Simulation II

3.2.2024
Sohun

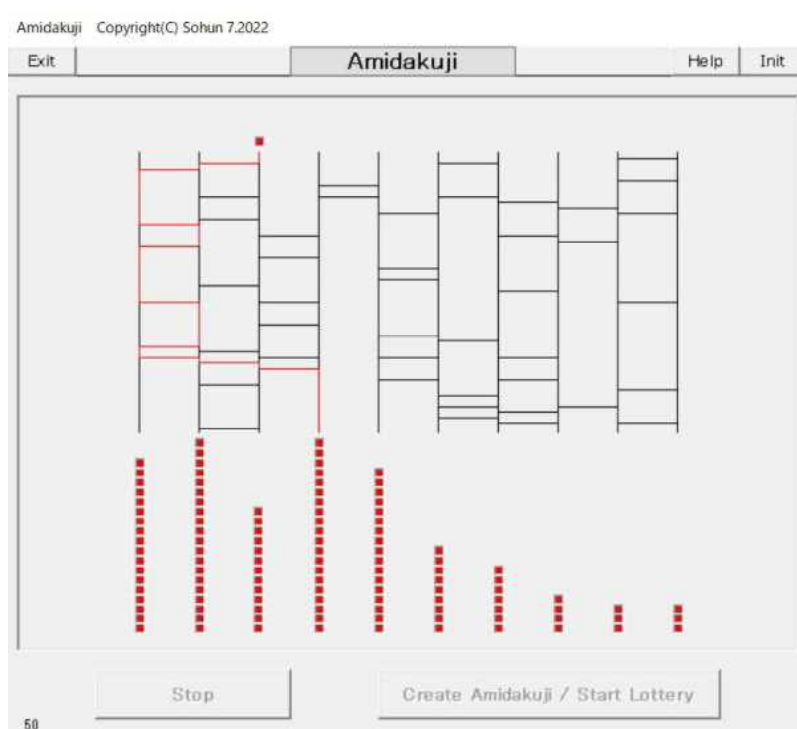
5 Is Amidakuji Equal ?

(2) Experimental result (VB version simulation)

(e) 5th experiment



(f) 6th experiment



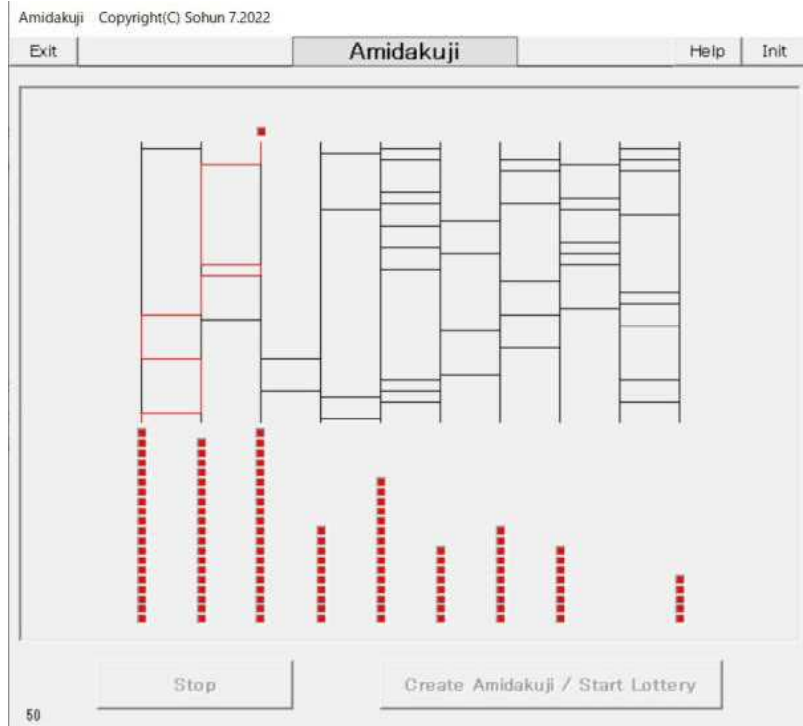
Modeling and Simulation II

3.2.2024
Sohun

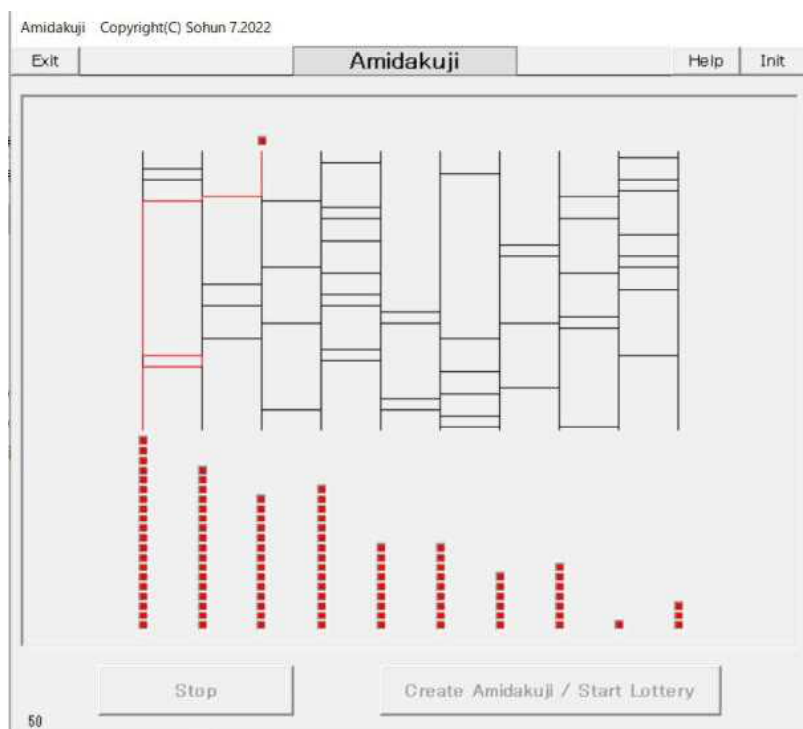
5 Is Amidakuji Equal ?

(2) Experimental result (VB version simulation)

(g) 7th experiment



(h) 8th experiment



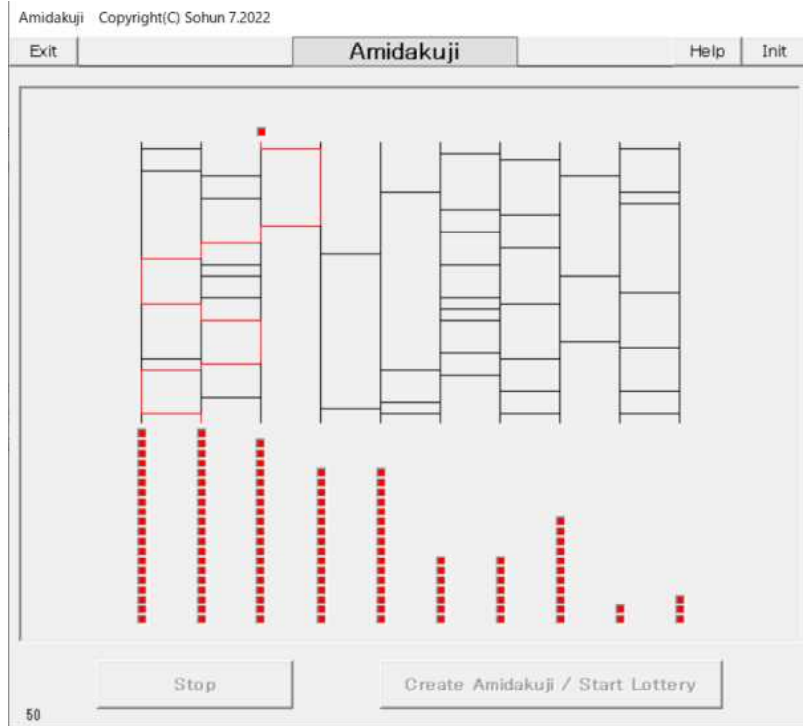
Modeling and Simulation II

3.2.2024
Sohun

5 Is Amidakuji Equal ?

(2) Experimental result (VB version simulation)

(i) 9th experiment



(j) 10th experiment

