3 Outer Billiard

Marcel Vosshans

September 21, 2017

Abstract

In the following report, we analyse the trajectory of a billiard ball if the ball tangents a corner of an equilateral triangle and moves the same distance forward, it needed to the corner. The question is: Can the ball escape? A Mathematica model simulates this model with the result, that the ball normally cant escape from the triangle. This is a simplified model about our solar system and the possibility of an escaping planet based on a billiard table.

Contents

1 Introduction

Imagine a billiard table with a triangle on it. A billiard ball runs toward to one of the corners with constant speed $v1$. The way $d1$ it needs to the corner, is also the distance it runs forward (from the corner away). Then the ball targets the next corner and got a new distance to the corner: $d2$ which is also the new distance from the corner away. This procedure it makes n times. The total distance amount to:

$$d_{sum} = \sum_{i=1}^{n} 2 \cdot d_i \tag{1}$$

The idea of the mathematical tool to use is the point reflection in a point unequal to the source(corner). The goal of the model is to make a statement:

Can the billiard ball escape from the triangle (divergent) or is there a border to reach which would be the local maximum circle around the triangle (convergent)? This theory describes a mathematical model of the solar system and the possibility of an escaping planet (for example the Pluto [currently a dwarf planet]). From the physical point of view can both cases signify that the planet escapes, it depends on the gravity of the solar system.

1.1 Setup

The CAS Mathematica from Wolfram is a very powerful tool to create Models like the regarded billiard table. So, for this model the latest (2017.09.21) version of Mathematica (v11.1.1.0) is used. The kernel runs on a Windows 10- 64bit OS.

1.2 Realization in Mathematica

The entire program is spitted in 4 basic parts as follows:

a) **Basic Framework** In the first part is it important to create the initial situation. The $circlePoint[n]$ function gives the positions of n points equally spaced around the unit circle [1]. With these three points is the construction of a triangle, with a side length of $d_1 = 1$, done. Further, will these points be saved to A, B and C. To detect where the starting point is, are the equations for the three sides of the triangle also needed:

$$f(x) = -0.5 \tag{2}$$
$$g(x) = 1 - \sqrt{3} \cdot x \tag{3}$$
$$h(x) = 1 + \sqrt{3} \cdot x \tag{4}$$

b) **Starting Point Initialization** The requirements concerning the model is that the starting point can be initialized everywhere outside the triangle. So, at first is here the area to fill in the changeable data for the Model: x-coordinate, y-coordinate and the number of repetitions. The validation follows in the next step. If the coordinates are inside the triangle, then will be the calculation step [c)] be skipped. The starting point$K$ must fulfil the following inequation:

$$K_y \leq h(K_x) \wedge K_y \leq g(K_x) \wedge K_y \geq f(K_x) \tag{5}$$

c) **Point Reflection Calculation** Now will the model calculate the points depending on the number of repetitions. The points are stored in a point list which is also the order for the plot connections because $AppendTo[]$ appends always the last point which is calculated each step.

Another difficult part is to find the right corner to reflect the current point P. Three inequations:

$$P_y < h(P_x) \wedge P_y \geq g(P_x) \rightarrow A \tag{6}$$
$$P_y \geq h(P_x) \wedge P_y > f(P_x) \rightarrow B \tag{7}$$
$$P_y < g(P_x) \wedge P_y \leq f(P_x) \rightarrow C \tag{8}$$

The position of the starting/current point is thus unique.

d) **Output** Now the model got the triangle, the starting point and the consequent points. The last step just plots the model and sets some preferences for the display.

2 Results

Basically, are there three types of trajectories. The depending factors for a different result are the

a) distance to the next corner
b) number of repetitions
c) position to the triangle

It is another type of trajectory if the pattern is different from the expected. The mainly factor is the distance to the next corner. Here are two options: close to the triangle (the point is located in the range of the dimensions of the triangle) or far away from the next corner (factor 100 and up):

2.1 Starting point in close vicinity to the triangle

The normal expected situation is to use the model in proportional way. So, we got a triangle with a side length of $d_1 = 1$, therefore it is a good start to locate the starting point also with a distance of $d_1 = 1$ to one of the corners. We start also with a relatively low number of repetitions in this case with: $n = 25$. It seems to be a loop after a while because there are not 25 lines.

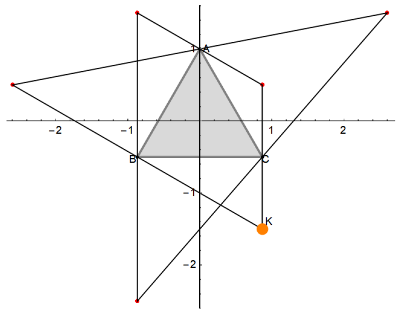

Figure 1: Close position of the point K with $d_1 = 1$ and $n = 25$

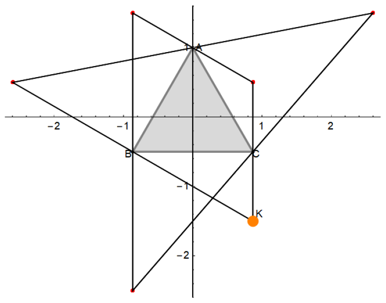

Figure 2: Close position with $d_1 = 1$ and $n = 1000$

Another try with $n = 1000$ repetitions tells more: So, both figures have the same pattern: it is a loop. First statement: the ball cant escape because it got only one trajectory, which it cant leave. Another starting point position approves it. It has another pattern (expected) and the loop has a longer way but it remains a loop:

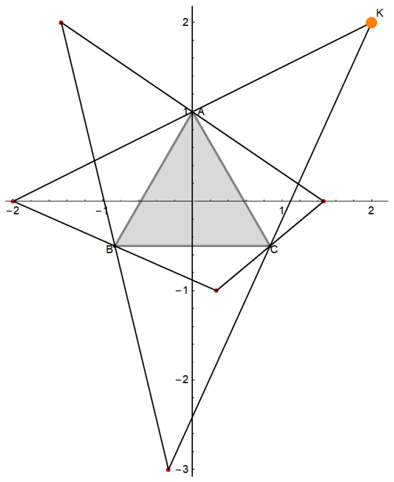

Figure 3: Another "normal" position of the start with $n = 1000$

2.2 Starting point is far away from the triangle

3 Discussion

References

[1] Wolfram Language & System, `http://reference.wolfram.com/language/`, 2017.09.20.