

# Outer Billiard Report

Evan Huynh, Marcel Vosschans

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 can't 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

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Method</b>	<b>2</b>
2.1	Implementing the reflectPoint function . . . . .	2
<b>3</b>	<b>Results</b>	<b>2</b>
<b>4</b>	<b>Discussion</b>	<b>2</b>

## 1 Introduction

Imagine a billiard table with a triangle on it. A billiard ball runs toward to one of the corners with constant speed  $v_1$ . The way  $d_1$  it needs to the corner, is also the distance it runs forward (from the corner). Then the ball targets the next corner and got a new distance to the corner  $d_2$ , which is also the new distance away from that corner. This procedure it makes  $n$  times. The total distance amount to:

$$n = d_1 + d_2 + \cdots + d_n$$

The question is: 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 an the possibility of an escaping planet (for example the Pluto [currently a dwarf planet]).

## 2 Method

In Mathematica, we first create a triangle by using CirclePoint[3]. This gives an output of three points needed to generate an equilateral triangle  $\triangle ABC$ , with the circumcenter at  $(0,0)$ . Then, the initial point  $K$  locates somewhere outside of  $\triangle ABC$ .

### 2.1 Implementing the reflectPoint function

According to the instruction, the ball moves toward a corner with distance  $d_1$ , then move another distance  $d_1$  before moving toward another corner without crossing the triangle. It can be solved by using vector. Since the ball has moved with a vector  $q\vec{i} + p\vec{j}$  toward one corner, it also move  $q\vec{i} + p\vec{j}$  away from it.

## 3 Results

## 4 Discussion

The program can be repeated as many times as possible within the degree of computational power. None the less, we checked all the possible location of  $K$ ; no single line had crossed the triangle.

As seen from the result by our calculations, the ball always returns to the initial point  $K$  after some movement. However, the smallest number of the movement it takes before returning to  $K$  is highly dependent on the coordinate of the original point. The least number of movements increase as  $K$  is nearer the triangle.

Exceptionally, if  $K$  is on a line or coincides with any two points of the triangle, by design of the algorithm, the movement will continue indefinitely in the different direction, meaning the ball will never return to  $K$  if we let the ball moves forever. Note that the program will not run if  $K$  is inside the triangle.

## References

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