

## 弹道反解及其数值解求解程序

# 目录

## 1 问题描述

### 1.1 背景

### 1.2 需求

## 2 问题建模

### 2.1 情景分析

### 2.2 正向求解

### 2.3 逆向求解

## 3 数值求解

### 3.1 求解分析

### 3.2 求解代码

### 3.3 误差分析

### 3.4 性能优化

## 4 附录

### 4.1 附录 1 $l(x)$ 函数值表

# 问题描述

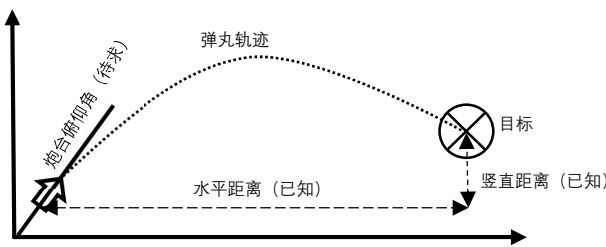
## 背景

在“机甲大师”的比赛中，参赛队伍自制的机器人将互相从炮台中发射弹丸，射击敌方机器人的装甲板，从而对敌方机器人造成伤害。现有自动战斗程序，能够自动控制炮台进行水平面上和俯仰上的旋转，从而自动攻击敌方机器人；炮台上安装的传感器和目标系统能够反馈敌方单位到被控制单位的相对位置，而为了使从炮台发射出球状弹丸能够命中目标，则需要根据弹丸的运动轨迹方程，反解出炮台的发射姿态。

在该问题中，敌方单位到被控制单位的相对位置为已知量，弹丸的出膛速度实际上是一个围绕给定值浮动的随机变量，但由于自动战斗程序通常工作在连发模式下，故简化为已知的变量，其取值由人为确定，通常有 15m/s、18m/s、30m/s 三个档位。由于炮台的水平转角并不涉及到弹道补偿，故不失一般地将其忽略，即在该问题中，总是认为炮台在水平方向上已经正对目标；而本问题中，需要求解的是炮台能够发射弹丸命中目标时的俯仰角。

根据目标的相对位置和弹道模型，反解出发射姿态即“弹道反解”。

其图示如下：



## 需求

### 精准性

在比赛中，只有精准地命中敌方装甲板才能造成有效伤害，而尺寸最小的装甲板仅 12 厘米高、13 厘米宽，加之弹丸发射系统本身存在一定的误差，故该反解模型反解出的精度必须尽可能高。

通常，普通机器人的作战距离为 2 到 6 米，其最低应当满足目标在 6 米以内时，不考虑发射系统误差时，弹丸实际落点与目标的竖直误差在 6 厘米以内。

### 实时性

自动攻击程序的锁定子程序以 1 秒钟 200 次（每次 5 毫秒）的速率更新目标的位置，而炮台控制程序以 1 秒钟 500 次（每次 2 毫秒）的速率对炮台的电动马达施加控制，故该反解程序最好能够在 3 毫秒内完成运算。

该反解模型的数值求解程序将运行在 nVIDIA Jetson TX2 迷你计算机或 STM32F4 微控制器上，其最好能够在微控制器上满足耗时要求，最低要能在迷你计算机上满足耗时要求。

# 问题建模

## 情景分析

**性质 1.1** 弹丸在空中飞行时的阻力可以忽略。

弹丸共分为两种：直径 17mm、重量 4.15g 的小弹丸和直径 42mm、平均重量 40g 的大弹丸。现在使用空气阻力计算公式  $f = 6\pi\eta Rv$  分别计算其最大阻力，其中，<sup>1</sup>空气粘滞系数  $\eta$  取  $1.983 \times 10^{-5} Pa \cdot s$ 。

在 30m/s 的飞行速度下，小、大弹丸均受到其三个速度档位的最大空气摩擦力，分别计算如下：

$$f_{small} = 6\pi \times 1.983 \times 10^{-5} \times \frac{1}{2} \times 17 \times 10^{-3} \times 30 \approx 9.532 \times 10^{-5} N,$$

$$f_{big} = 6\pi \times 1.983 \times 10^{-5} \times \frac{1}{2} \times 42 \times 10^{-3} \times 30 \approx 4.710 \times 10^{-4} N.$$

取  $g = 9.8N$ ，由于  $f_{small} \ll g$ ,  $f_{big} \ll g$ ，故忽略大、小弹丸在飞行过程中的空气阻力的影响，可视为其仅受到重力影响。

**性质 1.2** 弹丸在飞行过程中的马格努斯效应力可以忽略。

根据<sup>2</sup>球体马格努斯效应力的计算公式  $f = \frac{8}{3}\pi\omega v R^3\rho$ ，空气流体密度取  $\rho = 1.29 kg/m^3$ ，

取其一部分  $p = \frac{8}{3}\pi v R^3\rho$ ，则  $f = p\omega$ 。计算小、大弹丸的  $p$  系数如下：

$$p_{small} = \frac{8}{3}\pi v R_{small}^3\rho \approx 1.9910 \times 10^{-4},$$

$$p_{big} = \frac{8}{3}\pi v R_{small}^3\rho \approx 3.0025 \times 10^{-3},$$

$$\omega = \frac{2\pi}{T}.$$

若马格努斯力达到  $1 \times 10^{-2}$  级别，则有分别有  $\omega_{small} > 10^2$ ,  $\omega_{big} > 10$ ，即  $T_{small} < 0.0628s$ ,  $T_{big} < 0.628s$ 。

当前的发射系统为在弹路水平面上左右各安装一个摩擦轮，分别以逆时针和顺时针的方向相对转动，从而将弹丸挤出，该机构不刻意使弹丸旋转，弹丸仅可能因其下表面与枪管接触而发生旋转，由于弹丸上表面在设计上不存在挤压，故可认为其下表面摩擦力较小，该旋转不会达到上述的较高转速，从而马格努斯效应力可以忽略。

## 弹道方程

综合前述分析，可认为弹丸在室内的飞行过程中仅受重力影响，而忽略其它力的影

<sup>1</sup> 空气粘滞系数来自 [http://www.engineeringtoolbox.com/dynamic-absolute-kinematic-viscosity-d\\_412.html](http://www.engineeringtoolbox.com/dynamic-absolute-kinematic-viscosity-d_412.html)

<sup>2</sup> 马格努斯效应力公式来自 <https://www.grc.nasa.gov/WWW/K-12/airplane/beach.html>

响。则设： $\theta$ 为云台的俯仰角，以水平为 $0^\circ$ ，以向上竖直为 $90^\circ$ ； $v$ 为弹丸的出膛速度；弹道轨迹上任一点距离发射点水平距离 $d$ ，竖直距离 $h$ ，中间变量 $t$ 为从出膛到该位置的飞行时间，则建立如下的弹道方程：

$$\begin{aligned}d &= vt \cos \theta, \\h &= vt \sin \theta - \frac{1}{2}gt^2.\end{aligned}$$

## 正向求解

正向求解弹道方程，即以出膛速度 $v$ 、炮台俯仰角 $\theta$ 为自变量，以水平距离 $d$ 、竖直距离 $h$ 为因变量，通过参数 $v, \theta$ 得到弹道轨迹曲线，并研究该轨迹曲线的性质。

**性质 2.1** 轨迹水平距离 $d_M = \frac{v^2 \sin 2\theta}{2g}$ 的点的竖直距离为最大值 $h_{max} = \frac{v^2 \sin^2 \theta}{2g}$ 。

由弹道方程，得竖直距离对 $t$ 的一阶导数为：

$$\frac{dh}{dt} = v \sin \theta - gt$$

当 $h$ 取最值时，有 $\frac{dh}{dt} = 0$ ，即 $v \sin \theta = gt$ ；其中， $t = \frac{d}{v \cos \theta}$ ，则联立这些式子，得：

$$d = \frac{v^2 \sin \theta \cos \theta}{g}$$

化简，得：

$$d = \frac{v^2 \sin 2\theta}{2g}$$

将 $t = \frac{v \sin \theta}{g}$ 代入 $h = vt \sin \theta - \frac{1}{2}gt^2$ ，可得竖直高度的最大值 $h = \frac{v^2 \sin^2 \theta}{2g}$ 。同时，该

性质可直接得到后文中的性质 2.2。

即弹丸在该点之前的轨迹的竖直距离处于单调递增阶段，称该阶段为**上升段**，而在该点之后的轨迹上的点的竖直距离处于单调递减阶段，称该阶段为**下降段**；称该点，即上升段与下降段的分界点为**轨迹转折点**，记为 $d_M$ 。

在某些目标点上，可能同时存在上升段命中和下降段命中这两条可行的弹道轨迹，而该性质对于这种多解情况的处理而言较为重要。

**性质 2.2** 若某点 $h > \frac{v^2 \sin^2 \theta}{2g}$ 则弹道轨迹不可达该点。

**公式 1.1** 给定轨迹上某一点的水平距离 $d$ ，求其在轨迹上对应的竖直距离 $h$ 。

由弹道方程， $t = \frac{d}{v \cos \theta}$ ，代入 $h = vt \sin \theta - \frac{1}{2}gt^2$ 得：

$$h = d \tan \theta - \frac{gd^2 \sec^2 \theta}{2v^2}$$

**公式 1.2** 给定轨迹上某一点的竖直距离 $h$ ，求其在轨迹上对应的水平距离 $d$ 。

由弹道方程，得 $t$ 关于 $h$ 的根解析式为：

$$t = \frac{v \sin \theta \pm \sqrt{v^2 \sin^2 \theta - 2gh}}{g}, t \geq 0$$

当  $h > \frac{v^2 \sin^2 \theta}{2g}$  时，无解。该点即前文中的性质 2.2。

当  $0 \leq h < \frac{v^2 \sin^2 \theta}{2g}$  时，有  $0 < \sqrt{v^2 \sin^2 \theta - 2gh} < v \sin \theta$ ，则  $t$  存在两个解：

$$t_1 = \frac{v \sin \theta + \sqrt{v^2 \sin^2 \theta - 2gh}}{g}$$

$$t_2 = \frac{v \sin \theta - \sqrt{v^2 \sin^2 \theta - 2gh}}{g}$$

相应地，存在两个对应的  $d$ ：

$$d_1 = vt_1 \cos \theta = \frac{\frac{1}{2}v^2 \sin 2\theta + v \cos \theta \sqrt{v^2 \sin^2 \theta - 2gh}}{g}$$

$$d_2 = vt_2 \cos \theta = \frac{\frac{1}{2}v^2 \sin 2\theta - v \cos \theta \sqrt{v^2 \sin^2 \theta - 2gh}}{g}$$

值得注意的是，由于  $\theta \in [0^\circ, 90^\circ]$ ， $\cos \theta \in [0, 1]$ ，故  $v \cos \theta \sqrt{v^2 \sin^2 \theta - 2gh} \geq 0$ ， $d_1 \geq d_M$ ，即  $d_1$  位于轨迹的上升段；同理， $d_1 \geq d_M$ ， $d_2$  位于轨迹的下降段。该点即后文中的性质 2.3。

当  $h < 0$  或  $h = \frac{v^2 \sin^2 \theta}{2g}$  时，有  $\sqrt{v^2 \sin^2 \theta - 2gh} > v \sin \theta$ ，则  $t$  仅存在一个解：

$$t = \frac{v \sin \theta + \sqrt{v^2 \sin^2 \theta - 2gh}}{g}$$

$$d = vt \cos \theta = \frac{\frac{1}{2}v^2 \sin 2\theta + v \cos \theta \sqrt{v^2 \sin^2 \theta - 2gh}}{g}$$

**性质 2.3** 对于某个竖直距离满足  $0 \leq h < \frac{v^2 \sin^2 \theta}{2g}$  的点，同时存在两条可达该点的弹道轨迹，分别在上升段和下降段到达该点。

## 逆向求解

逆向求解弹道方程，即以水平距离  $d$ 、竖直距离  $h$  为自变量，以出膛速度  $v$ 、炮台俯仰角  $\theta$  为自变量，给定参数出膛速度  $v$  时，根据目标的相对位置  $d, h$  计算出炮台的发射姿态  $\theta$ 。求解的主要依据为由公式 1.1 得到的方程

$$d \tan \theta - \frac{gd^2 \sec^2 \theta}{2v^2} - h = 0$$

**变形**

将  $\tan \theta$  与  $\sec \theta$  分别用  $\sin \theta$  和  $\cos \theta$  表示，并简单化简，得：

$$d \sin \theta \cos \theta - \frac{gd^2}{2v^2} - h \cos^2 \theta = 0$$

因  $\theta \in [0^\circ, 90^\circ]$ , 故  $\sin \theta \geq 0$  将其中的  $\sin \theta$  以  $\sqrt{1 - \cos^2 \theta}$  代替, 并记  $\cos^2 \theta$  为  $x$ , 得到方程:

$$d\sqrt{x - x^2} - hx - \frac{gd^2}{2v^2} = 0, \quad x \in [0, 1]$$

### 分析

该方程的原方程为一元一次三角函数方程, 根据其形式来看, 很可能没有解析式解; 而变形化简后的方程的自变量次数分别为 1 次、0.5 次, 且最为棘手的是  $\sqrt{x - x^2}$ , 根号下同时存在无法约简的不同次数的自变量, 导致该方程很难给出解析解。并且由于  $x \in [0, 1]$ ,  $\sqrt{x - x^2}$  难以转换为易于求解的近似项, 故也很难通过将方程转化为近似方程来求解。

观察该方程发现, 其存在一个单独的 1 次项, 很容易化为  $x = f(x)$  的形式, 故首先考虑采取迭代法的方式进行求解, 转换后得到式子如下:

$$x = \frac{d}{h} \cdot \sqrt{x - x^2} - \frac{gd^2}{2hv^2}$$

对该式进行分析发现,  $f(x) = \frac{d}{h} \cdot \sqrt{x - x^2} - \frac{gd^2}{2hv^2}$  并不满足映内性, 故无法使用牛顿迭代法进行求解。从而使用二分法对该方程进行求解。

使用二分法对该方程进行求解时亦应当注意, 该函数表达式  $f(x)$  并非线性的, 在区间  $[0, 1]$  上也不一定为单调的, 因而不一定能直接从  $[0, 1]$  开始进行二分查找。

## 数值求解

### 求解分析

使用二分法进行求解, 记  $f(x) = d\sqrt{x - x^2} - hx - \frac{gd^2}{2v^2}, x \in [0, 1]$ , 其中  $d$ 、 $h$ 、 $g$ 、 $v$  均为已知量。对其求导, 得导数  $f'(x) = d \frac{1-2x}{\sqrt{x-x^2}} - h$ 。

当  $f'(x) = 0$ , 有  $\frac{1-2x}{\sqrt{x-x^2}} = \frac{h}{d}$ 。等式左右两边同时取平方, 并整理得方程:

$$\left(4 + \frac{h^2}{d^2}\right)x^2 - \left(4 + \frac{h^2}{d^2}\right)x + 1 = 0$$

该方程判别式  $\Delta = \left(4 + \frac{h^2}{d^2}\right)^2 - 4 \cdot \left(4 + \frac{h^2}{d^2}\right) = \frac{h^4}{d^4} + \frac{4h^2}{d^2}$ ,  $\Delta \geq 0$  恒成立。

当  $\Delta = 0$ , 即  $h = 0$ , 此时方程有且仅有一个实根, 此时根据  $f'(x)$  的表达式有  $f'(x) \geq 0$ , 即  $f(x)$  在  $[0, 1]$  上单调递增, 此时可以直接在  $[0, 1]$  上对  $f(x) = 0$  进行二分查找。

更一般的, 当  $\Delta \neq 0$ , 即  $h \neq 0$ , 此时方程有两个不等实根:

$$x_1 = \frac{4 + \frac{h^2}{d^2} + \frac{h}{d}\sqrt{4 + \frac{h^2}{d^2}}}{2 \cdot (4 + \frac{h^2}{d^2})} = \frac{1}{2} \cdot \left(1 - \frac{1}{\sqrt{1 + 4 \cdot \frac{d^2}{h^2}}}\right)$$

$$x_2 = \frac{4 + \frac{h^2}{d^2} - \frac{h}{d}\sqrt{4 + \frac{h^2}{d^2}}}{2 \cdot (4 + \frac{h^2}{d^2})} = \frac{1}{2} \cdot \left(1 + \frac{1}{\sqrt{1 + 4 \cdot \frac{d^2}{h^2}}}\right)$$

因  $\sqrt{1 + 4 \cdot \frac{d^2}{h^2}} \geq 1$  恒成立，故  $x_2 > x_1 \geq 0$ ，均为有效根。

则在  $[0, x_1]$  上， $f'(x) \geq 0$ ， $f(x)$  单调递增；在  $[x_1, x_2]$  上， $f'(x) \leq 0$ ， $f(x)$  单调递减；在  $[x_2, 1]$  上， $f'(x) \geq 0$ ， $f(x)$  单调递增。即  $x_1$  为  $f(x)$  的一个极大值点， $x_2$  为  $f(x)$  的一个极小值点。

根据上述分析的  $f(x)$  的性质，对  $f(x) = 0$  进行二分查找时，应当将完整区间  $[0, 1]$  分割为三段：

$$\left[0, \frac{1}{2} \cdot \left(1 - \frac{1}{\sqrt{1 + 4 \cdot \frac{d^2}{h^2}}}\right)\right], \left[\frac{1}{2} \cdot \left(1 - \frac{1}{\sqrt{1 + 4 \cdot \frac{d^2}{h^2}}}\right), \frac{1}{2} \cdot \left(1 + \frac{1}{\sqrt{1 + 4 \cdot \frac{d^2}{h^2}}}\right)\right], \left[\frac{1}{2} \cdot \left(1 + \frac{1}{\sqrt{1 + 4 \cdot \frac{d^2}{h^2}}}\right), 1\right]$$

综合步骤如下：

1. 根据导数方程计算这三段区间的两个端点的值  $x_1, x_2$ 。
2. 分别计算出  $f(0), f(x_1), f(x_2), f(1)$  的值，从  $\{0, x_1, x_2, 1\}$  选择最小的  $x_{lower}$  和最大  $x_{upper}$  满足  $f(x_{lower}) \cdot f(x_{upper}) \leq 0$ 。
3. 在区间  $[x_{lower}, x_{upper}]$  上对  $f(x) = 0$  进行二分查找。

## 求解代码

### MATLAB 代码

二分查找函数如下：

```
function x= binary_search(f, tol, a, b, d, h, v)
while (f(a, d, h, v)*f(b, d, h, v)<=0) && (abs(a - b) > tol)
    c=(a + b)/2
    if f(c, d, h, v)*f(b, d, h, v)<=0
        a=c;
    else
        b=c;
    end
end
x=(a + b)/2
end
```

被求表达式函数如下：

```
function f = expression(x, d, h, v)
f=d*sqrt(x-x^2)/h-4.9*d^2/(h*v^2)-x;
```



## C 代码

二分查找函数如下：

```
double binary_search(double begin, double end, double
tolerance, double d, double h, double v)
{
    double begin_value = formula(begin, d, h, v),
    end_value = formula(end, d, h, v);
    if (begin_value * end_value > 0.0f){
        return -1.0f;
    }
    double middle = 0, middle_value = 0;
    while (fabs(begin_value - end_value) > tolerance){
        middle = (begin + end) / 2.0f;
        middle_value = formula(middle_value, d, h, v);
        if (middle_value * end_value <= 0){
            begin = middle;
            begin_value = middle_value;
        }else{
            end = middle;
            end_value = middle_value;}
    }
    return middle;
}
```

被求表达式函数如下：

```
double formula(double x, double d, double h, double v)
{
    return d*sqrt(x - x*x) - h*x - 9.8*d*d / (2*v*v);
}
```

## 误差分析

该方法中的误差主要来自模型误差、方法误差、舍入误差。

其中，模型忽略了空气阻力、风力等其它力，因而存在误差，但这些力都较小，不大于  $10^{-3}\text{N}$ 。使用的二分查找方法的精度取决于迭代次数，但实时性要求使得其无法迭代足够多的次数，因而存在较大的方法误差。计算机存储的浮点数总是近似值，故存在舍入误差。

在这些误差中，方法误差是主要误差，即精度主要取决于迭代次数。

## 性能优化

一般情况下，炮台的陀螺仪精度，即炮台的姿态控制的精度为 $0.1^\circ$ ，故方程求解出的俯仰角的精度至少应当为 $0.1^\circ$ 。

由于每次追踪目标时，尤其是运动目标，会导致  $d$  和  $h$  发生变化，进而导致不能直接通过预存 $f(x)$ 函数值表来大幅优化性能，但 $-hx - \frac{gd^2}{2v^2}$  部分的运算不涉及到高耗时操作，可

以通过预存 $l(x) = \sqrt{x - x^2}$ 值表来提升性能；则此时 $f(x) = d \cdot l(x) - hx - \frac{gd^2}{2v^2}$ ，二分查找时通过在 $l(x)$ 函数值表上查找 $x$ 对应的函数值来减少耗时，提升实时性能。

现在分析 $l(x)$ 函数值表上 $x$ 的增量。

因为 $x = \cos^2 \theta$ ，而 $x$ 对 $\theta$ 的导数为 $x' = -2 \cdot \sin \theta$ ，当 $|x'| = 2 \cdot \sin \theta$ 最小，即 $x$ 变化最缓慢时，有 $\theta = 0^\circ$ 。则此时 $\Delta x = \cos^2(0^\circ) - \cos^2(0.1^\circ) \approx 3.0 \times 10^{-6}$ 。则从 $[0, 1]$ 总共有约 $3 \times 10^5$ 个预装载 $l(x)$ 值才能直接达到精度要求，而以单精度浮点数为例，这将占用 2.4MB 的内存，对于 nVIDIA Jetson TX2 来说足以接受，虽然 TX2 本身计算该表达式耗时也相差无几；而该内存占用量对于内存总共只有 1MB 的 STM32 微控制器而言无法接受。因而，应当采取预装载函数表与实时计算相结合的形式。

预装载 $l(x)$ 函数值表可参考附录 1  $l(x)$ 函数值表。

# 附录

附录 1  $l(x)$ 函数值表

$x$	$l(x)$	$x$	$l(x)$
0.01	0.099498743710662	0.51	0.4998999899979995
0.02	0.13999999999999999	0.52	0.49959983987187184
0.03	0.1705872210923198	0.53	0.4990991885387112
0.04	0.19595917942265426	0.54	0.4983974317750845
0.05	0.21794494717703367	0.55	0.49749371855331
0.06	0.23748684174075835	0.56	0.4963869458396343
0.07	0.2551470164434615	0.57	0.4950757517794625
0.08	0.2712931993250107	0.58	0.4935585071701226
0.09	0.2861817604250837	0.59	0.4918333050943175
0.10	0.3	0.60	0.4898979485566356
0.11	0.31288975694324034	0.61	0.4877499359302879
0.12	0.32496153618543844	0.62	0.4853864439804639
0.13	0.33630343441600474	0.63	0.4828043081829324
0.14	0.34698703145794946	0.64	0.48
0.15	0.3570714214271425	0.65	0.47696960070847283
0.16	0.3666060555964672	0.66	0.4737087712930804
0.17	0.375632799419859	0.67	0.47021271782034985
0.18	0.38418745424597095	0.68	0.466476151587624
0.19	0.39230090491866065	0.69	0.46249324319388707
0.20	0.4	0.70	0.45825756949558394
0.21	0.40730823708832603	0.71	0.4537620521815371
0.22	0.4142463035441596	0.72	0.44899888641287294
0.23	0.4208325082500163	0.73	0.44395945760846234
0.24	0.4270831300812525	0.74	0.4386342439892262
0.25	0.4330127018922193	0.75	0.4330127018922193
0.26	0.4386342439892262	0.76	0.4270831300812525
0.27	0.4439594576084623	0.77	0.4208325082500163
0.28	0.44899888641287294	0.78	0.41424630354415953
0.29	0.4537620521815371	0.79	0.4073082370883259
0.30	0.458257569495584	0.80	0.39999999999999999
0.31	0.46249324319388707	0.81	0.39230090491866054
0.32	0.46647615158762407	0.82	0.38418745424597084
0.33	0.4702127178203499	0.83	0.375632799419859
0.34	0.4737087712930804	0.84	0.3666060555964673
0.35	0.47696960070847283	0.85	0.3570714214271426
0.36	0.48	0.86	0.3469870314579495

0.37	0.48280430818293246	0.87	0.3363034344160047
0.38	0.4853864439804639	0.88	0.32496153618543844
0.39	0.4877499359302879	0.89	0.3128897569432403
0.40	0.4898979485566356	0.90	0.29999999999999993
0.41	0.4918333050943175	0.91	0.28618176042508364
0.42	0.4935585071701227	0.92	0.2712931993250107
0.43	0.49507575177946256	0.93	0.2551470164434613
0.44	0.4963869458396343	0.94	0.23748684174075813
0.45	0.49749371855331	0.95	0.21794494717703364
0.46	0.4983974317750845	0.96	0.19595917942265423
0.47	0.4990991885387112	0.97	0.17058722109231986
0.48	0.49959983987187184	0.98	0.14000000000000002
0.49	0.4998999899979995	0.99	0.0994987437106621
0.50	0.5	1.00	0.0