# Ballistics

**Root finding algorithm for hitting targets ballistically**

Laurin Meier

January 13, 2025

# Disclaimer:

This article was originally written to meet set LaTeX requirements, hence the sometimes out-of-place usage of LaTeX tools. It was also written in German, before i translated it into English, partly with the help of DeepL.

# Contents

# 1 What are Ballistics?

Ballistics is a collective term that describes the kinematics of a projectile. Primarily in firearms, one speaks of internal ballistics (inside the weapon), external ballistics (outside the weapon) and terminal ballistics ( within the target). This article deals with external ballistics.

Firstly, the basic concepts of external ballistics should be clarified:

- Firing point: The point at which the projectile leaves the barrel
- Launch angle: Angle at which the projectile leaves the barrel
  1. Azimuth: lateral angle
  2. Elevation: elevation angle
- Muzzle velocity: velocity at which the projectile leaves the barrel, also known as v0
- Point of impact: The point at which the projectile ends its trajectory
- Drag: Decelerates the projectile on its trajectory

External ballistics themselves can also be divided into two categories. On the one hand, the trajectory and impact point of the projectile can be calculated using a launch angle and a muzzle velocity (forward ballistics), and on the other hand, the launch angle required to hit the target can be calculated using an impact or aiming point and other aiming factors (inverse ballistics). The latter is the goal of this article.

# 2 The Algorithm

The basic idea of the algorithm for solving external ballistic problems is the simulation of all possible projectile positions and the discrete target position at time t. Based on this simulation, the smallest distance between the projectile set and the target can also be calculated as a function of t. The next step is to find suitable values for t for which this distance is zero.

The position of all projectiles can be described as a circle in 2-dimensional space and as a sphere in 3-dimensional space. To understand this, let's start with a 2-dimensional example.

In all the formulas mentioned, $t$ describes the time since the projectile was launched.

## 2.1 Scenario: Space station

We start in outer space. Far away from celestial bodies and their atmospheres. Here, we experience neither gravity nor air resistance. As the commander of a space station, we want to fend off an enemy spaceship. To do this, we have to calculate the right firing angle for our gun.

Let's imagine the gun firing in all directions at the same time. The result would be a circle of projectiles that has its centre on the gun and expands with the speed of the projectiles. This circle can be represented mathematically as follows

1. the function for the centre of the circle as a function of $t$:

$$c_{cntr}(t) = g\vec{u}n \tag{1}$$

assuming:

- $c_{cntr}(t)$: The centre of the circle as a function of $t$
- $gun$: The starting position of all projectiles (turret)

2. the function for the radius of the circle as a function of $t$:

$$c_r(t) = v_0 \cdot t \tag{2}$$

assuming:

- $c_r(t)$: The radius of the circle as a function of $t$
- $v_0$: The muzzle velocity

The target can also be determined mathematically. We know its starting position and speed and can therefore estimate its position at any future point in time $t$:

$$p_{tgt}(t) = tg\vec{t}_{pos} + tg\vec{t}_{vel} \cdot t \tag{3}$$

assuming:

- $p_{tgt}(t)$: The position of the target at any time $t$
- $tgt_{pos}$: The starting position of the target at t=0
- $tgt_{vel}$: The speed of the target at t=0

Now we need the distance between the centre of the circle and the target at time $t$

$$dist(t) = \|c_{cntr}(t) - p_{tgt}(t)\| \tag{4}$$

and form a difference, or an 'error', with the radius of the circle:

$$error(t) = dist(t) - c_r(t) \tag{5}$$

Now we just have to solve error(t) = 0.[1].

This gives us a $t$, which I call '$t_{hit}$', where the target lies on the circle. Since the circle represents a 'set of projectiles', there is exactly one projectile that hits the target at the time $t_{hit}$. This one projectile is a specific point on the circle, exactly the point on which the target lies. And this point can be represented as a direction vector from the centre of the circle to this point, i.e. the target:

$$\vec{dir} = p_{tgt}(\vec{t}_{hit}) - c_{cntr}(\vec{t}_{hit}) \tag{6}$$

The direction vector can now be normalised as $\hat{dir}$ and indicates the shooting direction required to hit the target.

This first example is shown here as a graphic. $t_{hit}$ is 3.

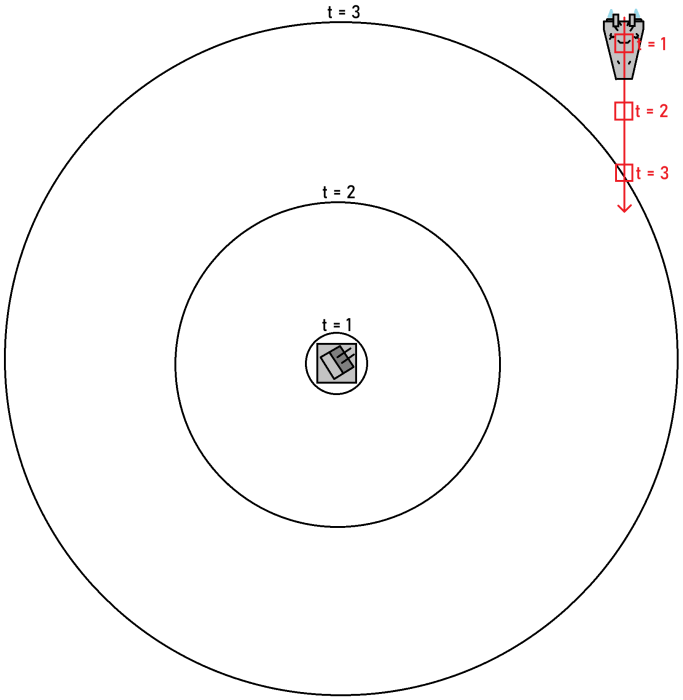



Figure 1: Top view of scenario 1

[1] $error(t) = 0$ can be solved by rearranging to $t$

## 2.2 Scenario: Convoy in indirect fire

Our base on a planet is in danger of being invaded by the enemy. Scouts have spotted an enemy convoy on the way to our base. There is a high mountain range between our cannons and the convoy. It would be in the way if we fired directly at the convoy. We therefore have to fire the projectile at an elevation angle of over 45°, so it flies over the mountain in a high arc.

This planet has no atmosphere, so there is no air resistance. There is, however, a gravitational pull towards the centre of the planet. For the sake of simplicity, this is assumed to be constant at all relevant altitudes. We are also now in 3 dimensions, which means that we are no longer talking about 'circles' but 'spheres'.

Not much changes here. The spheres are now accelerating downwards. All changes to the formulas are marked in red:

1. the function for the centre of the circle as a function of $t$ with gravity

$$c_{cntr}(t) = g\vec{u}n + \frac{1}{2}\vec{g} \cdot t^2 \tag{7}$$

assuming:

- What was assumed in equation 1
- g: Gravitational vector

The remaining equations 2-6 remain unchanged:

$$c_r(t) = v_0 \cdot t \tag{2}$$

$$p_{tgt}(t) = tg\vec{t}_{pos} + tg\vec{t}_{vel} \cdot t \tag{3}$$

$$dist(t) = \|c_{cntr}(t) - p_{tgt}(t)\| \tag{4}$$

$$error(t) = dist(t) - c_r(t) \tag{5}$$

$$d\vec{i}r = p_{tgt}(\vec{t}_{hit}) - c_{cntr}(\vec{t}_{hit}) \tag{6}$$

An important note when solving $error(t) = 0$ is that there are up to 2 solutions here. For example $t_1 = 4$ and $t_2 = 9$. $t_1$ would be a direct hit (flat trajectory), while $t_2$ would be an indirect hit (high trajectory).

The following graphics show a side view of the simulated scenario, with the 2 hitting projectiles marked in red and blue. The graphics are split up to make it easier to recognise what is happening.

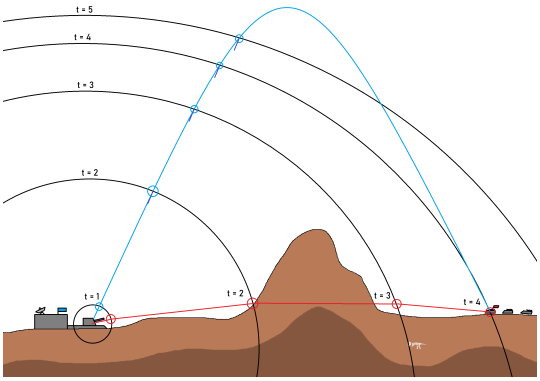



Figure 2: Scenario 2 | t=1-5

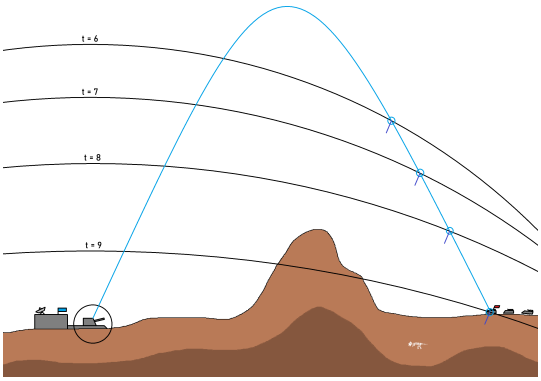



Figure 3: Scenario 2 | t=6-9

The 'falling' spheres, as well as the 2 possibilities of hitting the target, are clearly recognisable here. The dark blue lines, which show the direction to the centre of the sphere, are helpful for understanding. This direction is the same at any time and is also the result that we achieve by calculating $\vec{dir}$.

## 2.3 Scenario: Landing approach

On your return to Earth, you are intercepted by an enemy fighter shortly before landing. Your transporter has turrets on the deck, which you can use to fend off the enemy. Again, you want to calculate the direction of fire to hit the enemy. Within the atmosphere, the projectiles experience atmospheric drag. In addition, the projectiles are fired from your moving transporter. This movement also affects the projectiles. The enemy fighter is manoeuvrable. In addition to the velocity vector, your sensors can also detect acceleration, which makes it easier to describe the enemy's flight path.

We need an auxiliary formula to take atmospheric drag into account. For the sake of simplicity, this assumes linear air resistance. It would, however, also be possible to consider quadratic air resistance. The formula takes the following parameters:

- $t$ = Time
- $v0$ = Starting velocity
- $a$ = Constant acceleration
- $d$ = Drag coefficient ranging from 0 (none) to 1 (a lot)

$$offset(t, v0, a, d) = ((1-d)^t * (a - d * v0) + a * d * t - a + d * v0)/d^2 \tag{8}$$

The return value is a little more abstract.

Let's imagine a stone that we throw upwards with $5m/s$ before it is accelerated downwards with $-9.81m/s^2$ and an air resistance of 0.01. Now we want to know at what height the stone is after 3 seconds.

If we insert the values we get the following result

$$offset(3, 5, -9.81, 0.01) = -14.48 \tag{9}$$

The stone is therefore 14.48m below its starting position.

Here are 2 graphs to illustrate this:

The x-axis describes the time
The y-axis describes the altitude

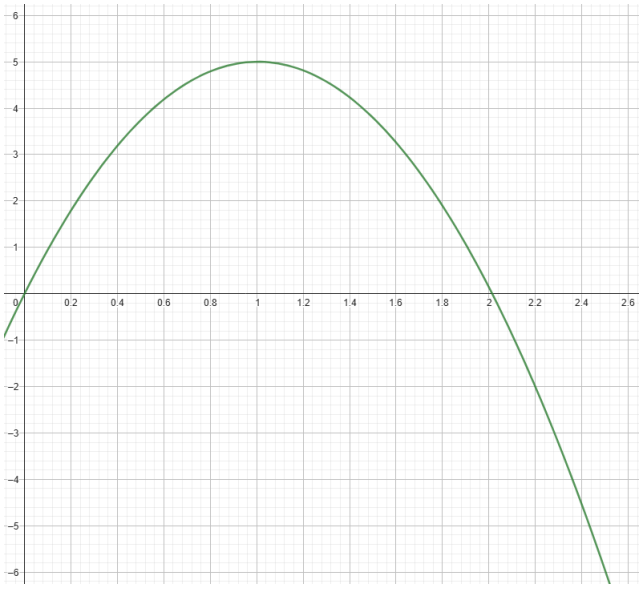



Figure 4: Plot of the offset-function: $offset(t, 5, -9.81, 0.01)$

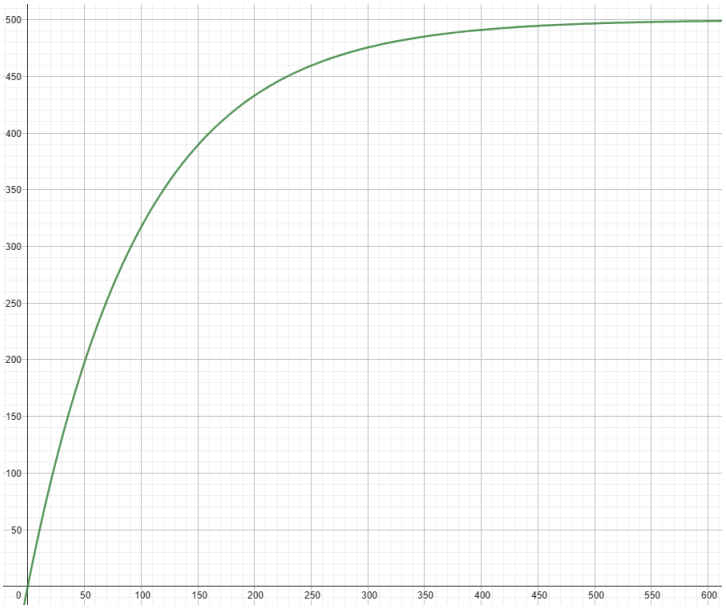



Figure 5: Plot of the offset-function without acceleration: $offset(t, 5, 0, 0.01)$

With the help of this function, we can describe the air resistance sufficiently to redefine our formulas. The following derivation is based on Scenario 2. Changes are marked in red. $d$ is the drag coefficient in all following formulas.
Sphere-centre

$$c_{cntr}(t) = g\vec{u}n + \begin{bmatrix} offset(t, v_x, 0, d) \\ offset(t, v_y, 0, d) \\ offset(t, v_z, g, d) \end{bmatrix} \tag{10}$$

assuming:

- What was assumed in equation 1
- g: Gravity
- $\vec{v}$: Velocity of your transporter

Circle-radius:

$$c_r(t) = offset(t, v0, 0, d) \tag{11}$$

assuming:

- What was assumed in equation 2

Target:

$$p_{tgt}(t) = \vec{tgt_{pos}} + \vec{tgt_{vel}} \cdot t + \frac{1}{2}\vec{tgt_{acc}} \cdot t^2 \tag{12}$$

assuming:

- What was assumed in equation 3
- $tgt_{acc}$: The acceleration of the target

The remaining equations 4-6 remain unchanged:

$$dist(t) = \|c_{cntr}(t) - p_{tgt}(t)\| \tag{4}$$

$$error(t) = dist(t) - c_r(t) \tag{5}$$

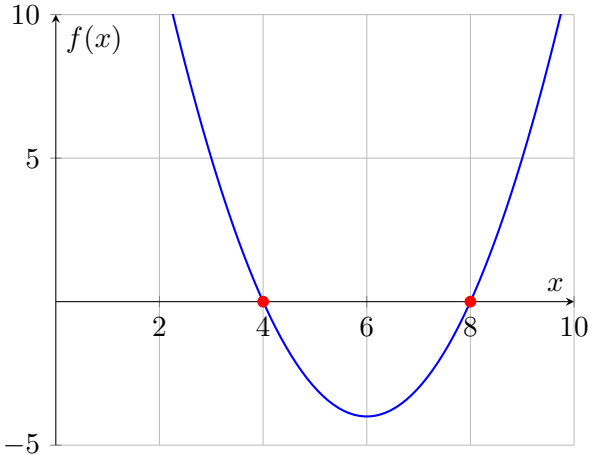
$$\vec{dir} = p_{tgt}(\vec{t_{hit}}) - c_{cntr}(\vec{t_{hit}}) \tag{6}$$

In this way, you can model almost any scenario. You only have to adapt the circle centre, radius and target functions to the given behaviour of the projectile and the target.

Here, however, you run into a problem if you calculate the roots of $error(t)$ by converting to $t$. Some functions are too complex to rearrange. Iterative algorithms such as the Newton-Raphson-method are suitable in such cases.

The Newton-Raphson-method is described on Wikipedia as follows
The idea is to start with an initial guess, then to approximate the function by its tangent line, and finally to compute the x-intercept of this tangent line. This x-intercept will typically be a better approximation to the original function's root than the first guess, and the method can be iterated.Wikipedia contributors, *Newton's method — Wikipedia, The Free Encyclopedia*.

Let's use the Newton-Raphson-method to find the left-hand root 4.




The following table shows the iterations:

Table 1: Individual steps of the Newton-Raphson-method

| step | x | f(x) |
|---|---|---|
| 1 | 5 | -3 |
| 2 | 3.5 | 2.25 |
| 3 | 3.95 | 0.202 |
| 4 | 3.999 | 0.002 |
| 5 | 3.999 | 0 |
| 6 | 3.999 | 0 |
| 7 | 4 | 0 |

Which of the two roots the method approaches depends primarily on the first estimate (in our example $x = 5$). The shape of the function can also deflect the estimate and cause the method to fail. However, since error(t) is approximately parabolic in most realistic cases, it is mainly the first estimate that plays a role. For which $x = 0$ is suitable in order to approach the first root, whereas $x = 3600$ is suitable for approaching the second root.

## 3 Limitations and Remarks

This method of calculating the launch angle assumes that all forces acting on the projectile are the same at all heights. There must therefore be no location-dependent forces. Otherwise, the projectiles could no longer be represented as spheres or circles.

To solve this problem, you would have to simulate random projectiles in all directions and use them to construct a sphere-like shape. I have not tested this approach.

I came up with this algorithm while trying to conquer the ballistics within a video-game/simulation. After implementing the first version of this method, I was made aware

of a project that solves the ballistic problem discussed here in the same way. This paper (b2studios, *The Absurdity of Projectile Aimbots*) describes the method very mathematically and can help with further understanding. In addition to the paper, the author also published a video on YouTube, with an intuitive visual explaination.

## List of Figures



## List of Tables



## References


b2studios. *The Absurdity of Projectile Aimbots*. URL: docs.google.com/document/d/1TKhiXzLMHVjDPX3a3U0uMvaiW1jWQWUmYpICjIDeMSA/edit.

Wikipedia contributors. *Newton's method — Wikipedia, The Free Encyclopedia*. [Online; accessed 13-January-2025]. 2025. URL: https://en.wikipedia.org/w/index.php?title=Newton%27s_method&oldid=1267293171.