ISSF Rifle Sighting Simulator

Mathematical Foundation, Physics Reference, and User Guide

Documentation for `simulator.js` / `index.html`

April 2026

Contents

1 Introduction

The ISSF Rifle Sighting Simulator is a browser-based pedagogical tool that models the optical and ballistic phenomena encountered in precision rifle marksmanship. In ISSF competition, the "sight picture" is not a static image but a dynamic alignment of four independent planes: the shooter's eye, the rear diopter aperture, the front globe sight, and the target [9, 3].

The simulator allows students and athletes to isolate, vary, and observe six fundamental effects that are difficult to separate on the range:

1. **Parallax error**—how head position relative to the rear aperture shifts the point of impact (section 3).
2. **Aperture and eye-relief optics**—how diopter size and distance affect depth of field and parallax sensitivity (section 4).
3. **Exterior ballistics**—gravity drop over the projectile's time of flight (section 5).
4. **The canting paradox**—why tilting the rifle introduces *both* a lateral shift and a vertical drop (section 6).
5. **Mechanical sight adjustment**—how "clicks" on the front sight translate into impact shifts on the target (section 7).
6. **Wind drift**—lateral and vertical deflection caused by atmospheric conditions (section 8).

Each phenomenon is developed in its own section with the competitive-shooting context, a physical explanation, a mathematical derivation, a reference to the implementing JavaScript code, and suggested exercises with the simulator.

2 User Manual

2.1 Running the Simulator

Open `index.html` in any modern web browser. The page loads `simulator.js`, which renders the simulation onto an HTML5 `<canvas>` element at 1000×600 pixels. No server or build step is required.

2.2 Interface Layout

The canvas is divided into three regions (fig. 1).

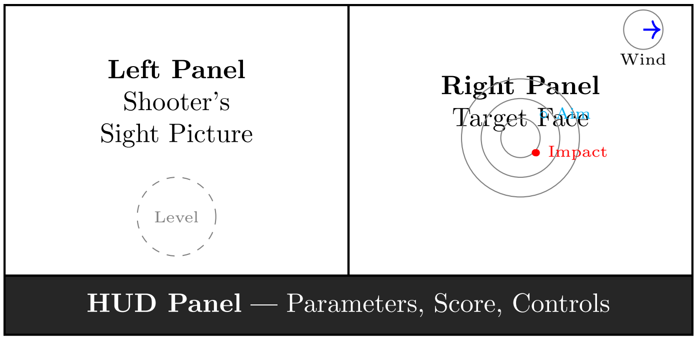


Figure 1: Schematic layout of the simulator interface.

- **Left panel** (0–500 px): the shooter's view through the rear diopter, showing the front tunnel, iris, target, and a spirit-level indicator at the bottom.

- **Right panel** (500–1000 px): the target face with scoring rings. A cyan circle marks the parallax-shifted aim point; a red dot marks the predicted impact after all ballistic and mechanical corrections. A wind-direction indicator is displayed in the upper-right corner.
- **HUD panel** (bottom 85 px): numerical readouts of all parameters, the decimal score, and a control-key reminder.

2.3 Controls Reference

Input	Action	Range	Section
Mouse position	Eye offset (parallax)	canvas area	section 3
A / **D**	Rear aperture ±0.1 mm	0.8–2.2 mm	section 4
Q / **E**	Front iris ±0.1 mm	2.4–7.0 mm	section 4
C / **V**	Iris ring thickness ±0.1 mm	0.5–5.0 mm	section 4
U / **J**	Eye relief ±1 mm	50–215 mm	section 4
W / **S**	Sight height ±1 mm	≥ 20 mm	section 5
Z / **X**	Rifle cant ±0.57°	unlimited	section 6
Arrow keys	Mechanical clicks ±1 px	unlimited	section 7
O / **P**	Wind speed ±0.5 m/s	0–10 m/s	section 8
K / **L**	Wind direction ±8.6°	full circle	section 8
T	Toggle 10 m / 50 m target	—	section 5
R	Reset all parameters to defaults	—	—
F	Toggle language (English / French)	—	—

Table 1: Complete keyboard and mouse controls.

2.4 Quick-Start Walkthrough

1. Open `index.html`. The default view shows a 10 m air-rifle sight picture centered on the target.
2. Move the mouse across the left panel and observe both the sight picture shift and the cyan circle moving on the right panel. This is *parallax* (section 3).
3. Press **Z** several times to cant the rifle. Watch the spirit level and the red impact dot on the right panel shift right *and* down (section 6).
4. Use the arrow keys to bring the red dot back to center—these are sight "clicks" (section 7).
5. Press **T** to switch to the 50 m target and observe how gravity drop and cant error grow dramatically (section 5).
6. Increase wind with **P** and rotate its direction with **K**/**L** to see wind drift (section 8).

3 Parallax Error

3.1 The Issue for Competitive Shooters

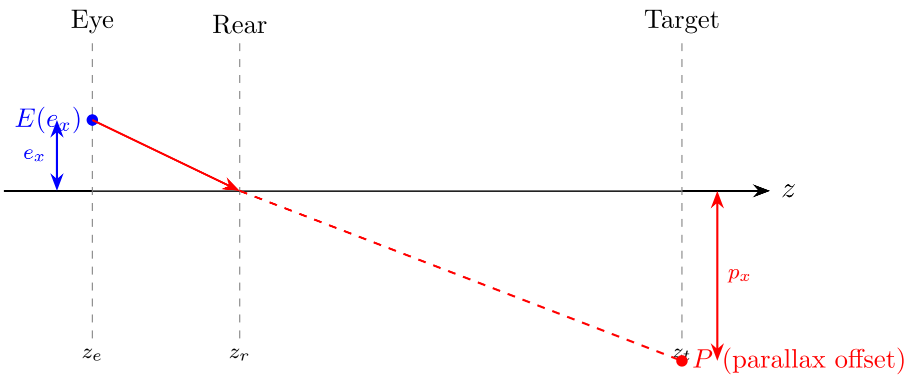
In ISSF rifle shooting, the shooter views the target through a small rear diopter aperture. If the eye is not perfectly centered behind this aperture, the line of sight through the front sight projects onto a different point on the target than intended. This displacement is called *parallax error* [9, 7].

Even sub-millimeter changes in cheek-weld position can shift the apparent aim point by a scoring ring or more on the target. Because the shooter *perceives* the sights as correctly aligned, the error is insidious: the sight picture looks acceptable, yet the point of impact has moved.

3.2 Why It Matters

At the 10 m air-rifle level, a 10.0 versus a 10.5 can determine a final ranking. Parallax is the single largest source of shot-to-shot dispersion for a well-trained athlete whose hold and trigger control are consistent [9]. Understanding the geometry allows coaches to diagnose inconsistent groups that are not explained by natural hold wobble.

3.3 Physical Explanation and Illustration



$$p_x = e_x \frac{z_r - z_t}{z_r - z_e}$$

Figure 2: Parallax geometry. An eye offset e_x projects through the rear-sight center onto the target, producing an error p_x in the opposite direction, magnified by the distance ratio.

The line of sight originates at the eye position (e_x, z_e), passes through the center of the rear aperture $(0, z_r)$, and intersects the target plane at z_t. By the intercept theorem (similar triangles), the target intersection is:

$$p_x = e_x \cdot \frac{z_r - z_t}{z_r - z_e}\,. \tag{1}$$

Since $z_t > z_r > z_e$, the fraction is negative, so the parallax offset on the target is in the *opposite* direction from the eye displacement and magnified by the ratio of distances.

Aperture Damping

A smaller rear aperture physically blocks off-axis light rays, constraining the eye to a narrow cone and reducing the range of parallax offsets that are optically possible [4, 5]. The simulator models this with a linear damping factor:

$$f = \max\!\left(0.05,\ \tfrac{d_{\text{rear}}}{4.5}\right), \qquad p_x \leftarrow p_x \cdot f\,, \tag{2}$$

where d_{rear} is the rear aperture diameter in mm. At the minimum aperture of 0.8 mm, $f \approx 0.18$, reducing parallax sensitivity by roughly 80 %.

3.4 Implementation

In `simulator.js`, the `compute()` function (lines 74–79):

```
const scale = (targetZ - eyeZ) / (rearZ - eyeZ);
let px = eyeXoff + (0 - eyeXoff) * scale;
let py = eyeYoff + (0 - eyeYoff) * scale;

const f = Math.max(0.05, rearAperture_mm / 4.5);
px *= f;  py *= f;
```

Here `scale` is the ratio $\frac{z_t - z_e}{z_r - z_e}$, and the expression `eyeXoff + (0 - eyeXoff) * scale` evaluates to $e_x(1 - \text{scale}) = e_x \cdot \frac{z_r - z_t}{z_r - z_e}$, which is eq. (1). The aperture damping factor f (eq. (2)) is applied to both axes.

3.5 Simulator Exercises

1. With the default 1.6 mm rear aperture, sweep the mouse across the canvas and note the cyan dot displacement on the right panel.
2. Reduce the aperture to 0.8 mm (**A**) and repeat. The cyan dot should move far less—this is the pinhole effect (eq. (2)).
3. Increase the aperture to 2.2 mm (**D**). Parallax sensitivity rises and the target image becomes blurred (see section 4).

4 Aperture Size and Eye Relief

4.1 The Issue for Competitive Shooters

The rear diopter and front iris form the optical frame of the sight picture. Their diameters and the distance from the eye to the rear sight (*eye relief*) determine three competing properties:

1. **Depth of field:** smaller apertures yield sharper images of both the front sight and the target, analogous to a pinhole camera [4].
2. **Light transmission:** too small an aperture reduces brightness and may cause diffraction artifacts, leading to eye fatigue.
3. **Parallax sensitivity:** as derived in section 3, smaller apertures reduce the magnitude of parallax error.

4.2 Why It Matters

Choosing the correct aperture is a balance: too wide and the front sight ring blurs, degrading centering precision; too narrow and the image darkens, especially in indoor ranges with poor lighting. Eye relief must also be consistent—changes in head position alter the apparent size of the rear aperture in the sight picture, which disrupts the concentricity judgment [9].

4.3 Physical Explanation and Illustration

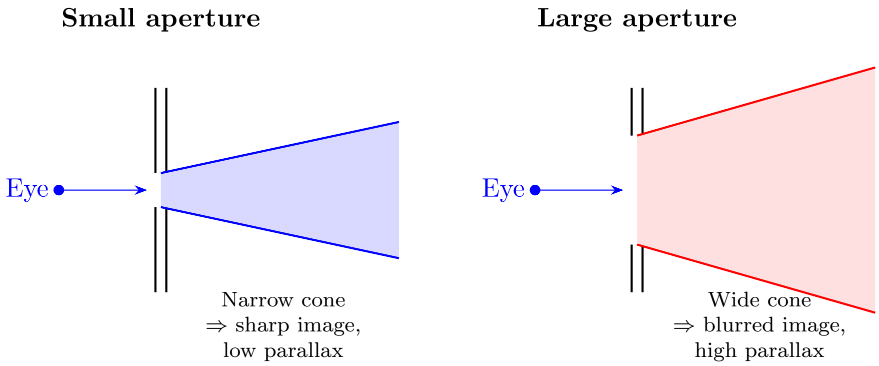


Figure 3: Pinhole analogy. A smaller rear aperture restricts the cone of accepted light rays, increasing depth of field and reducing parallax sensitivity.

Apparent Aperture Size and Eye Relief

The visual angle subtended by the rear diopter from the eye depends inversely on the relief distance $R = z_r - z_e$:

$$r_{\text{apparent}} \propto \frac{d_{\text{rear}}}{R}\,. \tag{3}$$

Larger relief makes the aperture appear smaller, narrowing the visible field, while smaller relief opens the view. The simulator implements this in the rendering code as:

$$r_{\text{px}} = (d_{\text{rear}} \times 10) \times \frac{250}{R}\,. \tag{4}$$

Target Blur

When the rear aperture is large, the depth of field shrinks and the target image appears blurred. The simulator applies a Gaussian blur proportional to the aperture size:

$$\sigma_{\text{blur}} = \max\bigl(0,\ (d_{\text{rear}} - 0.9) \times 4\bigr)\ \text{px}. \tag{5}$$

4.4 Implementation

In `simulator.js`, the rendering function `drawLeft()` (lines 171–173 and 120–121):

```
const relief = Math.max(5, rearZ - eyeZ);
const rearR = (rearAperture_mm * 10) * (250 / relief);
const targetBlur = Math.max(0, (rearAperture_mm - 0.9) * 4);
```

The front iris and ring thickness are rendered as concentric arcs whose radii are proportional to `frontIris_mm` and `frontThickness_mm`, providing a visual representation of the three-ring sight picture (rear diopter, front tunnel, front iris).

4.5 Simulator Exercises

1. Start with the default aperture (1.6 mm). Press **D** to widen the rear aperture to 2.2 mm: note the target blur increase and the parallax sensitivity rise.

2. Press **A** to narrow to 0.8 mm: the image sharpens and darkens, and parallax sensitivity drops.
3. Use **U**/**J** to change eye relief. Watch how the visible diameter of the sight picture changes (eq. (3)). A relief that is too short causes the rear aperture to fill the view; too long and the front tunnel cannot be framed concentrically.
4. Adjust front iris (**Q**/**E**) and thickness (**C**/**V**) to explore how the inner ring frames the target bull.

5 Exterior Ballistics: Gravity and Trajectory

5.1 The Issue for Competitive Shooters

The line of sight is perfectly straight, but a projectile follows a parabolic arc under gravity [1, 6]. To make the bullet hit the point of aim at a given distance, the barrel must be angled slightly upward relative to the sight line. The angular offset depends on the *sight height* (the vertical distance from bore center to the optical axis of the sights) and the *gravity drop* at the target distance.

5.2 Why It Matters

The total vertical compensation $C = H + h_g$ is the foundation on which canting error is built (section 6). At 50 m, gravity drop is an order of magnitude larger than at 10 m, explaining why canting and wind errors become far more severe in smallbore competition. Understanding the relationship between sight height, muzzle velocity, and gravity drop helps athletes interpret their equipment choices [2, 8].

5.3 Physical Explanation and Illustration

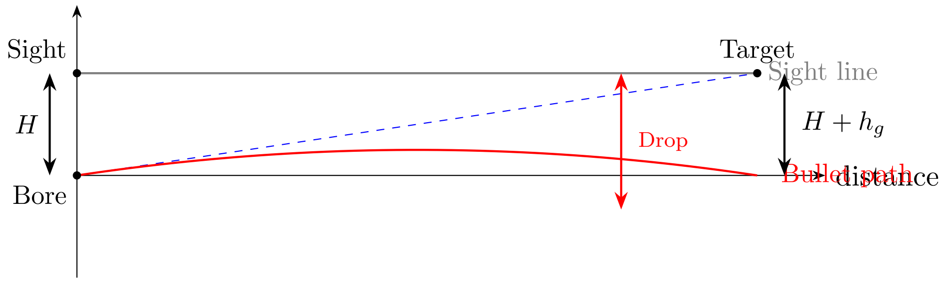


Figure 4: Sight height and gravity drop. The bore is angled upward by the combined compensation $C = H + h_g$ so that the parabolic bullet path intersects the line of sight at the target distance.

5.4 Mathematical Model

For a projectile with muzzle velocity v fired at a target at distance d, the time of flight is:

$$t = \frac{d}{v}\,. \tag{6}$$

The vertical drop due to gravity during this time is [1]:

$$h_g = \frac{1}{2}\,g\,t^2 = \frac{g\,d^2}{2\,v^2}\,. \tag{7}$$

The total vertical compensation required at the target is:

$$C = H + h_g\,, \tag{8}$$

where H is the sight height. Typical values used in the simulator:

Discipline	d (m)	v (m/s)	t (ms)	h_g (mm)
10 m Air Rifle	10	175	57.1	16.0
50 m Smallbore	50	330	151.5	112.6

Table 2: Time of flight and gravity drop for the two target modes.

5.5 Implementation

In `simulator.js`, lines 85–88 of `compute()`:

```
const dist_m = targetType === "10m" ? 10 : 50;
const velocity_ms = targetType === "10m" ? 175.0 : 330.0;
const time_s = dist_m / velocity_ms;
const gravityDrop_mm = 0.5 * GRAVITY * (time_s * time_s) * 1000;
```

The factor of 1000 converts from metres to millimetres. The total compensation C is formed on line 91:

```
const totalComp_mm = sightHeight_mm + gravityDrop_mm;
```

5.6 Simulator Exercises

1. In 10 m mode, note the gravity drop displayed in the HUD (≈16 mm). Press **T** to switch to 50 m and observe the drop increase to ≈113 mm.
2. Use **W**/**S** to raise or lower the sight height. Observe in section 6 how this changes the sensitivity to canting error.

6 Rifle Cant

6.1 The Issue for Competitive Shooters

Canting is the rotation of the rifle about the bore axis—a tilt to the left or right as viewed from behind. Because the barrel must be angled *upward* to compensate for gravity (section 5), any tilt rotates this vertical compensation vector out of the vertical plane. The result is a point-of-impact error that has *both* a lateral and a vertical component—the "canting paradox" [6, 7].

6.2 Why It Matters

Even small cant angles produce measurable errors, especially at longer distances where the total compensation $C = H + h_g$ is large. At 50 m, a 2° cant shifts the impact laterally by ≈ 6 mm—more than a full scoring ring. Because the rifle appears "almost level" to the shooter, canting errors are easy to introduce and difficult to detect without a spirit level. The impact **always drops** and shifts **toward the direction of the cant** [9, 6].

6.3 Physical Explanation and Illustration

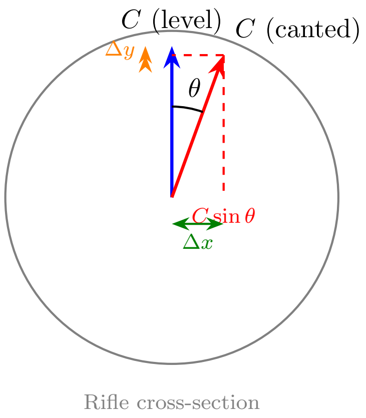


Figure 5: The canting paradox. When the rifle is tilted by θ, the compensation vector C rotates, producing a lateral error $\Delta x = C \sin\theta$ and a vertical loss $\Delta y = C(1 - \cos\theta)$.

6.4 Mathematical Model

When the rifle is level, the bore-to-sight compensation vector is purely vertical: $(0, C)$. When the rifle is canted by angle θ, this vector rotates to $(C \sin\theta,\ C \cos\theta)$. The errors on the target, relative to the uncanted aim point, are [6, 7]:

$$\Delta x = C\ \sin\theta\,, \tag{9}$$

$$\Delta y = C\,(1 - \cos\theta)\,, \tag{10}$$

where Δx is the lateral shift (positive in the direction of the cant) and Δy is the vertical drop (always positive—impact always falls).

For small angles ($\theta \ll 1$), the approximations $\sin\theta \approx \theta$ and $1 - \cos\theta \approx \theta^2/2$ reveal that the lateral error is first-order in θ while the vertical drop is second-order. This explains the well-known observation that cant primarily causes horizontal dispersion at small angles.

Numerical Example

At 50 m ($C = 60 + 112.6 = 172.6$ mm), a cant of $\theta = 2°$ (0.0349 rad):

$$\Delta x = 172.6 \times \sin(0.0349) = 6.0 \text{ mm}\,,$$

$$\Delta y = 172.6 \times (1 - \cos(0.0349)) = 0.1 \text{ mm}\,.$$

The horizontal shift is 60× larger than the vertical drop, consistent with competitive experience.

6.5 Implementation

In `simulator.js`, lines 91–93 of `compute()`. The cant errors are computed in **canvas coordinates** where $+y$ points downward, so a positive `shiftY_mm` moves the impact downward on screen (i.e., the impact drops, as expected physically):

```
const totalComp_mm = sightHeight_mm + gravityDrop_mm;
const shiftX_mm = totalComp_mm * Math.sin(cant);
const shiftY_mm = totalComp_mm * (1 - Math.cos(cant));
```

Note on coordinate convention. In a standard mathematical frame ($+y$ up), the vertical error is $C(\cos\theta - 1) \leq 0$ (negative means downward). In the HTML5 Canvas frame ($+y$ down), the same physical drop corresponds to a *positive* pixel offset. Equation (10) is expressed in the canvas convention used by the implementation: $\Delta y = C(1 - \cos\theta) \geq 0$.

6.6 Simulator Exercises

1. In 10 m mode, press **X** several times to introduce a clockwise cant. Observe the red dot shift right and slightly down on the right panel, and the spirit-level indicator move in the left panel.
2. Press **T** to switch to 50 m and repeat. The same cant angle produces a much larger lateral shift because C is larger.
3. Increase sight height with **W** and note how this amplifies the cant error (larger C in eqs. (9) and (10)).
4. Use the arrow keys to zero-out the cant error with clicks (section 7), then remove the cant with **Z**. The impact now moves to the opposite side—demonstrating why zeroing under a cant and then removing it doubles the error.

7 Mechanical Sight Adjustment (Clicks)

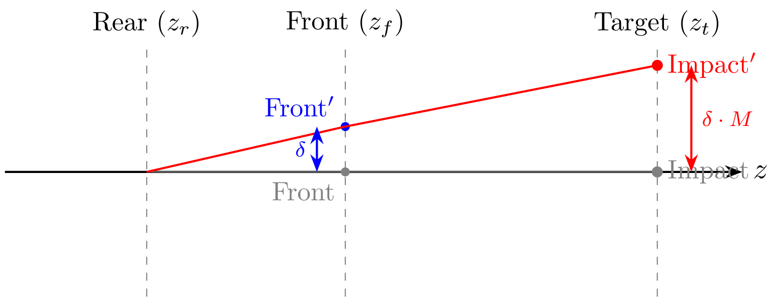
7.1 The Issue for Competitive Shooters

After establishing a natural point of aim, the shooter adjusts the front sight to bring the group center onto the target center. These adjustments—commonly called “clicks” because many mechanisms produce an audible click per increment—physically displace the front-sight element relative to the barrel, changing the angle between the bore and the line of sight [9].

7.2 Why It Matters

Correct use of sight adjustments is essential for zeroing and for responding to changing conditions (wind, light, temperature). Understanding the *lever-arm geometry* helps the shooter predict how many clicks are needed for a given correction and why front-sight adjustments move the group in the *same* direction as the adjustment.

7.3 Physical Explanation and Illustration



$$M = \frac{z_t - z_r}{z_f - z_r}$$

Figure 6: Mechanical sight adjustment geometry. A front-sight displacement δ is amplified by the lever-arm ratio M at the target.

7.4 Mathematical Model

The line of sight pivots about the rear sight. When the front sight is displaced by δ (in pixels or mm), the intercept theorem gives the impact shift at the target as:

$$\Delta_{\text{target}} = \delta \cdot M, \qquad M = \frac{z_t - z_r}{z_f - z_r}\,. \tag{11}$$

With the default values $z_r = 220$, $z_f = 350$, $z_t = 480$:

$$M = \frac{480 - 220}{350 - 220} = \frac{260}{130} = 2.0\,.$$

A one-pixel front-sight click produces a two-pixel impact shift at the target. Because the line of sight passes *through* the displaced front sight, the impact moves in the *same* direction as the front-sight adjustment.

7.5 Implementation

In `simulator.js`, lines 81–83:

```javascript
const mechanicalScale = (targetZ - rearZ) / (frontZ - rearZ);
const tx = frontXoff * mechanicalScale;
const ty = frontYoff * mechanicalScale;
```

7.6 Simulator Exercises

1. Press the right arrow key several times. The red dot moves right on the target—the front sight has been displaced rightward, tilting the line of sight rightward.
2. Combine clicks with cant (**Z**/**X**) to practice the real-world workflow of zeroing under imperfect conditions.

8 Wind Drift

8.1 The Issue for Competitive Shooters

Once the projectile leaves the barrel, it is subject to aerodynamic forces. A crosswind exerts a lateral force that deflects the bullet from its intended path [1, 2]. In outdoor ISSF disciplines (50 m rifle, 300 m rifle), wind reading is the dominant skill separating competitors of similar technical ability [2].

8.2 Why It Matters

Wind is invisible, variable, and non-uniform along the bullet's path. Shooters must estimate its speed and direction from range flags or mirage, then decide whether to adjust sights (clicks) or hold off the center. A 3 m/s crosswind at 50 m produces $\approx$ 31 mm of lateral drift—several scoring rings.

8.3 Physical Explanation and Illustration

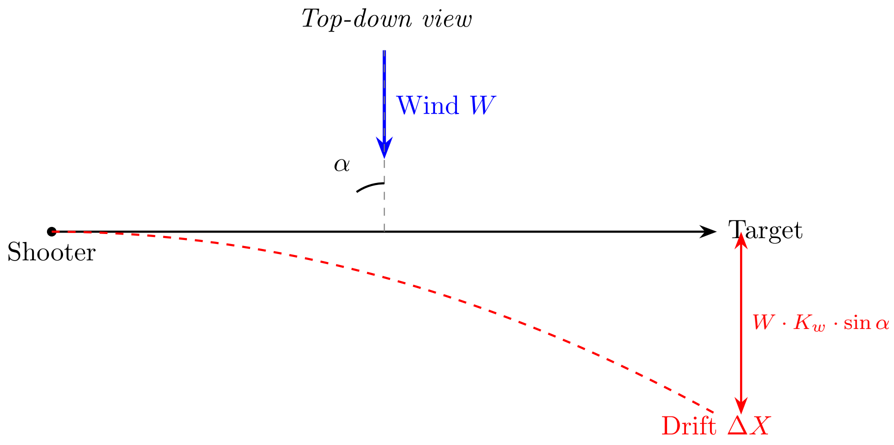


Figure 7: Wind deflection (top-down view). A crosswind with velocity component $W \sin\alpha$ pushes the bullet laterally over the full time of flight.

8.4 Mathematical Model

The full aerodynamic treatment of wind deflection involves the drag coefficient, projectile mass, and cross-sectional area integrated over the flight path [1]. For pedagogical purposes, the simulator uses a linearized empirical model:

$$\Delta X_{\text{wind}} = W \cdot K_w \cdot \sin\alpha\,, \tag{12}$$

$$\Delta Y_{\text{wind}} = W \cdot K_{w,y} \cdot \cos\alpha\,, \tag{13}$$

where W is the wind speed (m/s), α is the wind direction angle (0° = tailwind, 90° = full crosswind from the left), K_w is a lateral drift coefficient (mm per m/s of crosswind), and $K_{w,y} = 0.2\,K_w$ is a much smaller longitudinal coefficient representing the effect of head/tailwinds on time of flight and hence gravity drop.

Discipline	K_w (mm per m/s)	$K_{w,y}$ (mm per m/s)
10 m Air Rifle	1.2	0.24
50 m Smallbore	10.5	2.1

Table 3: Wind drift coefficients used in the simulator.

Wind Direction Convention

The wind angle α is measured from the shooter–target axis, with 0° representing a tailwind (wind blowing from behind the shooter toward the target) and 90° a crosswind blowing from left to right. The wind indicator on the right panel shows an arrow pointing in the direction of wind flow.

A tailwind ($\alpha = 0^\circ$) slightly increases the effective velocity, reducing time of flight and gravity drop, thus raising the impact. A headwind ($\alpha = 180^\circ$) has the opposite effect [2].

8.5 Implementation

In `simulator.js`, lines 97–100:

```
const windFactor = targetType === "10m" ? 1.2 : 10.5;
const windDriftX_mm = windSpeed_ms * windFactor
                    * Math.sin(windDir_rad);
const windDriftY_mm = windSpeed_ms * (windFactor * 0.2)
                    * Math.cos(windDir_rad);
```

In the final impact computation (line 107), `windDriftY_mm` is *subtracted* because it is computed in mathematical coordinates ($+y$ up) while the Canvas uses $+y$ down:

```
finalY: cy + py + ty
      + (shiftY_mm * PX_PER_MM)
      - (windDriftY_mm * PX_PER_MM)
```

8.6 Simulator Exercises

1. Set a 3 m/s crosswind: press **P** six times from the default. The red dot drifts laterally on the target.
2. Rotate the wind with **K**/**L** and observe how the drift direction follows the wind arrow on the upper-right indicator.
3. Compare drift at 10 m and 50 m (**T** to toggle): the 50 m drift coefficient is nearly 9× larger.
4. Set a pure headwind ($\alpha = 180°$, rotate until the arrow points downward on the clock face) and observe the subtle vertical impact shift.

9 Scoring Model

9.1 ISSF Decimal Scoring

In ISSF finals and electronic-target qualification, shots are scored to one decimal place. The maximum score per shot is 10.9 (center), and each integer ring boundary corresponds to a `.0` score. For the 10 m air-rifle target, ring boundaries are spaced 2.5 mm apart in radius [3].

9.2 Mathematical Model

The score is a linear function of radial distance r (in mm) from the target center, capped at the maximum:

$$S = \min\!\left(10.9,\ 11.0 - \tfrac{r}{2.5}\right), \qquad S \geq 0\,. \tag{14}$$

Key values:

- $r = 0$ mm: $S = 10.9$ (perfect center shot).
- $r = 0.25$ mm (inner-10 / X-ring edge): $S = 10.9$.
- $r = 2.5$ mm (10-ring edge): $S = 10.0$.
- $r = 5.0$ mm (9-ring edge): $S = 9.0$.
- $r \geq 27.5$ mm: $S = 0$ (off target).

9.3 Implementation

In `simulator.js`, lines 111–113:

```
function computeScore(x, y) {
  const dist_mm = Math.hypot(x - cx, y - cy) / PX_PER_MM;
  return Math.max(0, Math.min(10.9, 11.0 - (dist_mm / 2.5)));
}
```

The computation uses the physics-space impact position (before the 1.2× display magnification applied on the right panel), so the score reflects the true ballistic offset.

Note. The scoring ring spacing of 2.5 mm matches the ISSF 10 m air-rifle target. The 50 m small-bore target has a different ring geometry (approximately 4 mm radial spacing), but the simulator uses the same formula for both modes as a pedagogical simplification.

10 Implementation Notes

10.1 Coordinate System

The HTML5 Canvas places the origin at the top-left corner with $+x$ pointing right and $+y$ pointing *down*. All rendering and impact calculations use this convention. When converting from standard mathematical formulas (where $+y$ is up), vertical quantities must be negated. The key sign conventions in `compute()` are summarized in table 4.

Quantity	**Math frame** ($+y$ up)	**Canvas** ($+y$ down)
Cant Δy	$C(\cos\theta - 1) \le 0$	$C(1 - \cos\theta) \ge 0$
Wind ΔY (tailwind)	> 0 (up)	subtracted $\Rightarrow -$ (up)
Parallax p_y	opposite to e_y	same formula (eye offset is in canvas coords)

Table 4: Sign conventions for vertical quantities.

10.2 Rendering Pipeline

Each animation frame (via `requestAnimationFrame`) executes:

1. `compute()` — evaluates parallax, mechanical offset, cant error, wind drift, and composes the final impact coordinates.
2. `drawLeft()` — renders the sight picture: target (with depth-of-field blur), front tunnel and iris (with cant rotation), rear diopter mask, barrel silhouette, and level indicator.
3. `drawRight()` — renders the target face with scoring rings, the parallax aim point (cyan), the impact point (red), and the wind indicator.
4. `drawPanel()` — renders the HUD with numerical parameters and the decimal score.

10.3 Physical Constants and Defaults

Constant	**Description**	**Value**
`PX_PER_MM`	Display scale	6.0 px/mm
`rearZ`	Rear sight z	220
`frontZ`	Front sight z	350
`targetZ`	Target z	480
`GRAVITY`	Gravitational accel.	$9.81\,\mathrm{m/s^2}$
`eyeZ`	Eye position z (default)	190
`rearAperture_mm`	Rear diopter (default)	1.6 mm
`frontIris_mm`	Front iris (default)	3.8 mm
`sightHeight_mm`	Sight height (default)	60 mm

Table 5: Physical constants and default parameters.

11 Summary of Corrections Applied

During the preparation of this document, two corrections were identified and applied to `simulator.js`:

1. **Cant vertical error sign (line 93).** The original code computed `shiftY_mm = -totalComp_mm * (1 - Math.cos(cant))`, yielding a negative value that moved the impact *upward* on the canvas. Physically, canting always lowers the impact (section 6). In Canvas coordinates ($+y$ down), the correct expression is `shiftY_mm = totalComp_mm * (1 - Math.cos(cant))`, which is non-negative and moves the impact downward.

2. **Scoring offset (line 113).** The original formula `10.9 - dist_mm / 2.5` evaluated to 9.9 at the 10-ring boundary (2.5 mm), whereas ISSF rules score this as 10.0. The corrected formula `Math.min(10.9, 11.0 - dist_mm / 2.5)` aligns the ring boundaries with integer scores and caps the maximum at 10.9.

The backup file `simulator.best.js` retains the original code for reference.

References

[1] R. L. McCoy, *Modern Exterior Ballistics: The Launch and Flight Dynamics of Symmetric Projectiles*, 2nd ed. Schiffer Publishing, 2012. ISBN 978-0-7643-3825-0.

[2] B. Litz, *Applied Ballistics for Long-Range Shooting*, 3rd ed. Applied Ballistics LLC, 2015. ISBN 978-0-9909206-1-8.

[3] International Shooting Sport Federation, *ISSF Rule Book*, Edition 2026 (First Print 12/2025, effective 1 January 2026). https://www.issf-sports.org/rules

[4] E. Hecht, *Optics*, 5th ed. Pearson, 2017. ISBN 978-0-13-397722-6.

[5] W. J. Smith, *Modern Optical Engineering*, 4th ed. McGraw-Hill, 2007. ISBN 978-0-07-147687-4.

[6] R. A. Rinker, *Understanding Firearm Ballistics: Basic to Advanced Ballistics*, 6th ed. Mulberry House Publishing, 2005. ISBN 978-0-9645598-5-1.

[7] F. W. Mann, *The Bullet's Flight from Powder to Target*, Wolfe Publishing, 1909 (reprinted 2009). ISBN 978-0-935632-04-0.

[8] A. J. Pejsa, *Modern Practical Ballistics*, 2nd ed. Kenwood Publishing, 1991. ISBN 978-0-9612776-3-5.

[9] National Rifle Association, *NRA Guide to Rifle Marksmanship*, NRA Publications, 2014.

[10] J. S. Hatcher, *Hatcher's Notebook*, 3rd ed. Stackpole Books, 1962. ISBN 978-0-8117-0795-4.