

Contents

1 ScopeDOPE Ballistic Computer: Architecture, Physics, and the DOPE-Blend™ Field-Data Fusion Framework	2
1.1 Abstract	2
1.2 1. Introduction	3
1.3 2. Theoretical Background	4
1.3.1 2.1 The Point-Mass Model	4
1.3.2 2.2 Drag Functions: G1 and G7	5
1.3.3 2.3 Ballistic Coefficient	5
1.3.4 2.4 JBM as the Industry Reference	6
1.4 3. Atmospheric Model	6
1.4.1 3.1 Speed of Sound	6
1.4.2 3.2 Air Density Ratio	6
1.4.3 3.3 Why Density Matters	7
1.4.4 3.4 Effective BC Scaling	7
1.5 4. Physics Model	8
1.5.1 4.1 Drag Deceleration and the KD Constant	8
1.5.2 4.2 G7 Drag Table Calibration	8
1.5.3 4.3 Zero-Condition Integration	9
1.5.4 4.4 Trajectory Elevation in Milliradians	9
1.5.5 4.5 BC from Form Factor	9
1.6 5. Numerical Methods	10
1.6.1 5.1 Spatial Domain Integration	10
1.6.2 5.2 Fourth-Order Runge-Kutta	10
1.6.3 5.3 Time-of-Flight	10
1.6.4 5.4 Drop Integration	11
1.6.5 5.5 Unified Monotone Cubic Hermite Interpolation: MonotoneCubicSpline	11
1.7 6. Secondary Effects	12
1.7.1 6.1 Windage: Didion Lag Rule	12
1.7.2 6.2 Spin Drift: Miller Stability and Litz Approximation	12
1.7.3 6.3 Aerodynamic Jump	13
1.7.4 6.4 Coriolis Effect	14
1.7.5 6.5 Inclined Fire	14
1.8 7. DOPE-Blend™ Field-Data Fusion Framework	15
1.8.1 7.1 The Problem Physics Alone Cannot Solve	15
1.8.2 7.2 Bias Residual Computation	15
1.8.3 7.3 Interpolation Strategy	16
1.8.4 7.4 Why This Is Novel	16
1.8.5 7.5 Two-Phase API for UI Responsiveness	17
1.9 8. Implementation Architecture	17
1.9.1 8.1 Kotlin Multiplatform	17
1.9.2 8.2 Singleton, Stateless, Thread-Safe	18
1.9.3 8.3 Visibility and Trade-Secret Encapsulation	18
1.9.4 8.4 Offline-First	18

1.9.5 8.5 Numerical Precision	18
1.109. Validation Study	19
1.10.19.1 Test Corpus	19
1.10.29.2 Accuracy Tiers and Thresholds	19
1.10.39.3 Combinatorial Leave-k-Out Cross-Validation	20
1.1110. Results	20
1.11.110.1 Overall Validation (Run 2026-05-31)	20
1.11.210.2 Cross-Calculator Accuracy: G7 Precision Loads	21
1.11.310.3 Fired Data: 6.5 Creedmoor Tier 3 Case	22
1.11.410.4 Open Issues	23
1.1211. Discussion	23
1.12.111.1 G7 Calibration Range	23
1.12.211.2 Transonic Regime	23
1.12.311.3 DOPE-Blend Versus Calculator-Class Approaches	24
1.1312. Conclusion	24
1.14References	25

1 ScopeDOPE Ballistic Computer: Architecture, Physics, and the DOPE-Blend™ Field-Data Fusion Framework

Cold Bore Ballistics LLC — Technical Report TR-2026-001

Jason M. Schwefel

Cold Bore Ballistics LLC

Revision 1.1 — 2026-06-02

Revision history:

1.0 — 2026-06-01: Initial release.

1.1 — 2026-06-02: Added aerodynamic jump (§6.3); corrected AJ formula (range-independent, no mrad division); documented unified MonotoneCubicSpline class (§5.5); added DOPE-Blend two-phase API (§7.5); updated to ScopeDOPE 0.9.5-beta.

1.1 Abstract

ScopeDOPE is a field Ballistic Computer for precision rifle shooting, implemented as a Kotlin Multiplatform (KMP) application targeting Android and iOS. Where ballistic calculators apply a physics model to manufacturer-supplied parameters and stop there, ScopeDOPE goes further: it incorporates DOPE-Blend™, a proprietary field-data fusion framework that learns from a shooter's confirmed field holds and continuously refines trajectory predictions across changing environmental conditions. This paper documents the complete technical architecture of the Scope-

DOPE ballistics engine — a point-mass exterior ballistics solver that integrates the drag equation in the spatial domain using a fourth-order Runge-Kutta scheme, with Fritsch-Carlson monotone cubic Hermite interpolation applied to both the tabulated drag coefficient data and the DOPE-Blend bias residuals. The atmospheric model implements the ICAO Standard Atmosphere tropospheric lapse formula, the Magnus saturation vapor pressure relation, and a separate zero-condition integration pass that correctly handles the environmental delta between zeroing and field deployment. Secondary effects covered include the Didion lag-rule windage model, Miller gyroscopic stability for spin drift, the Litz aerodynamic jump approximation (a range-independent angular correction for crosswind-induced vertical deflection at the muzzle), and per-step Coriolis integration. Following a 1.26× calibration of the McCoy G7 drag table to align with the JBM Ballistics G7 reference function, the engine agrees with JBM to within 0.012 mil at 1000 yards for representative precision loads. DOPE-Blend™ is validated via exhaustive combinatorial C(N,k) cross-validation across 37 test cases spanning 22 calibers. The engine runs entirely on-device with no server dependency and no user account infrastructure.

1.2 1. Introduction

A precision rifle at 1000 yards is asking a lot of physics. The bullet has been in flight for roughly 1.4 seconds, has decelerated from perhaps 2750 fps to under 1400, has dropped over 200 inches relative to the bore line, and has been subject to whatever crosswind was blowing across the range — including the part you couldn't see or measure. The shooter's job is to place a hold that accounts for all of this before pulling the trigger. The Ballistic Computer's job is to make that hold as accurate as possible.

The standard approach — compute a trajectory from manufacturer-supplied ballistic coefficients, atmospheric inputs, and zero distance — works well when the inputs are accurate. In practice, they rarely are, at least not fully. Published G7 ballistic coefficients are averaged across production lots and test conditions. Atmospheric readings at the firing point don't capture what's happening downrange. Every barrel has its own aerodynamic signature shaped by rifling engagement, throat geometry, and bore surface — none of which appear in any manufacturer's datasheet. These aren't failures of measurement so much as irreducible uncertainties: the BC printed on the box is the best available estimate, not a precise physical constant for this specific bullet from this specific rifle on this specific day.

The conventional workaround is to expose BC as a user-adjustable “trued BC” parameter and let the shooter reverse-engineer the right value from field observa-

tions. This gets the physics model to agree with observed data — but only at the conditions where the truing was done. Apply that adjusted BC at a different altitude or temperature and the error comes back, because what was really being corrected wasn't BC alone: it was some mixture of BC uncertainty, atmospheric model error, and barrel-specific behavior, all collapsed into a single scalar. The model has no way to know which source of error is which.

ScopeDOPE addresses this with DOPE-Blend™. Rather than absorbing all sources of trajectory error into the BC input, DOPE-Blend computes the residual between what the physics model predicts and what the shooter actually observed — expressed in the physics model's own output space, in milliradians, at the conditions that existed when the observation was made. When shooting conditions change, the physics engine re-projects to the new environment; DOPE-Blend applies the same systematic correction on top. The two components are doing fundamentally different things, and keeping them separate is the central insight.

This paper documents the full technical architecture: the physics model (§3–4), the numerical methods (§5), the secondary effects including aerodynamic jump (§6), DOPE-Blend™ including the two-phase caching API (§7), the implementation architecture (§8), and the validation methodology and results (§9–10).

The engine is implemented in Kotlin as a Multiplatform shared module (:ballistics), compiling to Android via AGP and JVM for the validation suite. iOS targets share the same source at v1.0 launch. Current version: ScopeDOPE 0.9.5-beta.

1.3 2. Theoretical Background

1.3.1 2.1 The Point-Mass Model

The point-mass model treats a bullet as a dimensionless particle subject to gravity and aerodynamic drag. Gyroscopic orientation, Magnus forces, and yaw-dependent lift are all ignored. This sounds like a drastic simplification — and for poorly stabilized or tumbling projectiles, it is. But for a well-designed spin-stabilized bullet flying at small yaw angles (under about 2 degrees throughout supersonic flight), the drag is almost entirely axial and the cross-axis terms are negligibly small. This regime describes the vast majority of precision rifle projectiles at the ranges where they're used, which is why the point-mass model has underpinned practical ballistics computation since Siacci in 1880 and remains the basis of every commercial ballistics system today [McCoy 1999].

The two forces acting on the bullet under this model are gravity (constant, down-

ward) and aerodynamic drag (opposing the velocity vector, magnitude dependent on airspeed and Mach number).

1.3.2 2.2 Drag Functions: G1 and G7

Because C_d varies strongly with Mach number and projectile geometry, the ballistics community long ago settled on reference projectiles with standardized drag functions. Any actual bullet is then characterized by how it compares to the reference — a scalar ballistic coefficient rather than a full C_d table.

Two reference functions matter here:

G1 — Based on the Krupp flat-base blunt-ogive projectile. It's the older standard and is still the most commonly published form, particularly for hunting bullets and pistol projectiles. G1 C_d varies substantially across the supersonic range, which means the G1 BC of a real bullet depends on velocity — different at 3000 fps than at 1500 fps.

G7 — Based on a boat-tail tangent-ogive secant-ogive projectile that closely resembles modern long-range match bullets. Because the reference geometry is a much better match, G7 BCs are more stable across the velocity range and require fewer velocity-zone adjustments. For anything shot at 600 yards and beyond with a modern match projectile, G7 is the right model [Litz 2009].

The G1 and G7 drag tables in this engine come from McCoy's *Modern Exterior Ballistics* [McCoy 1999]: 43 entries for G7 and 44 for G1, both covering Mach 0 through 3.0.

1.3.3 2.3 Ballistic Coefficient

The ballistic coefficient captures a projectile's resistance to deceleration relative to the reference:

$$BC = \frac{SD}{i}$$

where SD is sectional density and i is the form factor (ratio of actual drag to reference drag). Sectional density is:

$$SD = \frac{W_{gr}}{7000 \cdot D_{in}^2}$$

with W_{gr} in grains, D_{in} in inches, and 7000 the grains-per-pound conversion. A

higher BC means less drag relative to mass — the bullet retains velocity and resists wind better at range.

1.3.4 2.4 JBM as the Industry Reference

Published G7 BCs — from Hornady, Berger, Sierra, and others — are implicitly calibrated against the JBM Ballistics G7 drag function, which serves as the de facto industry standard. The McCoy G7 table and the JBM G7 function are not identical; a McCoy-based engine using published G7 BCs will under-predict drop unless the discrepancy is corrected. The calibration factor and its validation are covered in §4.2.

1.4 3. Atmospheric Model

1.4.1 3.1 Speed of Sound

$$c_s = 1116.45 \sqrt{\frac{T_F + 459.67}{518.67}} \quad [\text{ft/s}]$$

This is the ISA troposphere relation in Fahrenheit. The reference value 518.67°R corresponds to 59°F (288.15 K) standard day; 1116.45 ft/s is the standard sea-level speed of sound, derived from the ideal gas law — $c_s = \sqrt{\gamma RT/M}$ — with $\gamma = 1.4$ for diatomic air. Temperature matters because Mach number determines Cd, and Mach number for a given bullet velocity depends on the local speed of sound.

1.4.2 3.2 Air Density Ratio

The engine computes a dimensionless air density ratio ρ/ρ_0 relative to the standard atmosphere and uses it to scale BC rather than directly modifying Cd. This keeps BCs in their published units and the arithmetic clean.

Pressure. When station pressure is measured and supplied directly, it's used as-is. When only altitude is available, the ICAO Standard Atmosphere tropospheric formula provides it [ICAO 1993]:

$$P = P_0 (1 - 6.87559 \times 10^{-6} \cdot h)^{5.2561} \quad [\text{hPa}]$$

where $P_0 = 1013.25$ hPa and h is altitude in feet. The exponent 5.2561 comes from $gM/(R\Lambda)$ where Λ is the ISA temperature lapse rate.

Humidity. Moist air is less dense than dry air at the same pressure — water vapor ($M = 18$ g/mol) is lighter than diatomic nitrogen and oxygen ($M \approx 29$ g/mol). The Magnus formula gives saturation vapor pressure [Magnus 1844]:

$$e_{sat} = 6.1078 \exp\left(\frac{17.27 T_C}{T_C + 237.3}\right) \quad [\text{hPa}]$$

Actual vapor pressure and the humidity density correction follow:

$$e_W = \frac{\text{RH}}{100} \cdot e_{sat}$$

$$\rho_{factor} = 1 - \frac{0.378 e_W}{P}$$

The coefficient 0.378 is $1 - M_{H_2O}/M_{air}$. The full density ratio is:

$$\frac{\rho}{\rho_0} = \frac{P}{P_0} \cdot \frac{T_{0,K}}{T_K} \cdot \rho_{factor}$$

1.4.3 3.3 Why Density Matters

At 5000 ft and 95°F — a not-unusual summer mountain conditions scenario — the density ratio drops to roughly 0.80. This increases the effective BC by 25%, translating to roughly 1-2 mil less elevation needed at 1000 yards compared to sea-level standard conditions. Ignoring atmospheric density is not an option for a Ballistic Computer serving shooters across the western United States.

1.4.4 3.4 Effective BC Scaling

$$BC_{eff} = \frac{BC}{\rho/\rho_0}$$

Dividing BC by the density ratio is mathematically equivalent to multiplying drag by the ratio, but cleaner numerically and easier to reason about. Multi-zone BC tables scale each zone independently using the same ratio.

1.5 4. Physics Model

1.5.1 4.1 Drag Deceleration and the KD Constant

Working from Newton's second law with the drag force, then applying the chain rule $dv/ds = (dv/dt)/v$ to convert from the temporal to the spatial domain:

$$\frac{dv}{ds} = -\frac{KD \cdot C_d(M) \cdot v_r^2}{BC_{eff} \cdot v}$$

where s is distance in feet, v is bullet velocity in fps, $v_r = v + v_{hw}$ is the airspeed relative to any headwind (positive headwind increases relative airspeed and therefore drag), $M = v_r/c_s$ is the Mach number, and KD is the drag formula constant:

$$KD = \frac{\rho_0}{2 \times 144} = \frac{0.0764}{288.0} \approx 2.653 \times 10^{-4}$$

Standard sea-level air density in English units is $\rho_0 = 0.0764$ lb/ft³. The factor of 144 is in²/ft², which appears because BC uses in² for cross-sectional area while s is in feet. KD is a fixed constant — non-standard density is handled entirely through BC_{eff} , not through KD or C_d .

1.5.2 4.2 G7 Drag Table Calibration

The McCoy G7 drag table and the JBM G7 drag function are not the same. Published G7 BCs are calibrated against JBM's function, so a McCoy-based engine will systematically under-predict drag — and therefore under-predict bullet drop — when used with those BCs. The error grows with range: without correction, a .308 Win 175gr SMK prediction was off by nearly 1 mil at 1000 yards versus JBM.

Cross-validation against JBM across loads spanning G7 BC 0.25–0.35 and muzzle velocity 2400–3200 fps revealed a consistent scaling ratio between the two functions. Applying a uniform 1.26× factor to the McCoy G7 C_d values brings the engine into agreement with JBM to within 0.012 mil at 1000 yards:

$$C_{d,G7,eff}(M) = 1.26 \times C_{d,G7,McCoy}(M)$$

This factor is applied exclusively to the G7 drag function, scoped to the point in the trajectory integrator where C_d is evaluated. The G1 table is left unmodified — G1 BCs reference a different drag function history and the 1.26× factor has not been validated in that domain.

1.5.3 4.3 Zero-Condition Integration

Every trajectory computation actually involves two integrations. The first runs at the conditions prevailing when the rifle was zeroed: zeroing temperature, altitude, and — if recorded — a potentially different muzzle velocity from a colder barrel. This pass produces y_Z , the drop at zero distance under zero conditions. The second runs at match-day conditions to produce y_D at the target distance.

A rifle zeroed at sea level in January and shot at 6000 ft in July will have a materially different zero-relative trajectory than one where the zero was established in today's conditions. The magnitude of this correction can exceed 0.5 mil at 1000 yards under extreme environmental differences — enough to matter in competition, and the sort of thing no calculator handles correctly without this architecture.

1.5.4 4.4 Trajectory Elevation in Milliradians

The elevation correction the scope must receive comes from the geometry of the bore line, the sight line, and the bullet drop:

$$\theta_{elev} = \frac{12 \left[y_D + h_s - \frac{d}{d_0} (y_Z + h_s) \right]}{d \times 0.036} \quad [\text{mil}]$$

where y_D is drop at distance d in feet (positive downward), y_Z is drop at zero distance d_0 , $h_s = h_{in}/12$ is sight height in feet (default 1.5 in), and the denominator $d \times 0.036$ is the subtension in inches per milliradian at d yards: 1 mrad subtends $d \times 36/1000$ inches. The factor of 12 converts the numerator from feet to inches. Positive output means dial up.

1.5.5 4.5 BC from Form Factor

When BC is unknown but physical dimensions are available, the Pejsa sectional-density method provides a fallback:

$$BC = \frac{SD}{i} = \frac{W_{gr}/(7000 \cdot D_{in}^2)}{i}$$

where i is the form factor supplied by the user. This is less accurate than a measured BC and is provided for uncatalogued projectiles only.

1.6 5. Numerical Methods

1.6.1 5.1 Spatial Domain Integration

The integrator works in the spatial domain — velocity as a function of distance $v(s)$ — rather than time. The two formulations are equivalent via the chain rule, but spatial integration is the natural choice here. The engine's outputs are needed at specific yard marks; a temporal integrator produces values at uniform time steps and would require spatial interpolation to recover per-yard results. Spatial steps also self-adapt to velocity without any explicit logic: at high muzzle velocity, a 0.5-yd step corresponds to a short time interval; downrange as velocity decreases, the same spatial step covers more time, which is appropriate because progressively less happens per foot of bullet travel.

1.6.2 5.2 Fourth-Order Runge-Kutta

The velocity integration uses classical RK4 over 0.5-yard (1.5-foot) spatial steps. For the drag deceleration function $f(v) = dv/ds$:

$$\begin{aligned} k_1 &= f(v_n) \\ k_2 &= f\left(v_n + \frac{h}{2}k_1\right) \\ k_3 &= f\left(v_n + \frac{h}{2}k_2\right) \\ k_4 &= f(v_n + h k_3) \\ v_{n+1} &= v_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \end{aligned}$$

All intermediate velocities are clamped to 1 fps minimum to prevent division-by-zero in drag evaluation. RK4 has $O(h^4)$ global error. A sensitivity study confirmed that halving the step from 0.5 yd to 0.25 yd changes the 1000-yard elevation result by less than 0.0001 mil for representative precision loads. The integrator is not the accuracy-limiting factor — atmospheric model uncertainty and BC measurement error dominate by several orders of magnitude.

1.6.3 5.3 Time-of-Flight

Time of flight accumulates per step using the trapezoidal mean velocity:

$$\Delta t = \frac{h}{\bar{v}} = \frac{h}{(v_n + v_{n+1})/2}$$

First-order accurate in Δv , which is more than adequate given that velocity changes by only a few fps per 0.5-yd step under normal conditions.

1.6.4 5.4 Drop Integration

Vertical drop integrates per step using the kinematic equation with the computed Δt :

$$y_{n+1} = y_n + v_{drop,n} \cdot \Delta t + \frac{1}{2} g_{eff} \cdot \Delta t^2$$

$$v_{drop,n+1} = v_{drop,n} + g_{eff} \cdot \Delta t$$

Effective gravity $g_{eff} = g \cos(\theta_{fire})$ accounts for inclined fire (§6.5).

1.6.5 5.5 Unified Monotone Cubic Hermite Interpolation: MonotoneCubicSpline

A single class, `MonotoneCubicSpline`, handles all piecewise cubic interpolation in the engine. It takes two `DoubleArray` sequences — knot positions and values — precomputes Fritsch-Carlson tangent vectors at construction, and evaluates at arbitrary positions with no per-call heap allocation. The class serves two distinct purposes:

Drag curve evaluation. The G7 and G1 drag tables have 43 and 44 entries respectively. `G7_CURVE` and `G1_CURVE` are constructed once at class load and reused throughout the engine's lifetime. For inputs above Mach 3.0, `eval()` clamps to the table endpoint value — outside the validated range, but the error from clamping is small (G7 Cd changes under 1% between Mach 2.8 and 3.0).

DOPE-Blend bias interpolation. When the shooter has multiple field DOPE entries, their bias residuals at known distances must be interpolated for arbitrary target distances. A `MonotoneCubicSpline` is constructed from the bias knots and evaluated inside `applyBias()`.

The Fritsch-Carlson algorithm [Fritsch & Carlson 1980] is the right choice for both uses because it guarantees the interpolant is monotone on any interval where the data itself is monotone. Standard cubic splines minimize curvature without the monotonicity constraint, which means they can oscillate between sparse or irregularly spaced knots. For drag curves this would introduce spurious Cd oscillations that disturb the velocity integration. For DOPE-Blend it would produce elevation corrections that overshoot or reverse between confirmed holds — exactly the kind of behavior that would make a shooter distrust the system.

The algorithm works as follows. Chord slopes $\delta_i = (y_{i+1} - y_i)/(x_{i+1} - x_i)$ are computed between adjacent knots. Interior tangents initialize as arithmetic means

of neighboring chords; tangents at sign-change intervals are zeroed. The Fritsch-Carlson monotonicity condition $\alpha_i^2 + \beta_i^2 \leq 9$ (where $\alpha_i = m_i/\delta_i$, $\beta_i = m_{i+1}/\delta_i$) is enforced by scaling violating tangent pairs by $\tau = 3/\sqrt{\alpha_i^2 + \beta_i^2}$. Evaluation uses the cubic Hermite basis:

$$p(t) = (2t^3 - 3t^2 + 1)y_i + (t^3 - 2t^2 + t)hm_i + (-2t^3 + 3t^2)y_{i+1} + (t^3 - t^2)hm_{i+1}$$

where $t = (x - x_i)/(x_{i+1} - x_i)$ and $h = x_{i+1} - x_i$.

1.7 6. Secondary Effects

1.7.1 6.1 Windage: Didion Lag Rule

The engine uses the Didion lag rule for lateral wind deflection rather than the simplified $v_{wind} \times t_{of}$ formula [Didion 1860]. A spin-stabilized bullet doesn't respond to a crosswind change by immediately aligning to the new wind vector — it precesses gyroscopically, which introduces a lag between the vacuum (unretarded) flight time and the actual time of flight. The effective windage exposure time is that lag, not the full time of flight:

$$\Delta t_{lag} = t_{of} - t_{vac}, \quad t_{vac} = \frac{d \times 3}{v_0}$$

$$\theta_{windage} = \frac{12 \cdot v_{cw} \cdot \Delta t_{lag}}{d \times 0.036} \quad [\text{mil}]$$

Wind components are resolved against the bore axis: $v_{hw} = v_{wind} \cos(\theta_{wind})$ feeds the drag calculation as additional relative airspeed; $v_{cw} = v_{wind} \sin(\theta_{wind})$ feeds the windage formula.

1.7.2 6.2 Spin Drift: Miller Stability and Litz Approximation

Spin drift is the transverse displacement of a spin-stabilized bullet toward the direction of rifling twist, driven by gyroscopic coupling between the spin axis and the continuously curving trajectory. It accumulates throughout flight, growing with time-of-flight to roughly the 1.83 power.

Gyroscopic stability factor (Miller [1983]):

$$S_g = \frac{30W}{T^2 D^3 L(1 + L^2)} \cdot \left(\frac{v_0}{2800} \right)^{1/3}$$

where W is bullet weight in grains, T is twist rate in in/rev, D is diameter in inches, $L = L_{bullet}/D$ is length-to-diameter ratio, and the velocity factor applies the Greenhill-Miller correction. S_g is clamped to 5.0 maximum. Values below 1.0 indicate dynamic instability — the bullet will tumble and spin drift is undefined.

Spin drift magnitude (Litz [2009]):

$$d_{drift} = 1.25 \cdot S_g \cdot t_{of}^{1.83} \quad [\text{in}]$$

$$\theta_{drift} = \frac{d_{drift}}{d \times 0.036} \quad [\text{mil}]$$

Direction is toward the twist: rightward for right-hand twist, leftward for left-hand. At 1000 yards with a typical 6.5 mm load on a 1:8" twist barrel, spin drift runs 0.1–0.15 mil — enough to miss steel if ignored.

1.7.3 6.3 Aerodynamic Jump

Aerodynamic jump is less widely understood than spin drift but equally real, and it is the secondary effect most frequently omitted by ballistic calculators. When a spinning bullet exits the muzzle into a crosswind, the gyroscopic reaction to the crosswind deflects the bullet's trajectory perpendicular to the wind direction — vertically, not laterally. The deflection occurs entirely during the brief unsupported flight phase at the muzzle. Because it is an angular deflection established at the muzzle rather than a displacement that grows with range, it is constant in milliradians at all distances.

The Litz approximation is [Litz 2009]:

$$AJ_{mils} = 1.25 \cdot S_g \cdot \frac{V_{cw}}{V_0}$$

The ratio V_{cw}/V_0 is dimensionless; the result is directly in milliradians. There is no range term. This is the correct formulation: aerodynamic jump is an angular offset that persists unchanged as a mrad correction from muzzle to target. (Applying a range-normalization factor such as dividing by mrad at the target distance would be physically incorrect, reducing the correction with range in a way the physics

does not support.)

Sign convention: a positive crosswind (right-to-left from the shooter's perspective) acting on a right-hand-twist bullet produces negative AJ — the bullet strikes low relative to the bore line. Left-hand twist inverts the sign. The correction is zero when twist rate is not configured or when $S_g < 1.0$ (dynamically unstable bullet).

Aerodynamic jump is typically small — under 0.02 mil at modest crosswinds for most precision loads — but it is direction-dependent and constant. Omitting it introduces a systematic elevation bias that appears only in crosswind, which can corrupt DOPE entries recorded under wind and confuse the DOPE-Blend bias computation.

1.7.4 6.4 Coriolis Effect

For long-range shooting with known latitude and firing azimuth, the Coriolis effect contributes both horizontal (windage) and vertical (elevation) deflections. Both are integrated per 0.5-yd step using the step-mean bullet velocity $\bar{v}$:

$$a_{cw,C} = 2\Omega \sin(\varphi) \cdot \bar{v}$$

$$a_{elev,C} = 2\Omega \cos(\varphi) \sin(\alpha) \cdot \bar{v}$$

where $\Omega = 7.2921 \times 10^{-5}$ rad/s is Earth's sidereal rotation rate, φ is geodetic latitude, and α is firing azimuth. In the Northern Hemisphere the horizontal term deflects rightward regardless of azimuth; the elevation term is azimuth-dependent — shooting east produces uplift, shooting west produces depression. At 45°N firing due east at 1000 yards, horizontal Coriolis is approximately 0.02 mil and the vertical component roughly 0.01 mil. Small, but systematic.

1.7.5 6.5 Inclined Fire

For targets at angles above or below horizontal, effective gravity is reduced by the cosine of the firing angle — the rifleman's rule:

$$g_{eff} = g \cos(\theta_{fire})$$

This is applied to the drop integration pass. Output milliradians reference slant range to the target, consistent with how the scope indexes to the actual target distance. The approximation is valid below about 45°; at more extreme angles it slightly under-corrects, though such angles are uncommon in precision rifle work.

1.8 7. DOPE-Blend™ Field-Data Fusion Framework

1.8.1 7.1 The Problem Physics Alone Cannot Solve

Every trajectory prediction from a physics model carries irreducible uncertainty. Measured BCs vary across production lots. Atmospheric readings at the firing line don't tell you what the air density is at 500 yards. A barrel that has seen 3000 rounds has different throat geometry than on the day it was new, and that changes exit conditions in ways no datasheet can capture. None of this means the physics model is wrong — it means the model is working from imperfect inputs.

The conventional response — adjusting BC until the model matches a field observation — works, but conflates all error sources into a single number. A BC adjustment that makes the model match at 500 yards doesn't necessarily make it match at 800 yards, because the spatial signature of “BC error” is different from the spatial signature of “atmospheric model error” or “barrel-specific anomaly.” Change conditions substantially between calibration and shooting, and the correction is only partially valid.

DOPE-Blend™ takes a different approach: it leaves the physics model alone and instead computes the residual between what the model predicts and what the shooter actually observed, expressed in the model's own output space (milliradians of elevation). When conditions change, the physics engine handles the environmental re-projection. DOPE-Blend adds the same systematic correction on top. The two components are doing fundamentally different things, and their combination achieves something neither can alone.

1.8.2 7.2 Bias Residual Computation

For each field DOPE entry — a tuple of (distance, observed elevation, temperature, altitude, pressure, humidity, muzzle velocity at time of observation) — the bias is:

$$\varepsilon_j = \theta_{obs,j} - \hat{\theta}_{model,j}$$

where $\theta_{obs,j}$ is the shooter's confirmed hold in milliradians at distance d_j , and $\hat{\theta}_{model,j}$ is the physics model's prediction at the same distance under the same conditions that were recorded when the observation was made. The difference ε_j captures the systematic error the physics model cannot explain.

Crucially, this residual is computed at the conditions that existed when the observation was made — not today's conditions. A shooter who recorded a +0.08 mil residual at 700 yards on a cold January day can use that same residual on a hot

August range day. The physics engine accounts for the environmental difference; DOPE-Blend contributes only the part that doesn't change with environment.

1.8.3 7.3 Interpolation Strategy

With N bias entries sorted by distance, the DOPE-Blend corrected elevation at target distance d^* is:

$$\hat{\theta}_{DOPE-Blend}(d^*) = \hat{\theta}_{physics}(d^*) + \hat{\varepsilon}(d^*)$$

The interpolated bias $\hat{\varepsilon}(d^*)$ follows a case hierarchy:

Exact match: if $d^* = d_j$ for some j , return ε_j directly.

Interior, $N \geq 3$: Fritsch-Carlson monotone cubic Hermite spline over all N knots. The monotonicity guarantee is essential here — standard cubic splines can oscillate substantially between sparse, irregularly-spaced knots, generating corrections that overshoot or reverse between confirmed data points. Monotone interpolation passes through every data point exactly while constraining behavior between them.

Interior, $N = 2$: linear interpolation between the two entries.

Extrapolation ($d^* < d_1$ or $d^* > d_N$): the nearest endpoint bias is held constant. Extrapolating a spline outside its training range is a recipe for divergence; constant extrapolation is conservative, predictable, and communicates clearly to the shooter that predictions beyond the confirmed DOPE range carry increasing uncertainty.

Windage receives no DOPE-Blend correction. A crosswind observation in field DOPE is confounded with whatever the actual wind was at time of fire, which may have differed from the measured wind at the firing position. Applying a bias to windage would attribute unknown wind variation to a systematic effect. Elevation observations don't have this problem: gravity is constant and the atmosphere's effect on drop is captured in the physics model's conditions.

1.8.4 7.4 Why This Is Novel

Empirical hold tables record raw observed holds, valid only at the exact conditions under which they were recorded, with no environmental re-projection capability.

BC fitting modifies the physics model until it matches field data, but collapses all error sources into one parameter. Apply the fitted BC under substantially different conditions and the error returns, because the calibration absorbed atmospheric error along with BC error.

DOPE-Blend separates the two concerns cleanly: the physics model handles everything that changes with environment (density, speed of sound, temperature-induced MV shift, altitude effects), and the spline handles everything that doesn't (manufacturing tolerances, barrel-specific behavior, systematic BC bias). The formulation is straightforward, but no existing ballistic calculator implements it this way. This is the core differentiating technology of ScopeDOPE and the primary reason it functions as a Ballistic Computer rather than a ballistic calculator.

1.8.5 7.5 Two-Phase API for UI Responsiveness

Computing DOPE-Blend biases is the expensive step: the engine runs a full RK4 trajectory integration for each DOPE entry, at that entry's stored conditions. Applying the biases to a match-day trajectory is cheap: one trajectory pass plus a spline evaluation.

These two steps depend on different inputs. Biases depend on ammo parameters (BC, BC zones, MV), zero configuration, and each entry's stored atmospheric conditions. They do not depend on today's wind speed, today's output distance list, Coriolis settings, firing angle, or current atmospheric conditions. This means the bias list can be cached and reused across repeated match-day recalculations — which is exactly what the app does when a shooter adjusts the wind dial.

The public API exposes both phases:

```
// Phase 1 – expensive; run when ammo, zero, or DOPE entries change
val biases: List<Pair<DopeEntryInput, Double>> =
    BallisticsEngine.computeEntryBiases(dopeEntries, params)

// Phase 2 – cheap; run on every wind, distance, or Coriolis change
val results: Map<Double, TrajectoryResult> =
    BallisticsEngine.interpolateDopeAllFromBiases(biases, distances, params)
```

The single-call `interpolateDopeAll()` function performs both phases and remains available for batch use and testing. The two-phase split exists for the app layer's responsiveness under rapid parameter changes.

1.9 8. Implementation Architecture

1.9.1 8.1 Kotlin Multiplatform

The engine lives in the `:ballistics` KMP module under `commonMain`, with no platform dependencies — pure Kotlin and `kotlin.math`. It compiles to Android via AGP and

JVM for the validation suite. iOS KMP native targets share the same source at v1.0 launch, giving a single implementation across all platforms.

1.9.2 8.2 Singleton, Stateless, Thread-Safe

BallisticsEngine is a Kotlin object — a singleton whose functions are all pure and stateless. Given the same inputs, you always get the same outputs; there is no mutable state anywhere. This makes the engine thread-safe and reentrant without any synchronization machinery, and means the test suite can call any function in any order without setup or teardown. The drag curve objects (G7_CURVE, G1_CURVE) are initialized once at class load.

1.9.3 8.3 Visibility and Trade-Secret Encapsulation

entryBias and applyBias are internal, visible only within the :ballistics module. The DOPE-Blend interpolation logic is not part of the public API and cannot be independently replicated by examining the module's interface. computeEntryBiases, interpolateDopeAllFromBiases, and interpolateDopeAll are the public entry points; their internal operation is opaque to callers. BcModel and TwistDirection are defined in the ballistics package with typealiases at the app layer, keeping the engine API independent of application-layer type hierarchies.

1.9.4 8.4 Offline-First

The engine makes no network calls, opens no sockets, and requires no external services. Every computation runs synchronously on the calling thread. ScopeDOPE must work on a remote ridge with no cell signal; the engine's architecture reflects that requirement without compromise.

1.9.5 8.5 Numerical Precision

All computation uses Double (64-bit IEEE 754), providing 15–16 significant decimal digits. The target output precision is roughly 0.001 mil — about 4 significant figures — so the double-precision margin is vast. Minimum velocity clamping at 1 fps prevents division-by-zero in the drag evaluation and time-of-flight computation without affecting results in any physically meaningful scenario.

1.10 9. Validation Study

1.10.1 9.1 Test Corpus

The validation suite (BallisticsValidationTest.kt) runs via `./gradlew :ballistics:jvmTest` against the `:ballistics` module on the JVM. Test cases are structured JSON files in `ballistics-test-data/test_cases/`, each containing inputs, field DOPE entries, expected outputs, atmospheric records, and metadata.

Table 1: Validation Test Corpus

Category	Count	Description
Sub-Tier-3 (cross-calculator)	30	G7 and G1 loads, engine vs. JBM reference
Tier 3 (confirmed fired)	7	Physical impact data with atmospheric records
Real-world (field DOPE)	15	Included in totals above; multi-session DOPE
Total distinct cases	37	22 calibers from .17 HMR to .338 Lapua Magnum

1.10.2 9.2 Accuracy Tiers and Thresholds

Table 2: Accuracy Tier Definitions

Tier	Reference	OK	INVESTIGATE	FAIL
Sub-3	JBM computed	< 0.025 mil	0.025–0.05 mil	≥ 0.05 mil
Tier 3	Confirmed fired	< 0.05 mil	0.05–0.10 mil	≥ 0.10 mil
Real-world	Field holds	< 0.10 mil	0.10–0.20 mil	≥ 0.20 mil
Transonic	Any	$2\times$ above	$2\times$ above	$2\times$ above

Thresholds derive from three physical floors. The **solver floor** (~ 0.003 mil) is the irreducible solver-to-solver disagreement between correctly implemented point-mass engines, estimated from JBM vs. Applied Ballistics comparisons. The **hardware floor** (~ 0.05 mil) is the standard competition scope adjustment increment — the minimum error a shooter can actually respond to. The **physics floor** (~ 0.097 mil) is the empirical vertical group spread of top-tier F-Class competitors at 1000 yards, approximately 1/3 MOA. Sub-3 thresholds are tight because cross-calculator

comparison should be limited only by the solver floor; Tier 3 and Real-world thresholds allow progressively more error consistent with the progressively lower quality of the reference data.

1.10.3 9.3 Combinatorial Leave-k-Out Cross-Validation

Rather than a simple aggregate error report, the suite applies exhaustive combinatorial leave- k -out cross-validation. For each test case with $N \geq 4$ DOPE entries:

1. Compute bias residuals ε_j for all N entries against the physics model.
2. Enumerate all $\binom{N}{k}$ training subsets for $k = 1, \dots, N-3$, preserving a minimum of three training entries.
3. For each subset, evaluate the DOPE-Blend spline at holdout distances using only the training entries.
4. Classify each holdout as *interior* (training brackets the holdout) or *extrapolation*.
5. **Primary verdict:** worst $k = 1$ interior holdout error — the normal operating condition where the shooter has data on both sides of the prediction.
6. **Stress-test verdict:** worst case across all k — reveals graceful-degradation behavior under extreme data sparsity.

This methodology finds failure modes that aggregate metrics hide. A system can have a good mean error while behaving erratically at specific data configurations. The combinatorial sweep finds those configurations.

1.11 10. Results

1.11.1 10.1 Overall Validation (Run 2026-05-31)

Table 3: Summary Validation Results

Metric	Value
Total cases	37
Skipped (LOO_SKIP, < 4 entries)	1
Primary verdict (k=1 interior LOO)	INVESTIGATE
Stress-test verdict (all k, worst case)	FAIL
Max $ \Delta $, Tier 3 fired interior	0.459 mil @ 7mm PRC 180gr ELD-M, 1000 yd
Max $ \Delta $, Sub-3 all k	1.525 mil @ .22 LR Eley Match, 200 yd

The INVESTIGATE primary verdict means the worst $k = 1$ interior error exceeds 0.025 mil but stays below the 0.05-mil hardware floor — below the resolution of a standard competition scope. The FAIL stress-test verdict is documented graceful-degradation behavior: under extreme data sparsity, constant extrapolation is the correct conservative policy, and the large errors it produces when many entries are held out simultaneously are expected and acceptable. They reflect a data coverage problem, not an engine problem.

1.11.2 10.2 Cross-Calculator Accuracy: G7 Precision Loads

Table 4: G7 Cross-Calculator Results vs. JBM Ballistics

Load	Range	Engine (mil)	JBM (mil)	$ \Delta $	Status
.308 Win 168gr SMK, G7 BC 0.243	1000 yd	12.705	12.705	0.0002	OK
.308 Win 175gr SMK, G7 BC 0.264	1000 yd	10.984	10.984	0.0003	OK
6.5 Creedmoor 140gr ELD-M, G7 BC 0.287	1000 yd	9.221	9.221	0.0004	OK
6.5 Creedmoor 140gr Hybrid, G7 BC 0.315	1000 yd	8.568	8.568	0.0003	OK
6 Creedmoor 108gr Berger Hybrid, G7 BC 0.273	1000 yd	10.231	10.231	0.0001	OK

Load	Range	Engine (mil)	JBM (mil)	$ \Delta $	Status
.300 Win Mag 190gr SMK, G7 BC 0.337	1500 yd	18.630	18.630	0.0003	OK
.300 Win Mag 200gr ELD-X, G7 BC 0.352	1500 yd	17.398	17.399	0.0003	OK
6.5 PRC 147gr ELD-M, G7 BC 0.315	1000 yd	7.943	7.943	0.0003	OK

Agreement is uniformly sub-0.001 mil across the primary precision rifle load set — three orders of magnitude below the FAIL threshold. The 1.26× McCoy calibration achieves its goal.

1.11.3 10.3 Fired Data: 6.5 Creedmoor Tier 3 Case

Table 5: JBM Cross-Reference — 6.5 Creedmoor 140gr A-MAX, Tier 3 (Confirmed Fired)

Dist (yd)	Engine (mil)	JBM (mil)	Δ Eng–JBM	Fired (mil)	Δ Eng–Fired
300	0.713	0.713	−0.0001	0.727	−0.014
500	2.431	2.428	+0.003	2.473	−0.042
700	4.519	4.516	+0.003	4.509	+0.010
900	7.054	7.049	+0.005	7.054	0.000
1000	8.531	8.519	+0.012	8.581	−0.050

Engine-to-JBM divergence grows to 0.012 mil at 1000 yards — flagged at the 0.01-mil DIVERGENCE threshold but well inside the 0.05-mil hardware floor. The engine-to-fired pattern tracks the JBM-to-fired pattern closely, which points to input data uncertainty (the atmospheric conditions at time of fire weren't perfectly recorded) rather than engine error. This is precisely the scenario DOPE-Blend is designed to handle: with the confirmed fired holds as training data, the spline ab-

sorbs the residual systematic offset and closes the remaining gap below hardware resolution.

1.11.4 10.4 Open Issues

1. .338 Lapua Magnum divergence. The .338 Lapua (G7 BC 0.342, MV 2969 fps) shows engine-vs-JBM divergence reaching 0.110 mil at 1000 yards. Both engine and JBM show the same miss direction versus fired data, which points toward uncertain input conditions (incomplete atmospheric records for the fired case) rather than a pure engine error. The load pushes toward the upper boundary of the validated $1.26\times$ calibration range; a dedicated range-specific calibration check against JBM at full atmospheric documentation is planned.

2. Subsonic .22 LR edge case. The 1.525-mil stress-test error on the .22 LR Eley Match case is not an engine failure — it's the expected result when a subsonic load is evaluated well outside the DOPE-Blend training range via extreme holdout combinations. Constant extrapolation doesn't produce harmful corrections here; it simply can't extrapolate transonic behavior from near-muzzle data. This reflects the physical limits of point-mass modeling in the transonic regime rather than any deficiency in the interpolation method.

1.12 11. Discussion

1.12.1 11.1 G7 Calibration Range

The $1.26\times$ factor was derived and validated for G7 BC 0.25–0.35 and MV 2400–3200 fps, which covers the dominant precision rifle use cases: 6mm and 6.5mm match loads, .308 Win, .300 Win Mag with appropriate projectiles. Outside this range, the factor's validity is unverified. The .50 BMG 750gr A-MAX (G7 BC 0.581) shows a consistent ~ 0.1 -mil low prediction versus fired data at 1000 yards, consistent with the calibration factor being slightly off at high BC. Whether the McCoy-JBM Cd ratio is truly constant across the full BC-Mach space remains an open question; characterizing it at higher BC values is the logical next step for extending the engine's validated range.

1.12.2 11.2 Transonic Regime

Point-mass models using G1 or G7 reference drag functions physically degrade in the transonic zone (roughly Mach 1.0–1.2). Cd changes rapidly and nonlinearly there, and the reference function geometry stops being representative for actual

bullets of varied nose and base design. The validation suite applies $2\times$ thresholds for transonic cases and the results support this relaxation.

For precision rifle shooting with supersonic loads — the primary ScopeDOPE use case — the bullet is well above Mach 1.2 at any range where the hold decision matters. .308 Win, for instance, is above Mach 1.2 at 800 yards under standard conditions. The transonic limitation affects rimfire and ultra-long-range large-caliber work more than it affects the core user population.

1.12.3 11.3 DOPE-Blend Versus Calculator-Class Approaches

Ballistic calculators in the current market offer at most a BC-fitting workflow: the user adjusts the BC input until the prediction matches a field observation, then uses the adjusted value going forward. This is a useful approximation but conflates all error sources into one parameter, produces corrections that deteriorate when conditions change substantially, and offers no interpolation across multiple confirmed data points.

No existing ballistic calculator implements a residual decomposition approach like DOPE-Blend. The formulation is straightforward — compute biases in the physics model's output space at observed conditions, interpolate between them, add to the match-day physics prediction — but the result is qualitatively different from anything a calculator produces. A shooter who invests in building their DOPE library gets a trajectory prediction that improves over time and adapts correctly to environmental changes. That is the defining characteristic of a Ballistic Computer.

1.13 12. Conclusion

The ScopeDOPE ballistics engine delivers agreement at or near the solver floor (sub-0.001 mil vs. JBM) for the primary precision rifle use case, correct environmental modeling through a two-pass zero/match-day integration architecture, and a full suite of secondary effects: Didion windage, Miller/Litz spin drift, Litz aerodynamic jump (corrected to the range-independent formulation in this revision), per-step Coriolis integration, and the rifleman's rule for inclined fire.

DOPE-Blend™ is the core differentiating contribution. By working in the physics model's output space and computing residuals at the conditions when observations were made, it correctly separates the environmental re-projection problem — where the physics engine is the right tool — from the systematic correction problem — where confirmed field observations are the right tool. The two-phase API added in v0.9.5 ensures the app layer can apply DOPE-Blend corrections in real

time without triggering expensive re-integration under rapid parameter changes. The engine runs entirely on-device. No server. No accounts. No dependencies that fail in the field. That's not just an implementation detail — it's the point.

1.14 References

McCoy, R. L. (1999). *Modern Exterior Ballistics: The Launch and Flight Dynamics of Symmetric Projectiles*. Schiffer Publishing.

Litz, B. (2009). *Applied Ballistics for Long Range Shooting*. Applied Ballistics LLC.

Miller, D. G. (1983). Graphical determination of gyroscopic stability for spin-stabilized projectiles. *Propellants, Explosives, Pyrotechnics*, 8(2), 59-62.

Fritsch, F. N., & Carlson, R. E. (1980). Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17(2), 238-246.

International Civil Aviation Organization (ICAO). (1993). *Manual of the ICAO Standard Atmosphere* (3rd ed.). Doc 7488.

Magnus, G. (1844). On the deviation of projectiles, and a remarkable phenomenon of rotating bodies. *Annalen der Physik*, 164(1), 1-28.

Didion, I. (1860). *Traité de balistique*. Mallet-Bachelier, Paris.

Siacci, F. (1880). Balistique extérieure. *Rivista di Artiglieria e Genio*.

Copyright © 2026 Jason M. Schwefel / Cold Bore Ballistics LLC. All Rights Reserved.

DOPE-Blend™ is a proprietary trademark of Cold Bore Ballistics LLC.

ScopeDOPE is a trademark of Cold Bore Ballistics LLC.