

Frozen design variables in simulation-based optimization: the cost of an unlocated threshold event

Adrián Klapp Fritz

`adrian.klapp.f@mail.pucv.cl`

Pontificia Universidad Católica de Valparaíso

José Luis Valín Rivera

Pontificia Universidad Católica de Valparaíso

Cristóbal Galleguillos Ketterer

Pontificia Universidad Católica de Valparaíso

Research Article

Keywords: design sensitivity analysis; simulation model verification; threshold event location; hybrid system sensitivity; complex-step differentiation; interior ballistics

Posted Date: October 23rd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10768900/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

Frozen design variables in simulation-based optimization: the cost of an unlocated threshold event

Adrián Klapp Fritz^{1,2,*}, José Luis Valín Rivera¹, Cristóbal Galleguillos Ketterer¹

¹Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile

²Academia Politécnica Militar, Santiago, Chile

*Corresponding author: Adrián Klapp Fritz, adrian.klapp.f@mail.pucv.cl

Abstract

We consider design optimisation in which the objective and constraints are readouts of a numerical integration rather than analytic functions, and ask whether a simulator refined until its outputs stop moving delivers a usable Jacobian. It need not. Across a state-dependent threshold event the discrete map can converge to the continuous one while its derivative does not; here we quantify what that costs a constrained engineering search and isolate the cost to the corrupted column. In a lumped-parameter interior ballistics model with one closed-form Jacobian column and a multi-precision derivative reference independently verified beyond twenty-five digits, reading the quantity of interest at the first node past the threshold, not at the crossing, makes one column zero almost everywhere, and three high-accuracy paths return exactly zero on it. We formalise when this becomes a frozen design variable: a projected first-order method whose direction lies in the span of the objective and constraint gradients leaves that coordinate where it started, and its constrained optimum is the box bound. Releasing only that coordinate from the returned design, the other six held bitwise fixed, recovers 4.19 % to 6.91 % of the objective at a bitwise-zero constraint change; refining the time step sixteenfold leaves that value unchanged to a part in five hundred, and a control varying only the Jacobian leaves the coordinate bitwise fixed in every run. A nine-step audit detects the defect before an optimiser consumes it.

Keywords: design sensitivity analysis; simulation model verification; threshold event location; hybrid system sensitivity; complex-step differentiation; interior ballistics

1 Introduction

Consider the constrained design problem an engineer poses to a simulator,

$$\max_{q \in Q} f(q) \quad \text{subject to} \quad c(q) \leq 0, \quad (1)$$

where f and c are not analytic functions but *readouts of a numerical integration* — the quantity of interest at the end of a simulated trajectory. A gradient-based method needs ∇f and ∇c , and it gets them by differentiating the discretised map F_h that the simulator actually evaluates. The step that ordinary output-convergence checks do **not** establish is the one that licenses this: that because the simulator has been refined until its outputs stop moving, its Jacobian is also close to the Jacobian of the model. Convergence of the map,

$$F_h(q) \longrightarrow F(q) \quad \text{as } h \rightarrow 0, \quad (2)$$

does **not** imply convergence of its derivative,

$$DF_h(q) \longrightarrow DF(q), \quad (3)$$

and when the model contains a threshold event that the integrator detects rather than locates, the two separate completely. That separation is a theorem (Stewart and Anitescu 2010) and it is not ours. What is ours is the consequence for the search, measured rather than argued.

The consequence is sharper than a loss of accuracy. In the terminal, state-continuous case studied here the affected column of the Jacobian is not merely biased but **zero almost everywhere, and exactly zero at every design tested**, and for a first-order method whose update in that coordinate is driven by the corresponding components of the objective and constraint gradients — the gradient-projection search of Section 4.4 is one — the coordinate is then not merely mis-stepped but *never stepped at all*:

$$\partial_L f_h = \partial_L c_h = 0 \implies L_k = L_0 \text{ for every iterate } k, \quad (4)$$

at any step length and from any starting point. Proposition 1 is elementary once the hypotheses are written down, and we present it as a formalisation rather than as a theoretical contribution: its job is to fix the exact conditions under which a numerical defect becomes a deterministically frozen variable, not to be difficult. It states those hypotheses precisely, and we are careful not to claim more: a quasi-Newton, preconditioned or momentum method *can* move the coordinate, because its update couples across coordinates — but it moves it on the strength of a Jacobian that is wrong in that column, so the defect is not thereby harmless, only harder to characterise in closed form. Because the constraint does not depend on that coordinate either, its constrained optimum *is* available in closed form (Proposition 2), which turns the cost into something that can be **measured** rather than inferred. Holding the other six coordinates of the returned design bitwise fixed and releasing only the frozen one recovers 4.19 % to 6.91 % of the objective at a change in the active constraint that is bitwise zero (Section 4.4). The optimiser gives no sign of this. It terminates at a feasible design near the active constraint boundary, exactly as it does when given the event-located Jacobian.

The paper closes the loop with a procedure. Section 7 states a nine-step **Event-Sensitivity Audit** that detects the defect before an optimiser consumes the Jacobian, using two cheap checks — a macroscopic secant compared against the pointwise derivative under step refinement, and an enumeration of the code’s threshold surfaces — and is deliberately diagnostic-then-confirmatory, so that a step-independent discrepancy raises a candidate rather than a verdict.

We work in interior ballistics because it is a small, standard, fifty-year-old model whose derivatives are consumed for exactly the purposes at stake — charge design, gun–ammunition matching, and the trade studies that precede a firing campaign (Corner 1950; Krier and Summerfield 1979; Ongaro et al. 2024) — and because it is small enough that a high-precision derivative reference can be computed and independently verified to more than twenty-five significant digits, one Jacobian column admits a closed-form reference, and every number in this paper can be recomputed from a released driver whose core reference model fits in about a hundred lines and evaluates a single design in seconds. That combination is what makes the cost of the defect attributable rather than merely visible. Simulation models across gun-system and ballistic domains are routinely validated against measured data — Selim et al. (2025) against measured recoil behaviour, Catovic (2026) against terminal-ballistics experiments — while their **derivatives** are not. This is a **verification** study in the standard V&V sense (Kuroswski et al. 2025): it establishes what the reported sensitivity coefficients are derivatives *of*. No firing data is required for that, and none is used. Section 6 states what would be required to *validate* the model, and why that is a different exercise.

Where this sits with respect to optimisation under inexact gradients. There is a mature literature on convergence when the gradient handed to the method is not the true one. Carter (1991) establishes global convergence of a trust-region method under a **relative** gradient-error condition — the error is required to stay small compared with the gradient itself. Heinkenschloss and Vicente (2002) do not

impose a single fixed inequality: they formulate **adaptive accuracy requirements** for inexact first-order information in trust-region SQP, tied to feasibility and optimality as the iteration proceeds. What the two share, and what matters here, is that the inexactness is assumed to be **controllable** — either bounded relative to the true gradient, or tightened on demand.

The failure reported here does not satisfy such a condition and cannot be made to. In the terminal, state-continuous case the reported component is $\tilde{g}_L = 0$ while $g_L \neq 0$, a relative error of exactly 100 % in that component. It is not a tolerance the user set too loosely, it is not noise that averages out, and — this is the part that matters for an algorithm designer — **it does not shrink when the time step is refined**, so the usual remedy of tightening the inner solve does not touch it. The inexact-gradient theory is not wrong here; its hypothesis simply fails, and it fails for a reason that lives in the simulator’s event handling rather than in the optimiser’s tolerances.

1.1 Prior work and the boundary of the present claim

The event trap is a theorem. Stewart and Anitescu (2010) proved that differentiating the output of a fixed-step integration across an event yields “gradients that have errors of a size independent of the stepsize”: the value converges under refinement, the sensitivity does not. Nurkanovic et al. (2024) state the consequence for optimal control — fixed-step direct methods “are doomed to fail” — and address it with Finite Elements with Switch Detection, a substantially more general complementarity-based discretisation in which the step sizes are themselves variables and additional conditions force exact switch detection while recovering the order of the underlying Runge–Kutta scheme. The present study uses simple linear event location, which addresses the same structural requirement — aligning the discretisation with the event — by the simplest means available, because the base scheme is first order. The correct sensitivity across an event, including the derivative of the event time itself, is given by hybrid-system sensitivity theory: Galán et al. (1999), Hiskens and Pai (2000), and, in current form, the saltation-matrix treatment of Kong et al. (2024). Event location as a matter of solution *accuracy* is older still (Shampine and Thompson 2000; Shampine et al. 2003). None of this is ours.

What we claim is narrower.

1. *We found no prior report of this failure in the interior-ballistics literature we reviewed.* The studies we reviewed predominantly employ derivative-free optimisation or one-factor-at-a-time trade studies, in which a zeroed Jacobian column never manifests; the search covered the sensitivity studies performed with the standard government codes. We state this as the outcome of a review rather than as a claim of absolute novelty, which is not demonstrable. What is demonstrable is the gap it implies: a result established in the numerical-analysis literature has not propagated into the applied practice of this domain.
2. *The terminal, state-continuous case is a particularly transparent specialisation of the general result.* The published analyses concern discontinuous right-hand sides or state jumps, where the gradient comes out $O(1)$ wrong. Here the pre-exit branch is smooth, the event is terminal only, and there is no state jump at it — so the derivative is not merely wrong but *zero almost everywhere*, with a relative error of exactly -100% at every step size, which makes the phenomenon visible and auditable without any saltation algebra.
3. *A verification suite built from differentiation methods cannot detect this class of defect, and we demonstrate that on ourselves.* Section 4.2 reports a second unlocated threshold in our own corrected code, found after the first was fixed, which biases four Jacobian columns by up to 50 % with an error that does not fall with the step size — while the two round-off-limited paths agree with each other to fourteen digits and the 40-digit reference returns the same slope. We give the two checks that do detect it.

4. *The consequence for a gradient-based design search is measured rather than asserted.* Section 4.4 poses a canonical pressure-constrained gun-system trade study and hands it to a textbook gradient-projection method, twice: once with the naive Jacobian and once with the located one. Under the naive Jacobian the affected variable never moves, from any starting point, and releasing that single coordinate from the design the search returns — with the other six held bitwise fixed — recovers 4.19 % to 6.91 % of muzzle velocity at a peak-pressure change that is bitwise zero. A Jacobian-only control isolates the cause: objective and constraint evaluations are held identical between the two branches and only the derivative information is changed, and the affected coordinate remains bitwise frozen in all six runs under the naive Jacobian and moves in all six under the event-located one. The obvious alternative is tested as well: a derivative-free coordinate search on the same problem does not lose the coordinate, provided its poll is wide compared with one tread of the staircase, and what it pays instead is the smaller value error of the same readout. Section 6 reports that measurement, including the case in which the derivative-free search also fails.

The axis of the paper is the second item. Muzzle velocity is read when travel first exceeds barrel length. With a fixed time step, the natural implementation reads the velocity at the first step past the muzzle, which makes the reported velocity a staircase in barrel length. Its derivative is therefore zero — not approximately and not merely small, but *exactly zero at every design tested, and almost everywhere in barrel length: everywhere except the isolated lengths at which the index of the first node past the muzzle changes, where the staircase steps and the readout is not differentiable at all.* This holds under all three of our high-accuracy differentiation paths, while the velocity itself is off by at most 0.04 %, two orders of magnitude below the model’s own discretisation error. A design study that consumes this Jacobian concludes that barrel length does not affect muzzle velocity.

2 The model and its numerical solution

We use the standard lumped-parameter formulation as it appears in the classical references (Corner 1950; Krier and Summerfield 1979); Ongaro et al. (2024) survey the modern landscape and place 0-D models as the appropriate tool for design-space exploration and optimisation. The state is the burnt fraction, the projectile velocity and its travel; the gas state follows from the Noble–Abel equation with a co-volume, and the burning rate follows the Vieille (Saint-Robert) law:

$$\frac{dZ}{dt} = \frac{a p^n}{e_1}, \quad \tilde{\psi}(Z) = \kappa Z (1 + \lambda Z), \quad \psi = \min(1, \tilde{\psi}), \quad (5)$$

where $\tilde{\psi}$ is the unsaturated form function and ψ the burnt mass fraction actually used in the gas state. The two are distinguished throughout, because the difference between them is exactly the burnout threshold of Section 4.2 and reporting the clipped variable alone would hide where that threshold fires.

$$p = \frac{f \omega \psi + f_{ig} \omega_{ig} - \frac{\theta}{2} m v^2}{V_0 - \frac{\omega(1-\psi)}{\rho_p} - \alpha \omega \psi + A x}, \quad (6)$$

$$m \frac{dv}{dt} = A p \quad (p > p_{\text{start}}), \quad \frac{dx}{dt} = v. \quad (7)$$

Seven design variables are exposed: charge mass, web thickness, burning-rate coefficient, projectile mass, chamber volume, bore area and barrel length. The thermochemical constants are held fixed and are those of a representative single-base propellant; projectile dynamics follow the standard treatment (Carlucci and Jacobson 2025). Online Resource 1, Table S1 tabulates every symbol with its unit and

value. The implementation is clean-room, written from the published equations; no proprietary code was consulted.

The discrete map. This paper is about differentiating the discretised model rather than the continuous one, so the discretisation is stated in full rather than named. With step h and state (Z_j, v_j, x_j) at the start of step j , one step is

$$\begin{aligned}\tilde{\psi}_j &= \kappa Z_j(1 + \lambda Z_j), & \psi_j &= \min(1, \tilde{\psi}_j), \\ V_j &= \max\left(V_{\min}, V_0 - \frac{\omega(1-\psi_j)}{\rho_p} - \alpha\omega\psi_j + Ax_j\right), \\ p_j &= \max\left(0, [f\omega\psi_j + f_{\text{ig}}\omega_{\text{ig}} - \frac{\theta}{2}mv_j^2]/V_j\right), \\ Z_{j+1} &= Z_j + h \frac{a p_j^n}{e_1} \mathbf{1}[\tilde{\psi}_j < 1], & v_{j+1} &= v_j + h_j \frac{A p_j}{m}, \\ x_{j+1} &= x_j + h_j v_{j+1}, & p_{\max} &= \max_j p_j,\end{aligned}\tag{8}$$

with $V_{\min} = 10^{-12} \text{ m}^3$ the free-volume floor and h_j the duration credited to acceleration in step j : zero before the projectile moves, h once it is moving, and $h(1 - \theta_j)$ at the single step in which shot start occurs, with θ_j the sub-step fraction defined in Section 4.2. Three features of this map matter for everything that follows. The position update uses the *updated* velocity v_{j+1} , so the projectile pair is a semi-implicit (symplectic) Euler step rather than a fully explicit one; the scheme is first order overall, which the observed order of 1.08 in Section 2.1 confirms. Every max and every indicator is decided on the real part of its argument, which is what makes the complex-step path of Section 3 valid within a branch. And the two quantities of interest are read off this recursion and not off the continuous solution: $p_{\max}$ is a maximum over the nodes, and muzzle velocity is read at the crossing $x_{j+1} \geq L$ — either at the node, which is the defect of Section 4.1, or by interpolation between x_j and x_{j+1} , which is the repair.

The design box. The seven design variables and the bounds within which every result in this paper is computed.

Table 1 The design box. Bounds within which every result in this paper is computed.

#	Symbol	Variable	SI unit	Minimum	Maximum
1	ω	propellant charge mass	kg	1.6×10^{-3}	2.2×10^{-3}
2	e_1	propellant web thickness	m	1.6×10^{-4}	2.5×10^{-4}
3	a	burning-rate coefficient	$\text{m s}^{-1} \text{ Pa}^{-n}$	3.5×10^{-9}	6.0×10^{-9}
4	m	projectile mass	kg	6.5×10^{-3}	1.1×10^{-2}
5	V_0	initial chamber volume	m^3	2.1×10^{-6}	2.6×10^{-6}
6	A	bore cross-sectional area	m^2	2.3×10^{-5}	2.7×10^{-5}
7	L	barrel length (total travel)	m	0.38	0.60

Online Resource 1, Section S2 states what sets each bound, the sampling procedure and the three separate senses in which the box is admissible; the one the results rest on is that every design actually simulated stays far from every branch surface, with the burnt fraction never approaching unity and the minimum free volume six orders of magnitude above its floor.

The choice of a 0-D formulation is deliberate and is part of the argument. The failure mode reported here is a property of the *interface* between a continuous model and its discrete realisation, and — for a design variable that enters the quantity of interest only through that terminal threshold — it is shared by every code of this family, including the higher-dimensional ones, *that share the two features that matter*: a fixed step and a quantity of interest read at the first state past a threshold. A solver that adapts its step to land on the event, or that already interpolates its readout, does not have the defect, and we do not claim it does. What the minimal model buys is that the derivative reference can be verified to more than twenty-five significant digits, that one column has a closed form, and that every number in this article can be recomputed from a hundred lines of code in seconds.

2.1 Verification of the solution

Refining the step from $8\ \mu\text{s}$ to $0.25\ \mu\text{s}$ changes muzzle velocity by 0.71 m/s and peak pressure by 0.48 MPa, with an observed order of 1.08 for velocity and 1.07 for pressure — first order, as expected for the scheme of (8). The convergence is not perfectly monotone, and the reason is instructive: the residual wobble is the muzzle event moving between time steps, which is the subject of Section 4. Separately, the production model is vectorised over the batch, and for verification we re-expressed the same algorithm as a scalar routine parameterised by its arithmetic; over eight designs the two agree to $5.7 \cdot 10^{-14}$ m/s in muzzle velocity and identify the same pressure-peak index in 8 of 8 cases, so any disagreement reported in Section 3 is attributable to the differentiation and never to the model.

3 Verification of the derivatives: four complementary paths and one closed form

A verification paper cannot establish that its model is right by comparing it against measured firings, and should not pretend to. What it can do is compute the same quantity along paths that fail differently and show that they agree to the level that floating-point arithmetic permits. The four paths execute the same control flow and take the same branches, so they share any error that lives in the *algorithm*; what they do not share is the *arithmetic* that carries the derivative, and that is the part under test.

The four paths are complex-step differentiation at a relative step of 10^{-100} (Lyness and Moler 1967; Squire and Trapp 1998; Martins et al. 2003), forward-mode algorithmic differentiation over hand-written dual numbers, central differences swept over fourteen relative step sizes from 10^{-1} to 10^{-9} , and a central difference carried out in 40-digit arithmetic, which is the reference. The first two are exact within a fixed active branch and share no derivative-propagation implementation with each other, so their agreement is informative rather than tautological. The reference is not exact either, and we measured it rather than assuming it: against a 120-digit gold reference cross-checked two independent ways to a median of 80 significant digits, the 40-digit reference carries a median of 25.9 correct digits for muzzle velocity and 26.7 for peak pressure. Two implementation details decide whether any of this works. Peak pressure is a maximum rather than a state, so its derivative is taken at the located argmax under the conditions of Danskin’s theorem (Bertsekas 2016; Danskin 1967), which we verify hold at 904 of 904 designs; and every branch and every clip is decided on the real part, since a comparison written on a complex value silently contaminates the imaginary channel. Online Resource 1, Section S5 gives all four paths, the precision study and both implementation details in full, and Section S3 the margin from every design to every branch surface, which is what makes the high-accuracy paths valid.

Generative-AI assistance used during code scaffolding and refactoring, language editing, bibliographic cross-checking and adversarial review of earlier drafts is disclosed in full in the *Statements and Declarations*; no AI system was used to design the study, or to generate, select or interpret any numerical result.

One column can also be checked against mathematics rather than against another computation. At muzzle exit the exact sensitivity of muzzle velocity to barrel length is $A p(t_e)/(m v(t_e))$, an identity that owes nothing to the discretisation. Over forty designs the located derivative matches it to a median relative error of 3.9×10^{-4} , which is the discretisation error of the located derivative itself and falls at the expected order under refinement, while the naive readout returns exactly zero against a located sensitivity never smaller than $231\ \text{m s}^{-1}$ per metre of barrel over those forty verification designs. Online Resource 1, Section S6 derives the identity and reports the comparison design by design.

3.1 Result of the verification

Over eight designs drawn across the box and all seven design variables, scored against the 40-digit reference, complex step reaches a maximum relative error of $4.1\text{e-}13$ in muzzle velocity and $4.2\text{e-}14$ in peak pressure and forward-mode AD $3.9\text{e-}13$ and $4.6\text{e-}14$, while the best of fourteen finite-difference steps reaches only $8.0\text{e-}08$ and $2.0\text{e-}08$ and the textbook $\sqrt{\varepsilon}$ step $1.1\text{e-}06$ and $1.5\text{e-}07$. The two round-off-limited paths agree with each other to $2.5\text{e-}14$ and $9.9\text{e-}15$ using independent derivative-carrying arithmetic; their residual against the reference is the accumulation of float64 round-off over three thousand integration steps and cannot be improved without changing the arithmetic of the solver itself. Exactly one of the fourteen finite-difference steps lands within a factor of ten of the optimum for muzzle velocity and two for peak pressure, the optimum is an order of magnitude away from $\sqrt{\varepsilon}$, and it is not the same step for the two quantities computed from the same run — so the right step is a narrow target that cannot be located without already knowing the answer. Online Resource 1, Section S9 gives the full sweep and Figure S1.

And none of it establishes that the derivative of the model is right. This distinction is central to the verification argument, and Section 4.2 is its demonstration. All four paths differentiate the same discretised map, take the same branches, and are evaluated at the same point; if that map carries a bias inherited from an unlocated event, the two round-off-limited paths report that bias with fourteen matching digits, the 40-digit reference reproduces it, and the float64 finite differences report it to within their own floor. Agreement between differentiation methods is evidence about the differentiation, and about nothing else.

4 A zeroed Jacobian column from an unlocated terminal event

Muzzle velocity is defined as the projectile velocity when travel reaches the barrel length. With a fixed time step, the natural implementation reads the velocity at the first step for which travel exceeds the barrel length. That readout is a staircase in barrel length: within one step's worth of travel the reported velocity does not change at all, and the consequence for the derivative is absolute rather than approximate.

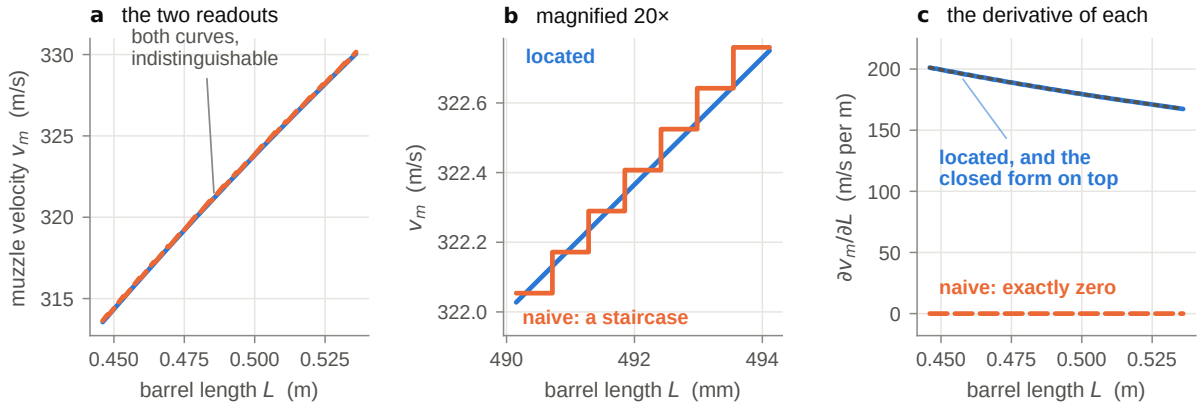


Fig. 1 The mechanism, in three panels. Left: muzzle velocity against barrel length under the two readouts over the whole sweep; the curves are indistinguishable at this scale, which is the entire difficulty. Centre: the same quantity magnified to four millimetres of barrel, where the fixed-step readout is visibly a staircase whose treads are one time step's worth of travel. Right: the corresponding sensitivity, computed by forward-mode algorithmic differentiation of each readout. The fixed-step curve is exactly zero at all 160 sampled lengths. The event-located curve is overlaid with the closed-form identity $A p(t_e)/(m v(t_e))$ of Section 3; the two are indistinguishable at this scale, differing by a median of $1.9\text{e-}04$ in relative terms

Over a sweep of barrel length across the design box, holding everything else fixed, the fixed-step

readout reports a sensitivity of **exactly zero at every one of the 160 points tested**, while the event-located readout averages 183.7 m/s per metre of barrel, ranging from 201.3 at the short end to 168.1 at the long end. An independent check on eight designs drawn across the whole box, not just along the barrel-length axis, gives a mean of 191.1 m/s per metre.

The zero is not an artefact of any one differentiation method, and this is the part worth dwelling on. Complex step, forward-mode AD and the 40-digit extended-precision reference **all three return exactly zero, in 8 of 8 designs**. They agree because they are all differentiating the same discretised map, and that map really is locally constant in barrel length. The defect is in the model's readout, not in the calculus applied to it — which is precisely why no amount of care in the sensitivity method can detect or repair it.

What makes this difficult to detect is how small the output error remains. The two readouts differ in the *value* of muzzle velocity by at most 0.179 m/s over the same sweep, around 0.04 %, far below the model's own discretisation error. A verification exercise that compares outputs will find nothing wrong. The fix is one line: interpolate linearly between the two bracketing steps at the crossing. Proper event location is a solved problem with a standard treatment (Shampine and Thompson 2000; Shampine et al. 2003); the point here is that omitting it costs nothing in the output and everything in the Jacobian, and that any implementation returning the first post-threshold state without locating the crossing exhibits the same staircase mechanism.

4.1 Convergence of the value without convergence of the derivative

The asymmetry is what makes this hard to catch, so we measure both orders on the same runs. Holding the design at the centre of the box and refining the time step from 8 to 0.25 microseconds, against a reference computed with event location at 0.0625 microseconds, the value error of the fixed-step readout falls from 0.484 to 0.008 m/s and that of the located readout from 0.0155 to 0.0004 m/s, while the fixed-step barrel-length sensitivity is 0.000 at all six steps and the located one stays between 183.07 and 183.20.

The value converges under **both** readouts, and the two converge differently in a way that is itself diagnostic. With both events located the error sequence is clean first order: the pairwise orders are 0.97, 1.07, 1.03, 1.11 and 1.22. With the muzzle event merely detected the sequence is erratic (1.28, 1.79, 0.33, 2.13 and 0.48) because the exit index migrates between time steps as the mesh is refined. The value still falls by a factor of sixty over the range, so a practitioner watching the output sees convergence and stops there; the ragged order is the first visible symptom, and it is worth treating as one.

The derivative is a different matter entirely. Under the fixed-step readout it is zero at every step size, so its relative error is exactly -100% regardless of h . Under event location it agrees with the analytic identity of Section 3 to better than 0.082 % at every step tested, monotonically.

This is the practical form of the theorem of Stewart and Anitescu (2010): refinement does not repair a sensitivity taken across an unlocated event. In the degenerate case here the statement sharpens: because the active branch is smooth up to the exit and the event carries no state jump, the discrete muzzle velocity is a genuine staircase in barrel length and the derivative is not approximately but exactly zero almost everywhere.

4.2 Recurrence of the defect in our own corrected code

The correction of Section 4.1 is one line and it is right. It is also incomplete, and we found that out by turning the diagnostic of this paper on the code the paper is about.

Shot start is a second threshold: the projectile begins to move when the pressure first exceeds p_{start} , and in the conventional implementation (ours included, until this section) the whole time step in which that happens is charged to the accelerated phase. The crossing occurs strictly inside the step,

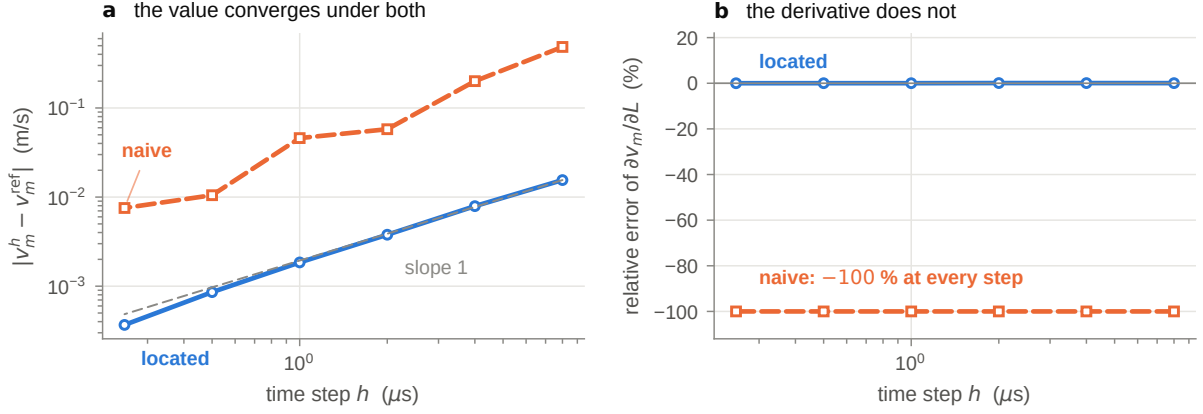


Fig. 2 The two panels are the argument. Left: the value error of muzzle velocity against the time step, under both readouts; both converge, and a practitioner who refines the step and watches the output sees a healthy first-order picture. Right: the relative error of the barrel-length sensitivity on the same runs; under the fixed-step readout it is exactly -100% at every step, and refining the step does not move it at all

so the implementation systematically over-credits the projectile with a fraction of a step’s worth of acceleration, and the size of that fraction jumps as any variable that moves the pressure history crosses a step boundary. The consequence is not the one Section 4.1 reports, and the difference is instructive: there, muzzle velocity was piecewise constant in barrel length and the derivative was exactly zero; here it is a *sawtooth* in charge mass, a smooth ramp with a jump at each step boundary. The value still converges — the sawtooth amplitude halves with the step, at 0.364, 0.199, 0.091 and 0.046 m/s for 4, 2, 1 and 0.5 microseconds — but the number of teeth grows like $1/h$, so the *slope* error does not converge at all.

We measure it by sweeping charge mass over a window spanning several teeth and comparing the pointwise derivative with the macroscopic secant through that window. The window is $\pm 0.5\%$ of the box-centre charge, sampled at 400 points, and the secant is an ordinary least-squares slope through all of them; the window holds between two and seventeen teeth over the range of steps tested, so the secant averages over whole teeth and is not an artefact of where the window begins. The reference is the located derivative at a time step of 0.125 microseconds:

Table 2 Shot start: the error of the charge-mass derivative under the two treatments, against a reference located at 0.125 microseconds. Detecting the crossing leaves an error that does not fall with the step.

time step (μs)	error with shot start detected	error with shot start located	sawtooth amplitude (m/s)
4.00	+50.12 %	−0.161 %	0.364
2.00	+49.92 %	−0.012 %	0.199
1.00	+50.19 %	−0.045 %	0.091
0.50	+50.17 %	−0.008 %	0.046
0.25	+50.21 %	−0.016 %	0.023

Refining the step by a factor of sixteen moves the unlocated error from 50.12 % to 50.21 %, while the located error falls by an order of magnitude. This is the theorem of Stewart and Anitescu (2010) in its generic form — an $O(1)$ error independent of the step size — where Section 4.1 was its degenerate form. The treatment is the sub-step analogue of the muzzle interpolation, and it is stated here in full because “the same interpolation” is not a specification. Let $p_{j-1} < p_{\text{start}} \leq p_j$ be the pressures at the two steps that bracket the crossing. The projectile begins to move at the instant given by linear interpolation of the pressure history,

$$\theta_j = \text{clip}\left(\frac{p_{\text{start}} - p_{j-1}}{p_j - p_{j-1}}, 0, 1\right), \quad h_j = (1 - \theta_j) h, \quad (9)$$

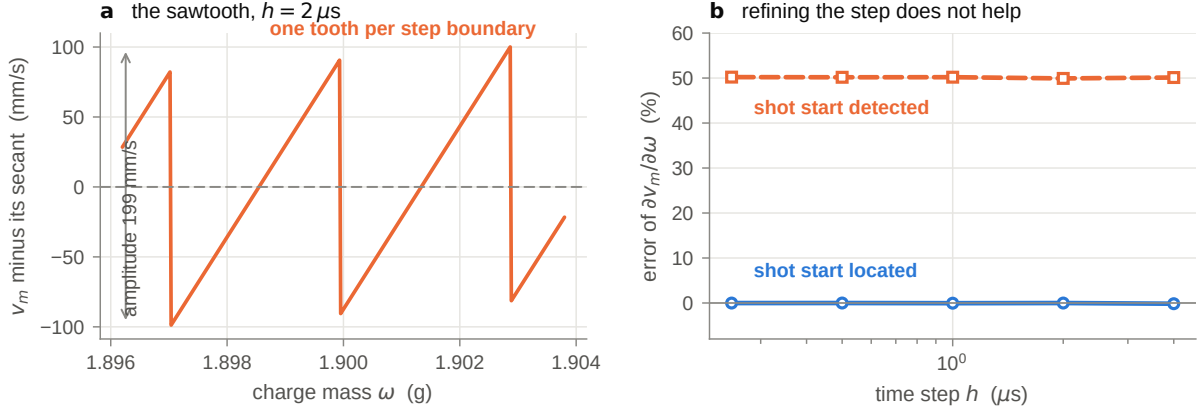


Fig. 3 The second event. Left: muzzle velocity as a function of charge mass with the shot start detected rather than located, plotted as the residual from its own macroscopic secant so that the mechanism is visible; each tooth is the crossing of a time-step boundary and the amplitude at this step is 199 mm/s. Right: the error of the charge-mass derivative against the time step under the two treatments, against a reference located at 0.125 microseconds. Detected, it sits at 50 % and refining does not move it; located, it falls with the step

so only the fraction $(1 - \theta_j)h$ of that step belongs to the accelerated phase. Substituting this h_j into the velocity and position updates of (8) is the whole of the correction; note that the position update there uses v_{j+1} , so the sub-step fraction enters twice and the two occurrences must agree. Every subsequent step is integrated in full, and the burnt fraction and the pressure are evaluated over the whole step under both treatments, so the only quantity that changes is the duration credited to acceleration. The sensitivity propagates through θ_j as through any other expression, and that is what restores the derivative.

Why the verification of Section 3 could not see it, and what does. All four arithmetic paths perturb the design by an amount chosen to be numerically negligible — the 40-digit reference uses a relative step of 10^{-13} , about 2×10^{-16} kg on the charge — while the period of the sawtooth is about 3×10^{-6} kg. Every path therefore sits on the same tread of the same staircase and reports the same slope. The two round-off-limited paths agree with each other to fourteen digits, the 40-digit central-difference reference returns the same slope, and the float64 central differences recover it to within their own floor of 8.0×10^{-8} . All four are wrong by half. This is not a defect of those methods; it is a statement about what they verify.

Two checks do detect it, and both are cheap. The first is the macroscopic secant above: a gap that does not shrink when the step is refined flags a candidate event-handling defect. The second is structural and requires no computation: enumerate every threshold in the code and check that each one is located rather than detected. The enumeration needs a definition, because not every non-smooth surface is a threshold event that requires location. What requires location is a *state-dependent crossing that changes the dynamics or the readout*. Two are active everywhere in the numerical fixture of Section 2 — shot start and muzzle exit — and the target-matched operating point of Section S4 activates burnout as a third. Burnout, $\psi = 1$, is a genuine switch but is never reached inside this box; the two numerical clips, on free volume and on pressure, are floors that guard against non-physical states and are never active. Section S3 reports the margin to all of them. We located muzzle exit and missed shot start, and we report it because the reader’s code will also have more than one. Except where an unlocated treatment is deliberately imposed as the experimental condition — Section 4.1, the naive branch of Section 4.3 and the search that consumes the fixed-step Jacobian in Section 4.4 — every reference and baseline calculation in this paper uses the corrected integrator, with both events located.

4.3 Attribution of the defect to the readout rather than the model or the integrator

Everything above happens inside one interior-ballistics model integrated by one first-order scheme, and a reader is entitled to ask how much of it is a property of either. **For the terminal-readout mechanism, the answer is none of it**, and the cleanest way to show that is to remove the ballistics and then raise the order of the integrator. The qualification matters: the *magnitude* of an event-induced error does depend on the integrator, on the step, on the phase of the crossing within the step and on whether the crossing resets the state, as Section 6 and Online Resource 1 both show. What does not depend on any of them is that, **for a design variable that enters only through the terminal threshold, as L does here**, reading the quantity of interest at a node instead of at the crossing zeroes its column.

An analytic benchmark with no ballistics in it. Take $\dot{x} = v$, $\dot{v} = a$ with a constant, and the same terminal event $x = L$. The closed form is immediate,

$$v_m(L) = \sqrt{v_0^2 + 2aL}, \quad \frac{dv_m}{dL} = \frac{a}{v_m(L)}, \quad (10)$$

which is the constant-acceleration case of the identity of Section 3. Integrate it with the scheme of (8), with Heun and with RK4. Velocity is linear in time and travel is quadratic, so **Heun and RK4 reproduce the discrete trajectory exactly at the nodes**: over five step sizes and five barrel lengths the largest relative node error is 4.0×10^{-15} , which is round-off. The integrator has no error left to blame in those two schemes: nodal integration error is reduced to round-off, which leaves the event readout as the only remaining source of the sensitivity failure. The naive readout still returns a derivative of **exactly zero**, in all fifteen combinations of scheme and step size — each the worst case over the five barrel lengths, seventy-five evaluations in all — while with the crossing located the derivative agrees with the closed form to between 1.4×10^{-4} and 1.5×10^{-3} **for Heun and RK4**, whose nodal trajectories are exact to round-off, and to between 2.2×10^{-4} and 2.8×10^{-3} for symplectic Euler, whose own first-order trajectory error adds to it. In the Heun and RK4 rows the residual is therefore the error of the linear event interpolation and nothing else. *An integrator with no error at all still reports a sensitivity that is wrong by -100% , if the quantity of interest is read at a node instead of at the crossing.*

Three integrators on the ballistics model. The same three schemes, applied to the full 0-D model at the centre of the design box with shot start located in all three, give the same verdict.

Table 3 The muzzle-exit defect under three integrators, on the analytic benchmark of Section 4.3 and on the ballistics model. Heun and RK4 reproduce the benchmark trajectory exactly at the nodes and still report a barrel-length sensitivity of exactly zero under the naive readout.

scheme	benchmark, node error	benchmark, located, rel. error	ballistics, located dv_m/dL
symplectic Euler, eq. (8)	4.3e-03	2.2e-04 to 2.8e-03	183.09 to 183.20
Heun (RK2)	4.0e-15	1.4e-04 to 1.5e-03	183.00 to 183.27
RK4	4.0e-15	1.4e-04 to 1.5e-03	183.00 to 183.27

The naive readout is omitted from the table because it has no spread to report: it is **exactly zero in every row of both experiments**, all fifteen of the benchmark and all nine of the ballistics model. Across the three schemes and three step sizes the located derivative agrees to 8.6×10^{-4} in relative terms. Raising the order of the integrator buys accuracy in the *value* — the gap between the two readouts falls from 0.19 m/s under symplectic Euler to 0.0002 m/s under RK4 at the finest step — and buys nothing whatever in the *derivative*. The same holds at shot start: repeating the macroscopic-secant test of Section 4.2 under each scheme, with the crossing detected the gap is between $+48.5\%$ and $+50.5\%$ under symplectic Euler, Heun and RK4 alike and does not fall as the step is refined, while with the crossing located it is at most 0.077% in every scheme and every step.

A state jump behaves differently, and the contrast is worth one paragraph. Online Resource 1,

Section S7 differentiates the height of a bouncing ball at a fixed horizon with respect to its initial height, over three initial heights, two restitution coefficients and four step sizes. Detecting the impact rather than locating it does not zero the derivative: it collapses it to the event-free value, exactly 1.000 in all twenty-four rows, against an exact range of 0.062 to 0.782. Locating the crossing repairs it only if the remaining sub-step is integrated; stopping at the located node leaves an error of -111% , worse in some rows than not locating the event at all. The defect of this article is therefore the state-continuous member of a family whose other members fail differently, and the repair is not one line in every case.

4.4 The cost of a zeroed column to a gradient-based search

The barrel-length column is not an academic entry. It is the column consulted when a weapon is shortened for a carbine or vehicle variant, when a common cartridge is qualified across weapons of different barrel length, and when a gun model is calibrated by adjusting parameters to match measured velocities. In every one of those uses the code, run with a fixed-step threshold readout, reports that barrel length has no effect at all — with a residual of exactly zero, which reads as certainty rather than as failure. The consequences differ in kind, and so does the evidence we have for each. In a calibration, the zero makes barrel length locally unidentifiable, and it may then force compensation through whichever parameters are still free; **we do not quantify that effect here** and state it as a possibility rather than a finding. In a qualification decision, the code reports no muzzle-velocity penalty from shortening the barrel; that follows directly from a derivative of exactly zero, but the decision consequence is a matter of how the number is used and we do not model it. **The design-search consequence, by contrast, is measured below**, and it is the only one of the three we assert.

We pose a canonical pressure-constrained gun-system trade study — maximise muzzle velocity subject to a peak-pressure limit, over the normalised design box — and solve it by **gradient projection** from six starting points: ascend v_m ; project the ascent direction onto the tangent of the pressure constraint while it is active; descend $p_{\max}$ whenever the iterate leaves the feasible set. The limit is set at 90 MPa, which is a *fixture-level active constraint* chosen inside the numerical box of Section 2 so that it binds; it is not a safety limit of any real system, and nothing below should be read as one. Nothing about the optimiser is unusual either; that is the point. It is run twice, once given the Jacobian of the fixed-step readout and once given the Jacobian of the event-located readout, and **both final designs are then scored with the located model**, because the design that leaves the optimiser is the one that gets built. All twelve endpoints satisfy the constraint and all twelve lie within 1.5 MPa of it, between 88.57 and 89.99 MPa, so the two searches are compared against an active constraint rather than in the interior. Online Resource 1, Section S10 states the algorithm, its parameters and its starting points in full.

Table 4 What a zeroed Jacobian column costs a gradient-based design search: maximise muzzle velocity subject to a 90 MPa peak-pressure limit, by gradient projection from six starting points. Both final designs are scored with the located model, because the design that leaves the optimiser is the one that gets built, and all twelve endpoints satisfy the constraint.

	naive Jacobian	located Jacobian
runs in which barrel length moved at all	0 of 6	6 of 6
largest excursion of L from its starting value	exactly 0	—
runs ending at the box bound $L = 0.600$ m	0 of 6	4 of 6
mean final barrel length	0.492 m (its starting value)	0.596 m
endpoints satisfying $p_{\max} \leq 90$ MPa	6 of 6	6 of 6

Two propositions delimit exactly what the zero column does. We state them because the interesting claim is not that our optimiser stalled — that is an observation about one run — but that a class of methods must stall, and that the optimum they miss is available in closed form.

Proposition 1 (frozen coordinate). Let a search generate iterates by $u_{k+1} = \Pi(u_k +$

$\eta d_k / \|d_k\|$), where Π is the projection onto a box and is therefore separable, and where the direction d_k lies in the span of the objective and constraint gradients, $d_k \in \text{span}\{\nabla_u f_h(u_k), \nabla_u c_h(u_k)\}$. If $\partial_L f_h = \partial_L c_h = 0$ at every iterate, then $L_k = L_0$ for every k , at any step length η and from any starting point. The update is read at iterations with $d_k \neq 0$; if $d_k = 0$ the direction is undefined, the method has no step to take and terminates, and the conclusion holds trivially. *Proof.* Every vector in the span has zero L -component, so $(d_k)_L = 0$ and the unnormalised update leaves L unchanged; a separable box projection acts coordinatewise, so it cannot reintroduce a change in L . Induction on k . $\square$

The gradient-projection search of this section satisfies both hypotheses exactly: its three cases — ascend, project onto the constraint tangent, descend — are all combinations of $\nabla_u v_m$ and $\nabla_u p_{\max}$, and Π is the elementwise clip to $[0, 1]^7$. **We claim no more than that.** A quasi-Newton step, a non-diagonal preconditioner, momentum carrying state from earlier iterates, or a trust-region model with coupling can all give L a non-zero component even when $\partial_L f_h = 0$; what they cannot do is give it the *right* one, since the information that would set it is exactly what the zeroed column has destroyed.

Proposition 2 (the omitted optimum). If $\partial_L f > 0$ and $\partial_L c = 0$ throughout the box, then for any fixed values of the remaining coordinates the constrained optimum in L is the upper bound, $L^* = L_{\max}$. *Proof.* Since $\partial_L c = 0$, the constraint value $c(q_{1:6}, L)$ is constant in L on the admissible interval $[L_{\min}, L_{\max}]$, so for fixed values of the remaining coordinates feasibility with respect to c is unchanged throughout that interval — every admissible L is feasible whenever any is; f is strictly increasing in L ; the maximum over a compact interval of a strictly increasing function is at the upper end. $\square$

The two hypotheses are of different kinds, and we treat them differently.

The first is analytic. The closed-form identity of Section 3 gives $\partial v_m / \partial L = A p(t_e) / (m v(t_e))$ under the located readout, so $\partial v_m / \partial L > 0$ follows from $p(t_e) > 0$ and $v(t_e) > 0$ — conditions on the state at exit, not on a sample. Over 1152 evaluations described below the smallest pressure at exit is 5.49 MPa and the smallest velocity 193.9 m/s, so the identity delivers strict positivity rather than our having sampled for it; the smallest located $\partial v_m / \partial L$ measured directly is 20.8 m/s per metre.

The second is not analytic and we do not present it as such. $\partial p_{\max} / \partial L = 0$ holds when the pressure peak occurs before the projectile has travelled the shortest admissible barrel. We check that **where it would break first** rather than on a random sample: at $2^6 = 64$ **near-corner stress points**, each coordinate offset 2 % inside one bound or the other, plus a 64-point Halton fill of the interior, each design swept over nine barrel lengths spanning the box — 1152 evaluations, all of which exit within the horizon. The largest travel to the pressure peak anywhere in that sweep is **0.0503 m against a shortest admissible barrel of 0.380 m, a margin of a factor of 7.6**, and the measured $|\partial p_{\max} / \partial L|$ is **exactly zero at every point checked**. The worst case is at a near-corner point (0.132) and not in the interior fill (0.113), which is why the near-corner set is there: a random fill concentrates in the middle of the box, and the middle is not where we would expect this hypothesis to be closest to failing.

This is a directed sweep and not a proof, and we say so. Proposition 2 therefore gives $L^* = L_{\max} = 0.600$ m **at every design at which we invoke it** — the six the search returns and every design in the reported sweeps — and we do not claim it for every point of Q . That is enough for what the proposition is used for here, which is to make the cost of the frozen column measurable rather than inferable.

Coordinate release: the causal cost of the frozen column. Take the design the naive search returned, hold its other six coordinates bitwise fixed — the same float64 values, not re-drawn and not re-optimised — and move only barrel length from where the search left it to L^* . Nothing else varies, so the difference is attributable to the frozen column and to nothing else.

Table 5 Coordinate release: the causal cost of the frozen column. The other six coordinates of the design the naive search returned are held bitwise fixed and only barrel length is moved to its constrained optimum, so the difference is attributable to that column alone.

coordinate release from the naive design	measured over 6 runs
velocity recovered, Δv_L	+18.5 to +29.4 m/s (mean +24.9)
as a fraction of the design’s own velocity	+4.19 % to +6.91 % (mean +5.61 %)
change in peak pressure, Δp_L	bitwise zero in 6 of 6
constraint still satisfied after release	6 of 6

And the cost does not go away when the step is refined, which is the check that separates this from ordinary discretisation error. Holding each of the six designs bitwise fixed and halving the time step four times with the physical horizon constant — a factor of sixteen in h — the recovered velocity moves by less than a part in five hundred, and fitting $\Delta v_L(h) = a + b h^p$ gives an observed order $p = 1.00$ and a limit a between 18.51 and 29.37 m/s, mean 24.91, **non-zero at every design**. Refinement removes the error in the value and leaves the cost of the zeroed column exactly where it was. The refinement is done at a fixed design on purpose: re-optimising at each step would change the map, the Jacobian, the search path and the active set at once, and the effect of h could no longer be separated from the effect of the trajectory. The same sweep also records that the naive barrel-length derivative is **exactly zero** at all thirty design-step combinations, so the premise of Proposition 1 holds throughout the sweep rather than only at the step at which it was first measured.

The Jacobian alone is responsible. The two searches compared above differ in everything at once: the naive one evaluates *and* differentiates with the fixed-step readout, so its objective values, its constraint values, its Jacobian, its active set and its whole trajectory are all different from the located one’s. A reader is entitled to ask whether the coordinate is frozen because the column is zero or because the values were wrong too. We answer it with a control in which only one thing differs. In both branches the objective and the constraint are evaluated with the **event-located** model at every iterate; the Jacobian is taken from the fixed-step readout in one branch and from the located readout in the other. Every other element — starting points, step length, iteration budget, projection, feasibility restoration, seed — is identical.

Table 6 The Jacobian alone: a control in which the objective and the constraint are evaluated with the event-located model in both branches at every iterate, and only the Jacobian differs. Starting points, step length, iteration budget, projection, feasibility restoration and seed are identical. Under the naive Jacobian the barrel-length coordinate is bitwise unchanged from its starting value in all six runs.

same values, two Jacobians	naive J	located J
barrel length moved at all largest normalised excursion of L , smallest over the six runs	never, in 6 of 6 exactly 0	in 6 of 6 0.287 (range 0.287 to 0.604)
mean final L	0.494 m	0.593 m
endpoint feasible	6 of 6	6 of 6

Under the naive Jacobian the coordinate is *bitwise* unchanged from its starting value in every run, with correct objective and constraint values throughout. The freezing is therefore attributable to the derivative information and to nothing else. This branch is a control and not a workflow: nobody holds correct values together with wrong derivatives on purpose, and anyone who had the located model would differentiate with it. Its only job is to leave one variable loose.

The frozen column costs between 4.19 % and 6.91 % of muzzle velocity, and it costs it **without any additional peak-pressure margin**: the peak pressure of the released design is not approximately but bitwise identical to that of the frozen one, so the recovered velocity is not bought against the active constraint. The end-to-end comparison of the two searches gives a larger gap — 25.8 to 42.7 m/s, mean

33.5 — but it confounds the frozen column with the whole trajectory of the optimiser: the step size, the iteration budget, the starting point and the behaviour of the other six coordinates. We report it as context and not as the cost.

The failure is not exposed by the optimiser’s standard convergence or feasibility outputs. The optimiser terminates at a feasible design with a plausible objective, near the active constraint boundary, exactly as it does when given the event-located Jacobian. Nothing in its output says that one of its seven design variables was never moved. The check costs minutes: sweep the threshold variable across a range of a few integration steps and confirm that the reported quantity of interest is not piecewise constant. If it is, event location is missing.

5 A second null direction: model structure rather than discretisation error

The verified, event-located Jacobian exposes a second asymmetry in the same model, and we report it in Online Resource 1, Section S8 rather than here because it is a statement about the physics as the 0-D model represents it and not about the discretisation. Barrel length moves muzzle velocity at about 189 m/s per metre while leaving peak pressure invariant to floating-point zero at every one of 600 designs — the pressure peak occurs after at most 45.8 mm of travel against a shortest barrel of 380 mm, so the barrel end is not in the pressure history. This is not a rank deficiency: under the row normalisation declared there the two-by-seven Jacobian is close to rank one, with 92.6 % of its squared energy in a single singular direction. What forbids the inference is the exact zero, not the conditioning. We separate the two results deliberately, because the first is a defect of the code and the second is a property of the model, and only the first is repaired by locating the event.

6 Scope of the claim: what is and is not established

The design box is a numerical fixture, not a weapon, and the paper should not be read as if it were. Over the box the model produces a 5.4–5.9 mm bore driving a 6.5–11.0 g projectile to 237–480 m/s at peak pressures of 39–132 MPa, with only 11–38 % of the charge burnt at exit. That is an under-burnt regime and not a fielded operating point; we chose the box so that every trajectory stays far from the numerical floors and so that the burnout branch is never entered, which is what makes the derivative study clean, and the price is that the operating point is not a fielded system. Every statement in this paper is a statement about the discretisation and about the structure of the equations, and both transfer to a realistically parameterised box.

That is a promise, and a reviewer is entitled to the demonstration. Online Resource 1, Section S4 builds a second design point whose bore diameter (5.56 mm) and pressure limit (4300 bar) are taken from a published standard (C.I.P. 1984), matches it to 930 m/s and 430 MPa by adjusting two parameters against two targets, and repeats the experiments there. Three results carry over. The realistic case activates a **third** threshold, burnout, because the charge is essentially consumed. The muzzle-exit defect survives the move unchanged: over 401 barrel lengths at 25 μm spacing the naive readout is bitwise constant across 98.8 % of the intervals, while the located readout gives 239.0 m/s per metre. And the shot-start defect does **not** survive unchanged — it biases the charge-mass column by 3.3 % there against 50 % in the fixture box, because at a realistic loading density the pressure crosses the threshold far more steeply and the discarded sub-step fraction is a much smaller part of the step. In the three events examined the magnitude of the error tracked the local crossing dynamics and the size of the unresolved sub-step fraction rather than the mere presence of the threshold. This refines the recommendation of Section 7 rather than weakening it: enumerating the thresholds remains necessary, but which of them costs anything has to be measured.

A derivative-free method does not inherit the frozen column, and the escape has a precondition. The immediate response to a corrupted Jacobian is to stop computing one, and that deserves a measurement rather than an opinion. We ran the design problem of Section 4.4 again with a coordinate search over the positive spanning set $\pm e_i$, an extreme barrier for the pressure constraint, the poll doubled on success and halved on failure, from the same six starting points and under both readouts, scoring every branch at the end with the event-located model. **Under the naive readout the search does not lose the coordinate:** barrel length reaches the box bound in all six runs, as it does under the located readout, at true muzzle velocities differing by at most 0.304 m/s. We state that first because it is the result a reader could hold against the paper.

The escape is conditional on the poll size, and the condition is measurable. One tread of the staircase is $v \Delta t$ — an identity we check against a bisection between two consecutive tread boundaries, agreeing to four parts in ten thousand — which is 0.53 to 0.75 mm at the six starts and grows with velocity along the search. While the poll is wide compared with a tread, every step that crosses one improves the objective and the search climbs to the bound. Started instead at a poll of 0.002 in normalised coordinates, which is 0.67 of a tread, the naive branch reaches the bound in none of the six runs and finishes at a mean barrel length of 0.494 m, against 0.551 m for the located branch on the same starts.

The quantisation is visible throughout, in a statistic that needs no stopping tolerance to state. Classifying every poll in the barrel-length direction by the ratio of its realised step to the local tread, the fraction returning a **bitwise identical** velocity falls from 82 % below a quarter of a tread, through 56 % and 17 %, to **zero in all 361 polls wider than one tread**; under the located readout it is zero in all 746 polls of both problems. That, and not a poll direction that stops improving, is the signature a practitioner should look for: in the target-matching problem below, the barrel-length poll fails to improve under *both* readouts.

Moving the optimum into the interior — matching a target muzzle velocity of 322 m/s, everything else unchanged — the naive branch converges as tightly as the located one, to within 1.2×10^{-5} m/s of the target *on its own objective*, and is nonetheless 0.011 to 0.120 m/s away from it in truth, worse in every run. That residual is the bias between the two readouts, at most 0.037 % of the velocity and consistent with the value error reported in Section 4.1; it is not a derivative effect, and barrel length fails to move in five of the six runs under both readouts. **The defect this article reports is therefore specific to methods that consume the Jacobian.** What a derivative-free method pays instead is the value error of the same readout: smaller, already quantified here, and it does not remove a design variable from the search.

We make no claim about absolute accuracy against firings, and none should be inferred. This paper compares a model against itself, against an independent re-implementation, and against a reference carried in forty-digit arithmetic and verified to more than twenty-five significant digits. In the standard simulation V&V vocabulary (Kuroswiski et al. 2025), that is **verification**; **validation** — whether those equations describe a real gun within a stated tolerance — is a separate exercise requiring an instrumented campaign with a traceable measurement chain and an uncertainty budget for both the experiment and the simulation. The distinction is not a hedge. The muzzle-exit trap is a statement about a discretised map, decided by the fact that three computationally distinct high-accuracy arithmetic paths return the same zero; measured velocities could not strengthen it, because the defect lies between the continuous model and its discrete realisation and is present whatever the model’s absolute accuracy.

7 The Event-Sensitivity Audit for simulation-based optimization

The nine-step audit below is the practical methodological contribution of this study. It is stated as a procedure so that it can be run rather than agreed with, it costs a handful of high-accuracy Jacobian evaluations — steps 6 and 9 repeat them at several time steps and again after the repair — together with a few one-dimensional sweeps, and every step of it is executed on the released code. Its structure

is deliberately **diagnostic-then-confirmatory**: steps 1 to 7 raise a candidate, and only steps 8 and 9 establish the cause.

1. **Enumerate.** List every surface $g_i(y, q) = 0$ at which the code takes a decision — every comparison of a state variable or a state function against a constant. Include clips and floors; exclude nothing at this stage.
2. **Classify.** For each surface, decide whether crossing it changes the *dynamics*, changes the *readout*, or neither. Only the first two are threshold events that require location. A floor that is never active is not an event; a readout taken at the first node past a threshold is one, even though nothing about the dynamics changes there.
3. **Measure the margin.** For every design in the study, record the distance to each surface, in the units in which the crossing is decided. A surface that is never approached can be set aside, and the fact that it was checked is what makes setting it aside legitimate (Section S3).
4. **Take the pointwise derivative.** Complex step or forward-mode AD, so that no finite-difference differentiation step has to be tuned. The integration step h of course remains, and is the variable of step 6. Record it.
5. **Take the macroscopic secant, and check that it does not depend on the window.** Sweep each design variable over a window wide enough to contain several crossings of the surface under test, and fit an ordinary least-squares slope. The width is a real degree of freedom with two opposing failure modes — too narrow and the window holds too few crossings, so the slope is dominated by where the window happens to start inside one of them; too wide and the physical curvature of the quantity of interest turns the secant into a chord — so repeat the fit at two or three widths and require the *verdict* of step 7 to be unchanged. Two cheap diagnostics probe these two failure modes: count the crossings the window holds, and compare the least-squares slope with the chord through the window endpoints. We report both. Over five half-widths spanning a factor of sixteen, holding between 1 and 49 crossings, the fitted slope moves by 0.18 % in relative terms, the chord diagnostic stays below 2.6×10^{-4} , the diagnosed bias of the unlocated derivative stays between 50.04 % and 50.45 %, and the located derivative agrees with the secant to better than 0.08 % at every width.
6. **Refine.** Repeat steps 4 and 5 at two or three time steps spanning at least a factor of four.
7. **Flag a candidate.** If $|DF_h - DF_{\text{sec}}|$ falls consistently with h , the behaviour is consistent with ordinary discretisation error and does not by itself flag a persistent event-handling defect; it does not prove the absence of one either, since another h -dependent mechanism can converge too. **If it does not fall, flag a candidate event-handling defect** and record the size of the gap, which is an empirical estimate of the magnitude of the suspected bias and not a bound on it: no theorem here makes $|DF_h - DF_{\text{sec}}|$ an upper or lower bound for $|DF_h - DF|$, and the secant carries curvature, window choice and any other active surface along with it. This is a diagnosis and not yet an attribution: a step-independent gap can also come from a second active surface that step 1 missed, from a discontinuity of another kind, from a window that step 5 chose badly, or from an unrelated implementation error. A gap that is exactly -100 % — a pointwise derivative of exactly zero against a non-zero secant — is the characteristic pattern of the terminal, state-continuous case of Section 4.1, and after the structural check of steps 1 to 3 it is particularly strong evidence for that mechanism. It is not a unique signature: quantisation, dead zones, clipping and discretised lookup tables can all produce a zero pointwise derivative against a non-zero secant.
8. **Locate the crossing, and integrate the remaining sub-step.** Locate it to an accuracy commensurate with the integration scheme; linear interpolation between the bracketing steps is the lowest-order procedure and is what the present implementation uses, which is adequate for the

first-order scheme of (8) but will limit the order of the sensitivity under a higher-order integrator. Then integrate the remainder of the step with the same scheme. Both halves are necessary; Section S7 shows what relocating the event without integrating the sub-step costs.

9. **Confirm, and record.** Re-run steps 4 to 7 on the corrected code. Only if the step-independent gap now converges with h is the discrepancy attributable to the event handling; if it does not, the candidate of step 7 was something else and the enumeration of step 1 has to be revisited. Record the cost of *not* locating each event, per column, because it is not predictable from the presence of the threshold: in this model the same defect costs 50 % at one operating point and 3.3 % at another (Section S4).

8 Conclusion

Sensitivities of a simulation code deserve the same scrutiny as its outputs, and they do not receive it. Verifying them along four complementary arithmetic paths, and one column against a closed-form identity, costs seconds of computation and settles a question that a convergence study cannot reach: complex-step differentiation and forward-mode AD make the derivatives exact to solver round-off *within a fixed active branch*, and remove differentiation-step tuning from the derivative calculation — the integration step, of course, remains. Neither, however, can tell that the map being differentiated inherited a bias from an unlocated event: the two round-off-limited paths agree to fourteen digits on the biased slope, and the multiprecision and float64 finite-difference references reproduce that same slope to within their own accuracy. What settles that is the audit of Section 7: enumerate the state-dependent thresholds and confirm each crossing is located rather than detected; compare the pointwise derivative with a macroscopic secant over a window spanning several crossings, and treat a gap that does not shrink under refinement as a **candidate** event-handling defect, confirming the attribution by locating the event, integrating the remaining sub-step and repeating the test; and inspect the verified Jacobian for directions along which one output moves and another does not.

The cost of getting this wrong is not abstract. In the design problem of Section 4.4 the frozen coordinate is worth 4.19 % to 6.91 % of muzzle velocity, recovered at a peak-pressure change that is bitwise zero — the velocity is free of the binding constraint, and Proposition 2 puts the optimum of that coordinate at the box bound in closed form. The attribution is not inferred from the end-to-end comparison: in a control that holds every objective and constraint evaluation identical between the two branches and changes only the derivative information, the coordinate moves in none of six runs under the naive Jacobian and in all six under the event-located one. A textbook optimiser fed the naive Jacobian therefore leaves that velocity on the table, without any indication in its output that a design variable was never moved.

9 Statements and Declarations

Funding. This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval and consent. Not applicable. This study involves no human participants, no animal subjects and no personal data.

Author contributions. A.K.F.: conceptualisation, methodology, software, formal analysis, investigation, writing — original draft. J.L.V.R.: supervision, methodology, writing — review and editing. C.G.K.: validation, writing — review and editing. All authors read and approved the final manuscript.

Data availability. No experimental data were generated or analysed in this study; every quantity reported here is produced by the released code, which is described under *Replication of results* below.

Use of generative artificial intelligence. Generative AI assistance was used for four tasks: scaffolding and refactoring Python code in the released repository; language editing of the manuscript text; cross-checking bibliographic records against Crossref and publisher pages; and adversarial review of earlier drafts, which is how two of the defects corrected in this version were found. The tools used were Anthropic Claude and OpenAI ChatGPT, both accessed in August 2026. It was not used to design the study, to generate, select or interpret results, or to draw conclusions. Every numerical claim in this manuscript is recomputed from the released code by an automated ledger that is part of the deposit, and every reference was verified by the authors against the publisher record. The authors reviewed all AI-assisted output and take full responsibility for the content of this article.

ORCID iDs. Adrián Klapp Fritz, <https://orcid.org/0009-0005-9534-8695>; José Luis Valín Rivera; Cristóbal Galleguillos Ketterer.

Corresponding author. Adrián Klapp Fritz, Pontificia Universidad Católica de Valparaíso, Chile. adrian.klapp.f@mail.pucv.cl

10 Replication of results

Every number in this article and in Online Resource 1 can be recomputed from the released deposit, archived under the MIT licence. It contains the reference implementation, the scalar re-implementation used for the four derivative paths, the extended-precision reference, the drivers for each experiment, the scripts that regenerate every table and figure, and the exact design sets, so that no result depends on a random draw made at read time. The core reference model fits in about a hundred lines and evaluates a single design in seconds; the *complete* replication suite — every sweep, the precision study, the optimisation runs and the figure generation — takes of the order of an hour on one core. The only dependencies are NumPy, SciPy, Matplotlib and mpmath.

Two mechanisms make the claim checkable rather than merely stated. A **traceability ledger** re-reads each numerical claim out of the result files and compares it against the manuscript source, so a figure that changes in the code and not in the text is reported as a mismatch. A **manifest and checker** verify that the files are the ones we deposited, treating two kinds of file differently and deliberately: code and input files are checked by exact SHA-256, because they should be bit-identical, while result files are checked by their column schema and by numerical comparison within a declared tolerance and **not** by digest — a floating-point result regenerated on a different platform, BLAS build or library version is not expected to be bit-identical, and a checker that demanded it would report a failure where none exists. We say this because our first version of the manifest did demand it, and it produced exactly that false alarm.

The code, verification scripts and result files needed to reproduce every numerical claim in this article are archived at Zenodo, doi:10.5281/zenodo.22032878 (Klapp Fritz, Valín Rivera & Galleguillos Ketterer, 2026).

Supplementary information. Online Resource 1 contains the nomenclature table, the admissibility analysis of the design box, the margins to every branch surface, the four arithmetic paths in full, the closed-form identity, the bouncing-ball contrast, the velocity–pressure asymmetry, the standards-constrained target-matched operating point, the complete specification of the gradient-projection search, and the window-width study of the macroscopic secant.

References

- Bertsekas, D. P. (2016). *Nonlinear Programming*, 3rd ed. Athena Scientific, Belmont. ISBN 978-1-886529-05-2.
- Brunet, L. E., Fitas, T., Cuvelier, S., Fontaine, A., Nouguez, B. (2023). Deep and machine learning architecture for high accuracy prediction of interior ballistics for modular charges. *Propellants, Explosives, Pyrotechnics* 48(10), e202300151. <https://doi.org/10.1002/prep.202300151>
- Carlucci, D. E., Jacobson, S. S. (2025). *Ballistics: Theory and Design of Guns and Ammunition*, 4th ed. CRC Press, Boca Raton. ISBN 978-1-032-90945-5.
- Carter, R. G. (1991). On the global convergence of trust region algorithms using inexact gradient information. *SIAM Journal on Numerical Analysis* 28(1), 251–265. <https://doi.org/10.1137/0728014>
- Catovic, A. (2026). Estimation of terminal ballistics parameters for several 7.62×51 mm projectiles using numerical simulations. *The Journal of Defense Modeling and Simulation* 23(1), 29–39. <https://doi.org/10.1177/15485129241240768>
- Commission Internationale Permanente pour l’Epreuve des Armes a Feu Portatives (C.I.P.). *Tableau des Dimensions des Cartouches et Chambres (TDCC)*, sheet 223 Rem., created 14 June 1984. Available at https://bobp.cip-bobp.org/fr/tdcc_public (accessed August 2026). The sheets carry the notice “*Reproduction forbidden as well as in the form of extracts without approval of C.I.P.*”; the present work cites the standard and reproduces no part of it.
- Corner, J. (1950). *Theory of the Interior Ballistics of Guns*. John Wiley & Sons, New York.
- Danskin, J. M. (1967). *The Theory of Max-Min and its Application to Weapons Allocation Problems*. Ökonometrie und Unternehmensforschung 5. Springer, New York. <https://doi.org/10.1007/978-3-642-46092-0>
- Galán, S., Feehery, W. F., Barton, P. I. (1999). Parametric sensitivity functions for hybrid discrete/continuous systems. *Applied Numerical Mathematics* 31(1), 17–47. [https://doi.org/10.1016/S0168-9274\(98\)00125-1](https://doi.org/10.1016/S0168-9274(98)00125-1)
- Heinkenschloss, M., Vicente, L. N. (2002). Analysis of inexact trust-region SQP algorithms. *SIAM Journal on Optimization* 12(2), 283–302. <https://doi.org/10.1137/S1052623499361543>
- Hiskens, I. A., Pai, M. A. (2000). Trajectory sensitivity analysis of hybrid systems. *IEEE Transactions on Circuits and Systems I* 47(2), 204–220. <https://doi.org/10.1109/81.828574>
- Klapp Fritz, A., Valín Rivera, J. L., Galleguillos Ketterer, C. (2026). Frozen design variables in simulation-based optimization: the cost of an unlocated threshold event — code and data [Software]. Zenodo. <https://doi.org/10.5281/zenodo.22032878>
- Kong, N. J., Payne, J. J., Zhu, J., Johnson, A. M. (2024). Saltation matrices: the essential tool for linearizing hybrid dynamical systems. *Proceedings of the IEEE* 112(6), 585–608. <https://doi.org/10.1109/JPROC.2024.3440211>
- Krier, H., Summerfield, M., eds. (1979). *Interior Ballistics of Guns*. Progress in Astronautics and Aeronautics 66. AIAA, New York. ISBN 0-915928-32-9.
- Kuroswiski, A. R., Medeiros, F. L. L., De Marchi, M. M., Passaro, A. (2025). Beyond visual range air combat simulations: validation methods and analysis using agent-based models. *The Journal of Defense Modeling and Simulation* 22(4), 387–404. <https://doi.org/10.1177/15485129231211915>
- Lyness, J. N., Moler, C. B. (1967). Numerical differentiation of analytic functions. *SIAM Journal on Numerical Analysis* 4(2), 202–210. <https://doi.org/10.1137/0704019>
- Martins, J. R. R. A., Sturdza, P., Alonso, J. J. (2003). The complex-step derivative approximation. *ACM Transactions on Mathematical Software* 29(3), 245–262. <https://doi.org/10.1145/838250.838251>
- Nagler, J. (2025). Thermoelastic impact modeling for projectile–target–muzzle components during penetration start of motion. *The Journal of Defense Modeling and Simulation* 22(4), 405–427. <https://doi.org/10.1177/15485129231210300>
- Nurkanović, A., Sperl, M., Albrecht, S., Diehl, M. (2024). Finite Elements with Switch Detection for direct optimal control of nonsmooth systems. *Numerische Mathematik* 156(3), 1115–1162. <https://doi.org/10.1007/s00211-024-01412-z>
- Ongaro, F., Robbe, C., Papy, A., Stirbu, B., Chabotier, A. (2024). Modelling of internal ballistics of gun systems: a review. *Defence Technology* 41, 35–58. <https://doi.org/10.1016/j.dt.2024.05.004>
- Selim, M. S., Elsaady, W. A., Abdelsalam, O. R., Ibrahim, A. Z. (2025). Modeling and design optimization

- of concentric recoil systems for mortar and tank guns. *The Journal of Defense Modeling and Simulation*, OnlineFirst 23 May 2025. <https://doi.org/10.1177/15485129251335042>
- Shampine, L. F., Thompson, S. (2000). Event location for ordinary differential equations. *Computers & Mathematics with Applications* 39(5–6), 43–54. [https://doi.org/10.1016/S0898-1221\(00\)00045-6](https://doi.org/10.1016/S0898-1221(00)00045-6)
- Shampine, L. F., Gladwell, I., Thompson, S. (2003). *Solving ODEs with MATLAB*, §2.3.1. Cambridge University Press. ISBN 978-0-521-82404-0.
- Sobol', I. M., Kucherenko, S. (2009). Derivative based global sensitivity measures and their link with global sensitivity indices. *Mathematics and Computers in Simulation* 79(10), 3009–3017. <https://doi.org/10.1016/j.matcom.2009.01.023>
- Squire, W., Trapp, G. (1998). Using complex variables to estimate derivatives of real functions. *SIAM Review* 40(1), 110–112. <https://doi.org/10.1137/S003614459631241X>
- Stewart, D. E., Anitescu, M. (2010). Optimal control of systems with discontinuous differential equations. *Numerische Mathematik* 114(4), 653–695. <https://doi.org/10.1007/s00211-009-0262-2>
- U.S. Army Materiel Command (1965). *Engineering Design Handbook, Ballistics Series: Interior Ballistics of Guns*, AMCP 706-150, Chapter 2 (differential coefficients). Washington DC. DTIC AD0462060. The primary source of the tabulated coefficients is H. P. Hitchcock, *Differential Coefficients for Interior Ballistics*, BRL Report 169, Aberdeen Proving Ground, 1940.