# Machine-Checked Scalar Foundations for SGD Analysis

A Reusable Corpus of Conditional Algebraic Lemmas

Tamás Nagy, Ph.D.

tnagyphd@gmail.com

Working Paper • 2026-08-06

Build-std v2.0 | 2026-08-06 11:04 | 6910c9f4

## Overview

Familiar optimization arguments often compress many small algebraic steps into a few lines. This makes the exposition efficient, but it can also hide which assumptions are actually used at each step. A reusable checked inventory can make those local dependencies visible without pretending to replace the surrounding mathematics.

The artifact studied here contains 78 named real-arithmetic declarations. Forty-three are selected for the paper because they resemble recurring local steps in analyses of descent, convexity, noise, momentum, variance reduction, schedules, averaging, and regularization. The remaining declarations provide support identities and recovery variants.

The scope is deliberately narrow. The formal source does not define a stochastic process, gradients, convex functions, probability spaces, iterate sequences, or an end-to-end convergence theorem. Optimization notation in the paper is mnemonic notation for real scalars, and every displayed conclusion is conditional on the premises printed with it.

The value of the corpus is therefore auditability and reuse. Each paper-facing statement is linked to a named declaration, the complete source can be replayed, and the limitations identify the semantic layers that a future formalization would need to add before claiming an algorithmic SGD result.

## Abstract

We present a machine-checked source containing 78 named real-arithmetic declarations that occur as local steps in standard analyses of stochastic gradient descent (SGD). This declaration count includes support identities, recovery variants, and direct restatements; it is not a count of novel or publication-credit theorems. The corpus covers sign and ordering facts, scalar forms of convexity and smoothness consequences, one-step distance rearrangements, elementary momentum and mini-batch algebra, assumed control-variate bounds, rate-shape inequalities, schedule monotonicity, and regularization terms. Forty-three selected declarations are exposed in the paper. The verification replay checks 109 registered items: 31 real-symbol registrations and 78 theorem declarations. The kernel introduces no domain axioms, but every declaration remains conditional on the assumptions displayed in its statement. In particular, the development does not define an SGD process, a probability space, a convex function, or an arbitrary-horizon iteration, and it does not prove end-to-end convergence of SGD or its variants. This is an artifact and corpus report, not a claim of new SGD mathematics; its contribution is transparent organization, replayability, and theorem-level traceability for a scalar foundation used in such analyses.

# 1. Introduction

## 1.1 The Problem

Stochastic gradient descent is a standard optimization method in modern machine learning. Its convergence theory is classical, but textbook arguments repeatedly depend on small algebraic transformations whose hypotheses can be obscured by surrounding functional and probabilistic notation. This paper isolates a scalar layer of those arguments and checks each implication mechanically.

## 1.2 Main Results

This paper presents a machine-checked scalar lemma corpus supporting standard SGD analyses. Its central scope claim is:

**Corpus claim (informal).** *Under the scalar premises displayed in each statement, the kernel verifies one-step, rearrangement, sign, monotonicity, and rate-shape consequences used in familiar analyses of SGD and related methods.*

The paper-facing selection covers:

1. **Descent step bounds**: scalar sign and ordering facts (§2)
2. **Convexity-shaped inequalities**: consequences of supplied scalar premises (§3–§4)
3. **Smoothness-shaped descent algebra**: consequences of supplied one-step bounds (§5)
4. **Distance and noise rearrangements**: one-step scalar implications (§6)
5. **Momentum and variance-reduction algebra**: elementary consequences, not algorithmic convergence (§7–§8)
6. **Rate and schedule shapes**: reciprocal and contraction inequalities (§9–§10)
7. **Averaging, scaling, and regularization terms**: scalar identities and inequalities (§11–§12)

## 1.3 Proof Architecture

The source groups theorems by their intended analytical role. These labels provide organization; they do not add function-space or probabilistic semantics absent from the formal statements:

1. **Real arithmetic layer** — Kernel-checked equalities and inequalities over $\mathbb{R}$
2. **Descent primitives** — Basic step bounds (descent_step, loss_decreases)
3. **Convexity layer** — Suboptimality from convexity (suboptimality_from_convexity)
4. **Strong convexity** — Quadratic growth (strong_convex_quadratic_growth)
5. **Stochastic layer** — Noise bounds (second_moment_upper_bound, noise_term_nonneg)
6. **Method-shaped algebra** — Momentum, variance-bound, and schedule consequences

## 1.4 Formalization Statistics

| Metric | Value |
|---|---|
| Theorem declarations | 78 |
| Registered real symbols | 31 |
| Paper-facing selected results | 43 |
| Domain axioms introduced by the field | 0 |
| Total registered items checked | 109 |
| Errors | 0 |

| Metric | Value |
|---|---|
| Proof files | 1 |

The counting rule is disjoint: the 109 replayed items consist of 31 registered real symbols plus 78 theorem declarations. The 43 paper-facing results are a selected subset of the 78 declarations, not an additional category to add to the total. Declaration count is kept separate from mathematical-credit grading.

### 1.5 Notation and Semantic Convention

Every symbol in a displayed theorem denotes a real scalar in the source. Labels such as $\|\nabla f\|^2$, $\langle \nabla f, x - x^* \rangle$, $E[\|g\|^2]$, Var, and $\|x_t - x^*\|^2$ are mnemonic paper notation for those scalars; the source does not construct a normed space, gradient, expectation, variance, function, or iterate sequence. Subscripts label one-step quantities and are not quantified time indices. Consequently, the displayed formulas are paper-level substitutions into the associated scalar declarations, not literal transcriptions of richer formal objects.

| Paper notation | Source scalar or expression | Status of the bridge |
|---|---|---|
| $\|\nabla f\|^2$ | grad_sq | mnemonic rename only |
| $\langle \nabla f, x - x^* \rangle$ | grad_inner | mnemonic rename only |
| $(\mu/2)\|x - x^*\|^2$ | mu_half $*$ dist_sq | mu_half and dist_sq are independent real scalars; no norm or $\mu/2$ definition is checked |
| $(L\eta^2/2)\|\nabla f\|^2$ | L_half_eta_sq $*$ grad_sq | independent scalar factor; no checked composite definition |
| $E[\|g\|^2]$ and Var | second_moment and variance | independent real scalars related only by the displayed theorem premise |
| $\|x_t - x^*\|^2$, $\|x_{t+1} - x^*\|^2$ | dist_curr, dist_next | one-step scalar labels, not members of a sequence |

The exact checked statements are therefore the source expressions named in paper_claim_map.json. The optimization-shaped notation records an intended interpretation and does not add a formal bridge theorem.

## 2. Descent Step Bounds

This section records elementary sign and ordering facts used inside scalarized descent arguments. It does not formalize the analytic descent lemma for a smooth function.

**Theorem 2.1** (descent_step). *For scalars $\eta > 0$, $\|\nabla f\|^2 \geq 0$, and $L > 0$, if $\eta \leq L$, then $\eta \cdot \|\nabla f\|^2 \geq 0$.*

This establishes only the non-negativity of the scalar product. The source premise $\eta \leq L$ is unused for that conclusion and is not the conventional smoothness restriction $\eta \leq c/L$; this theorem supplies no learning-rate validity or progress guarantee.

**Theorem 2.2** (loss_decreases). *For any decrease d ≥ 0, if f_{t+1} = f_t - d, then f_{t+1} ≤ f_t.*

**Theorem 2.3** ( smaller_lr_smaller_step ). *For $\|\nabla f\|^2 \geq 0$ and $0 \leq \eta_2 \leq \eta$, we have $\eta_2 \cdot \|\nabla f\|^2 \leq \eta \cdot \|\nabla f\|^2$.*

This monotonicity compares two scalar products; validity of an optimization step is outside the theorem.

**Theorem 2.4** (zero_grad_stationary). *If $\|\nabla f\|^2 = 0$, then $\eta \cdot \|\nabla f\|^2 = 0$.*

In the source's mnemonic scalar notation, the premise grad_sq = 0 makes the product eta ∗ grad_sq equal to zero. No stationary-point predicate, gradient vector, norm, or update step is formalized.

**Theorem 2.5** (lr_grad_product_nonneg). *For $\eta \geq 0$ and $\|\nabla f\|^2 \geq 0$, we have $\eta \cdot \|\nabla f\|^2 \geq 0$.*

## 3. Convexity Analysis

The kernel does not define convex functions. It accepts scalar inequalities corresponding to familiar first-order convexity consequences and verifies their rearrangements.

**Theorem 3.1** (suboptimality_from_convexity). *If f_t - ⟨∇f, x - x*⟩ ≤ f*, then f_t - f* ≤ ⟨∇f, x - x*⟩.*

Given the supplied scalar premise, the conclusion is its algebraic rearrangement.

**Theorem 3.2** (nonneg_suboptimality). *If f* ≤ f_t, then 0 ≤ f_t - f*.*

**Theorem 3.3** (scaled_suboptimality). *If f_t - ⟨∇f, x - x*⟩ ≤ f* and η ≥ 0, then η(f_t - f*) ≤ η · ⟨∇f, x - x*⟩.*

**Theorem 3.4** (double_scaled_suboptimality). *If f_t - ⟨∇f, x - x*⟩ ≤ f* and η ≥ 0, then 2η(f_t - f*) ≤ 2η · ⟨∇f, x - x*⟩.*

## 4. Strong Convexity

The kernel does not define strong convexity. This section verifies consequences of supplied scalar inequalities shaped like strong-convexity bounds.

**Theorem 4.1** (strongly_convex_implies_convex). *For real scalars $m > 0$ and $d \geq 0$, if $f_t + g + md \leq f^*$, then $f_t + g \leq f^*$.*

The non-negative product can be dropped from the supplied scalar bound. Here $g$, $m$, and $d$ correspond to the neutral source scalars grad_inner, mu_half, and dist_sq. Under the global notation bridge, the sign of grad_inner does not match the usual strong-convexity inequality, so this declaration is not presented as that inequality itself.

**Theorem 4.2** (strong_convex_quadratic_growth). *If f* + (μ/2)‖x - x*‖² ≤ f_t, then (μ/2)‖x - x*‖² ≤ f_t - f*.*

The conclusion is an algebraic rearrangement of the supplied quadratic-growth premise.

**Theorem 4.3** (strong_convex_tighter_suboptimality). *If f_t - ⟨∇f, x - x*⟩ + (μ/2)‖x - x*‖² ≤ f*, then f_t - f* ≤ ⟨∇f, x - x*⟩ - (μ/2)‖x - x*‖².*

## 5. Gradient Lipschitz and Stochastic Bounds

The kernel does not define Lipschitz gradients. It verifies consequences after the relevant scalar one-step and comparison inequalities are supplied as premises.

**Theorem 5.1** (gradient_step_descent). *If $f_{t+1} \le f_t - \eta\|\nabla f\|^2 + (L\eta^2/2)\|\nabla f\|^2$ and $(L\eta^2/2)\|\nabla f\|^2 \le \eta\|\nabla f\|^2$, then $f_{t+1} \le f_t$.*

**Theorem 5.2** (stochastic_step_upper_bound). *If $f_{t+1} \le f_t - \eta\langle\nabla f, x - x^*\rangle + (L\eta^2/2)\|\nabla f\|^2$, $\eta\langle\nabla f, x - x^*\rangle \ge 0$, and $(L\eta^2/2)\|\nabla f\|^2 \le \eta\langle\nabla f, x - x^*\rangle$, then $f_{t+1} \le f_t$.*

**Theorem 5.3** (second_moment_upper_bound). *If $Var \ge 0$ and $E[\|g\|^2] = \|\nabla f\|^2 + Var$, then $\|\nabla f\|^2 \le E[\|g\|^2]$.*

**Theorem 5.4** (noise_term_nonneg). *For $\eta \ge 0$ and $\sigma^2 \ge 0$, we have $\eta^2\sigma^2 \ge 0$.*

## 6. Distance-Bound Rearrangements

The symbols in this section are real scalars labeled as squared distances. There is no formal sequence of iterates; the results are one-step rearrangements.

**Theorem 6.1** (suboptimality_from_distance). *If $\|x_{t+1} - x^*\|^2 \le \|x_t - x^*\|^2 - 2\eta(f_t - f^*) + \eta^2\sigma^2$, then $2\eta(f_t - f^*) \le \|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2 + \eta^2\sigma^2$.*

**Theorem 6.2** ( drop_final_distance ). *If $\|x_{t+1} - x^*\|^2 \ge 0$ and $2\eta(f_t-f^*) \le \|x_t-x^*\|^2 - \|x_{t+1}-x^*\|^2 + \eta^2\sigma^2$, then $2\eta(f_t-f^*) \le \|x_t-x^*\|^2 + \eta^2\sigma^2$.*

**Theorem 6.3** (contraction_preserves_bound). *For $\mu \ge 0$ and $\eta \ge 0$, if $\|x_{t+1} - x^*\|^2 \le \|x_t - x^*\|^2 - \mu\eta\|x_t - x^*\|^2 + \eta^2\sigma^2$ and $\|x_t - x^*\|^2 \ge 0$, then $\|x_{t+1} - x^*\|^2 \le \|x_t - x^*\|^2 + \eta^2\sigma^2$.*

**Theorem 6.4** (gradient_dominates_gap). *For $\mu > 0$, $f_t - f^* \ge 0$, if $2\mu(f_t - f^*) \le \|\nabla f\|^2$, then $\mu(f_t - f^*) \le \|\nabla f\|^2$.*

**Theorem 6.5** (condition_number_ge_one, legacy source identifier). *For $\mu > 0$, $L > 0$, $\mu \le L$, we have $\mu \le L$.*

This theorem directly repeats one premise. The identifier is retained only for exact source traceability; it neither defines a condition number nor proves a lower bound for one.

## 7. Momentum-Shaped Algebra

These results establish elementary sign and upper-bound facts for scalar expressions labeled as momentum terms. Here $\beta$ is a scalar discount factor constrained to [0,1]. The results do not prove acceleration or heavy-ball convergence.

**Theorem 7.1** (momentum_accumulation_nonneg). *For $0 \le \beta \le 1$, $m_{prev} \ge 0$, $\|\nabla f\| \ge 0$, we have $\beta \cdot m_{prev} + \|\nabla f\| \ge 0$.*

**Theorem 7.2** (momentum_bounded_by_sum). *For $0 \le \beta \le 1$, $m_{prev} \ge 0$, $\|\nabla f\| \ge 0$, we have $\beta \cdot m_{prev} + \|\nabla f\| \le m_{prev} + \|\nabla f\|$.*

**Theorem 7.3** (heavy_ball_step_nonneg). *For $\alpha > 0$ and $m_t > 0$, we have $\alpha \cdot m_t > 0$.*

## 8. Mini-Batch and Assumed Variance Bounds

The first two results compare the scalar expression $\sigma^2/B$ across positive batch sizes. The control-variate results assume the relevant variance ordering; they do not derive it from an estimator construction and do not justify a learning-rate change.

**Theorem 8.1** (minibatch_variance_reduction). *For $\sigma^2 > 0$ and batch size $B > 1$, we have $\sigma^2/B < \sigma^2$.*

**Theorem 8.2** ( larger_batch_less_noise ). *For $\sigma^2 > 0$, $0 < B_1 < B_2$, we have $\sigma^2/B_2 < \sigma^2/B_1$.*

**Theorem 8.3** (batch_gradient_nonneg). *For batch size $B > 0$ and batch gradient sum $\geq 0$, the average gradient is non-negative.*

**Theorem 8.4** (svrg_variance_bound). *For $0 \leq Var_{SVRG} \leq Var_{SGD}$, we have $Var_{SGD} - Var_{SVRG} \geq 0$ and $Var_{SVRG} \leq Var_{SGD}$.*

**Theorem 8.5** ( control_variate_reduces_variance ). *For $0 \leq reduction \leq Var_{orig}$, we have $Var_{orig} - reduction \geq 0$ and $Var_{orig} - reduction \leq Var_{orig}$.*

## 9. Rate-Shape Algebra

This section verifies monotonicity of reciprocal and contraction-shaped scalar expressions. Here $\rho$ is a scalar contraction parameter constrained to [0,1). The section does not connect these expressions to an SGD iterate sequence.

**Theorem 9.1** (sublinear_rate_decreases). *For $C > 0$, $0 < T_1 < T_2$, we have $C/T_2 < C/T_1$.*

The reciprocal expression C/T decreases with T; no asymptotic bound for an algorithm is established.

**Theorem 9.2** ( linear_rate_contraction ). *For $0 \leq \rho < 1$ and $gap \geq 0$, we have $\rho \cdot gap \leq gap$.*

Multiplication by a scalar in [0,1) does not increase a non-negative quantity.

**Theorem 9.3** ( contraction_iterated_shrinks ). *For $0 \leq \rho < 1$ and $gap \geq 0$, we have $\rho^2 \cdot gap \leq \rho \cdot gap$.*

A second multiplication by the same scalar does not increase the expression. Arbitrary-horizon iteration is not formalized.

## 10. Learning Rate Schedules

These theorems concern scalar formulas commonly used as learning-rate schedules; they do not analyze a training process.

**Theorem 10.1** (lr_decay_monotone). *For $\eta_0 > 0$, $0 < t_1 < t_2$, we have $\eta_0/t_2 < \eta_0/t_1$.*

The scalar formula $\eta_0/t$ decreases as the positive real scalar t increases.

**Theorem 10.2** (warmup_lr_bounded). *For $\eta_{max} > 0$, $t > 0$, $T_{warmup} > 0$, $t \leq T_{warmup}$, we have $(\eta_{max} \cdot t)/T_{warmup} \leq \eta_{max}$.*

Under the displayed real-scalar premises, the expression $(\eta_{max} \cdot t)/T_{warmup}$ is at most $\eta_{max}$.

**Theorem 10.3** (warmup_lr_nonneg). *For $\eta_{max} > 0$, $t \geq 0$, $T_{warmup} > 0$, we have $(\eta_{max} \cdot t)/T_{warmup} \geq 0$.*

# 11. Polyak Averaging and Batch Size Scaling

The same scalar monotonicity patterns also appear in formulas labeled as averaging weights and batch-scaling rules.

**Theorem 11.1** (polyak_average_convex_combination). *For $T > 1$, the expression $((T-1)/T)x_{avg} + (1/T)x_{new}$ equals itself.*

The current source theorem is a direct restatement. It does not prove that the weights are non-negative or sum to one, and it is retained to make that trust limitation explicit.

**Theorem 11.2** (averaging_weight_decreases). *For $1 < T_1 < T_2$, we have $1/T_2 < 1/T_1$.*

The scalar weight 1/T decreases with T.

**Theorem 11.3** ( linear_scaling_rule ). *For $\eta_{base} > 0$ and $k > 1$, we have $\eta_{base} < k \cdot \eta_{base}$.*

This proves only the scalar ordering $\eta_{base} < k\eta_{base}$. It does not prove that a larger learning rate is permitted by an optimization method.

**Theorem 11.4** ( sqrt_scaling_intermediate ). *For $\eta_{base} > 0$ and $1 < k_{sqrt} < k$, we have $\eta_{base} < k_{sqrt} \cdot \eta_{base} < k \cdot \eta_{base}$.*

The intermediate positive multiplier gives the stated scalar ordering; no batch-size rule is derived.

# 12. Gradient Noise and Proximal SGD

Finally, the corpus records scalar consequences of supplied bias–variance decompositions and regularization terms.

**Theorem 12.1** (biased_gradient_error_decomposition). *If $MSE = bias^2 + variance$ with $bias^2 \geq 0$ and $variance \geq 0$, then $bias^2 \leq MSE$ and $variance \leq MSE$.*

**Theorem 12.2** (unbiased_estimator_mse_equals_variance). *If $MSE = 0 + variance$, then $MSE = variance$.*

**Theorem 12.3** (proximal_step_nonneg). *For $\lambda \geq 0$ and $\|x\|^2 \geq 0$, we have $\lambda \cdot \|x\|^2 \geq 0$.*

**Theorem 12.4** (regularization_shrinks_norm). *For $\|x\| > 0$, $0 < \lambda < 1$, we have $(1 - \lambda)\|x\| < \|x\|$.*

These are scalar regularization-term facts. No proximal operator or proximal-SGD update is defined.

# 13. Discussion

## 13.1 Scope and Limitations

This formalization covers scalar real arithmetic only. It does not include:

- functions, gradients, norms, inner products, or convexity predicates;
- random variables, expectations, probability spaces, or variance definitions;
- an SGD update relation, iterate sequence, or arbitrary-horizon convergence theorem;
- derivations of momentum acceleration, SVRG variance reduction, Polyak averaging guarantees, batch-scaling validity, or proximal-SGD convergence;
- matrix/tensor adaptive methods, non-convex analysis, distributed SGD, or high-probability bounds.

### 13.2 Related Work

The scalar patterns are motivated by classical optimization analyses due to Nesterov, Bottou, Johnson and Zhang, Polyak, and Robbins and Monro [1–6]. The mathematical claims themselves are standard; this paper does not claim theorem-level originality.

Mechanized optimization already extends beyond the scope of this corpus. Li et al. formalize complexity results for gradient, subgradient, proximal-gradient, and accelerated methods in Lean 4 [7]. Vajjha et al. formalize Dvoretzky's stochastic-approximation convergence theorem in Coq [8]. Isabelle/HOL has reusable foundations for unconstrained optimization and optimality conditions [9]. Most directly, Cassie's 2026 sgd–lean artifact defines an SGDSetup over a real inner-product space, assumes a bounded-noise gradient model, proves a one-step distance inequality, and derives an averaged $O(1/\sqrt{K})$ bound in Lean 4 [10]. It therefore occupies the algorithmic and semantic layer that this corpus does not reach. This manuscript also supersedes an earlier working draft, *SGD Is Right: A Machine-Checked Proof That Stochastic Gradient Descent Converges*, built from the same 78-theorem artifact; the present scalar framing replaces, rather than independently corroborates, that draft's overstrong convergence interpretation. The present contribution is neither a competing convergence result nor a claim of superiority: it is a narrower auditable inventory of small conditional scalar implications, explicit trust boundaries, and a claim map for possible reuse or later semantic lifting.

### 13.3 Future Directions

1. **Semantic lift**: replace scalar labels with functions, inner-product spaces, gradients, and convexity predicates.
2. **Stochastic lift**: define probability spaces, conditional expectations, unbiased estimators, and variance.
3. **Iterative lift**: define SGD trajectories and prove finite-horizon and asymptotic convergence results.
4. **Lean export hardening**: reduce the generated preamble neutralization surface and replace asserted declarations with narrower imported foundations where practical.

## 14. Conclusion

The verified artifact is a corpus of conditional real-arithmetic declarations, not a formal convergence theory for SGD. Its 43 paper-facing results expose the most recognizable analytical steps; 35 additional declarations provide algebraic support and recovery variants. The exact replay checks all 78 declarations together with 31 registered real symbols. This narrower framing turns a potential overclaim into a useful trust surface: readers can see precisely which implications are checked, which assumptions are supplied, and which semantic layers remain to be formalized.

## Acknowledgments

No external acknowledgments are declared for this version.

### Declaration of Generative AI Use

*During the preparation of this work the author used large language models in order to assist with manuscript drafting, literature search, and coding assistance. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.*

---

## References


1. Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT 2010*, 177–186. https://doi.org/10.1007/978-3-7908-2604-3_16.
2. Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, 315–323.
3. Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers.
4. Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5), 1–17. https://doi.org/10.1016/0041-5553(64)90137-5.
5. Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4), 838–855. https://doi.org/10.1137/0330046.
6. Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3), 400–407. https://doi.org/10.1214/aoms/1177729586.
7. Li, C., Wang, Z., He, W., Wu, Y., Xu, S., & Wen, Z. (2024). Formalization of complexity analysis of the first-order algorithms for convex optimization. arXiv:2403.11437. https://doi.org/10.48550/arXiv.2403.11437.
8. Vajjha, K., Trager, B., Shinnar, A., & Pestun, V. (2022). Formalization of a stochastic approximation theorem. In *13th International Conference on Interactive Theorem Proving (ITP 2022)*, LIPIcs 237, Article 31, 31:1–31:18. https://doi.org/10.4230/LIPIcs.ITP.2022.31.
9. Bryant, D. (2026). Unconstrained optimization. *Archive of Formal Proofs*. https://isa-afp.org/entries/Unconstrained_Optimization.html.
10. Cassie, B. (2026). *Sgd-lean: Formal Proofs of Bounded-Noise SGD Convergence in Lean 4*. Zenodo. https://doi.org/10.5281/zenodo.20475582.


## Appendix A: Formal Verification Details

### A.1 Proof Architecture

The authoritative companion source is proofs/sgd/sgd_proof.py. A replay on 6 August 2026 checked:

- 78 theorem declarations;
- 31 registered real symbols;
- 109 total registered items, all accepted: the preceding 31 symbols plus 78 theorems;

- zero domain axioms introduced by this field.

The phrase "zero domain axioms" does not mean "unconditional": theorem-local premises encode every sign, ordering, convexity-shaped, smoothness-shaped, noise, and variance-bound condition.

The trust layers are distinct. First, verify_all () replays the declarations in the custom proof environment and checks that the submitted proof steps close the encoded real-arithmetic statements under the bootstrapped arithmetic rules and each theorem's local premises. The trusted computing base for that result includes the Python runtime and the custom expression, elaboration, tactic, kernel, and real-arithmetic bootstrap implementation. The field has no second independent checker for this replay.

Second, the current proof source has a successful source-bound Rust replay, but it does not currently have a source-bound independent Lean compilation certificate. The live seal records compiled: false, l4_pass: false, and independent_compile: false. An earlier proof-source version had historical Lean evidence, but that evidence does not certify the present source. The raw current export also collides with imported foundational declarations and therefore does not compile without the publication pipeline's preamble neutralization. A fresh source-bound publication export, Lean compilation, and transitive axiom audit are required before an independent-Lean claim can be restored.

Third, ordinary mathematical and peer review remains responsible for matters outside syntactic proof replay: whether the scalar premises faithfully model the intended optimization setting, whether the paper notation is an appropriate substitution for the source scalars, whether assumptions and references are complete, and whether the selected elementary lemmas are relevant or useful. Machine replay does not decide those semantic or scholarly questions.

## A.2 Paper-Facing Theorem Index and Semantic Grades

A full-declaration publication-credit audit classifies the 78 declarations as follows:

| A | B | C | D | E | F | Total |
|---|---|---|---|---|---|---|
| 0 | 0 | 65 | 4 | 9 | 0 | 78 |

Here Grade C denotes correct scalar or order arithmetic, Grade D a direct premise return or tautological conclusion, and Grade E a thin wrapper around an identical library theorem. Thus the strict A+B publication-result count is zero; the correct positive claim is 65 checked arithmetic implications, reported separately from four direct restatements and nine library aliases.

The grades below classify relative roles inside the 43-row paper-facing selection, not mathematical novelty, depth, or the repository's A–F publication-credit grades. **Substantive** is a corpus-relative label for a recognizable analytical implication with a nontrivial rearrangement; **derived** means a useful but immediate consequence of stronger supplied premises; **structural** means domain-neutral sign, product, identity, or rate-shape algebra; **direct-restatement** means the conclusion repeats all or part of a premise without adding mathematical content. Even the 11 corpus-relative substantive rows are short consequences of supplied scalar inequalities.

### Premise-utilization note

The declarations are true under their stated premises, but a proof-body audit found redundant premises in at least ten paper-facing rows: 2.1, 2.3, 5.2, 6.5, 7.1, 7.2, 8.4, 9.2, 10.2, and 11.1. This

is premise debt rather than logical vacuity: the conclusions remain satisfiable and checked, but not every displayed hypothesis is needed by the current proof. The publication passport records these rows explicitly so that vacuity_warnings: 0 is not misread as a complete premise-utilization certificate.

| Theorem Name | Section | Grade | Description |
|---|---|---|---|
| descent_step | §2 | derived | Product non-negativity |
| loss_decreases | §2 | substantive | One-step loss ordering |
| smaller_lr_smaller_step | §2 | derived | Learning-rate product ordering |
| zero_grad_stationary | §2 | structural | Zero-product identity |
| lr_grad_product_nonneg | §2 | structural | Product sign |
| suboptimality_from_convexity | §3 | substantive | Convexity-shaped rearrangement |
| nonneg_suboptimality | §3 | derived | Non-negative gap |
| scaled_suboptimality | §3 | derived | Scaled supplied inequality |
| double_scaled_suboptimality | §3 | derived | Double-scaled supplied inequality |
| strongly_convex_implies_convex | §4 | substantive | Drop non-negative quadratic term |
| strong_convex_quadratic_growth | §4 | substantive | Quadratic-growth rearrangement |
| strong_convex_tighter_suboptimality | §4 | substantive | Strong-convexity-shaped rearrangement |
| gradient_step_descent | §5 | substantive | Combine supplied descent bounds |
| stochastic_step_upper_bound | §5 | substantive | Combine supplied stochastic-step bounds |
| second_moment_upper_bound | §5 | substantive | Moment-decomposition consequence |
| noise_term_nonneg | §5 | structural | Noise-product sign |
| suboptimality_from_distance | §6 | substantive | Distance-bound rearrangement |
| drop_final_distance | §6 | substantive | Drop non-negative final term |
| contraction_preserves_bound | §6 | derived | Weaken supplied contraction bound |
| gradient_dominates_gap | §6 | derived | Weaken supplied domination bound |
| condition_number_ge_one | §6 | direct-restatement | Repeats $\mu \le L$ premise |
| momentum_accumulation_nonneg | §7 | structural | Scalar sum sign |
| momentum_bounded_by_sum | §7 | derived | Scalar momentum upper bound |
| heavy_ball_step_nonneg | §7 | structural | Positive product |
| minibatch_variance_reduction | §8 | derived | Positive reciprocal comparison |
| larger_batch_less_noise | §8 | derived | Reciprocal batch comparison |
| batch_gradient_nonneg | §8 | structural | Scalar average sign |
| svrg_variance_bound | §8 | derived | Consequence of assumed variance ordering |
| control_variate_reduces_variance | §8 | derived | Consequence of assumed reduction bound |
| sublinear_rate_decreases | §9 | derived | Reciprocal rate-shape comparison |
| linear_rate_contraction | §9 | structural | One-factor contraction shape |
| contraction_iterated_shrinks | §9 | derived | Two-factor contraction shape |
| lr_decay_monotone | §10 | derived | Reciprocal schedule comparison |
| warmup_lr_bounded | §10 | derived | Linear warmup bound |
| warmup_lr_nonneg | §10 | structural | Warmup expression sign |
| polyak_average_convex_combination | §11 | direct-restatement | Expression equals itself |
| averaging_weight_decreases | §11 | derived | Reciprocal-weight comparison |
| linear_scaling_rule | §11 | structural | Positive scalar multiplication |
| sqrt_scaling_intermediate | §11 | derived | Intermediate scalar ordering |
| biased_gradient_error_decomposition | §12 | substantive | Error-decomposition consequences |
| unbiased_estimator_mse_equals_variance | §12 | derived | Zero-bias substitution |
| proximal_step_nonneg | §12 | structural | Regularization-product sign |
| regularization_shrinks_norm | §12 | derived | Scalar shrinkage inequality |

Grade totals: **11 substantive, 20 derived, 10 structural, and 2 direct-restatement**, for 43 paper-facing results.

## A.3 Verification Command

From the root of the pinned public repository revision named in the companion reproduction instructions, the following command replays the authoritative field:

```
uv run python -m proofs.sgd.sgd_proof
```

Expected output: `s_g_d: verify 109/109 OK, 0 errors`