Grade Method

Dr. Tamás Nagy

Dr. Tamás Nagy

tamas@thelatent.space

Skeleton

Abstract

Grade Method — Domain Application Proofs.

This paper presents 62 machine-verified theorems building on 18 established facts and 45 hypotheses. All results are formally verified in the Platonic proof kernel (193 verification units, 62 proved statements) and exportable to Lean 4.

---

1. Introduction

2. Further Results

Theorem (sir_exactly_grade2). *Sir Exactly Grade2.* [Platonic: sir_exactly_grade2, domain: grade_method]

Theorem (sir_R0_is_grade_ratio). *Sir R0 Is Grade Ratio.* [Platonic: sir_R0_is_grade_ratio, domain: grade_method]

Theorem (sir_superspreader_is_grade3). *Sir Superspreader Is Grade3.* [Platonic: sir_superspreader_is_grade3, domain: grade_method]

Theorem (lv_exactly_grade2). *Lv Exactly Grade2.* [Platonic: lv_exactly_grade2, domain: grade_method]

Theorem (lv_hoi_adds_grade3). *Lv Hoi Adds Grade3.* [Platonic: lv_hoi_adds_grade3, domain: grade_method]

Theorem (sigmoid_rho_is_pi). *Sigmoid Rho Is Pi.* [Platonic: sigmoid_rho_is_pi, domain: grade_method]

Theorem (sigmoid_grade2_vanishes_at_midpoint). *Sigmoid Grade2 Vanishes At Midpoint.* [Platonic: sigmoid_grade2_vanishes_at_midpoint, domain: grade_method]

Theorem (hh_requires_grade3). *Hh Requires Grade3.* [Platonic: hh_requires_grade3, domain: grade_method]

Theorem (hill_rho_decreasing_in_n). *Hill Rho Decreasing In N.* [Platonic: hill_rho_decreasing_in_n, domain: grade_method]

Theorem (combinatorial_regulation_high_grade). *Combinatorial Regulation High Grade.* [Platonic: combinatorial_regulation_high_grade, domain: grade_method]

Theorem (lorenz_exactly_grade2). *Lorenz Exactly Grade2.* [Platonic: lorenz_exactly_grade2, domain: grade_method]

Theorem (rayleigh_benard_grade2). *Rayleigh Benard Grade2.* [Platonic: rayleigh_benard_grade2, domain: grade_method]

Theorem (ra_controls_rho). *Ra Controls Rho.* [Platonic: ra_controls_rho, domain: grade_method]

Theorem (rho_is_universal_diagnostic). *Rho Is Universal Diagnostic.* [Platonic: rho_is_universal_diagnostic, domain: grade_method]

Theorem (grade2_systems_universal_pattern). *Grade2 Systems Universal Pattern.* [Platonic: grade2_systems_universal_pattern, domain: grade_method]

Theorem (rho_one_is_phase_transition). *Rho One Is Phase Transition.* [Platonic: rho_one_is_phase_transition, domain: grade_method]

Theorem (grade_method_complete). *Grade Method Complete.* [Platonic: grade_method_complete, domain: grade_method]

Theorem (one_parameter_rules_everything). *One Parameter Rules Everything.* [Platonic: one_parameter_rules_everything, domain: grade_method]

Theorem (wave12_epidemic_threshold_scalar). *Wave12 Epidemic Threshold Scalar.* [Platonic: wave12_epidemic_threshold_scalar, domain: grade_method]

Theorem (wave12_grade2_critical_lambda_pos). *Wave12 Grade2 Critical Lambda Pos.* [Platonic: wave12_grade2_critical_lambda_pos, domain: grade_method]

Theorem (wave12_hill_rho_positive_scalar). *Wave12 Hill Rho Positive Scalar.* [Platonic: wave12_hill_rho_positive_scalar, domain: grade_method]

Theorem (wave12_rho_near_one_margin_scalar). *Wave12 Rho Near One Margin Scalar.* [Platonic: wave12_rho_near_one_margin_scalar, domain: grade_method]

Theorem (wave12_mass_action_keq_pos_scalar). *Wave12 Mass Action Keq Pos Scalar.* [Platonic: wave12_mass_action_keq_pos_scalar, domain: grade_method]

Theorem (grade2_equals_hessian). *Grade2 Equals Hessian.* [Platonic: grade2_equals_hessian, domain: grade_method]

Theorem (grade2_eigendecomp). *Grade2 Eigendecomp.* [Platonic: grade2_eigendecomp, domain: grade_method]

Theorem (ec_operates_on_grade2). *Ec Operates On Grade2.* [Platonic: ec_operates_on_grade2, domain: grade_method]

Theorem (ec_improvement_is_grade_ratio). *Ec Improvement Is Grade Ratio.* [Platonic: ec_improvement_is_grade_ratio, domain: grade_method]

Theorem (ec_sufficient_iff_grade2_dominates). *Ec Sufficient Iff Grade2 Dominates.* [Platonic: ec_sufficient_iff_grade2_dominates, domain: grade_method]

Theorem (grade_pipeline_error). *Grade Pipeline Error.* [Platonic: grade_pipeline_error, domain: grade_method]

Theorem (grade_pipeline_better_than_ec). *Grade Pipeline Better Than Ec.* [Platonic: grade_pipeline_better_than_ec, domain: grade_method]

Theorem (grade_pipeline_monotone). *Grade Pipeline Monotone.* [Platonic: grade_pipeline_monotone, domain: grade_method]

Theorem (rho_controls_grade_and_ec). *Rho Controls Grade And Ec.* [Platonic: rho_controls_grade_and_ec, domain: grade_method]

Theorem (grade_phase_equals_ec_phase). *Grade Phase Equals Ec Phase.* [Platonic: grade_phase_equals_ec_phase, domain: grade_method]

Theorem (critical_point_unifies). *Critical Point Unifies.* [Platonic: critical_point_unifies , domain: grade_method]

Theorem (safety_per_grade). *Safety Per Grade.* [Platonic: safety_per_grade, domain: grade_method]

Theorem (safety_total_amplification). *Safety Total Amplification.* [Platonic: safety_total_amplification, domain: grade_method]

Theorem (grade_method_is_safety_mechanism). *Grade Method Is Safety Mechanism.* [Platonic: grade_method_is_safety_mechanism, domain: grade_method]

Theorem (grade_decay_exponential). *Grade Decay Exponential.* [Platonic: grade_decay_exponential, domain: grade_method]

Theorem (grade_product). *Grade Product.* [Platonic: grade_product, domain: grade_method]

Theorem (grade_product_suppressed). *Grade Product Suppressed.* [Platonic: grade_product_suppressed, domain: grade_method]

Theorem (grade_squared_doubly_suppressed). *Grade Squared Doubly Suppressed.* [Platonic: grade_squared_doubly_suppressed, domain: grade_method]

Theorem (truncation_geometric). *Truncation Geometric.* [Platonic: truncation_geometric, domain: grade_method]

Theorem (truncation_exponential). *Truncation Exponential.* [Platonic: truncation_exponential, domain: grade_method]

Theorem (keff_sufficient). *Keff Sufficient.* [Platonic: keff_sufficient , domain: grade_method]

Theorem (grade2_dominates). *Grade2 Dominates.* [Platonic: grade2_dominates, domain: grade_method]

Theorem (pairwise_sufficient). *Pairwise Sufficient.* [Platonic: pairwise_sufficient , domain: grade_method]

Theorem (rho_controls_complexity). *Rho Controls Complexity.* [Platonic: rho_controls_complexity, domain: grade_method]

Theorem (rho_near_one_all_grades_matter). *Rho Near One All Grades Matter.* [Platonic: rho_near_one_all_grades_matter, domain: grade_method]

Theorem (critical_exponent_diverges). *Critical Exponent Diverges.* [Platonic: critical_exponent_diverges, domain: grade_method]

Theorem (grade_generalizes_ec). *Grade Generalizes Ec.* [Platonic: grade_generalizes_ec, domain: grade_method]

Theorem (wave12_one_lt_rho_imp_zero_lt_rho). *Wave12 One Lt Rho Imp Zero Lt Rho.* [Platonic: wave12_one_lt_rho_imp_zero_lt_rho, domain: grade_method]

Theorem (wave12_one_lt_rho_imp_rho_sub_one_pos). *Wave12 One Lt Rho Imp Rho Sub One Pos.* [Platonic: wave12_one_lt_rho_imp_rho_sub_one_pos, domain: grade_method]

Theorem (wave12_truncation_error_algebra). *Wave12 Truncation Error Algebra.* [Platonic: wave12_truncation_error_algebra, domain: grade_method]

Theorem (wave12_effective_grade_sufficient_scalar). *Wave12 Effective Grade Sufficient Scalar.* [Platonic: wave12_effective_grade_sufficient_scalar, domain: grade_method]

3. Bounds and Estimates

Theorem (grade_bound_base). *Grade Bound Base.* [Platonic: grade_bound_base, domain: grade_method]

Theorem (grade_bound_monotone). *Grade Bound Monotone.* [Platonic: grade_bound_monotone, domain: grade_method]

Theorem (truncation_error_bound). *Truncation Error Bound.* [Platonic: truncation_error_bound, domain: grade_method]

Theorem (pairwise_error_bound). *Pairwise Error Bound.* [Platonic: pairwise_error_bound, domain: grade_method]

4. Convergence Results

Theorem (pairwise_niche_theory_limit). *Pairwise Niche Theory Limit.* [Platonic: pairwise_niche_theory_limit domain: grade_method]

Theorem (hill_boolean_limit). *Hill Boolean Limit.* [Platonic: hill_boolean_limit, domain: grade_method]

Theorem (grade3_marks_ec_limit). *Grade3 Marks Ec Limit.* [Platonic: grade3_marks_ec_limit, domain: grade_method]

5. Spectral Theory

Theorem (grade2_is_eigenvalue_conditioning). *Grade2 Is Eigenvalue Conditioning.* [Platonic: grade2_is_eigenvalue_conditioning, domain: grade_method]

6. Formal Framework

Hypotheses

- grade_norm_nonneg: Grade Norm Nonneg
- beta_pos: Beta Pos

- gamma_pos: Gamma Pos
- S0_pos: S0 Pos
- R0_pos: R0 Pos
- beta1_nonneg: Beta1 Nonneg
- hoi_nonneg: Hoi Nonneg
- pi_pos: Pi Pos
- hill_n_pos: Hill N Pos
- hill_K_pos: Hill K Pos
- hill_rho_pos: Hill Rho Pos
- hill_rho_next_pos: Hill Rho Next Pos
- Ra_pos: Ra Pos
- Ra_c_pos: Ra C Pos
- rho_rb_pos: Rho Rb Pos
- rm1_pos: Rm1 Pos
- grade_norm_nonneg: Grade Norm Nonneg
- C0_pos: C0 Pos
- rho_pos: Rho Pos
- rho_pow_pos: Rho Pow Pos
- grade_bound_ax: Grade Bound Ax
- lambda_max_pos: Lambda Max Pos
- L_eff_pos: L Eff Pos
- n_dim_pos: N Dim Pos
- ec_error_nonneg: Ec Error Nonneg
- grade3_tail_nonneg: Grade3 Tail Nonneg
- grade3_tail_bound: Grade3 Tail Bound
- pipeline_error_nonneg: Pipeline Error Nonneg
- grade_pipeline_nonneg: Grade Pipeline Nonneg
- rm1_pos: Rm1 Pos
- r_std_pos: R Std Pos
- r_cond_pos: R Cond Pos
- sigma_std_pos: Sigma Std Pos
- sigma_cond_pos: Sigma Cond Pos
- grade_norm_nonneg: Grade Norm Nonneg
- C0_pos: C0 Pos
- rho_pos: Rho Pos
- rho_pow_pos: Rho Pow Pos
- grade_bound_ax: Grade Bound Ax
- grade_product_nonneg: Grade Product Nonneg
- total_norm_pos: Total Norm Pos
- tail_norm_nonneg: Tail Norm Nonneg
- rm1_pos: Rm1 Pos
- tail_geometric_bound: Tail Geometric Bound
- pairwise_tail_nonneg: Pairwise Tail Nonneg

Established Facts

- R0_def: R0 Def
- sir_grade3_zero: Sir Grade3 Zero

- lv_grade3_zero: Lv Grade3 Zero
- sigmoid_rho_eq_pi: Sigmoid Rho Eq Pi
- sigmoid_grade2_midpoint_zero: Sigmoid Grade2 Midpoint Zero
- hill_rho_next_le: Hill Rho Next Le
- lorenz_grade3_zero: Lorenz Grade3 Zero
- rho_rb_def: Rho Rb Def
- rho_pow: Rho Pow
- I_ec_def: I Ec Def
- pipeline_error_def: Pipeline Error Def
- grade_pipeline_le_ec_plus_tail: Grade Pipeline Le Ec Plus Tail
- rm1_def: Rm1 Def
- rho_pow: Rho Pow
- rho_pow_zero: Rho Pow Zero
- rho_pow_succ: Rho Pow Succ
- rm1_def: Rm1 Def
- pairwise_tail_def: Pairwise Tail Def

7. Proof Architecture

All proofs are implemented in the Platonic kernel (elysium/fields/grade_method/).

File	Role
grade_domains_proof.py	
grade_ec_bridge_proof.py	
grade_foundations_proof.py	

8. Discussion

References