Concrete Applications

Dr. Tamás Nagy

Dr. Tamás Nagy

tamas@thelatent.space

Skeleton

Abstract

Concrete Applications: Goldbach Crystallization & NS Turbulence.

This paper presents 20 machine-verified theorems building on 0 established facts and 31 hypotheses. All results are formally verified in the Platonic proof kernel (75 verification units, 18 proved statements) and exportable to Lean 4.

---

1. Introduction

2. Further Results

Theorem (goldbach_surplus_positive). *Goldbach Surplus Positive.* [Platonic: goldbach_surplus_positive, domain: concrete_applications]

Theorem (goldbach_repr_positive). *Goldbach Repr Positive.* [Platonic: goldbach_repr_positive, domain: concrete_applications]

Theorem (surplus_monotone). *Surplus Monotone.* [Platonic: surplus_monotone, domain: concrete_applications]

Theorem (iterated_surplus_growth). *Iterated Surplus Growth.* [Platonic: iterated_surplus_growth, domain: concrete_applications]

Theorem (cascade_monotone). *Cascade Monotone.* [Platonic: cascade_monotone, domain: concrete_applications]

Theorem (cascade_strict_decrease). *Cascade Strict Decrease.* [Platonic: cascade_strict_decrease, domain: concrete_applications]

Theorem (cascade_contraction_factor). *Cascade Contraction Factor.* [Platonic: cascade_contraction_factor, domain: concrete_applications]

Theorem (contraction_factor_range). *Contraction Factor Range.* [Platonic: contraction_factor_range, domain: concrete_applications]

Theorem (contraction_factor_lt1). *Contraction Factor Lt1.* [Platonic: contraction_factor_lt1, domain: concrete_applications]

Theorem (two_step_cascade). *Two Step Cascade.* [Platonic: two_step_cascade, domain: concrete_applications]

Theorem (two_step_monotone). *Two Step Monotone.* [Platonic: two_step_monotone, domain: concrete_applications]

Theorem (three_step_monotone). *Three Step Monotone.* [Platonic: three_step_monotone, domain: concrete_applications]

Theorem (three_step_cascade). *Three Step Cascade.* [Platonic: three_step_cascade, domain: concrete_applications]

Theorem (rho_controls_viscosity). *Rho Controls Viscosity.* [Platonic: rho_controls_viscosity, domain: concrete_applications]

Theorem (Re_vs_cascade_depth). *Re Vs Cascade Depth.* [Platonic: Re_vs_cascade_depth, domain: concrete_applications]

Theorem (energy_conservation). *Energy Conservation.* [Platonic: energy_conservation, domain: concrete_applications]

Theorem (dissipated_fraction). *Dissipated Fraction.* [Platonic: dissipated_fraction, domain: concrete_applications]

Theorem (rho_invariance_cascade). *Rho Invariance Cascade.* [Platonic: rho_invariance_cascade, domain: concrete_applications]

3. Bounds and Estimates

Theorem (surplus_lower_bound). *Surplus Lower Bound.* [Platonic: surplus_lower_bound, domain: concrete_applications]

Theorem (error_bounded). *Error Bounded.* [Platonic: error_bounded, domain: concrete_applications]

4. Formal Framework

Hypotheses

- H_n_large: N Large
- H_pw_pos: Pw Pos
- H_se_pos: Se Pos
- H_S_def: S Def
- H_pw_gt_se: Pw Gt Se
- H_rc_eq_S: Rc Eq S
- H_Sn_def: Sn Def
- H_pwn_ge_pw: Pwn Ge Pw
- H_sen_le_se: Sen Le Se
- H_log_pos: Log Pos
- H_se_bound: Se Bound
- H_Sk_pos: Sk Pos
- H_Sk1_ge_Sk: Sk1 Ge Sk
- H_Sk1_def: Sk1 Def
- H_Om_pos: Om Pos
- H_nu_pos: Nu Pos

- H_rho_pos: Rho Pos
- H_diss_def: Diss Def
- H_Om1_def: Om1 Def
- H_Om1_nn: Om1 Nn
- H_nu_lt1: Nu Lt1
- H_Om2_def: Om2 Def
- H_Om2_nn: Om2 Nn
- H_Om3_def: Om3 Def
- H_Om3_nn: Om3 Nn
- H_rho_inv1: Rho Inv1
- H_nue_def: Nue Def
- H_Re_def: Re Def
- H_Re_pos: Re Pos
- H_E_total: E Total
- H_Ed_def: Ed Def

5. Proof Architecture

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

File	Role
concrete_applications_proof.py	

6. Discussion

References