Applied Math Theorem Framework

A reusable template for stating proofs, visualizing results, and sharing findings

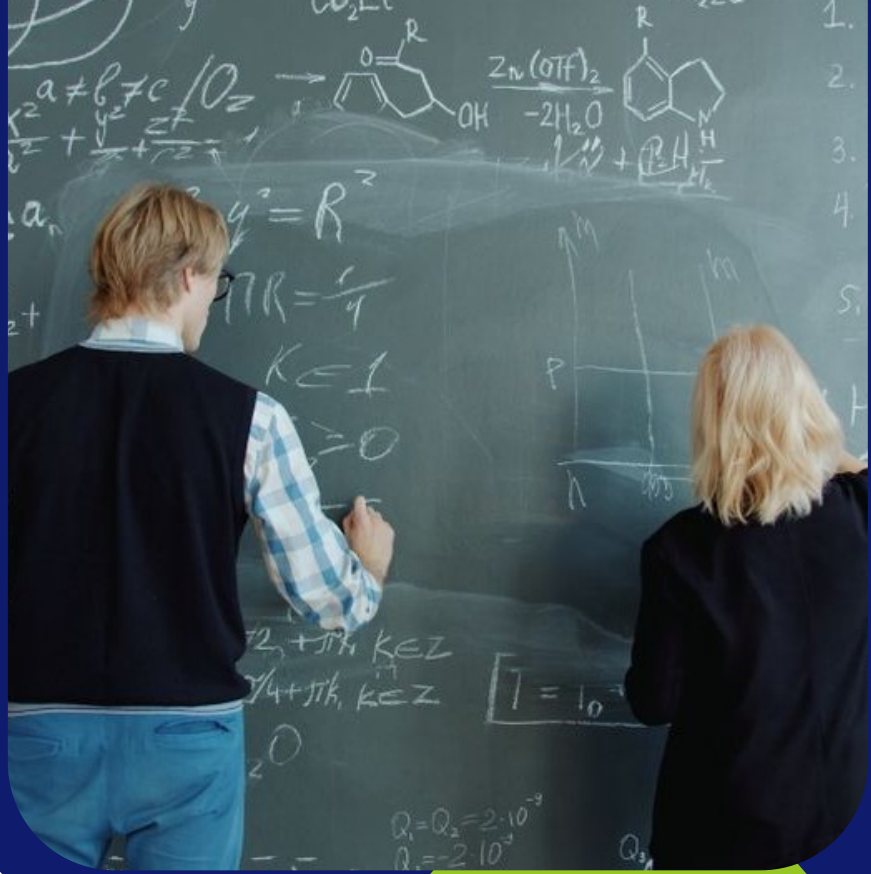


Table of Contents

01 Theorem Template for Applied Math

02 Abstract and Problem Formulation

03 Definitions, Variables, and Axioms

04 Core Theorem Statement Card

05 Proof Architecture with Lemma Chain

06 Geometric or Functional Visualization

07 Algorithmic Simulation and Numerical Results

08 Open Problems and Q&A

Theorem Template for Applied Math

Theorem Name & Field

Theorem or model name plus mathematical field (e.g., stochastic optimization, PDEs, statistical inference)

Author Attribution

Presenter name and institution placeholders for author attribution

Version Date

Date placeholder for versioning and citation context

Template Intent

Proof-ready structure with theorem, lemmas, and numerical evidence

Visual Styling

Chalkboard gray background, chalk-white text, goldenrod and cyan annotations



Abstract and Problem Formulation

Application context

Motivating application: specify the real-world system or theoretical setting and the measurable objective

Target property

Baseline question: state what is to be proven or computed, including the target quantity or property

Model assumptions

Assumptions overview: list regularity, distributional, or boundary-condition requirements at a high level

Key notation

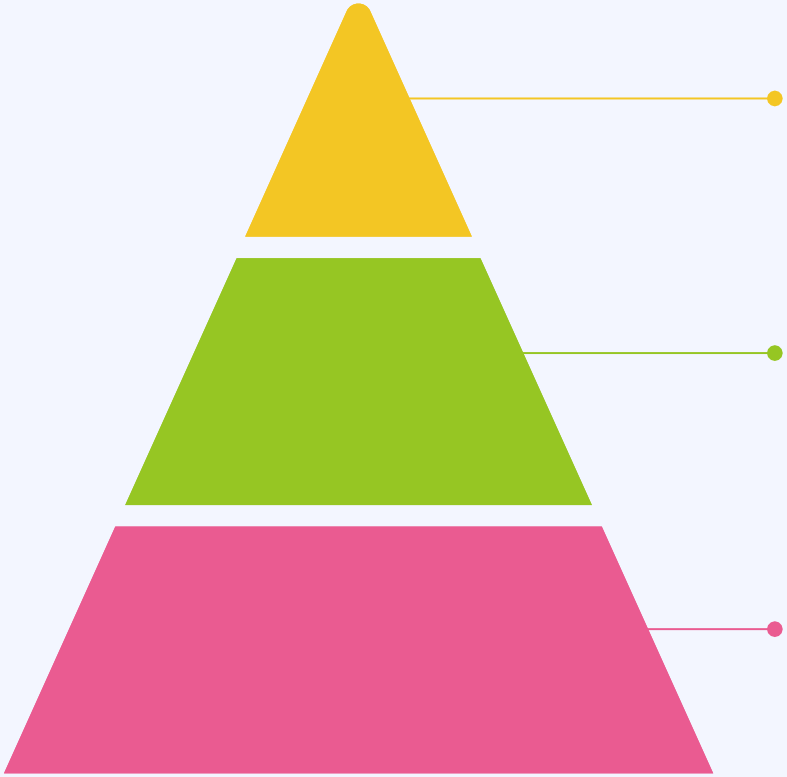
Notation preview: define the main objects (variables, operators, parameters) used throughout

Expected conclusion

Outcome preview: describe the expected theorem conclusion and what the numerical section will validate



Definitions, Variables, and Axioms



Derived quantities

Derived quantities (risk/energy functional, estimators, residuals, norms, metrics)

Governing properties

Axioms or governing properties (assumptions on operators, smoothness, identifiability)

Domain objects

Domain objects and spaces (state space, parameter set, function classes, sigma-algebras)

Core Theorem Statement Card

Theorem header card

Formal claim with explicit hypotheses and a precise conclusion in LaTeX-ready form

Emphasized conditions

Highlight regularity, boundedness, convexity, or moment assumptions as theorem constraints

Conclusion formatting

Separate main result, corollary-style refinement, and edge-condition notes if needed

Parameter dependencies

State how constants depend on problem data and which terms are universal

Proof pointer

Add a short roadmap linking theorem components to later lemmas and derivation steps



Proof Architecture with Lemma Chain

01

Lemma 1 role

Establish the key intermediate bound or structural property used downstream

02

Lemma 2 role

Prove the supporting estimate, stability claim, or decomposition result

03

Core derivation

Combine lemmas to obtain the theorem inequality or identity

04

Final verification

Conclude with the requested rate, convergence, or characterization and check all conditions

Geometric or Functional Visualization

Visualization placeholder

Choose a placeholder: coordinate plane for trajectories, manifold sketch for topology, or graph for functions

Coordinate and parameter annotations

Annotate coordinates and parameters: label axes, constraint sets, and key regions supporting the proof intuition

Critical points mapping

Mark critical points: fixed points, decision boundaries, singularities, or minimizers with consistent symbols

Hypotheses-to-conclusion link

Link to theorem: visually indicate which region corresponds to hypotheses and where the conclusion applies

LaTeX-ready notation

LaTeX compatibility: keep labels math-ready and consistent with notation from earlier slides



Algorithmic Simulation and Numerical Results

Experiment	Metric	Result summary	Complexity note
Convergence test	Error vs iterations	Observed decay rate	Empirical scaling
Robustness check	Worst-case or variance	Stability under noise	Sensitivity summary
Runtime benchmark	Wall-clock time	Breakdown by component	Asymptotic class

Open Problems and Q&A

Edge-case assumptions

Specify which assumptions may be relaxed and where counterexamples could arise

Computational implications

Discuss scalability bottlenecks, conditioning issues, or discretization artifacts

Future extensions

Propose generalizations to broader operators, distributions, or constraints

Citations and references

List formal sources supporting theorem tools and numerical baselines