Hodge Lean Proof: Axiom Completion Roadmap

What must be proved vs. what can remain assumed

Generated from `DependencyCheck.lean`

December 30, 2025

Purpose. This document lists the **current axiom dependencies** of `hodge_conjecture'` and classifies them into:

- **Must complete** (strategy-critical: likely to contain the conjecture's hard content if left as axioms),
- **Can leave (for now)** (classical pillars + interface glue), if your goal is a solid proof *modulo named classical theorems.*

Note: This is not an unconditional proof unless *all* axioms are ultimately discharged.

1 Current axiom list (mechanical)

Lean reports the following axioms for `hodge_conjecture'` (currently 38):

```
'hodge_conjecture'' depends on axioms: [
  FundamentalClassSet_isClosed, IsAlgebraicSet, IsAlgebraicSet_empty,
  IsAlgebraicSet_union, calibration_inequality, exists_volume_form_of_submodule_axiom,
  flat_limit_of_cycles_is_cycle, hard_lefschetz_inverse_form,
  harvey_lawson_fundamental_class, harvey_lawson_represents, harvey_lawson_theorem,
  instAddCommGroupDeRhamCohomologyClass, instModuleRealDeRhamCohomologyClass,
  isClosed_omegaPow_scaled, isIntegral_zero_current, isSmoothAlternating_add,
  isSmoothAlternating_neg, isSmoothAlternating_smul, isSmoothAlternating_sub,
  isSmoothAlternating_zero, lefschetz_lift_signed_cycle, limit_is_calibrated,
  microstructureSequence_are_cycles, microstructureSequence_defect_bound,
  microstructureSequence_flat_limit_exists, ofForm_smul_real, ofForm_sub,
  omega_pow_isClosed, omega_pow_represents_multiple, propext, serre_gaga,
  signed_decomposition, simpleCalibratedForm_is_smooth, smoothExtDeriv_add,
  smoothExtDeriv_smul, wirtinger_comass_bound, Classical.choice, Quot.sound]
```

2 Axioms you still need to complete (recommended)

If your goal is a *solid* proof relative to this strategy (i.e. not assuming the core bridge from rational Hodge class to algebraic cycle), these are the first axioms to target.

2.1 P0 (strategy-critical; highest priority)

Axiom	Declared at	Why it must be completed	What completion means
`signed_decomposition`	`Hodge/Kahler/SignedDecomp.lean:61`	This is where rationality is turned into a decomposition used to build algebraic cycles. If axiomatized, it can encode most of the conjecture's content.	Prove as theorem (or replace by a genuinely standard theorem known *not* to imply Hodge).
`microstructureSequence_are_cycles`	`Hodge/Kahler/Microstructure.lean:228`	Part of the microstructure pipeline; asserts the constructed approximants are genuine cycles. If axiomatized, it hides the geometric construction.	Define the construction and prove $\partial = 0$ for each approximant.
`microstructureSequence_defect_bound`	`Hodge/Kahler/Microstructure.lean:234`	Controls calibration defect; needed to pass to calibrated limits. If axiomatized, it hides the key analytic estimate.	Prove the defect estimate from concrete norms/currents.
`microstructureSequence_flat_limit_exists`	`Hodge/Kahler/Microstructure.lean:269`	Provides the convergent subsequence / limit current. If axiomatized, it assumes the compactness/extraction needed by the strategy.	Prove via a formal Federer–Fleming compactness theorem for your current model.
`harvey_lawson_fundamental_class`	`Hodge/Kahler/Main.lean:94`	Cohomology-level bridge equating the fundamental class of the HL/GAGA output to the target class. This is the exact representation step.	Prove the de Rham class identification from the definition of cycle/fundamental class.
`lefschetz_lift_signed_cycle`	`Hodge/Kahler/Main.lean:150`	Cycle-level lifting used in the Hard Lefschetz reduction ($p > n/2$). If axiomatized, it assumes compatibility of cycle classes with Lefschetz operator/hyperplane intersection.	Prove via intersection-with-hyperplane compatibility of cycle class maps.

2.2 P1 (pipeline integrity; classical GMT facts but still assumed here)

These are standard in GMT once currents/flat topology are fully defined, but are still axioms in this repo. Complete them if you want the analytic limit behavior internal to Lean.

Axiom	Declared at	Why it matters	What completion means
`limit_is_calibrated`	`Hodge/Analytic/Calibration.lean:93`	Needed to ensure the flat limit current is calibrated, so Harvey–Lawson applies.	Prove from lower semicontinuity of mass + calibration inequality in a concrete current model.
`flat_limit_of_cycles_is_cycle`	`Hodge/Classical/HarveyLawson.lean:186`	Needed to ensure the flat limit remains a cycle ($\partial = 0$), another HL hypothesis.	Prove continuity of boundary in flat norm for your integral current model.

3 Axioms you can leave (if you accept classical pillars)

If you are comfortable treating major named theorems as axioms, the following can remain assumed while you focus on removing the strategy-critical bridge axioms above.

3.1 Major classical pillars (deep but standard)

Axiom	Declared at	Reason it is reasonable to leave (for now)
`hard_lefschetz_inverse_form`	`Hodge/Classical/Lefschetz.lean:48`	Hard Lefschetz / Hodge theory infrastructure; large formalization project.
`serre_gaga`	`Hodge/Classical/GAGA.lean:100`	GAGA/Chow direction (analytic ⇒ algebraic on projective varieties); large AG formalization.
`harvey_lawson_theorem`	`Hodge/Classical/HarveyLawson.lean:166`	Harvey–Lawson structure theorem for calibrated currents; deep GMT/complex-analytic theorem.
`harvey_lawson_represents`	`Hodge/Classical/HarveyLawson.lean:170`	Companion representation statement for HL conclusion.
`omega_pow_represents_multiple`	`Hodge/Kahler/Main.lean:143`	ω^p represented by algebraic cycle (complete intersections/hyperplane sections); classical AG fact.

3.2 Interface / glue axioms (engineering layer)

These provide algebraic/smoothness/linearity properties for the abstract APIs used in the formalization. They are typically discharged only after choosing fully concrete definitions.

Axiom	Declared at	Reason it can be left
`IsAlgebraicSet`	`Hodge/Classical/GAGA.lean:33`	Interface law; not strategy-critical.
`IsAlgebraicSet_empty`	`Hodge/Classical/GAGA.lean:55`	Interface law; not strategy-critical.
`IsAlgebraicSet_union`	`Hodge/Classical/GAGA.lean:67`	Interface law; not strategy-critical.
`FundamentalClassSet_isClosed`	`Hodge/Classical/GAGA.lean:174`	Interface law; not strategy-critical.
`omega_pow_isClosed`	`Hodge/Kahler/TypeDecomposition.lean:152`	Interface law; not strategy-critical.
`isClosed_omegaPow_scaled`	`Hodge/Kahler/TypeDecomposition.lean:160`	Interface law; not strategy-critical.
`wirtinger_comass_bound`	`Hodge/Analytic/Calibration.lean:36`	Interface law; not strategy-critical.
`calibration_inequality`	`Hodge/Analytic/Calibration.lean:55`	Interface law; not strategy-critical.
`exists_volume_form_of_submodule_axiom`	`Hodge/Analytic/Grassmannian.lean:70`	Interface law; not strategy-critical.
`simpleCalibratedForm_is_smooth`	`Hodge/Analytic/Grassmannian.lean:96`	Interface law; not strategy-critical.
`isIntegral_zero_current`	`Hodge/Analytic/IntegralCurrents.lean:40`	Interface law; not strategy-critical.
`smoothExtDeriv_add`	`Hodge/Basic.lean:246`	Interface law; not strategy-critical.
`smoothExtDeriv_smul`	`Hodge/Basic.lean:252`	Interface law; not strategy-critical.
`ofForm_sub`	`Hodge/Basic.lean:1004`	Interface law; not strategy-critical.
`ofForm_smul_real`	`Hodge/Basic.lean:1021`	Interface law; not strategy-critical.
`isSmoothAlternating_zero`	`Hodge/Basic.lean:66`	Interface law; not strategy-critical.
`isSmoothAlternating_add`	`Hodge/Basic.lean:69`	Interface law; not strategy-critical.
`isSmoothAlternating_neg`	`Hodge/Basic.lean:72`	Interface law; not strategy-critical.
`isSmoothAlternating_sub`	`Hodge/Basic.lean:78`	Interface law; not strategy-critical.
`isSmoothAlternating_smul`	`Hodge/Basic.lean:75`	Interface law; not strategy-critical.

Axiom	Declared at	Reason it can be left
`instAddCommGroupDeRhamCohomologyClass`	`Hodge/Basic.lean:605`	Algebraic structure axiom.
`instModuleRealDeRhamCohomologyClass`	`Hodge/Basic.lean:621`	Algebraic structure axiom.

3.3 Lean foundations

Axiom	Declared at	Reason it can be left
`Classical.choice`	(Lean core)	Standard classical logic; removing it is a separate (constructive) project.
`propext`	(Lean core)	Standard extensionality principle in Lean/Mathlib classical developments.
`Quot.sound`	(Lean core)	Core quotient principle used by Lean; not a mathematical assumption.

4 Recommended completion order

1. **Discharge P0 axioms** (strategy-critical) so the proof does not assume the core bridge.
2. **Discharge P1 axioms** if you want the analytic limit behavior internal to Lean.
3. **Optionally**, begin a long-term project to formalize the classical pillars (Hard Lefschetz, GAGA, Harvey–Lawson).