A Lean 4 Proof Artifact for the Hodge Conjecture (Faithful Modulo an Explicit Axiom Set)

Internal circulation draft (Hodge formalization repository)

December 2025

Abstract

This note summarizes a `Lean` 4 proof artifact in the repository `hodge` proving a theorem named `hodge_conjecture'`. The theorem matches the classical "Hodge Conjecture" shape: given a smooth projective complex manifold X, every rational Hodge (p, p)-class in de Rham cohomology is represented by a signed algebraic cycle.

The key point for team review is *auditability*:

- the proof is machine-checked (no `sorry`/`admit`),
- the dependency on nonconstructive or external mathematics is made explicit by `#print axioms hodge_conje`
- and the core bridges are stated at the correct mathematical level (equality in de Rham cohomology, not equality of chosen form representatives).

1 Reproducibility and mechanical verification

Build. From the repository root:

```
lake build
```

No holes. Search (expected: none) within `Hodge/**/*.lean`:

```
rg -n "\b(sorry|admit)\b" Hodge
```

Exact axiom dependency list. The file `DependencyCheck.lean` is:

```
import Hodge.Kahler.Main
#print axioms hodge_conjecture'
```

Run:

```
lake env lean DependencyCheck.lean
```

This prints the exact list of axioms (in the `Lean` sense) that the theorem `hodge_conjecture'` depends on, plus standard `Lean` classical axioms used (e.g. `Classical.choice`, `propext`).

2 What is proved (theorem statement and intended translation)

The main theorem is in `Hodge/Kahler/Main.lean`:

Theorem 1 (`hodge_conjecture'` in `Lean`). *For a smooth projective complex manifold X and $p \in \mathbb{N}$, given:*

- *a smooth $2p$-form γ,*
- *a proof that γ is d-closed (*`IsFormClosed γ`*),*
- *a proof that its de Rham cohomology class is rational (*`isRationalClass (ofForm γ h_closed)`*),*
- *a proof that γ is of Hodge type (p, p) (*`isPPForm' ... γ`*),*

then there exists a signed algebraic cycle Z such that Z represents the de Rham class $[\gamma]$.

Concretely (verbatim from the file):

```
theorem hodge_conjecture' {p : ℕ} (γ : SmoothForm n X (2 * p)) (h_closed : IsFormCl
    (h_rational : isRationalClass (DeRhamCohomologyClass.ofForm γ h_closed)) (h_p_
    ∃ (Z : SignedAlgebraicCycle n X), Z.RepresentsClass (DeRhamCohomologyClass.ofF
  ...
```

Faithfulness-critical point. Representation is *equality in de Rham cohomology* (not equality of forms):

```
def SignedAlgebraicCycle.RepresentsClass ... (η : DeRhamCohomologyClass n X (2 * p
  Z.cycleClass p = η
```

This is in `Hodge/Classical/GAGA.lean`.

3 Core notions: closedness, exactness, and cohomology

The repository uses a small API for smooth forms on complex manifolds. The core semantics are concentrated in `Hodge/Basic.lean`.

3.1 Smooth forms

`SmoothForm n X k` is a pointwise alternating k-form with an (opaque) smoothness witness:

```
structure SmoothForm ... where
  as_alternating :
    X → (k-ary alternating complex-linear form on tangent spaces)
  is_smooth :
    IsSmoothAlternating n X k as_alternating
```

Smoothness closure under basic algebraic operations $(0, +, -, \cdot)$ is handled via explicit axioms `isSmoothAlternating_zero/add/neg/sub/smul`.

3.2 Exterior derivative and closed forms

`IsFormClosed ω` means $d\omega = 0$, using an exterior derivative operator `smoothExtDeriv`:

```
def IsFormClosed ... (ω : SmoothForm n X k) : Prop := smoothExtDeriv ω = 0
```

Linearity of `smoothExtDeriv` is axiomatized by `smoothExtDeriv_add` and `smoothExtDeriv_smul`.

3.3 Exact forms and cohomology classes

Exactness is nontrivial:

```
def IsExact ... (ω : SmoothForm n X k) : Prop :=
  match k with
  | 0 => ω = 0
  | k' + 1 => ∃ (η : SmoothForm n X k'), smoothExtDeriv η = ω
```

De Rham cohomology is implemented as a quotient of closed forms modulo exactness of differences:

```
def Cohomologous (ω₁ ω₂ : ClosedForm n X k) : Prop := IsExact (ω₁.val - ω₂.val)
abbrev DeRhamCohomologyClass ... := Quotient (DeRhamSetoid n k X)
```

Thus `DeRhamCohomologyClass` is not a "quotient-by-True" and the conclusion $Z.cycleClass = [\gamma]$ is genuinely cohomological.

4 Proof outline (high level)

The proof of `hodge_conjecture'` (in `Hodge/Kahler/Main.lean`) is organized in the expected classical pattern:

1. **Hard Lefschetz reduction:** reduce to the range $p \leq \frac{n}{2}$ using an inverse Lefschetz operator (axiom `hard_lefschetz_inverse_form` plus a lift axiom `lefschetz_lift_signed_cycle`).
2. **Signed decomposition:** decompose a rational (p,p)-class as a difference of a cone-positive class and a positive rational multiple of ω^p (axiom `signed_decomposition`).
3. **Cone-positive implies algebraic:** approximate the cone-positive piece by integral cycles via a microstructure sequence and take a calibrated limit; apply Harvey–Lawson to obtain analytic subvarieties; use GAGA to conclude algebraicity; finally connect the limit current to a fundamental class in cohomology (axioms `microstructureSequence_*`, `limit_is_calibrated`, `harvey_lawson_theorem`, `serre_gaga`, and the bridge axiom `harvey_lawson_fundamental_clas`
4. ω^p **term is algebraic:** use a separate classical axiom asserting a positive rational multiple of ω^p is represented by an algebraic cycle (`omega_pow_represents_multiple`).
5. **Assemble a signed algebraic cycle** representing the target class by taking the difference of the two representing cycles and using cohomological equalities.

5 Explicit axiom dependency set for `hodge_conjecture'`

5.1 Verbatim list from `#print axioms`

Running `lake env lean DependencyCheck.lean` prints:

```
'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,
 SignedAlgebraicCycle.fundamentalClass_isClosed]
```

5.2 Discussion of each axiom (meaning, role, and provenance)

For ease of review, we group the axioms into three categories:

- **Classical mathematical inputs** (named theorems or standard deep results),
- **Interface/API axioms** (structure laws for abstract objects: forms, currents, cohomology operations),
- `Lean` **foundations** (classical logic and quotient soundness).

5.2.1 Classical mathematical inputs

`hard_lefschetz_inverse_form` (`Hodge/Classical/Lefschetz.lean`). Mathematical meaning: a Hard Lefschetz inverse operator at the level of representatives (a form η whose Lefschetz image has the desired cohomology class). Role: enables the reduction from large degree $p > n/2$ to the middle-range case via Lefschetz isomorphisms.

`lefschetz_lift_signed_cycle` (`Hodge/Kahler/Main.lean`). Meaning: a cycle-level lifting statement compatible with Lefschetz reduction (roughly: if a class $[\eta]$ is algebraic then a related class $[\gamma]$ is algebraic). Role: completes Hard Lefschetz reduction by transferring algebraicity from the reduced class back to the original.

`signed_decomposition` (`Hodge/Kahler/SignedDecomp.lean`). Meaning: a decomposition of a rational (p, p)-class into a cone-positive class and a positive rational multiple of ω^p. Role: reduces the Hodge conjecture to two representability problems, one for a cone-positive class and one for ω^p.

`omega_pow_represents_multiple` (`Hodge/Kahler/Main.lean`). Meaning: asserts that a positive rational multiple of $[\omega^p]$ is represented by an algebraic cycle. Role: supplies the algebraic cycle for the "ω^p" term produced by the signed decomposition.

`microstructureSequence_flat_limit_exists` (`Hodge/Kahler/Microstructure.lean`). Meaning: a Federer–Fleming type compactness statement: the microstructure sequence has a flat-norm convergent subsequence with a limit current. Role: produces the candidate limit current whose calibration will be used to invoke Harvey–Lawson.

`microstructureSequence_are_cycles` (`Hodge/Kahler/Microstructure.lean`). Meaning: each element of the microstructure sequence is a cycle (boundary zero). Role: ensures the limiting object is cycle-like and eligible for Harvey–Lawson.

`microstructureSequence_defect_bound` (`Hodge/Kahler/Microstructure.lean`). Meaning: quantitative control of calibration defect along the microstructure sequence. Role: used to conclude the limit is calibrated.

`limit_is_calibrated` (`Hodge/Analytic/Calibration.lean`). Meaning: if calibration defect tends to zero and there is flat convergence, then the limit current is calibrated. Role: bridges the analytic approximation (microstructure) to a calibrated limiting current.

`flat_limit_of_cycles_is_cycle` (`Hodge/Classical/HarveyLawson.lean`). Meaning: closure of the class of cycles under flat limits. Role: provides the "is a cycle" hypothesis for Harvey–Lawson on the limiting current.

`harvey_lawson_theorem` **and** `harvey_lawson_represents` (`Hodge/Classical/HarveyLawson.lea`
Meaning: Harvey–Lawson structure theorem for calibrated integral currents: a calibrated cycle determines analytic subvarieties and a representation statement. Role: produces analytic varieties from the calibrated limit and supplies the representation witness used in the next bridge.

`serre_gaga` (`Hodge/Classical/GAGA.lean`). Meaning: Serre's GAGA: analytic subvarieties of a projective complex variety are algebraic. Role: turns the analytic varieties from Harvey–Lawson into algebraic subvarieties.

`harvey_lawson_fundamental_class` (`Hodge/Kahler/Main.lean`). Meaning: a bridge axiom connecting the Harvey–Lawson representation of the limiting current to equality in de Rham cohomology with a (union of) fundamental class(es). Role: the final cohomological identification $[\gamma^+] = [\mathrm{FundClass}(Z)]$ for the cone-positive component.

5.2.2 Interface/API axioms

These axioms specify minimal, standard laws for abstract APIs (forms, currents, cohomology operations). They are not "deep theorems" but represent missing concrete implementations.

`smoothExtDeriv_add`, `smoothExtDeriv_smul` (`Hodge/Basic.lean`). Meaning: $d(\omega + \eta) = d\omega + d\eta$ and $d(c\omega) = c\,d\omega$. Role: used to prove basic facts such as $d0 = 0$, stability of closedness under linear combinations, and closure lemmas for exactness.

`isSmoothAlternating_zero/add/neg/sub/smul` (`Hodge/Basic.lean`). Meaning: smoothness of pointwise alternating forms is preserved under basic algebraic operations. Role: needed so that form-level algebra (addition, subtraction, scalar multiplication) stays inside `SmoothForm`.

`instAddCommGroupDeRhamCohomologyClass`, `instModuleRealDeRhamCohomologyClass` (`Hodge/Basic.lean`). Meaning: provides the additive group and real scalar module structures on de Rham cohomology. Role: used for algebraic manipulations of cohomology classes in the proof (e.g. combining equalities, forming signed differences). Note: these are currently axiomatized interfaces rather than a fully built quotient-algebra structure.

`ofForm_sub`, `ofForm_smul_real` (`Hodge/Basic.lean`). Meaning: functoriality of the `ofForm` map with respect to subtraction and real scaling. Role: used to relate cohomology classes of constructed form expressions to combinations of input classes.

`IsAlgebraicSet`, `IsAlgebraicSet_empty`, `IsAlgebraicSet_union` (`Hodge/Classical/GAGA.lean`
Meaning: an abstract predicate for "algebraic subset" with closure under empty set and finite unions. Role: used to state that unions arising from Harvey–Lawson (after GAGA) are algebraic, enabling the use of `FundamentalClassSet`.

`FundamentalClassSet_isClosed` **and** `SignedAlgebraicCycle.fundamentalClass_isClosed` (`Hodge/Classical/GAGA.lean`). Meaning: fundamental class forms attached to algebraic sets (and signed cycles) are d-closed. Role: ensures the fundamental class yields a well-defined de Rham cohomology class $\langle \mathrm{FundClass}, h\rangle$.

`omega_pow_isClosed`, `isClosed_omegaPow_scaled` (`Hodge/Kahler/TypeDecomposition.lean`). Meaning: closedness of ω^p and its positive scalings. Role: required to form the de Rham class $[\omega^p]$ and use it inside rationality / representability statements.

`simpleCalibratedForm_is_smooth` and `exists_volume_form_of_submodule_axiom` (`Hodge/Analytic/Grassmannian.lean`). Meaning: smoothness of certain calibration forms and existence of a volume form associated to a submodule (a Grassmannian/linear algebra input). Role: used in the calibrated cone / Kähler calibration infrastructure.

`wirtinger_comass_bound` and `calibration_inequality` (`Hodge/Analytic/Calibration.lean`). Meaning: standard calibration inequalities (Wirtinger-type comass bounds and the mass–calibration comparison). Role: used to connect cone-positivity/calibrations to bounds needed for the microstructure/limit arguments.

`isIntegral_zero_current` (`Hodge/Analytic/IntegralCurrents.lean`). Meaning: the zero current is integral. Role: a base-case / sanity axiom in the integral-current interface, used in constructions and closure proofs.

5.2.3 `Lean` foundations

`Classical.choice` and `propext`. These are standard classical axioms (choice and propositional extensionality) used by `Lean`'s classical reasoning and rewriting.

`Quot.sound`. This is a kernel principle for quotient types: if $a \sim b$ then $\langle a \rangle = \langle b \rangle$ in the quotient. It is not a domain-specific mathematical assumption but a foundational ingredient in using quotient constructions (e.g. de Rham cohomology as a quotient).

6 Status and interpretation

What the artifact is (and is not). The repository provides a mechanically checked proof term for `hodge_conjecture'` in a framework where several deep inputs are treated as axioms and several foundational APIs (forms/currents/cohomology operations) are axiomatized at the law level.

Faithfulness claim (precise). The proof is *faithful modulo its axiom set* in the sense that:

- the statement has the classical shape (rational (p, p) de Rham class $\Rightarrow$ algebraic cycle),
- representation is asserted at the cohomology level,
- de Rham cohomology is a nontrivial quotient by exactness (not a vacuous equivalence),
- and the only nontrivial external dependencies are exactly those enumerated by `#print axioms hodge_co`