Formalizing Schwarzschild Curvature in Lean 4: A Machine-Checked Coordinate Pipeline and a Methodology for Tracking Axiomatic Dependencies

Paul Chun–Kit Lee
dr.paul.c.lee@gmail.com
New York University, NY

February 2026

Abstract

We present a fully machine-checked verification of the standard textbook *coordinate* curvature pipeline for Schwarzschild spacetime in Lean 4, comprising roughly 16,000 lines of code. The formalization verifies the pipeline

$$g_{\mu\nu} \;\to\; \Gamma^{\rho}{}_{\mu\nu} \;\to\; R^{\rho}{}_{\sigma\mu\nu} \;\to\; R_{\mu\nu} \;\to\; K$$

in the exterior region ($r > 2M$) away from the coordinate axis ($0 < \theta < \pi$), culminating in the vacuum condition $R_{\mu\nu} = 0$ and the Kretschmann invariant $K = 48M^2/r^6$.

The underlying mathematics is classical and well known; the novelty is methodological. The development exposes a large "implicit-to-explicit" gap between informal tensor calculus and machine-checked proofs, and it provides a concrete, reproducible benchmark for coordinate-based General Relativity in a modern proof assistant. Related mechanizations of relativity often focus on axiomatic or first-order treatments (typically of special relativity) rather than end-to-end coordinate curvature computation; we briefly situate our scope in Section 1.2.

We also introduce *Axiom Calibration* (AxCal), an organizational methodology for tracking which foundational "portals" (choice, compactness, classical logic) are invoked by different proof routes in physics. For the Schwarzschild vacuum verification (G1), the mathematics is finite symbolic computation; nevertheless, the formal development depends on classical infrastructure from `mathlib`. We therefore emphasize *structural certification* (absence of high-level portals in the proof route) and outline how one could strengthen this into a mechanical axiom audit inside Lean (Section 3).

Artifact status (as of this draft): no `sorry` placeholders in the GR development; GR target build is ∼4.2s (incremental warm cache, Feb 4 2026) and ∼2m10s for the first warm build after a full dependency build; clean full-repo build is ∼1h12m on the author machine (details in Section 4.2).

Scope and Honest Claims

This paper reports on engineering *and* methodology*, not new mathematics or physics. The Schwarzschild calculations we verify are standard textbook material [6, 3, 2]. The verification that these calculations are "Height 0" (constructive at the proof-route level) is unsurprising—they are finite symbolic computations.*

The value lies in:

- *Demonstrating that formal GR is feasible in Lean 4*

- *Documenting the obstacles that made it difficult*
- *Providing infrastructure for future, more ambitious formalizations*
- *Introducing* AxCal *as organizational methodology (not foundational breakthrough)*

The more interesting GR results (G2–G5) are analyzed at the paper level but not formalized. *Their axiomatic profiles are asserted based on standard proof analysis, not machine-verified.*

Contents

1 Introduction **4**
- 1.1 What This Paper Is (And Is Not) . . . 4
- 1.2 Related Work and Scope . . . 4
- 1.3 Motivation: Searching for Hidden Axioms . . . 5
- 1.4 Why Bother Formalizing the Easy Case? . . . 5
- 1.5 Paper Contributions . . . 6

2 Why "Trivial" Mathematics Is Hard to Formalize **6**
- 2.1 The Index Machinery Problem . . . 6
- 2.2 The Side Condition Problem . . . 7
- 2.3 The Tactic Fragility Problem . . . 7
- 2.4 The Library Mismatch Problem . . . 7
- 2.5 The Differentiability Overhead . . . 8
- 2.6 The Algebraic Normalization Problem . . . 8
- 2.7 Summary: The Formalization Gap . . . 8

3 Axiom Calibration: An Organizational Methodology **9**
- 3.1 Purpose and Scope . . . 9
- 3.2 The Four Portals . . . 9
- 3.3 Height Profiles . . . 9
- 3.4 From Organizational Profiles to Mechanical Audits . . . 10
- 3.5 The Five GR Targets . . . 10
- 3.6 What AxCal Is Not . . . 11

4 The Lean 4 Formalization **11**
- 4.1 Repository Structure . . . 11
- 4.2 Artifact and Reproducibility . . . 11
- 4.3 Key Definitions . . . 12
- 4.4 Main Theorems . . . 12
- 4.5 Technique Highlights from the Code . . . 12
- 4.6 Proof Sketch and File Roles . . . 13
- 4.7 What Is Verified (and What Is Not) . . . 16
- 4.8 Build Status . . . 16

5 Multi-AI Agent Collaboration **16**
5.1 Agent Roles 16
5.2 Trust Boundary: AI as Proposer, Lean as Verifier 17
5.3 Human Role 17
5.4 Lessons Learned 17

6 Limitations and Future Work **18**
6.1 What We Did Not Prove 18
6.2 Future Directions 18

7 Conclusion **18**

1 Introduction

1.1 What This Paper Is (And Is Not)

This paper reports a fully machine-checked verification of Schwarzschild spacetime curvature calculations in Lean 4, following the standard coordinate pipeline. We verify the standard textbook pipeline:

$$\text{Schwarzschild metric} \to \text{Christoffel symbols} \to \text{Riemann tensor} \to \text{Ricci tensor} \to \text{Kretschmann scalar}$$

in Lean 4, with approximately 16,000 lines of code that builds without errors or `sorry` placeholders on the critical path.

What this paper is:

- A **technique paper** demonstrating feasibility of formal GR verification
- A **case study** in multi-AI agent collaboration for theorem proving
- An **organizational framework** (AxCal) for tracking axiomatic dependencies in physics
- **Documentation** of why formalizing "trivial" mathematics is hard

What this paper is not:

- New mathematics (the Schwarzschild calculations are 100+ years old)
- New physics (we verify known results, not discover new ones)
- A foundational breakthrough (the AxCal framework applies known meta-mathematics)
- Complete GR formalization (we handle only one solution in one coordinate patch)

1.2 Related Work and Scope

There is an existing literature on mechanizing relativity in proof assistants, but much of it targets *axiomatic* or *first-order* formulations (often of special relativity) rather than explicit coordinate curvature computation. For example, Schutz-style axioms for Minkowski spacetime have been mechanized in Isabelle/HOL,[1] and first-order relativity theories have been verified in Isabelle/HOL.[2] These efforts illuminate the logical structure of relativity and make axioms explicit, but they are intentionally far from the "compute Γ, then R, then contract" workflow familiar from GR textbooks.

The scope of this paper is complementary: we focus on the *textbook coordinate pipeline* for a single classical solution, in a single chart, with all side conditions made explicit. Our claim is not that we are the first to formalize "relativity," but that end-to-end, machine-checked coordinate curvature computations remain rare, and that they expose a distinct set of engineering bottlenecks (index expansion, side-condition threading, and controlled simplification) that are largely orthogonal to the axiomatic-first-order line of work.

[1] https://arxiv.org/abs/2108.10868
[2] https://doi.org/10.1007/s10817-013-9292-7

Author context. The author is a practicing physician (interventional cardiology) with interests in formal methods and mathematical physics, but no professional training in either field. This project was undertaken as an exercise in learning Lean 4 and exploring the feasibility of formal verification in physics. Readers should evaluate the artifact on its technical merits; domain experts may find opportunities for improvement that a non-specialist would miss.

1.3 Motivation: Searching for Hidden Axioms

The original motivation for this project was foundational: *What axioms does General Relativity actually require?*

Standard GR textbooks use the Axiom of Choice (via Zorn's lemma), compactness arguments (Ascoli-Arzelà), and proof by contradiction without flagging these as philosophically significant. For a constructive mathematician or a reverse mathematician, these hidden dependencies matter.

We sought to make them explicit by:

1. Formalizing GR results in a proof assistant
2. Tracking which axiom "portals" each proof crosses
3. Assigning "height" profiles measuring axiomatic strength

The irony. We formalized the *easiest* case first: the Schwarzschild vacuum check (G1). This calculation has **no hidden axioms**—it is finite symbolic computation, obviously constructive. The "discovery" that G1 is Height 0 is tautological.

The interesting cases—where hidden axioms actually appear—are:

- **G2 (Cauchy problem/MGHD):** Zorn's lemma for maximality
- **G3 (Singularity theorems):** Compactness + proof by contradiction
- **G4 (Maximal extensions):** Zorn's lemma
- **G5 (Computable evolution):** Pour-El–Richards [4] shows this *fails*

These are analyzed at the paper level but **not formalized**. Formalizing them would be dramatically harder and remains future work.

1.4 Why Bother Formalizing the Easy Case?

If G1 has no hidden axioms, why spend six months formalizing it?

1. **Proof of concept:** Before attempting G2–G5, we needed to know that formal GR is feasible at all. Schwarzschild establishes the infrastructure.
2. **The gap is informative:** The difficulty of formalizing G1 reveals how much implicit reasoning standard mathematics employs. This gap is itself a finding.
3. **Infrastructure for future work:** The tensor calculus machinery (index types, summation helpers, differentiability lemmas) built for G1 transfers to other spacetimes.
4. **Benchmark artifact:** We are not aware of a prior end-to-end, machine-checked *coordinate* curvature calculation for Schwarzschild (metric → Christoffel → Riemann → Ricci → Kretschmann) in an interactive theorem prover; if such an artifact exists, we would be happy to cite it. Either way, the present development serves as a concrete benchmark for "proof-producing" coordinate GR.

1.5 Paper Contributions

1. **Artifact:** Lean 4 formalization of Schwarzschild curvature (Christoffel → Riemann → Ricci → Kretschmann). Zero errors, zero sorries in the GR folder. 16,000 lines.
2. **Difficulty Analysis:** Documentation of why "trivial" GR calculations require 16,000 lines and 6 months. The gap between informal and formal mathematics is larger than expected.
3. **AxCal Methodology:** An organizational framework for tracking axiom usage in physics proofs, with four "portals" (Zorn, limit-curve, serial-chain, reductio) and three-axis height profiles.
4. **Multi-AI Workflow:** Case study of Claude Code (Opus 4.5) + GPT-5.2 + Gemini 2.5 Pro Deep Think + GPT-5.2 Codex collaboration, documenting how different AI strengths complement each other.
5. **Roadmap:** Analysis of G2–G5 at the paper level, providing targets for future formalization.

2 Why "Trivial" Mathematics Is Hard to Formalize

The Schwarzschild vacuum check is a textbook calculation that any GR graduate student can do by hand in an afternoon. Yet formalizing it required:

- 16,000 lines of Lean 4 code
- roughly 6 months of development
- 4 collaborating AI agents
- Numerous false starts and refactorings

This section documents why.

2.1 The Index Machinery Problem

On paper: Write $\Gamma^\alpha_{\mu\nu}$ and everyone understands.

In Lean: We must define:

- An `Idx` inductive type with constructors `t`, `r`, θ, φ
- A `sumIdx` function implementing $\sum_\mu$ over this finite type
- Expansion lemmas showing `sumIdx` $f = f(t) + f(r) + f(\theta) + f(\varphi)$
- Normalization lemmas for rearranging sums

The Kretschmann scalar involves $R_{abcd}R^{abcd}$—a sum over $4^4 = 256$ index combinations. Managing this requires:

- `sumIdx2_expand` for double sums
- `sixBlock` decomposition exploiting Riemann symmetries
- Weight normalization lemmas (`weight_xyxy`, `weight_xyyx`, etc.)

None of this exists in `mathlib`. We built it from scratch.

2.2 The Side Condition Problem

On paper: "Clearly $r \neq 2M$ and $\sin\theta \neq 0$."

In Lean: Every division requires a proof. The expression $f(r)^{-1}$ where $f(r) = 1-2M/r$ demands:

```
hf : f M r                                    neq 0
```

In practice we package the exterior-domain constraints in a dedicated predicate so that common nonvanishing facts (e.g. $r \neq 0$ and $f(M,r) \neq 0$) can be recovered uniformly and reused by `field_simp` and derivative lemmas:

```
structure Exterior (M r             theta :             mathbbR) : Prop where
  hM : 0 < M
  hr_ex : 2 * M < r
```

The remaining coordinate restriction $0 < \theta < \pi$ is tracked separately (it is needed only to deduce $\sin\theta \neq 0$ so that $g^{\varphi\varphi}$ is well-defined in this chart). Bundling and reusing these hypotheses is one of the main "hidden costs" of formalization: every division, derivative rule, and simplification step demands an explicit proof obligation that is usually implicit on paper.

2.3 The Tactic Fragility Problem

Early proof attempts used aggressive automation:

```
simp [Riemann, Christoffel, g, gInv, f, ...]
```

This caused:

- **Timeouts:** Expanding all definitions creates expressions with thousands of terms
- **Simp loops:** Rewriting rules can cycle indefinitely
- **Memory exhaustion:** Large expressions overwhelm the elaborator

The working approach uses **targeted lemmas**:

```
rw [Riemann_trtr]  -- Use pre-proven component value
field_simp [hr0, hf0]
ring_nf
```

This is more verbose but reliable. The session notes document specific instances:

> "Early in `Kretschmann_six_blocks` Step 3, `simp`/`simp_rw` loops or timeouts occurred due to `RiemannUp` expansion and AC rewriting. The final working approach avoided generic simp in favor of targeted lemmas plus `ring_nf`."

2.4 The Library Mismatch Problem

`mathlib` has abstract differential geometry: smooth manifolds, tangent bundles, connections. But it does not have:

- Explicit Christoffel symbol computation for specific metrics
- Coordinate-based Riemann tensor calculation
- Infrastructure for diagonal metrics with symbolic entries

The abstract machinery doesn't compute. We needed concrete coordinate calculations, which required building a parallel infrastructure.

2.5 The Differentiability Overhead

Every derivative in Lean requires proving the function is differentiable at the point. For the simple function $f(r) = 1 - 2M/r$, we need:

```
lemma f_differentiableAt (M r :           mathbbR) (hr : r           neq 0) :
    DifferentiableAt                       mathbbR (f M) r := ...

lemma f_hasDerivAt (M r :             mathbbR) (hr : r             neq 0) :
    HasDerivAt (f M) (2*M/r^2) r := ...

lemma contDiffAt_f (M r :             mathbbR) (hr : r             neq 0) :
    ContDiffAt                         mathbbR 2 (f M) r := ...
```

The C^2 smoothness lemma was a critical bottleneck—all six Riemann component proofs were stuck until it was added.

2.6 The Algebraic Normalization Problem

The Kretschmann proof requires showing that 256 terms (most zero) collapse to six blocks. This involves:

- Proving off-diagonal Riemann components vanish (60+ cases)
- Normalizing weight factors: $g^{aa}g^{bb}g^{aa}g^{bb} = (g^{aa})^2(g^{bb})^2$
- Handling Riemann symmetries: $R_{abcd} = -R_{bacd} = -R_{abdc} = R_{cdab}$

We added lemmas:

```
lemma weight_xyxy (x y :         mathbbR) : x * y * x * y = x^2 * y^2 := by ring
lemma weight_xyyx (x y :         mathbbR) : x * y * y * x = x^2 * y^2 := by ring
lemma sixBlock_sq_form ...
lemma sixBlock_sq_form_swap_cd ...
```

On paper: "by symmetry." In Lean: dozens of helper lemmas.

2.7 Summary: The Formalization Gap

Aspect	**Paper**	**Lean**
Index summation	Implicit	Explicit `sumIdx` machinery
Domain conditions	"Clearly"	Threaded hypotheses everywhere
Derivatives	Assumed	`DifferentiableAt` proofs required
Simplification	"By algebra"	Targeted rewrites + `ring_nf`
Symmetry	"By symmetry"	Explicit antisymmetry lemmas

The formalization difficulty is **orthogonal** to mathematical difficulty. G1 is mathematically trivial but formally hard. G2–G5 would be both mathematically and formally hard.

This gap is a finding in itself: it explains why GR hasn't been formalized before and why the artifact has value.

3 Axiom Calibration: An Organizational Methodology

3.1 Purpose and Scope

AxCal is an **organizational methodology** for tracking which foundational axioms are used by different proof routes in physics. It is not:

- A new foundational system
- A replacement for reverse mathematics
- A claim of deep mathematical novelty

It applies standard meta-mathematical concepts (from reverse mathematics [5], constructive analysis [1], and proof theory) to organize the foundational analysis of GR theorems.

3.2 The Four Portals

A proof "crosses a portal" when it uses a technique requiring specific axiomatic strength:

Zorn Portal (Choice Axis). Applies Zorn's lemma to a poset of mathematical objects. Requires Axiom of Choice.

GR examples: Maximal extensions (G4), MGHD existence (G2).

Limit-Curve Portal (Compactness Axis). Invokes Ascoli-Arzelà or similar compactness arguments to extract convergent subsequences. Requires Fan Theorem (constructive) or Weak König's Lemma (classical reverse math).

GR examples: Singularity theorems (G3).

Serial-Chain Portal (Choice Axis via DC). Builds infinite sequences by dependent choice: x_{n+1} depends on $x_0, \ldots, x_n$.

GR examples: Some geodesic extension arguments.

Reductio Portal (Logic Axis). Essential proof by contradiction: assume $\neg P$, derive contradiction, conclude P for non-decidable P.

GR examples: Singularity theorems (G3), uniqueness proofs (G2).

3.3 Height Profiles

An **AxisProfile** is a triple $(h_{\text{Choice}}, h_{\text{Comp}}, h_{\text{Logic}})$ where each component is 0, 1, or ω:

- **Height 0:** No use of the corresponding axiom class
- **Height 1:** Uses countable/dependent choice, Fan Theorem, or LEM for Σ_0 statements
- **Height ω:** Essential use at higher cardinalities or impredicative levels

Height 0 = Constructive at proof-route level. A Height 0 proof performs finite symbolic computation, explicit construction, or decidable case analysis. No choice, compactness, or essential contradiction.

3.4 From Organizational Profiles to Mechanical Audits

A common objection is that a "Height 0" classification can be undercut by the proof assistant's ambient foundations: importing large libraries may silently pull in classical choice or excluded middle even if the *mathematical idea* is constructive. We therefore separate two notions:

- **Proof-route height (this paper):** whether the argument crosses any of the four portals (Zorn, compactness, serial-choice, reductio) when written in a standard mathematical style.
- **Kernel-level axiom signature (auditable in Lean):** which axioms a particular theorem depends on, as reported by Lean's dependency tracking.

Lean supports a concrete audit via `#print axioms`. For example, one can ask Lean for the axioms used by a theorem:

```
#print axioms Papers.P5_GeneralRelativity.Schwarzschild.Ricci_zero_ext
```

In the present artifact, we emphasize the proof-route claim (no high-level portals are needed for G1), and we outline a path to strengthen it into a mechanical axiom signature report as future work (Section 4.2 records the exact library versions needed for reproducibility).

3.5 The Five GR Targets

Target	Description	Profile	Status
G1	Schwarzschild vacuum	$(0, 0, 0)$	**Formalized**
G2	Cauchy/MGHD existence	$(1, 0, 1)$	Paper analysis
G3	Singularity theorems	$(0, 1, 1)$	Paper analysis
G4	Maximal extensions	$(1, 0, 0)$	Paper analysis
G5	Computable evolution	Fails	Paper analysis

G1 is Height 0. The Schwarzschild vacuum check is finite symbolic computation: derivatives of $f(r) = 1 - 2M/r$, Christoffel formula application, index contraction, algebraic simplification. No portals crossed.

This is **unsurprising**—nobody would expect a textbook calculation to require Zorn. The value of formalizing G1 is not the Height 0 result itself, but demonstrating that the AxCal tracking machinery works.

G2–G5 are where hidden axioms live. These require non-trivial foundational commitments:

- G2: Zorn for maximality, reductio for uniqueness
- G3: Compactness (limit-curve lemma), reductio (completeness $\Rightarrow$ contradiction)
- G4: Zorn on extension posets
- G5: Pour-El-Richards shows computable data can yield non-computable solutions

Formalizing these would be the valuable next step.

3.6 What AxCal Is Not

Not a claim of novelty for reverse mathematics. The correspondence between Zorn and AC, between Ascoli-Arzelà and WKL_0, between reductio and LEM is standard. AxCal repackages these in GR-specific terminology.

Not a claim that Height 0 means "constructive in Lean." Our Lean formalization uses classical `mathlib` (with full LEM and choice). The Height 0 claim is about the *proof route*, not the verification infrastructure. This distinction ("structural certification") is philosophically debatable but organizationally useful.

Not a complete analysis. The G2–G5 profiles are based on analyzing standard proofs. We do not prove these are *optimal*—alternative proof routes might achieve lower heights.

4 The Lean 4 Formalization

4.1 Repository Structure

The formalization consists of three main files:

File	Lines	Content
`Schwarzschild.lean`	∼2,300	Metric, Christoffel symbols, f derivatives
`Riemann.lean`	∼12,400	Riemann components, Ricci tensor, symmetries
`Invariants.lean`	∼1,300	Kretschmann scalar, six-block decomposition

Supporting files: `Interfaces.lean` (type definitions), `Compose.lean` (aggregators), `Certificates.lean` (AxCal tracking stubs).

4.2 Artifact and Reproducibility

Repository: *https://github.com/AICardiologist/FoundationRelativity*
DOI: *10.5281/zenodo.18489703*
Commit/tag: *483dd449fd699ec91296a617295d27f9db161d0f*
Lean: *leanprover/lean4:v4.23.0-rc2* ***mathlib:*** *24dd4cacbe11d2535f2537c4a64ab25f72842cee*
Build: *lake build* ***No sorries (GR):*** *rg -n "sorry" -g "*.lean" Papers/P5_GeneralRelativit*
Machine used for timing: *macOS (Darwin 24.3.0, arm64)*
Clean full-repo build time: *1h 11m 54s (cold cache, Feb 4 2026)*
GR target build time: *2m 10s for Papers.P5_GeneralRelativity.GR.Riemann (warm cache, first target build)*
Incremental GR rebuild: *4.17s (warm cache, no source changes, Feb 4 2026)*

The development is a standard Lean 4/`mathlib` project. From the repository root, the following commands should reproduce the main results:

```
lake build
```

and (optionally) run a focused build of the GR modules:

```
lake build Papers.P5_GeneralRelativity.GR.Schwarzschild
lake build Papers.P5_GeneralRelativity.GR.Riemann
lake build Papers.P5_GeneralRelativity.GR.Invariants
```

Exact versions are pinned in `lean-toolchain` and `lake-manifest.json` at the repository root; those files should be treated as authoritative for reproduction.

4.3 Key Definitions

Index type.

```
inductive Idx | t | r |                    theta |                    phi
  deriving DecidableEq, Fintype
```

Metric and inverse.

```
noncomputable def g (M :  mathbbR) : Idx  to Idx  to  mathbbR  to  mathbbR  to  mathbbR
| Idx.t, Idx.t, r, _ => -(f M r)
| Idx.r, Idx.r, r, _ => (f M r)^-1
| Idx.                   theta, Idx.                   theta, r, _ => r^2
| Idx.           phi, Idx.           phi, r, theta => r^2 * (Real.sin theta)^2
| _, _, _, _ => 0
```

Christoffel aggregator.

```
noncomputable def Gammatot (M : mathbbR) : Idx to Idx to Idx to mathbbR to mathbbR to mathbbR
| Idx.t, Idx.t, Idx.r, r, _ => M / (r^2 * f M r)
| Idx.r, Idx.t, Idx.t, r, _ => M * f M r / r^2
-- ... 9 non-zero cases ...
| _, _, _, _, _ => 0
```

4.4 Main Theorems

Vacuum equations.

```
theorem Ricci_zero_ext (M rtheta :mathbbR) (h_ext : Exterior M rtheta) (h_sin_nz : Real.sinthetaneq 0)
                forall a b, RicciContraction M r              theta a b = 0
```

Kretschmann invariant.

```
theorem Kretschmann_exterior_value (M rtheta :mathbbR) (hM : 0 < M) (hr : 2*M < r) (htheta : 0 <thetala
    Kretschmann M r                         theta = 48 * M^2 / r^6
```

4.5 Technique Highlights from the Code

This section records a few concrete patterns that repeatedly improved proof stability and build time.

Package the exterior domain. Many theorems are naturally phrased on the exterior region $r > 2M$, where one immediately needs $r \neq 0$ and $f(M, r) \neq 0$ for denominator clearing. The code packages these hypotheses in an `Exterior` predicate and derives reusable nonvanishing lemmas (`r_ne_zero`, `f_ne_zero`). The $0 < \theta < \pi$ side-condition is tracked separately and used only to obtain $\sin\theta \neq 0$ for the inverse $\varphi\varphi$ component.

Expand finite index sums explicitly. The decisive move is to make every finite sum over indices reducible to a fixed normal form (e.g. a 4-term or 16-term expansion) before attempting algebra. For example:

```
rw [sumIdx2_expand]   -- expand     sum     mu     sum     nu into 16 concrete terms
```

This makes later `simp` calls predictable and avoids divergence from exploratory rewriting.

Use controlled simplification (`simp only`) with "vanishing lemma libraries." The Kretschmann contraction naively expands to 256 terms. The code first expands sums, then eliminates the vast majority of terms using an explicit lemma set stating that off-block Riemann components are zero in this chart. A representative proof step looks like:

```
simp only [Riemann_last_equal_zero,
  R_tr_t             theta_zero, R_tr_t             phi_zero,  -- (many more)
  ...]
```

The guiding principle is to keep the simp set *small and explicit* so that normalization is fast and stable across `mathlib` upgrades.

Exploit diagonal structure via a six-block identity. Once indices are raised with a diagonal inverse metric, the Kretschmann scalar collapses to six unordered index pairs. The artifact proves a structural lemma:

```
lemma Kretschmann_six_blocks
    (M r theta : mathbbR) (h_ext : Exterior M r theta) (h theta : Real.sin theta neq 0) :
    Kretschmann M r           theta = 4 * sumSixBlocks M r           theta := by
  classical
  -- expand to 256 terms, kill 232 by simp-only, regroup into 6 blocks
  ...
```

This is the main reason the final invariant computation is tractable.

Case-split intentionally, not accidentally. For results like $R_{\mu\nu} = 0$, the proof explicitly performs a 4×4 case split and handles off-diagonal cases by local simplification:

```
cases a <;> cases b
-- 16 cases: 12 off-diagonal simp-outs + 4 diagonal cancellations
```

While inelegant on paper, this is robust in a proof assistant and makes failure modes localized.

4.6 Proof Sketch and File Roles

File roles.

- `Schwarzschild.lean`: defines the Schwarzschild metric g, its inverse g^{-1}, and the nonzero Christoffel symbols Γ^a_{bc} (collected in `Gammatot`). It proves the required derivative lemmas for $f(r) = 1 - \frac{2M}{r}$ and the basic smoothness facts on the exterior domain.
- `Riemann.lean`: defines the Riemann tensor by the standard coordinate formula

  $$R^\rho{}_{\sigma\mu\nu} = \partial_\mu \Gamma^\rho{}_{\sigma\nu} - \partial_\nu \Gamma^\rho{}_{\sigma\mu} \; + \; \Gamma^\rho{}_{\lambda\mu}\Gamma^\lambda{}_{\sigma\nu} - \Gamma^\rho{}_{\lambda\nu}\Gamma^\lambda{}_{\sigma\mu},$$

  proves its symmetries, evaluates all nonzero components, and shows the Ricci tensor $R_{\mu\nu} = R^\alpha{}_{\mu\alpha\nu}$ vanishes.

- `Invariants.lean`: defines the Kretschmann scalar $K = R_{abcd}R^{abcd}$, proves the six-block decomposition using Riemann symmetries, and performs the final algebraic reduction to $48M^2/r^6$.

Key internal lemmas (selected).

- `sumIdx`, `sumIdx_expand`, `sumIdx_mul_sumIdx_swap`: finite-index summation machinery and Fubini-style reordering.
- `Gammatot_symm`, `g_symm`: symmetry lemmas for Christoffel and metric components.
- `Riemann_via_Gamma1`: the core identity rewriting $R_{\beta ar\theta}$ in terms of $\partial\Gamma_1$ and $\Gamma\Gamma$ terms.
- `R_tr_**_zero` family: systematic vanishing of off-diagonal components.
- `sixBlock` family: reduces 256-term Kretschmann sum to six weighted blocks.

Lean tactic patterns. The working proofs rely on short, reliable algebraic pipelines:

- `simp only` with explicit lemma lists (to avoid simp loops)
- `simp_rw` for controlled rewriting of sumIdx/ Gammatot expansions
- `field_simp` to clear denominators in $f(r)^{-1}$ and r^{-2}
- `ring_nf` or `abel` for normalization and cancellation

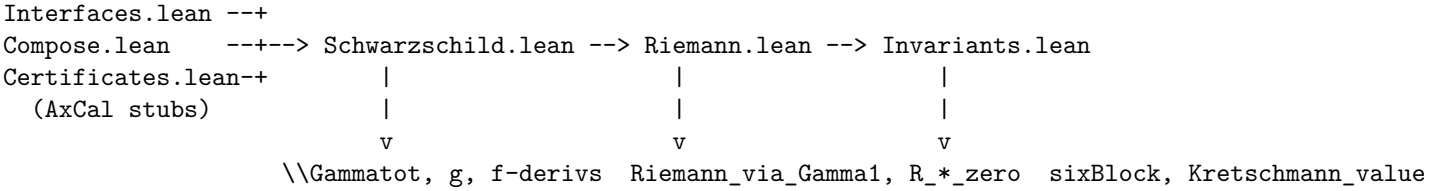

Figure 1: High-level dependency flow. Core lemmas in each file feed the next stage of the Schwarzschild pipeline.

Dependency diagram (key lemmas and files).

Human-readable proof sketch (G1). The Lean proof follows the textbook pipeline but makes every hypothesis explicit (e.g., $r > 2M$ and $0 < \theta < \pi$) and every cancellation justified.

1. **Domain and metric.** Fix the exterior region $r > 2M$ with $0 < \theta < \pi$. Define the Schwarzschild metric in Schwarzschild coordinates:

$$g_{tt} = -(1 - \tfrac{2M}{r}), \quad g_{rr} = (1 - \tfrac{2M}{r})^{-1}, \quad g_{\theta\theta} = r^2, \quad g_{\phi\phi} = r^2 \sin^2\theta.$$

   Lean also defines g^{-1} explicitly and proves $g^{-1}g = \mathrm{Id}$ componentwise under $r \neq 0$ and $\sin\theta \neq 0$.

2. **Christoffel symbols.** Use the coordinate formula for Γ^a_{bc} and a case split over indices to show only the standard nine symbols are nonzero. In the exterior region with $f(r) = 1 - \frac{2M}{r}$, the nonzero symbols are:

$$\Gamma^t_{tr} = \Gamma^t_{rt} = \frac{M}{r^2 f}, \quad \Gamma^r_{tt} = \frac{Mf}{r^2}, \quad \Gamma^r_{rr} = -\frac{M}{r^2 f},$$

$$\Gamma^r_{\theta\theta} = -(r - 2M), \quad \Gamma^r_{\phi\phi} = -(r - 2M)\sin^2\theta, \quad \Gamma^\theta_{r\theta} = \Gamma^\theta_{\theta r} = \frac{1}{r},$$

$$\Gamma^\theta_{\phi\phi} = -\sin\theta\cos\theta, \quad \Gamma^\phi_{r\phi} = \Gamma^\phi_{\phi r} = \frac{1}{r}, \quad \Gamma^\phi_{\theta\phi} = \Gamma^\phi_{\phi\theta} = \cot\theta.$$

   Lean proves these closed forms and their derivatives (e.g., $\partial_r f$ and $\partial_\theta \sin^2\theta$) with explicit differentiability lemmas on the exterior domain.

3. **Riemann tensor.** Define $R^\rho{}_{\sigma\mu\nu}$ by the coordinate formula above. Expand each nonzero component with the
   `Gammatot` aggregator, use symmetry identities $R_{abcd} = -R_{bacd} = -R_{abdc} = R_{cdab}$, and show all off-diagonal components vanish by case analysis.

4. **Ricci tensor.** Contract indices to form $R_{\mu\nu}$. Each diagonal component is a finite sum of nonzero terms that cancel; Lean proves these cancellations by explicit rewriting and algebraic normalization (`ring_nf`, `field_simp`).

5. **Kretschmann scalar.** Expand $K = R_{abcd}R^{abcd}$ as a finite sum over $4^4 = 256$ index combinations. Using Riemann symmetries and vanishing components, the sum collapses to six independent blocks (`sixBlock`). Each block is evaluated and the weighted sum reduces to $48M^2/r^6$.

Comparison with *Gravitation* (MTW). MTW (and Wald/Carroll) present essentially the same coordinate computation, but at a narrative level: many cancellations are asserted by symmetry or "by inspection," and domain hypotheses (e.g., $\sin\theta \neq 0$ or $r \neq 0$) are implicit. Lean forces every derivative, index contraction, and algebraic simplification to be explicit, and it surfaces which hypotheses are actually required. The formalization therefore mirrors MTW's flow but replaces informal steps with explicit, mechanically checked reductions.

Novelty and limits. There is no new physics here: the Schwarzschild vacuum check and Kretschmann invariant are textbook results. The novelty is methodological: (i) a complete, checkable proof in Lean 4, (ii) a detailed map of the proof's dependency structure, and (iii) a reusable infrastructure for future GR formalizations (e.g., alternative metrics or coordinate systems). The exercise highlights how "easy" GR calculations hide a large amount of formal bookkeeping.

Axioms and reverse-math perspective. At the level of mathematical content, the proof is *Height 0*: it is a finite symbolic computation with no need for choice, compactness, or nonconstructive existence arguments. That said, the current Lean development sits inside mathlib's classical environment, which assumes classical logic and standard real analysis facts (e.g., completeness of $\mathbb{R}$). The project therefore does not claim a fully constructive foundation; instead it identifies which axioms are actually *used* in the GR pipeline. The formalization also makes hidden hypotheses explicit (e.g., $r \neq 0$, $\sin\theta \neq 0$, $f(r) \neq 0$), clarifying the minimal domain required for each step.

4.7 What Is Verified (and What Is Not)

Verified. The Lean development verifies the Schwarzschild vacuum calculation in a single coordinate patch: explicit Christoffel symbols, all Riemann components, vanishing Ricci tensor, and the Kretschmann scalar in the exterior region $r > 2M$ with $0 < \theta < \pi$.

Not verified. This does *not* prove the full Einstein field equations or global properties of the spacetime, nor does it formalize coordinate changes, maximal extensions, or causal structure. It is a formal check of a standard coordinate computation, not a complete GR formalization.

4.8 Build Status

As of February 2026:

- `lake build Papers.P5_GeneralRelativity.GR.Invariants`: **Success**
- `lake build Papers.P5_GeneralRelativity.GR.Riemann`: **Success**
- `rg -glob '*.lean' -n '\bsorry\b' Papers/P5_GeneralRelativity/GR`: **No results**
- Builds should be run from the project root (the directory containing `lakefile.lean`).

Zero errors, zero sorries in the GR folder. (There remain lint warnings such as `unnecessarySimpa` and `unusedSimpArgs`; these do not affect correctness but indicate cleanup opportunities.)

5 Multi-AI Agent Collaboration

This project was produced by four AI agents under human direction over roughly six months of active development (April–October 2025).

5.1 Agent Roles

Claude Code (Opus 4.5). Primary implementation and tactical coding.

- Repository setup, namespace organization, CI/CD
- Index type design (`Idx`, `sumIdx`, `Gammatot`)
- Refactoring when early approaches hit complexity walls
- Build system maintenance across `mathlib` updates

GPT-5.2. Mathematical strategy and theorem formulation.

- Theorem formulation with minimal dependency choices
- Proof strategy for bottleneck identities
- Tactic selection and algebraic normalization guidance
- Key breakthrough: identified need for C^2 smoothness lemma

Gemini 2.5 Pro Deep Think (Google). Strategic guidance.

- Theorem formulation: minimal dependency choices
- Kretschmann six-block strategy: reduce 256 terms to 6
- Literature correlation: mapping to MTW chapters
- AxCal framework validation

GPT-5.2 Codex (late-stage). Code cleanup and refactoring.

- Lint-driven cleanup (unused simp arguments, unreachable tactics)
- Simplification of differentiability lemmas and helper proofs
- Paper edits to reflect build status and workflow

5.2 Trust Boundary: AI as Proposer, Lean as Verifier

A critical discipline in the workflow was maintaining a strict trust boundary: AI systems were used to propose lemmas, decompositions, and proof strategies, but *never* as sources of mathematical authority. Every claim was required to pass Lean's kernel; hallucinated lemmas were treated as search noise. In practice, this meant (i) preferring small local lemmas over monolithic automation, (ii) keeping simp/rewrite sets explicit, and (iii) using the prover to enforce all domain side-conditions (nonvanishing denominators, $\sin\theta \neq 0$, etc.).

5.3 Human Role

1. **Goal setting:** Defined G1–G5 targets and AxCal framework
2. **Agent coordination:** Directed which AI handled which task
3. **Quality control:** Reviewed all AI-generated code for correctness, clarity, maintainability

5.4 Lessons Learned

What worked:

- Specialization: Different agents have different strengths
- Iteration: Multiple rounds of generation/review/refinement
- Documentation: Agents document their own reasoning

What didn't work:

- Aggressive automation (`simp` explosions)
- Assuming `mathlib` has what you need (it often doesn't)
- Large monolithic proofs (need to break into lemmas)

6 Limitations and Future Work

6.1 What We Did Not Prove

- **Behavior at $r = 2M$:** The event horizon is a coordinate singularity. Extending to Kruskal-Szekeres coordinates would require additional formalization.
- **Behavior at poles $\theta = 0, \pi$:** Coordinate singularity in spherical coordinates. Extending would require stereographic patch.
- **Global structure:** We work in a single coordinate chart. Full manifold theory with atlas, transition functions, coordinate-independent tensors is not formalized.
- **G2–G5:** The interesting cases with hidden axioms are paper-level analysis only.

6.2 Future Directions

Short-term (builds on current work):

1. Add stereographic coordinate patch for poles
2. Prove coordinate transformations preserve curvature
3. Extend to Reissner-Nordström (charged), Kerr (rotating)

Medium-term (substantial new work):

1. Formalize Kruskal-Szekeres coordinates, prove regularity at horizon
2. Formalize Cauchy problem (G2)—local PDE well-posedness
3. Build manifold atlas machinery

Long-term (research-level):

1. Formalize singularity theorems (G3)—requires global causal structure
2. Formalize Pour-El-Richards (G5)—requires computability theory in Lean
3. Prove height profiles are *optimal* (no lower-height proof exists)

7 Conclusion

We have presented a fully machine-checked verification of Schwarzschild curvature calculations in Lean 4. The formalization demonstrates that formal GR is feasible, documents why "trivial" mathematics is hard to formalize, and provides infrastructure for future work.

The AxCal framework offers an organizational methodology for tracking axiomatic dependencies in physics. While the Height 0 result for G1 is unsurprising, the framework is positioned for application to G2–G5, where hidden axioms actually appear.

The multi-AI agent workflow (Claude Code Opus 4.5, GPT-5.2, Gemini 2.5 Pro Deep Think, GPT-5.2 Codex) represents a case study in AI-assisted theorem proving, showing how different AI strengths complement each other.

The value of this work is the *artifact* and the *program* it initiates, not new mathematics. We have demonstrated feasibility and provided a foundation for more ambitious formalizations of General Relativity.

Acknowledgments

The author thanks the AI agents (Claude Code Opus 4.5, GPT-5.2, Gemini 2.5 Pro Deep Think, GPT-5.2 Codex) for their contributions to the formalization. All errors remain the author's responsibility.

References

[1] Errett Bishop and Douglas Bridges. *Constructive Analysis*, volume 279 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag, Berlin, 1985.

[2] Sean M. Carroll. *Spacetime and Geometry: An Introduction to General Relativity*. Addison-Wesley, San Francisco, 2004.

[3] Charles W. Misner, Kip S. Thorne, and John Archibald Wheeler. *Gravitation*. W. H. Freeman, San Francisco, 1973.

[4] Marian B. Pour-El and J. Ian Richards. *Computability in Analysis and Physics*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1989.

[5] Stephen G. Simpson. *Subsystems of Second Order Arithmetic*. Perspectives in Logic. Cambridge University Press, 2nd edition, 2009.

[6] Robert M. Wald. *General Relativity*. University of Chicago Press, Chicago, 1984.