

# Formalizing Schwarzschild Spacetime in Lean 4: A First Complete Verification of General Relativistic Curvature 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 the first complete formal verification of Schwarzschild curvature calculations in a proof assistant, comprising approximately 16,000 lines of Lean 4 code. The formalization verifies the full pipeline from metric definition through Christoffel symbols, Riemann tensor components, Ricci tensor vanishing, and Kretschmann invariant ( $K = 48M^2/r^6$ ) in the exterior region ( $r > 2M$ ). Despite the underlying mathematics being textbook material, the formalization required six months and three collaborating AI agents (Claude Code, GPT-5-Pro, Gemini 2.5 Pro), revealing a significant gap between informal and formal mathematical practice.

We introduce *Axiom Calibration* (AXCAL), an organizational methodology for tracking which foundational axioms (choice, compactness, classical logic) are required by different proof routes in physics. The formalization structurally certifies that the Schwarzschild vacuum verification is “Height 0”—requiring no Axiom of Choice, compactness arguments, or essential proof by contradiction—a result that is expected but now machine-verified. We analyze four additional GR targets (Cauchy problem, singularity theorems, maximal extensions, computable evolution) at the paper level, identifying where non-constructive axioms enter standard proofs.

The primary contributions are: (1) the artifact itself, demonstrating feasibility of formal GR verification; (2) documentation of why “trivial” mathematics is surprisingly hard to formalize; (3) a multi-AI agent workflow as a case study in AI-assisted theorem proving; and (4) the AXCAL framework as a tool for organizing foundational analysis of physics theorems.

**Artifact:** 16,000 lines Lean 4 | **Build time:** <30 seconds | **Status:** Zero errors, zero sorries on critical path

## 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 [13, 8, 3]. 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

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	What This Paper Is (And Is Not)	3
1.2	Motivation: Searching for Hidden Axioms	3
1.3	Why Bother Formalizing the Easy Case?	4
1.4	Paper Contributions	4
<b>2</b>	<b>Why “Trivial” Mathematics Is Hard to Formalize</b>	<b>5</b>
2.1	The Index Machinery Problem	5
2.2	The Side Condition Problem	5
2.3	The Tactic Fragility Problem	6
2.4	The Library Mismatch Problem	6
2.5	The Differentiability Overhead	6
2.6	The Algebraic Normalization Problem	7
2.7	Summary: The Formalization Gap	7
<b>3</b>	<b>Axiom Calibration: An Organizational Methodology</b>	<b>7</b>
3.1	Purpose and Scope	7
3.2	The Four Portals	7
3.3	Height Profiles	8
3.4	The Five GR Targets	8
3.5	What AXCAL Is Not	9
<b>4</b>	<b>The Lean 4 Formalization</b>	<b>9</b>
4.1	Repository Structure	9
4.2	Key Definitions	9
4.3	Main Theorems	10
4.4	Proof Sketch and File Roles	10
4.5	Build Status	11
<b>5</b>	<b>Multi-AI Agent Collaboration</b>	<b>12</b>
5.1	Agent Roles	12
5.2	Human Role	12
5.3	Lessons Learned	12
<b>6</b>	<b>Limitations and Future Work</b>	<b>13</b>
6.1	What We Did Not Prove	13
6.2	Future Directions	13
<b>7</b>	<b>Conclusion</b>	<b>14</b>

# 1 Introduction

## 1.1 What This Paper Is (And Is Not)

This paper reports on the first complete formal verification of Schwarzschild spacetime curvature calculations in any proof assistant. We verify the standard textbook pipeline:

Schwarzschild metric  $\rightarrow$  Christoffel symbols  $\rightarrow$  Riemann tensor  $\rightarrow$  Ricci tensor  $\rightarrow$  Kretschmann scalar

in Lean 4 [5], 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 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 [10] shows this *fails*

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

### 1.3 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. **First of its kind:** No prior complete formalization of GR curvature calculations exists in any proof assistant. Being first has value even for routine mathematics.

### 1.4 Paper Contributions

1. **Artifact:** First complete Lean 4 formalization of Schwarzschild curvature (Christoffel  $\rightarrow$  Riemann  $\rightarrow$  Ricci  $\rightarrow$  Kretschmann). Zero errors, zero sorries. 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 + GPT-5-Pro + Gemini 2.5 Pro 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
- 6 months of development
- 3 collaborating AI agents
- Numerous false starts and refactorings

This section documents why.

### 2.1 The Index Machinery Problem

**On paper:** Write  $\Gamma_{\mu\nu}^{\alpha}$  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(θ) + f(φ)`
- 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 ≠ 0`

This hypothesis must be threaded through *every lemma* that involves the metric. The `Exterior M r θ` predicate bundles the domain conditions:

```
structure Exterior (M r : ℝ) : Prop where
  hM : 0 < M
  hr : 2 * M < r
```

Every Christoffel lemma, every Riemann component, every Ricci contraction carries this predicate as a hypothesis. The overhead is substantial.

## 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` [12] has abstract differential geometry: smooth manifolds, tangent bundles, connections [2]. 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 : ℝ) (hr : r ≠ 0) :
```

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 : ℝ$) : x * y * x * y = x^2 * y^2 := by ring
lemma weight_xyyx (x y : ℝ$) : 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 <code>sumIdx</code> machinery
Domain conditions	“Clearly”	Threaded hypotheses everywhere
Derivatives	Assumed	<code>DifferentiableAt</code> proofs required
Simplification	“By algebra”	Targeted rewrites + <code>ring_nf</code>
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 [11], 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 [7].

*GR examples:* Maximal extensions (G4) [6], MGHD existence (G2) [13, 4].

**Limit-Curve Portal (Compactness Axis).** Invokes Ascoli-Arzelà or similar compactness arguments to extract convergent subsequences. Requires Fan Theorem (constructive) [1] or Weak König’s Lemma (classical reverse math) [11].

*GR examples:* Singularity theorems (G3) [9, 6].

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

*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  $\omega$ :

- **Height 0:** No use of the corresponding axiom class
- **Height 1:** Uses countable/dependent choice, Fan Theorem, or LEM for  $\Sigma_0$  statements
- **Height  $\omega$ :** 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 The Five GR Targets

Target	Description	Profile	Status
G1	Schwarzschild vacuum	(0, 0, 0)	<b>Formalized</b>
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 [10] shows computable data can yield non-computable solutions

Formalizing these would be the valuable next step.

### 3.5 What AxCal Is Not

**Not a claim of novelty for reverse mathematics.** The correspondence between Zorn and AC, between Ascoli-Arzelà and WKL<sub>0</sub>, 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
<code>Schwarzschild.lean</code>	~2,300	Metric, Christoffel symbols, $f$ derivatives
<code>Riemann.lean</code>	~12,400	Riemann components, Ricci tensor, symmetries
<code>Invariants.lean</code>	~1,300	Kretschmann scalar, six-block decomposition

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

### 4.2 Key Definitions

**Index type.**

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

**Metric and inverse.**

```
noncomputable def g (M : ℝ) : Idx → Idx → ℝ → ℝ → ℝ
| Idx.t, Idx.t, r, _ => -(f M r)
| Idx.r, Idx.r, r, _ => (f M r)¹
| Idx., Idx., r, _ => r²
| Idx., Idx., r, _ => r² * (Real.sin )²
| _, _, _, _ => 0
```

**Christoffel aggregator.**

```
noncomputable def tot (M : ℝ) : Idx → Idx → Idx → ℝ → ℝ → ℝ
| Idx.t, Idx.t, Idx.r, r, _ => M / (r² * f M r)
| Idx.r, Idx.t, Idx.t, r, _ => M * f M r / r²
-- ... 9 non-zero cases ...
| _, _, _, _, _ => 0
```

### 4.3 Main Theorems

**Vacuum equations.**

```
theorem Ricci_vanishes (M r : ℝ)
```

**Kretschmann invariant.**

```
theorem Kretschmann_value (M r : ℝ)
```

### 4.4 Proof Sketch and File Roles

**File roles.**

- **Schwarzschild.lean**: defines the Schwarzschild metric  $g$ , its inverse  $g^{-1}$ , and the nonzero Christoffel symbols  $\Gamma_{bc}^a$  (collected in `tot`). 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  $48M^2/r^6$ .

**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:

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

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

2. **Christoffel symbols.** Use the coordinate formula for  $\Gamma_{bc}^a$  and a case split over indices to show only the standard nine symbols are nonzero. Prove 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 `tot` 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 [8] (and Wald [13]/Carroll [3]) 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 non-constructive 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.

## 4.5 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**

Zero errors, zero sorries on the critical path.

## 5 Multi-AI Agent Collaboration

This project was produced by three AI agents under human direction over six months (April–October 2025).

### 5.1 Agent Roles

**Claude Code (Anthropic).** Primary implementation agent.

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

**GPT-5-Pro (OpenAI, 2025).** “Junior Tactics Professor.”

- Tactic selection: when to use `field_simp` vs `ring` vs `simp`
- Error diagnosis: type mismatch, missing hypothesis, timeout
- Pattern recognition: canonical tactic sequences that work
- 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

### 5.2 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

### 5.3 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 ( 50–100 hours).
- **Behavior at poles  $\theta = 0, \pi$ :** Coordinate singularity in spherical coordinates. Extending would require stereographic patch ( 40–60 hours).
- **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 the first complete formal 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, GPT-5-Pro, Gemini 2.5 Pro) 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, GPT-5-Pro, Gemini 2.5 Pro) 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] Anthony Bordg and Nicolò Cavalleri. Elements of differential geometry in Lean: A report for mathematicians. *arXiv preprint*, 2021.
- [3] Sean M. Carroll. *Spacetime and Geometry: An Introduction to General Relativity*. Addison-Wesley, San Francisco, 2004.
- [4] Yvonne Choquet-Bruhat. *General Relativity and the Einstein Equations*. Oxford Mathematical Monographs. Oxford University Press, 2009.
- [5] Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28*, pages 625–635. Springer, 2021.
- [6] Stephen W. Hawking and George F. R. Ellis. *The Large Scale Structure of Space-Time*. Cambridge Monographs on Mathematical Physics. Cambridge University Press, 1973.
- [7] Thomas Jech. *Set Theory*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 3rd edition, 2003.
- [8] Charles W. Misner, Kip S. Thorne, and John Archibald Wheeler. *Gravitation*. W. H. Freeman, San Francisco, 1973.
- [9] Roger Penrose. Gravitational collapse and space-time singularities. *Physical Review Letters*, 14(3):57–59, 1965.
- [10] Marian B. Pour-El and J. Ian Richards. *Computability in Analysis and Physics*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1989.

- [11] Stephen G. Simpson. *Subsystems of Second Order Arithmetic*. Perspectives in Logic. Cambridge University Press, 2nd edition, 2009.
- [12] The mathlib Community. The Lean mathematical library. <https://github.com/leanprover-community/mathlib4>, 2020. Accessed: 2026.
- [13] Robert M. Wald. *General Relativity*. University of Chicago Press, Chicago, 1984.