Abstract

We calibrate the logical strength of Banach space non-reflexivity over Bishop-style constructive mathematics (BISH). Writing "bidual gap" for the failure of surjectivity of the canonical embedding $J_X : X \to X^{**}$, we show—*in a classical metatheory*—that over BISH the Weak Limited Principle of Omniscience (WLPO) is equivalent to the existence of a bidual gap for some real Banach space (our `BidualGapStrong`). A classical corollary is the non-surjectivity of J_{ℓ^∞}; this corollary is *not* formalized in Lean. The forward implication proceeds by classically extracting an Ishihara kernel (a finite tuple of functionals and a positive threshold) from a gap; the kernel-to-WLPO step is intuitionistic. Both directions of WLPO $\leftrightarrow$ BidualGapStrong are formalized in Lean 4, and the WLPO$\to$gap direction isolates a small witness-extraction step; we formalize both a WLPO-based lemma (using classical Bool extraction) and an LPO-based variant that provides a constructive witness. We also give finite-dimensional surrogates via Cesàro means (paper-level) and include, in Appendix A, a Stone window sketch identifying idempotents of ℓ^∞/c_0 with almost-equality classes of subsets of $\mathbb{N}$. This sketch is not used in the main equivalence and is not formalized in Lean. These results sharpen the folklore that the bidual gap is "non-constructive" by locating it precisely at WLPO over BISH.

The Bidual Gap and WLPO: A Constructive Calibration of Banach Space Non-Reflexivity

Paul Chun-Kit Lee*
New York University
dr.paul.c.lee@gmail.com

2026-02-05

IMPORTANT DISCLAIMER

A Case Study: Using Multi-AI Agents to Tackle Formal Mathematics

This entire Lean 4 formalization project was produced by multi-AI agents working under human direction. All proofs, definitions, and mathematical structures in this repository were AI-generated. This represents a case study in using multi-AI agent systems to tackle complex formal mathematics problems with human guidance on project direction.

Contents

*Lean 4 formalization available at https://github.com/AICardiologist/FoundationRelativity. Code archive available at Zenodo: DOI 10.5281/zenodo.17107493.

AI Assistance and Responsibility

This project used multiple AI systems in a supervised workflow. Initial Lean code and scaffolding were produced with Claude Code (Opus 4.5). Refactoring and consolidation were performed with GPT-5.2-Codex (Extra High). Proof-strategy exploration for the hardest steps was assisted by Gemini 2.5 Pro (Deep Think). All results were reviewed, edited, and integrated under human oversight; the author retains full responsibility for correctness.

The author is a practicing cardiologist rather than a professional logician or analyst. We expect domain experts may find errors or suggest improvements, and we welcome constructive feedback.

1 Introduction: Foundation Relativity and Logical Calibration

1.1 The foundation-sensitive bidual gap

For any Banach space X, the canonical embedding $J_X : X \to X^{**}$ maps $x \in X$ to the evaluation functional $J_X(x)(f) = f(x)$ for $f \in X^*$. A central question in functional analysis is whether J_X is surjective; if not, X is non-reflexive, and there exists a "bidual gap."

Definition 1.1 (Non-reflexivity and bidual gap)**.** *A Banach space X is **non-reflexive** if the canonical embedding J_X is not surjective. In this case, we say X **has a bidual gap**.*

For the space ℓ^∞ of bounded sequences, the answer depends on the foundational setting. In classical mathematics (ZFC), the Hahn–Banach theorem implies that J_{ℓ^∞} is not surjective. This ℓ^∞

corollary is not formalized in the Lean bundle; the formalization focuses on the abstract equivalence and the c_0 witness.

However, in Bishop-style constructive mathematics (BISH)—a system based on intuitionistic logic without the Law of Excluded Middle (LEM) or full Choice—one cannot prove that this gap exists. This dependence of provability on foundations exemplifies *foundation relativity*.

1.2 Motivation: From independence to precise calibration

This project began by probing whether functional-analytic "pathologies" might mirror independence phenomena. While no direct connection to Gödelian incompleteness emerged, the Lean 4 formalization led to a precise calibration result: the exact logical strength of the bidual gap.

The bidual gap is not independent in the Gödelian sense, but its provability over BISH is exactly that of a specific, weak non-constructive principle. This situates the problem within *Constructive Reverse Mathematics* (CRM), whose goal is to determine minimal axioms needed for classical theorems over constructive bases.

1.3 The Weak Limited Principle of Omniscience (WLPO)

Definition 1.2 (WLPO)**.** *For any binary sequence* $\alpha : \mathbb{N} \to \{0, 1\}$*,*

$$(\forall n,\ \alpha(n) = 0) \quad \vee \quad \neg(\forall n,\ \alpha(n) = 0).$$

WLPO is strictly weaker than LEM but not provable in BISH. It captures the minimal decision strength needed to determine whether an infinite sequence is identically zero.

Definition 1.3 (LPO)**.** *For any binary sequence* $\alpha : \mathbb{N} \to \{0, 1\}$*,*

$$(\forall n,\ \alpha(n) = 0) \quad \vee \quad \exists n,\ \alpha(n) = 1.$$

LPO implies WLPO and is strictly stronger. In this paper it appears only as an explicit witness-extraction principle (see the reverse direction and the Constructivity caveat).

1.4 Main result and contributions

Theorem 1.4 (Main Theorem)**.** *Over* BISH*, the following are equivalent:*

1. WLPO*.*
2. ***Gap*$_\exists$*:*** *There exists a real Banach space* X *such that* $J_X : X \to X^{**}$ *is not surjective (*`BidualGapStrong`*).*

Moreover, this equivalence is formalized in Lean 4 (see Section 4).

Corollary 1.5 (Classical corollary (not formalized in Lean))**.** *In* ZFC*, the embedding* $J_{\ell^\infty} : \ell^\infty \to (\ell^\infty)^{**}$ *is not surjective. The Lean bundle includes an abstract Option B bridge from a nonzero functional on* ℓ^∞/c_0 *to a bidual gap, but it does not specialize this bridge to* ℓ^∞*.*

Our contributions are:

1. **Logical calibration:** an exact equivalence within CRM.
2. **Formal verification:** a Lean 4 formalization of WLPO $\leftrightarrow$ `BidualGapStrong` (existence of a gap), plus an abstract Option B bridge.

3. **Methodology:** a Prop-level Ishihara kernel and a robust csSup treatment of partial sums.

4. **Constructive surrogates:** explicit finite-dimensional approximations via Cesàro means (paper-level; not formalized in Lean).

2 Main Mathematical Results: The Equivalence Theorem

We explain the equivalence between the bidual gap and WLPO.

2.1 Forward direction: Gap implies WLPO

To bridge a bidual-gap witness and a logical decision, we use an "Ishihara kernel", adapted from constructive reverse mathematics.

Definition 2.1 (Ishihara kernel)**.** *An Ishihara kernel consists of a normed space X, an element $y \in X^{**} \setminus J(X)$, a functional $f \in X^*$, a family $g : (\mathbb{N} \to \{0,1\}) \to X^*$, and a constant $\delta > 0$ such that for all binary sequences α:*

1. *$|y(f + g(\alpha))| = 0$ or $|y(f + g(\alpha))| \geq \delta$ (dichotomy);*

2. *$(\forall n,\ \alpha(n) = 0) \ \Leftrightarrow\ y(f + g(\alpha)) = 0$ (decision property).*

Lemma 2.2 (Kernel $\Rightarrow$ WLPO)**.** *If an Ishihara kernel exists, then* WLPO *holds.*

Proof. Given α, compute $s = |y(f + g(\alpha))|$. By dichotomy, either $s = 0$ or $s \geq \delta > 0$. The decision property equates $s = 0$ with $(\forall n, \alpha(n) = 0)$, deciding the WLPO instance. □

Theorem 2.3 (Gap implies WLPO (meta-classical))**.** *Working in a classical metatheory: if some Banach space X has $J : X \to X^{**}$ not surjective, then* WLPO *holds over* BISH*.*

Proof. Choose $y \in X^{**} \setminus J(X)$ with $y \neq 0$. A half-norm attainment lemma yields $h \in X^*$ with $\|h\| \leq 1$ and $|y(h)| > \|y\|/2$. Let $f = 0$, $\delta = |y(h)|/2$ (so $\delta > 0$ by $0 \leq \|y\|/2 < |y(h)|$), and

$$g(\alpha) = \begin{cases} 0 & \text{if } \forall n, \alpha(n) = 0, \\ h & \text{otherwise.} \end{cases}$$

Then the dichotomy and decision properties hold, so the kernel exists. □

2.2 Foundation-theoretic interlude: Classical logic in reverse mathematics

Before proceeding to the reverse direction, we must address a subtle but crucial point about the use of classical logic in constructive reverse mathematics.

Remark 2.4 (Classical meta-logic vs. object logic)**.** *The definition of $g(\alpha)$ in the proof case-splits on the undecidable proposition $(\forall n, \alpha(n) = 0)$. In* BISH*, we cannot constructively perform this case split. However, in reverse mathematics, we work in a* classical meta-logic *to prove statements* about BISH.

Concretely, we prove the implication:

(classically) $\big[$*there exists a Banach space with a bidual gap (a* BISH*-provable statement)*$\big] \implies \big[$BISH $\vdash$ WLPO$\big]$

This is the standard proof strategy in Constructive Reverse Mathematics (CRM), which follows a two-part structure that we frame using a producer/consumer *analogy [6]. First, a* classical producer*,*

operating in a meta-theory, uses non-constructive reasoning to extract a concrete "witness"—in our case, an Ishihara kernel—from a classical theorem. Second, a constructive consumer *analyzes this witness using only the logic of the constructive base theory (*BISH*) to derive a non-constructive principle like* WLPO*. This pattern is conceptually analogous to models of linear logic that separate the unlimited production of resources from their disciplined consumption [10], thereby clarifying how the non-constructive content of a theorem is isolated before its logical strength is measured.*

*In the formalization, we fence the meta-classical reasoning (witness extraction and the case split in g) inside a dedicated block, and keep the kernel consumer (*Ishihara kernel ⇒ WLPO*) intuitionistic. No classical axiom is added to the object theory. This approach follows Ishihara's original pattern for extracting logical principles from functional analysis results [7].*

We stress that LPO *is* not *obtained by this meta-classical case split. Whenever* LPO *is used, it is assumed explicitly as an object-level principle to supply a concrete index (a witness-extraction step). Thus the only meta-classical reasoning in the paper is the kernel extraction in Gap ⇒* WLPO*; the LPO-dependent step belongs entirely to the reverse direction.*

Special note (differences from an earlier draft). Two technical changes make the argument robust and CRM-compliant:

1. **Gap parameter.** We set
$$\delta := \frac{|y(h^\star)|}{2},$$
where $h^\star \in X^*$ satisfies $\|h^\star\| \le 1$ and $\frac{\|y\|}{2} < |y(h^\star)|$. This yields $\delta > 0$ by elementary order facts ($0 \le \frac{\|y\|}{2} < |y(h^\star)|$ implies $0 < |y(h^\star)|$, hence $0 < \delta$). In particular, we do *not* need to derive $\|y\| > 0$ from $y \neq 0$, which would otherwise force norm-positivity lemmas at the bidual type and introduce foundational friction in a constructive setting.

2. **Fencing classical reasoning.** The only classical steps are: picking $y \in X^{**}\backslash j(X)$, obtaining $h^\star$, and defining $g(\alpha)$ by a global case split. The kernel consumer (from the separation statement and the zero-characterization to WLPO) is intuitionistic and does not depend on classical principles.

3. **Reverse direction aligned with Lean.** Earlier drafts sketched the WLPO ⇒ gap direction via a quotient functional on ℓ^∞/c_0. While correct classically, that presentation obscured a witness-extraction step needed to control ℓ^1 norms constructively. In v6 we instead use the explicit c_0 witness (matching the Lean development) and make the required LPO-style extraction step explicit; see the "Constructivity caveat" below.

Both changes are invisible at the level of classical mathematics, but they are important for constructive reverse mathematics: the consumer no longer relies on bidual-specific norm facts or undecidable case splits.

2.3 Reverse direction: WLPO implies gap (via the c_0 witness)

Theorem 2.5 (WLPO implies Gap (with explicit witness-extraction))**.** *Assuming* WLPO *and the near-supremum coefficient step for ℓ^1 (obtained either by classical Bool extraction under WLPO or constructively from* LPO*), the canonical embedding $J : c_0 \to (c_0)^{**}$ is not surjective.*

Proof sketch. Work with the concrete witness $X = c_0$. Let (e_n) be the standard basis of c_0 and $\delta_n \in (c_0)^*$ the coordinate functionals. For $f \in (c_0)^*$, define

$$G(f) := \sum_{n=0}^{\infty} f(e_n).$$

Using the isometric identification $(c_0)^* \cong \ell^1$, the series converges absolutely and satisfies $|G(f)| \leq \|f\|$, so $G \in (c_0)^{**}$. The key analytic step is a near-supremum coefficient lemma for ℓ^1; constructively this requires extracting an index where $|a_n|$ is close to $\sup_n |a_n|$, which is exactly the LPO-style witness-extraction hypothesis stated above (in Lean, this is the lemma `exists_coeff_near_sup_LPO`).

Finally, G is not in the range of the canonical embedding $J : c_0 \to (c_0)^{**}$: if $G = J(x)$, then $G(\delta_n) = x_n$ for all n, but by construction $G(\delta_n) = 1$ for every n, forcing $x = (1, 1, 1, \dots) \notin c_0$. □

Constructivity caveat (WLPO vs. LPO). The explicit c_0 witness relies on a near-supremum coefficient lemma in ℓ^1. WLPO alone yields only a disjunction "all zero" or "not all zero" and does *not* produce an index witnessing a large coefficient. We therefore isolate this step in Lean and supply two variants: a WLPO-based lemma `exists_coeff_near_sup_WLPO` that uses classical Bool reasoning to extract an index, and an LPO-based lemma `exists_coeff_near_sup_LPO` that provides the witness constructively. The LPO lemma yields `dual_is_banach_l1_from_LPO` and `dual_is_banach_c0_dual_from_LPO` in `Papers.P2_BidualGap.HB.WLPO_DualBanach`. When emphasizing constructive witness extraction, one should read the WLPO ⇒ gap direction as conditional on this explicit LPO-level step (or a Markov-style substitute).

2.4 A structural byproduct

Theorem 2.6 (Gap structure). *(partial)Assuming* WLPO*, the quotient ℓ^∞/c_0 is nontrivial and carries a Boolean algebra of idempotents with rich structure.*

This is a paper-level statement and is not formalized in Lean; it connects naturally with the Stone window sketch in Appendix A (not used in the main equivalence).

3 Constructive Algorithms: Finite Approximations

Finite-dimensional surrogates clarify the constructive boundary: they are computable and illuminating, but their infinite limit would decide WLPO-type predicates.

3.1 Cesàro mean surrogates

Definition 3.1 (Cesàro mean). *For $n \geq 1$, define $f_n : \mathbb{R}^n \to \mathbb{R}$ by $f_n(x) = \frac{1}{n}\sum_{i=1}^n x_i$.*

Theorem 3.2 (Constructive finite Hahn–Banach). *(partial)With the sup norm on $\mathbb{R}^n$, the average $f_n(x) = \frac{1}{n}\sum_{i=1}^n x_i$ is the unique linear functional of norm 1 satisfying $f_n(1, \dots, 1) = 1$ and* $\ker(f_n) = M_n := \{x \in \mathbb{R}^n : \sum_{i=1}^n x_i = 0\}$.

Proof. If $\|x\|_\infty \leq 1$, then $|f_n(x)| \leq \frac{1}{n}\sum_{i=1}^n |x_i| \leq 1$, with equality at $x = (1, \dots, 1)$, so $\|f_n\| = 1$. The kernel claim is immediate from linearity and the definition of M_n; uniqueness follows because any linear functional with this kernel and value on $(1, \dots, 1)$ must agree with f_n on a direct-sum decomposition $\mathbb{R}^n = M_n \oplus \operatorname{span}\{(1, \dots, 1)\}$. □

Note. This finite-dimensional lemma is included as a paper-level surrogate and is not currently formalized in Lean.

3.2 Computational implementation

No numerical implementation is included in this bundle; Theorem 3.2 is intended as a paper-level illustration.

3.3 Convergence and the constructive boundary

Theorem 3.3 (Non-constructive limit)**.** *Viewing* (f_n) *as functionals on* ℓ^∞ *via projection, any constructive pointwise limit that separates the encodings described below would decide a* WLPO*-level predicate, and hence does not exist in* BISH*.*

Proof sketch. Encode a WLPO instance α by $v^\alpha \in \ell^\infty$; in a simple restricted case (at most one 1) let $v_n^\alpha = (-1)^{\sum_{k \le n} \alpha_k}$. Then $f_n(v^\alpha)$ converges to different limits depending on whether α is all zeros or has a single 1. A constructive limit would decide that dichotomy. □

4 Formalization in Lean 4

4.1 Overview and axiom profile

Our Lean 4 formalization contains about 5,600 lines of Lean code in this bundle (excluding mathlib).

- **Bidirectional proof:** both Gap ⇒ WLPO and WLPO ⇒ Gap.
- **Auxiliaries:** near-constructive development of $(c_0)^* \cong \ell^1$; a Stone window sketch appears in Appendix A (not formalized in Lean).
- **Option B (abstract):** a bridge from a nonzero functional on a quotient to a bidual gap; the ℓ^∞ specialization is not formalized.
- **Axioms:**

```
#print axioms Papers.P2.HB.gap_equiv_wlpo
-- [propext, Classical.choice, Quot.sound] (typical output)
```

  Here, `Quot.sound` underlies quotient constructions used in the development; `propext` is used extensively at Prop level; and `Classical.choice` (with `open Classical`) provides classical case splits in the meta-logic, as discussed in Remark 2.4.

4.2 CRM methodology and axiom hygiene

Following standard CRM (Constructive Reverse Mathematics) methodology, our formalization separates the *classical producer* from the *constructive consumer*:

- **Producer (classical):** The construction of the Ishihara kernel from the bidual gap, including extraction of $y \in X^{**} \setminus J(X)$, finding $h^\star$ with $\|y\|/2 < \|y(h^\star)\|$, and defining $g(\alpha)$ via case-splitting on the undecidable predicate $(\forall n, \alpha(n) = 0)$. This is fenced in `section ClassicalMeta` in `Ishihara.lean`.
- **Consumer (constructive):** The theorem `WLPO_of_kernel` that derives WLPO from the abstract kernel properties. This is placed *outside* any `noncomputable` or `Classical` sections, ensuring pure intuitionistic reasoning.

This separation can be mechanically verified in Lean:

```
#print axioms Papers.P2.Constructive.WLPO_of_kernel
-- (no classical axioms)

#print axioms Papers.P2.Constructive.WLPO_of_gap
-- [propext, Classical.choice, Quot.sound]
```

The result properly demonstrates that in a classical metatheory, BISH $\vdash$ BidualGapStrong $\Rightarrow$ WLPO.

4.3 Reproducibility information

Reproducibility Box

- **Repository**: https://github.com/AICardiologist/FoundationRelativity
- **Lean toolchain (root)**: `leanprover/lean4:v4.23.0-rc2`
- **Lake**: `Lake 5.0.0-src+ad1a017 (Lean 4.23.0-rc2)`
- **mathlib4 commit**: `24dd4cacbe11d2535f2537c4a64ab25f72842cee`
- **Project tag**: `p2-crm-v0.2`; **current commit**: `483dd449fd699ec91296a617295d27f9db161d0f`
- **Build (Paper 2 full)**: `lake build Papers.P2_BidualGap.P2_Full`
- **Build (Paper 2 minimal)**: `lake build Papers.P2_BidualGap.P2_Minimal`
- **Full repo build**: `lake build` (long; compiles mathlib)
- **Code Archive DOI**: `10.5281/zenodo.17107493`
- **Status**: `P2_Full` and `P2_Minimal` build. WIP files contain sorries and are excluded. The WLPO→gap direction isolates a witness-extraction step; we provide an LPO-based variant for that step (see Constructivity caveat).

4.4 Key technical solutions

Prop-level kernel technique. We implement the Ishihara kernel entirely at Prop level.

```
structure IshiharaKernel (X : Type _) [NormedAddCommGroup X] [NormedSpace ℝ X]
    where
  y      : (X →L[ℝ] ℝ) →L[ℝ] ℝ
  f      : X →L[ℝ] ℝ
  g      : (ℕ → Bool) → (X →L[ℝ] ℝ)
  δ      : ℝ
  δpos   : 0 < δ
  sep    : ∀ α : ℕ → Bool, |y (f + g α)| = 0 ∨ δ ≤ |y (f + g α)|
  zero_iff_allFalse :
    ∀ α : ℕ → Bool, (∀ n, α n = false) ↔ y (f + g α) = 0

theorem WLPO_of_kernel
  {X : Type _} [NormedAddCommGroup X] [NormedSpace ℝ X]
  (K : IshiharaKernel X) : WLPO := by
  intro α
  have h := K.sep α
  rcases h with h0 | hpos
  · have yz0 : K.y (K.f + K.g α) = 0 := (abs_eq_zero.mp h0)
```

```
    exact Or.inl ((K.zero_iff_allFalse α).mpr yz0)
  · have hne : K.y (K.f + K.g α) ≠ 0 := by
      intro yz0; have : |K.y (K.f + K.g α)| = 0 := by simp [yz0]
      exact (ne_of_gt (lt_of_lt_of_le K.δpos hpos)) this
    exact Or.inr (by
      intro hall
      exact hne ((K.zero_iff_allFalse α).mp hall))
```

Listing 1: Ishihara kernel (illustrative Lean snippet)

Robust csSup for partial sums. We avoid fragile complete-lattice instance resolution by working directly with conditional suprema:

```
private lemma tsum_eq_csSup_sum_of_nonneg
  {ι : Type*} (u : ι → ℝ) (h0 : ∀ i, 0 ≤ u i) (hs : Summable u) :
  (∑' i, u i) = sSup (Set.range (fun s : Finset ι => ∑ i ∈ s, u i)) := by
  have nonempty : (Set.range _).Nonempty := ⟨0, ∅, by simp⟩
  have bdd : BddAbove (Set.range _) := by
    use ∑' i, u i
    rintro x ⟨s, rfl⟩
    exact sum_le_tsum s (fun i _ => h0 i) hs
  apply le_antisymm
  · apply tsum_le_of_sum_le hs; intro s; exact le_csSup bdd ⟨s, rfl⟩
  · apply csSup_le nonempty; rintro x ⟨s, rfl⟩
    exact sum_le_tsum s (fun i _ => h0 i) hs
```

Listing 2: tsum equals csSup of finite partial sums

HasWLPO architecture. A lightweight typeclass separates WLPO-dependent arguments from constructive cores.

```
class HasWLPO : Prop :=
  (wlpo : ∀ (α : ℕ → Bool), (∀ n, α n = false) ∨ ¬(∀ n, α n = false))

namespace HasWLPO
theorem em_all_false [HasWLPO] (α : ℕ → Bool) :
    (∀ n, α n = false) ∨ ¬ (∀ n, α n = false) :=
  (inferInstance : HasWLPO).wlpo α
end HasWLPO

noncomputable instance instHasWLPO_of_Classical : HasWLPO := by
  classical
  refine ⟨?_⟩
  intro α
  simpa using Classical.em (∀ n, α n = false)
```

Listing 3: WLPO typeclass sketch

4.5 The bidirectional theorem (Lean)

```
open Papers.P2.HB

theorem gap_equiv_wlpo : BidualGapStrong.{0} ↔ WLPO := by
  constructor
  · exact gap_implies_wlpo
  · exact wlpo_implies_gap
```

Listing 4: WLPO ↔ Gap (top-level equivalence)

5 Significance and Conclusion

5.1 Precise logical calibration

The result places a familiar analytic phenomenon exactly at WLPO, demonstrating how constructive reverse mathematics can sharpen vague "non-constructive" labels into precise equivalences.

5.2 Foundation relativity

Foundation	Statement about the gap ($\exists y \notin \mathrm{im}(J)$)	Comment
ZFC	*Holds*	Hahn–Banach yields witnesses
BISH	*Equivalent to* WLPO	(Main Theorem)
BISH + DC	*Equivalent to* WLPO	DC does not imply WLPO
BISH + WLPO	*Holds*	by equivalence
BISH + ¬WLPO	*Not provable*	fails in models of BISH + ¬WLPO

5.3 The role of formalization

The Lean development clarified axiom usage, suggested robust approaches to conditional suprema, and yielded reusable patterns (Ishihara kernel, HasWLPO) for future formalized reverse mathematics.

5.4 Conclusion

We establish and formalize that the existence of a bidual gap has exactly the logical strength of WLPO over BISH. Finite constructive surrogates illuminate the obstruction at the infinite limit. The approach integrates constructive analysis, reverse mathematics, and mechanized verification.

Acknowledgments

I thank colleagues and the Lean community for discussions around constructive analysis and formal verification.

AI assistance disclosure. During development we used large language models as programming assistants: Claude Code (Opus 4.5) for initial Lean scaffolding, GPT-5.2-Codex (Extra High) for refactoring and consolidation, and Google Gemini 2.5 Pro (Deep Think) for difficult proof-strategy exploration. All suggested code and text were reviewed, edited, and verified by the author; the final mathematical claims are machine-checked by Lean 4 against mathlib. No model is credited as an author, and the author bears full responsibility for all content and any errors.

References

[1] E. Bishop. *Foundations of Constructive Analysis*. McGraw-Hill, 1967.

[2] H. Ishihara. Reverse mathematics in Bishop's constructive mathematics. *Philosophia Scientiae*, Cahier Spécial 6:43–59, 2006.

[3] F. Albiac and N. J. Kalton. *Topics in Banach Space Theory*. Springer, 2nd edition, 2016.

[4] D. S. Bridges and F. Richman. *Varieties of Constructive Mathematics*. Cambridge University Press, 1987.

[5] D. K. Brown and S. G. Simpson. Which set existence axioms are needed to prove the separable Hahn–Banach theorem? *Annals of Pure and Applied Logic*, 31(2–3):123–144, 1986.

[6] H. Diener. Constructive Reverse Mathematics. *arXiv:1804.05495*, 2018.

[7] H. Ishihara. An omniscience principle, the König lemma and the Hahn–Banach theorem. *Mathematical Logic Quarterly*, 36(3):237–240, 1990.

[8] The Lean 4 theorem prover. https://leanprover.github.io/

[9] The mathlib4 mathematical library. https://github.com/leanprover-community/mathlib4

[10] A. Pous and S. Zdancewic. A linear/producer/consumer model of classical linear logic. In *Proceedings of the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10. IEEE, 2014.

A Stone Window (Sketch)

This appendix records a sketch of the Stone window correspondence. It is independent of the main WLPO $\leftrightarrow$ gap equivalence and is not formalized in Lean.

A.1 Almost-equality and idempotents

Definition A.1 (Almost equality)**.** *For $A, B \subseteq \mathbb{N}$, write $A \sim B$ if the symmetric difference $A \triangle B$ is finite.*

Definition A.2. *Write* $\mathrm{Idem}(B)$ *for the set of idempotents of a (commutative) Banach algebra B under the induced Boolean operations $e \wedge f := ef$, $e \vee f := e + f - ef$, and $\neg e := 1 - e$.*

Theorem A.3 (Stone window)**.** *(partial)Equip ℓ^∞/c_0 with the quotient Banach algebra structure induced by pointwise operations. The map $\Phi : \mathcal{P}(\mathbb{N})/\sim \to \mathrm{Idem}(\ell^\infty/c_0)$, $[A] \mapsto [\chi_A]$, is a Boolean algebra isomorphism.*

Proof sketch. (1) $[\chi_A] = [\chi_B]$ iff $\chi_A - \chi_B \in c_0$, i.e. $A \sim B$. (2) Boolean operations are respected. (3) Every idempotent is equivalent (mod c_0) to an indicator via thresholding at $1/2$. □

Corollary A.4 (Finite distributive lattices)**.** *(partial)Every finite distributive lattice embeds into* $\mathrm{Idem}(\ell^\infty/c_0)$.

A.2 Lean sketch

(Sketch; not formalized in the Lean development.)

```
/-- Almost-equality of subsets of ℕ -/
def AlmostEq (A B : Set ℕ) : Prop := (A \ B ∪ B \ A).Finite

/-- Indicator function in ℓ∞ -/
def chi (A : Set ℕ) : ℕ → ℝ := fun n => if n ∈ A then 1 else 0

/-- The quotient ℓ∞/c₀ has a Banach algebra structure -/
instance : Algebra ℝ (ℓ∞ ⧸ c₀) where
  -- Multiplication is pointwise, well-defined modulo c₀
  mul := Quotient.map₂ (fun f g n => f n * g n) (by sorry)
```

```
  one := Quotient.mk (fun _ => 1)
  -- Other fields omitted

/-- Boolean algebra of idempotents -/
def Idem (B : Type*) [Mul B] := { e : B // e * e = e }

instance [Mul B] [One B] [Add B] [Sub B] : BooleanAlgebra (Idem B) where
  inf e f := ⟨e.val * f.val, by simp [e.property, f.property]⟩
  sup e f := ⟨e.val + f.val - e.val * f.val, by ring_nf; simp [e.property, f.
    property]⟩
  compl e := ⟨1 - e.val, by ring_nf; simp [e.property]⟩
  -- Other fields follow from these operations

/-- Stone window isomorphism -/
def StoneWindow : (Set ℕ) / AlmostEq ≃ Idem (ℓ∞ / c₀) where
  toFun := Quotient.map (fun A => ⟨Quotient.mk (chi A), by sorry⟩) (by sorry)
  invFun := sorry  -- Inverse via support extraction
  left_inv := by sorry
  right_inv := by sorry
```

Listing 5: Stone window: Boolean algebra of idempotents (sketch)

B Selected Lean Snippets

This appendix presents key excerpts from our Lean 4 formalization, demonstrating the producer/-consumer architecture and axiom hygiene discussed in the main text. These snippets are illustrative but align with the actual implementation available at https://github.com/AICardiologist/FoundationRelativity.

B.1 Constructive consumer (Ishihara kernel → WLPO)

The following shows the purely constructive consumer that derives WLPO from an Ishihara kernel without using classical axioms:

```
/-! Constructive consumer (Ishihara.lean). -/
structure IshiharaKernel (X : Type _) [NormedAddCommGroup X] [NormedSpace ℝ X]
    where
  y      : (X →L[ℝ] ℝ) →L[ℝ] ℝ
  f      : X →L[ℝ] ℝ
  g      : (ℕ → Bool) → (X →L[ℝ] ℝ)
  δ      : ℝ
  δpos   : 0 < δ
  sep    : ∀ α : ℕ → Bool, |y (f + g α)| = 0 ∨ δ ≤ |y (f + g α)|
  zero_iff_allFalse :
    ∀ α : ℕ → Bool, (∀ n, α n = false) ↔ y (f + g α) = 0

theorem WLPO_of_kernel
  {X : Type _} [NormedAddCommGroup X] [NormedSpace ℝ X]
  (K : IshiharaKernel X) : WLPO := by
  intro α
  have h := K.sep α
  rcases h with h0 | hpos
  · have yz0 : K.y (K.f + K.g α) = 0 := (abs_eq_zero.mp h0)
    exact Or.inl ((K.zero_iff_allFalse α).mpr yz0)
  · have pos : 0 < |K.y (K.f + K.g α)| := lt_of_lt_of_le K.δpos hpos
    have hne : K.y (K.f + K.g α) ≠ 0 := by
```

```
      intro yz0; have : |K.y (K.f + K.g α)| = 0 := by simp [yz0]
      exact (ne_of_gt pos) this
    have : ¬ (∀ n, α n = false) := by
      intro hall
      have yz0 : K.y (K.f + K.g α) = 0 := (K.zero_iff_allFalse α).mp hall
      exact hne yz0
    exact Or.inr this
```

Listing 6: Constructive consumer: no classical axioms

B.2 Classical producer (Gap → kernel)

The classical producer extracts an Ishihara kernel from a bidual gap. Note the explicit `noncomputable` and `open Classical` declarations:

```
/-! Classical producer: fenced in `section ClassicalMeta`. -/
noncomputable section
section ClassicalMeta
open Classical

lemma exists_on_unitBall_gt_half_opNorm
  {E} [NormedAddCommGroup E] [NormedSpace ℝ E]
  (T : E →L[ℝ] ℝ) (hT : T ≠ 0) :
  ∃ x : E, ‖x‖ ≤ 1 ∧ (‖T‖ / 2) < ‖T x‖ := by
  classical
  by_contra h
  push_neg at h
  -- ... proof uses scaling on the unit ball and opNorm_le_bound
  -- ... then derives ‖T‖ = 0, contradiction

end ClassicalMeta
```

Listing 7: Classical producer: extracting kernel from gap

B.3 Gap → WLPO (full bridge)

This shows the complete meta-classical construction:

```
/-- Meta-classical: from a bidual gap, build a kernel and consume it. -/
theorem WLPO_of_gap (hGap : BidualGapStrong) : WLPO := by
  classical
  rcases hGap with ⟨X, Xng, Xns, Xc, _dualBan, _bidualBan, hNotSurj⟩
  letI : NormedAddCommGroup X := Xng
  letI : NormedSpace ℝ X := Xns
  letI : CompleteSpace X := Xc
  let j := NormedSpace.inclusionInDoubleDual ℝ X
  have : ∃ y : (X →L[ℝ] ℝ) →L[ℝ] ℝ, y ∉ Set.range j := by
    have : ¬ (∀ y, y ∈ Set.range j) := by
      simpa [Function.Surjective, Set.mem_range] using hNotSurj
    push_neg at this
    exact this
  rcases this with ⟨y, hy⟩
  have hy0 : y ≠ 0 := by intro h0; subst h0; exact hy ⟨0, by simp⟩
  obtain ⟨hstar, hstar_le1, hstar_big⟩ :
      ∃ h : (X →L[ℝ] ℝ), ‖h‖ ≤ 1 ∧ (‖y‖ / 2) < ‖y h‖ := by
    simpa using
      (exists_on_unitBall_gt_half_opNorm (E := (X →L[ℝ] ℝ)) y hy0)
  let δ : ℝ := ‖y hstar‖ / 2
```

```
  have δpos : 0 < δ := by
    -- uses 0 ≤ ‖y‖/2 < ‖y hstar‖
    have : 0 < ‖y hstar‖ := by
      have h0 : 0 ≤ ‖y‖ / 2 := div_nonneg (norm_nonneg y) (by norm_num)
      exact lt_of_le_of_lt h0 hstar_big
    simpa [δ] using half_pos this
  let f : X →L[ℝ] ℝ := 0
  let g : (ℕ → Bool) → (X →L[ℝ] ℝ) := fun α =>
    if (∀ n, α n = false) then 0 else hstar
  have sep : ∀ α, |y (f + g α)| = 0 ∨ δ ≤ |y (f + g α)| := by
    intro α; by_cases hall : ∀ n, α n = false
    · left; simp [f, g, hall]
    · right
      have : δ ≤ ‖y hstar‖ := by
        have hnn : 0 ≤ ‖y hstar‖ := norm_nonneg _
        simpa [δ] using half_le_self hnn
      simpa [f, g, hall, zero_add] using this
  have zero_iff_allFalse : ∀ α, (∀ n, α n = false) ↔ y (f + g α) = 0 := by
    intro α; constructor
    · intro hall; simp [f, g, hall]
    · intro h0; by_contra hnot
      have yh_eq : y (f + g α) = y hstar := by simpa [f, g, hnot, zero_add]
      have yh0 : y hstar = 0 := by simpa [yh_eq] using h0
      have pos : 0 < ‖y hstar‖ := by
        have h0' : 0 ≤ ‖y‖ / 2 := div_nonneg (norm_nonneg y) (by norm_num)
        exact lt_of_le_of_lt h0' hstar_big
      have zero : ‖y hstar‖ = 0 := by simpa [yh0]
      have : (0 : ℝ) < 0 := by simpa [zero] using pos
      exact lt_irrefl _ this
  exact WLPO_of_kernel (X := X)
    { y := y, f := f, g := g, δ := δ, δpos := δpos
      sep := sep, zero_iff_allFalse := zero_iff_allFalse }
```

Listing 8: Meta-classical: from bidual gap to WLPO

B.4 WLPO → Gap (Lean formalization)

The Lean development proves the reverse direction by a direct construction using the explicit bidual witness for c_0 and WLPO-based dual-Banach lemmas:

```
local notation "c₀" => C₀(ℕ, ℝ)

/-- WLPO ⇒ BidualGapStrong via direct construction. -/
lemma wlpo_implies_gap : WLPO → BidualGapStrong.{0} := by
  intro hWLPO
  have gap_c0 : ¬Function.Surjective (inclusionInDoubleDual ℝ c₀) :=
    c0_not_reflexive_via_direct
  use c₀
  exact ⟨inferInstance, inferInstance, inferInstance,
         dual_is_banach_c0_from_WLPO_struct hWLPO,
         dual_is_banach_c0_dual_from_WLPO hWLPO,
         gap_c0⟩
```

Listing 9: WLPO implies gap (Lean)

This appears in `Papers.P2_BidualGap.HB.WLPO_to_Gap_HB`.

B.5 OpNorm core (classical LUB)

The operator norm construction shows the classical use of suprema:

```
/-- Classical: existence of operator norm via sSup of values on unit ball. -/
namespace OpNorm
variable {X : Type*} [NormedAddCommGroup X] [NormedSpace ℝ X]

def valueSet (h : X →L[ℝ] ℝ) : Set ℝ :=
  { r | ∃ x, ‖x‖ ≤ 1 ∧ r = ‖h x‖ }

lemma valueSet_nonempty (h : X →L[ℝ] ℝ) : (valueSet h).Nonempty := by
  refine ⟨0, ?_⟩
  exact ⟨0, by simp, by simp⟩

lemma valueSet_bddAbove (h : X →L[ℝ] ℝ) : BddAbove (valueSet h) := by
  refine ⟨‖h‖, ?_⟩
  intro r hr
  rcases hr with ⟨x, hx, rfl⟩
  calc
    ‖h x‖ ≤ ‖h‖ * ‖x‖ := h.le_opNorm x
    _     ≤ ‖h‖ * 1   := mul_le_mul_of_nonneg_left hx (norm_nonneg _)
    _     = ‖h‖       := by simp

def HasOpNorm (h : X →L[ℝ] ℝ) : Prop :=
  ∃ N : ℝ, IsLUB (valueSet h) N

lemma hasOpNorm_CLF (h : X →L[ℝ] ℝ) : HasOpNorm h := by
  classical
  use sSup (valueSet h)
  exact isLUB_csSup (valueSet_nonempty h) (valueSet_bddAbove h)

lemma hasOpNorm_zero : HasOpNorm (0 : X →L[ℝ] ℝ) :=
  hasOpNorm_CLF (0 : X →L[ℝ] ℝ)
end OpNorm
```

Listing 10: Classical operator norm via supremum

B.6 Axiom hygiene verification

Finally, we can verify the axiom usage as discussed in Section 4.1:

```
#print axioms Papers.P2.Constructive.WLPO_of_kernel
-- (no classical axioms)

#print axioms Papers.P2.Constructive.WLPO_of_gap
-- [propext, Classical.choice, Quot.sound]

#print axioms Papers.P2.HB.wlpo_implies_gap
-- [propext, Quot.sound]  -- uses WLPO hypothesis, not Classical.choice

#print axioms Papers.P2.HB.gap_equiv_wlpo
-- [propext, Classical.choice, Quot.sound]
```

Listing 11: Axiom audit commands

The output confirms that the constructive consumer (`WLPO_of_kernel`) uses no classical axioms, while the meta-classical producer (`WLPO_of_gap`) requires classical choice. This validates our CRM methodology claim that the consumer remains purely constructive.