Cubical Higher Inductive–Inductive Types and the Cauchy Reals

A Cubical Agda Completion of the Book HoTT Construction

YonedaAI Research
Univalent Correspondence Project
Magneton Foundational Studies
mlong168@gmail.com

4 May 2026

Abstract

The Cauchy reals $\mathbb{R}_c$ admit a higher inductive–inductive presentation in Book HoTT (HoTT Book §11.3, Booij 2020), and this presentation underwrites the unique-existence definitions of π and e used throughout the *Univalent Correspondence* series (Paper V §8). However, the construction is given in Book HoTT, where path constructors are *postulated* together with their associated β-rules and where function extensionality is derived from univalence. A computational analogue in Cubical Agda—in which paths are functions $\mathbb{I} \to A$, univalence reduces, and higher inductive types compute—has been *in progress but incomplete*: the existing Cubical Agda standard library exposes set quotients and several truncations, but not a clean HIIT presentation of $\mathbb{R}_c$ with the closeness-relation constructor.

This paper completes the cubical version. We give a Cubical Agda specification of the simultaneous higher inductive–inductive type $(\mathbb{R}_c, \sim)$, where $\sim\colon \mathbb{Q}^{>0} \to \mathbb{R}_c \to \mathbb{R}_c \to \mathsf{Type}$ is the ε-closeness predicate of Booij. The four point/path/square constructors are expressed as primitive cubical operations: the limit constructor uses PathP rather than `Path`, the closeness path constructor uses an interval-typed family, and the propositional truncation of $\sim$ is replaced by a PathP-valued square constructor that we prove satisfies the same universal property as the truncated Book HoTT version. We verify that the cubical $\mathbb{R}_c$ is an h-set, that it carries the structure of an Archimedean ordered field, and that it is equivalent (via a cubical `Glue` type) to the Dedekind reals $\mathbb{R}_d$ when the latter is restricted to located Dedekind cuts. We also extract, via Cubical Agda's reduction discipline, an executable map $\mathsf{run} : \mathbb{N} \to \mathbb{Q}$ approximating $\sqrt{2}$, π, and e; the rational approximants we obtain by reducing the constructively-defined unique-existence centres are checked numerically against Haskell prototypes (Section 7). We close with the open problems that remain: judgemental η-rules for the relator constructor, full coherence of the IIT (inductive–inductive) elimination principle in the presence of Glue types, and integration with the `Cubical.HITs` hierarchy (`Cubical.HITs.CauchyReals` as a candidate library entry).

Contents

1 Introduction

The constructive real numbers occupy a privileged position in the foundations of analysis. In a setting without the law of excluded middle, no canonical definition of $\mathbb{R}$ exists; instead, several non-equivalent candidates compete: *Cauchy reals* $\mathbb{R}_c$, defined as a quotient of Cauchy sequences of rationals; *Dedekind reals* $\mathbb{R}_d$, defined as located cuts; and *streams* or *redundant binary signed-digit representations*, definable as final coalgebras (Paper III, §5). In classical mathematics these collapse to a single object, but constructively they differ by countable choice, fan-theorem-like principles, or modulus-of-convergence considerations [10]. The *Cauchy* presentation is the most useful for executable analysis: it admits a direct definition of limits, supports the unique-existence definitions of transcendentals via Picard-style fixed points, and yields an Archimedean ordered field on the nose.

In Homotopy Type Theory (HoTT) [1], an additional dimension is available: the *path* structure of identity types. §11.3 of the HoTT Book, refined by Booij [2], defines $\mathbb{R}_c$ not as a *quotient* but as a *higher inductive–inductive type* (HIIT): the carrier $\mathbb{R}_c$ and a parameterised closeness relation $\sim_\varepsilon\colon \mathbb{R}_c \to \mathbb{R}_c \to \mathsf{Type}$ are introduced *simultaneously*, with constructors that include both points and *path constructors* identifying close-enough representatives. The result is an h-set on which the limit operation $\mathsf{lim} : (\mathbb{Q}^{>0} \to \mathbb{R}_c) \to \mathbb{R}_c$ exists *by construction*, with no choice principle invoked.

This Book HoTT construction is the foundation of Paper V's treatment of π and e (Definitions 8.1, 8.2, here recalled in Section 3), which expresses each as the centre of a contractible type of solutions to its characteristic structural condition. As long as $\mathbb{R}_c$ is a *type*—and not merely a proposition—these definitions are mathematically meaningful even before one has computational content for the rationals approximating π to a given precision.

The gap. The synthesis of the prior series [13, §8, item 6] flagged the following gap:

> *The Cauchy reals construction of Paper V is given in Book HoTT; a clean cubical version (in Cubical Agda) is in progress but incomplete.*

The gap matters for three reasons. First, Book HoTT is a postulational extension of MLTT: path constructors are added with their β-rules postulated, breaking canonicity in the absence of a model with a computational interpretation. Cubical type theory (CCHM, [3]) restores canonicity by introducing the *interval* $\mathbb{I}$ as a primitive judgemental object with a de Morgan algebra structure and replacing path postulates with operations on $\mathbb{I}$-functions. Second, in Cubical Agda [4], univalence is *not* an axiom but a *theorem*: `ua` is a defined term whose β-rule reduces. The same machinery suggests that $\mathbb{R}_c$ should be expressible as an honest cubical HIIT, with all of its eliminators computing. Third, downstream applications (transcendentals, complex analysis, ζ-as-HoTT-native-statement; cf. [13, §7.3]) need a *computational* $\mathbb{R}_c$, not a postulated one.

What we do. We give a Cubical Agda specification of $(\mathbb{R}_c, \sim)$ as a HIIT with four constructors: `rat`, `lim`, `eq` (closeness-induced path), and a square constructor expressing propositional truncation of $\sim$. The novelty over the Book HoTT presentation is that *every* constructor is given as either a Type, a Path, or a PathP (a path over a path); we never invoke set-truncation or propositional-truncation as black boxes. The closeness predicate is defined simultaneously by an inductive-inductive scheme whose well-formedness in Cubical Agda we justify by reduction to the inductive-inductive schema of Altenkirch–Kaposi [6].

We then prove four results, *cubically*:

(R1) $\mathbb{R}_c$ **is an h-set.** (Theorem 6.2)

(R2) $\mathbb{R}_c$ **has the universal property of the Cauchy completion of $\mathbb{Q}$.** (Theorem 6.3)

(R3) **There is a cubical equivalence $\mathbb{R}_c \simeq \mathbb{R}_d^{\mathbf{loc}}$, where $\mathbb{R}_d^{\mathbf{loc}}$ is the type of located Dedekind cuts.** (Theorem 7.2)

(R4) $\mathbb{R}_c$ **extracts: the rational approximants to $\sqrt{2}$, π, and e are computed by reduction of normal forms in Cubical Agda.** (Section 9)

The remainder of the paper makes (R1)–(R4) precise, gives the Cubical Agda code for the central definitions and theorems, and identifies the remaining gaps—chiefly the absence of a fully judgemental η-rule for the closeness constructor and the lack of an integrated `Cubical.HITs.CauchyReals` module in the standard library [8].

Outline. Section 3 recalls the Book HoTT construction of $\mathbb{R}_c$ from [1, 2]. Section 4 reviews CCHM cubical type theory: the interval, Kan operations, Glue types, computational univalence. Section 5 discusses cubical HIITs in general, focusing on what changes when one moves from Book HoTT path-constructors to Cubical Agda `Path`/`PathP` constructors. Section 6 gives the Cubical Agda HIIT of $\mathbb{R}_c$. Section 7 sketches the equivalence with located Dedekind cuts. Section 9 discusses computational content and extraction of rational approximants. Section 10 lists open problems and the path forward.

Audience. We assume familiarity with HoTT at the level of [1], Chapters 1–3 and 6, and basic constructive analysis at the level of [10]. Cubical type theory at the level of [3] is helpful but is reviewed here. Familiarity with Cubical Agda [4] is helpful for reading the code blocks but is not strictly required.

2 Mathematical Framework and Notation

This section fixes notation and recalls the type-theoretic vocabulary used throughout. Readers familiar with the HoTT Book may safely skip to Section 3.

2.1 Universes and Truncation Levels

We work in a tower of universes $\mathcal{U}_0 : \mathcal{U}_1 : \mathcal{U}_2 : \ldots$, with the Tarski-style cumulativity $\mathcal{U}_n \subseteq \mathcal{U}_{n+1}$. We write Type for the generic universe; the index is left implicit unless required.

For $X : \mathsf{Type}$, the predicates $\mathsf{isContr}(X), \mathsf{isProp}(X), \mathsf{isSet}(X)$ are defined as in [1, §3]:

$$\begin{aligned}
\mathsf{isContr}(X) &:\equiv \Sigma_{x:X}\, \Pi_{y:X}\, x = y, \\
\mathsf{isProp}(X) &:\equiv \Pi_{x,y:X}\, x = y, \\
\mathsf{isSet}(X) &:\equiv \Pi_{x,y:X}\, \mathsf{isProp}(x = y).
\end{aligned}$$

A type X with $\mathsf{isSet}(X)$ is called an *h-set* (homotopy set). The type $\mathbb{R}_c$ to be defined will be an h-set, but its definition crucially uses the path structure between paths: it is not just a quotient.

2.2 Identity Types and Path Types

In Book HoTT, the identity type $a =_A b$ is generated by reflexivity and eliminated by the J-rule. In Cubical Agda, the type $\mathsf{Path}\, A\, a\, b$ replaces this and is judgmentally $\mathbb{I} \to A$ with a, b at the endpoints. The two are equivalent ([3, §6]), but the cubical formulation makes function extensionality, univalence, and HIT β-rules computable.

Definition 2.1 (Equivalence)**.** For $f : A \to B$, $\mathsf{isEquiv}(f) :\equiv \Pi_{b:B}\, \mathsf{isContr}(\Sigma_{a:A}\, f\, a = b)$. The type of equivalences is $A \simeq B :\equiv \Sigma_{f:A\to B}\, \mathsf{isEquiv}(f)$.

2.3 Higher Inductive Types and HIITs

A *higher inductive type* (HIT) is presented by a list of point constructors and path constructors (paths in the resulting type), with an eliminator that respects all of these. The canonical example is the circle S^1, with $\mathsf{base} : S^1$ and $\mathsf{loop} : \mathsf{base} = \mathsf{base}$.

A *higher inductive–inductive type* (HIIT) introduces both a type A and a family B over A (or over $A \times A$, etc.) with constructors that may reference inhabitants of either. The formal schema for HIITs in Book HoTT is given by Altenkirch–Kaposi [6] via quotient inductive–inductive types (QIITs); the cubical version is treated by [3] for QITs and extended in [4] to handle the HIIT case.

2.4 Conventions

- $\Pi_{x:A}\, B(x)$ is the dependent product (function) type, also written $(x : A) \to B(x)$ in Agda syntax.
- $\Sigma_{x:A}\, B(x)$ is the dependent sum (pair) type.
- We write $f \circ g$ for function composition, f^{-1} for the inverse of an equivalence.
- We use the convention that bound variables in Π and Σ types may be introduced silently when no ambiguity arises (so we write $\Pi q.\, P\, q$ for $\Pi_{q:T}\, P\, q$ when T is clear from context).

3 The Book HoTT Cauchy Reals: Recap of Paper V

We follow [1, §11.3], reorganising slightly to align with Booij's PhD thesis [2], which is the most detailed source for this construction.

3.1 Rationals and Closeness

We assume the type $\mathbb{Q}$ of rational numbers is given as a set quotient $(\mathbb{Z} \times \mathbb{Z}_{>0})/{\sim}$, where $\sim$ is the cross-multiplication equivalence. We write $\mathbb{Q}^{>0}$ for the type of strictly positive rationals, equivalently $\Sigma_{q:\mathbb{Q}}\, q > 0$. The standard operations $+, -, \cdot, |\cdot|$ are inherited from $\mathbb{Q}$; we treat them as judgmentally well-defined.

Definition 3.1 (ε-closeness on $\mathbb{Q}$)**.** For $\varepsilon : \mathbb{Q}^{>0}$ and $p, q : \mathbb{Q}$, define $p \sim_\varepsilon q :\equiv |p - q| < \varepsilon$.

3.2 The HIIT Presentation

A brief orientation. The Cauchy condition we use, $x_\delta \sim_{\delta+\varepsilon} x_\varepsilon$ for all $\delta, \varepsilon : \mathbb{Q}^{>0}$, is the *constructive* formulation: a sequence carries with it an *explicit modulus of convergence*, given by the index ε itself. Classically, one writes "$\forall\varepsilon\, \exists N\, \forall n \geq N\, |x_n - x_m| < \varepsilon$," but this requires both choice (to extract N from the existence claim) and a $\mathbb{N}$-indexed sequence. We avoid both: the sequence is indexed by $\mathbb{Q}^{>0}$ (so the rate of convergence is encoded in the indexing itself), and the Cauchy condition is Π-typed (so the modulus is an internal datum, not extracted via choice). This is the formulation adopted by Booij [2] and by the HoTT Book [1, §11.3].

Definition 3.2 (Cauchy reals as a HIIT, Booij)**.** The pair $(\mathbb{R}_c, \sim)$ is the simultaneous higher inductive–inductive type with the following constructors.

Carrier constructors of $\mathbb{R}_c$:

- $\mathsf{rat} : \mathbb{Q} \to \mathbb{R}_c$.
- $\mathsf{lim} : \Pi_{x:\mathbb{Q}^{>0} \to \mathbb{R}_c} \mathsf{isCauchyApprox}(x) \to \mathbb{R}_c$, where

$$\mathsf{isCauchyApprox}(x) :\equiv \Pi_{\delta,\varepsilon:\mathbb{Q}^{>0}}\, x_\delta \sim_{\delta+\varepsilon} x_\varepsilon.$$

- $\mathsf{eq}_{\mathbb{R}_c} : \Pi_{u,v:\mathbb{R}_c} (\Pi_{\varepsilon:\mathbb{Q}^{>0}}\, u \sim_\varepsilon v) \to u = v$.

Constructors of $\sim_\varepsilon$ (one per pair-shape). A small but crucial detail: the constructors below take subtractions of strictly positive rationals as $\mathbb{Q}^{>0}$-precisions, so each is implicitly conditioned on the relevant strict inequality ($\varepsilon > \delta$, $\varepsilon > \delta + \eta$). We carry these preconditions along throughout the paper; equivalently, one may reparameterise so that the gap is given as an additional $\mathbb{Q}^{>0}$ argument.

- **rat-rat.**

$$\Pi_{q,r:\mathbb{Q}}\, \Pi_{\varepsilon:\mathbb{Q}^{>0}}\, |q - r| < \varepsilon \;\longrightarrow\; \mathsf{rat}\, q \sim_\varepsilon \mathsf{rat}\, r.$$

- **rat-lim.**

$$\Pi_{q:\mathbb{Q}}\, \Pi_{y,\mathsf{cy}}\, \Pi_{\varepsilon,\delta:\mathbb{Q}^{>0}}\, \Pi_{h_>:\delta<\varepsilon},\quad \mathsf{rat}\, q \sim_{\varepsilon-\delta} y_\delta \;\longrightarrow\; \mathsf{rat}\, q \sim_\varepsilon \mathsf{lim}(y, \mathsf{cy}).$$

- **lim-rat.** Symmetric, with precondition $\delta < \varepsilon$.
- **lim-lim.**

$$\Pi_{x,\mathsf{cx},y,\mathsf{cy}}\, \Pi_{\varepsilon,\delta,\eta:\mathbb{Q}^{>0}}\, \Pi_{h_>:\delta+\eta<\varepsilon},\quad x_\delta \sim_{\varepsilon-\delta-\eta} y_\eta \;\longrightarrow\; \mathsf{lim}(x, \mathsf{cx}) \sim_\varepsilon \mathsf{lim}(y, \mathsf{cy}).$$

The preconditions are essential: $\mathbb{Q}^{>0}$ is not closed under subtraction, and $\varepsilon - \delta$ is well-defined in $\mathbb{Q}^{>0}$ exactly when $\delta < \varepsilon$. Booij's thesis [2, Definition 6.2.1] carries these inequalities as part of the constructor signatures; we adopt the same convention. An alternative formulation, equivalent up to coercion, takes the *positive gap* $\theta : \mathbb{Q}^{>0}$ as an explicit argument and writes $\varepsilon = \delta + \theta$ in place of $\varepsilon - \delta$; this avoids the precondition entirely.

Path constructor of $\sim$:

- For each ε, u, v, the type $u \sim_\varepsilon v$ is a proposition: any two inhabitants are equal.

Set-truncation:

- $\mathbb{R}_c$ is an h-set: for any $u, v : \mathbb{R}_c$ and $p, q : u = v$, $p = q$.

The dependent eliminator for $(\mathbb{R}_c, \sim)$ provides simultaneous induction principles for any motive $A : \mathbb{R}_c \to \mathsf{Type}$ and $B : \Pi_{u,v:\mathbb{R}_c} A(u) \to A(v) \to \mathbb{Q}^{>0} \to \mathsf{Type}$ respecting all the constructors and path constructors. For a full statement, see [1, Lemma 11.3.10].

3.3 Universal Property

Theorem 3.3 (Universal property of $\mathbb{R}_c$, [1] Thm. 11.3.5)**.** *For any complete metric space Y over $\mathbb{Q}$ that is an h-set, every $\mathbb{Q}$-uniformly continuous map $f : \mathbb{Q} \to Y$ extends uniquely to $\bar{f} : \mathbb{R}_c \to Y$.*

This is the Cauchy-completion universal property and the core technical result needed for the unique-existence definitions of π and e.

3.4 Why a HIIT and Not a Quotient

In *set-theoretic* foundations one defines $\mathbb{R}_c$ as a quotient of Cauchy sequences of rationals modulo the equivalence $x \sim y \iff \forall\varepsilon\,\exists N\,\forall n \geq N\,|x_n - y_n| < \varepsilon$. In type theory *without* countable choice (e.g. propositions-as-types MLTT), this quotient is *not* Cauchy-complete: lifting a Cauchy sequence of equivalence classes to a Cauchy sequence of representatives requires the choice principle. The HIIT construction of Definition 3.2 avoids this problem by introducing lim as a *primitive constructor* and identifying representatives via the path constructor $\mathsf{eq}_{\mathbb{R}_c}$.

3.5 π and e as Unique Existence Centres

Definition 3.4 (π, after [1] 11.3 and Paper V Def. 8.1)**.** $\pi : \mathbb{R}_c$ is the centre of the contractible type

$$P_\pi :\equiv \Sigma_{p:\mathbb{R}_c}\,(\sin p =_{\mathbb{R}_c} 0) \times (p > 0) \times (\Pi_{x:\mathbb{R}_c}\, 0 < x < p \to \sin x > 0).$$

Here $\sin : \mathbb{R}_c \to \mathbb{R}_c$ is the unique solution of $\sin'' + \sin = 0$, $\sin 0 = 0$, $\sin' 0 = 1$, again presented as a unique-existence centre.

Definition 3.5 (e, after Paper V Def. 8.2)**.** $e :\equiv \exp 1$, where $\exp : \mathbb{R}_c \to \mathbb{R}_c$ is the centre of

$$P_{\exp} :\equiv \Sigma_{f:\mathbb{R}_c \to \mathbb{R}_c}\,(f' = f) \times (f\,0 = 1).$$

The crucial property of these definitions is that, although π and e are introduced as *centres of contractible types*, the centres are *terms of type* $\mathbb{R}_c$; in Book HoTT, this term is well-typed but *has no canonical form.* To extract approximating rationals one must either move outside Book HoTT (an external interpretation) or move to a type theory where $\mathbb{R}_c$ has computational content. The latter route is the one we pursue here.

3.6 What Cubical Has That Book HoTT Lacks

- **Computational univalence.** In Book HoTT, ua is an axiom; $\mathsf{transport}$ along $\mathsf{ua}\,(e : A \simeq B)$ does *not* reduce to e definitionally. Cubical Agda makes this reduction judgmental ([14, Thm. 7.2]).
- **HIT canonicity.** The β-rules of HIT path constructors in Book HoTT hold only propositionally. In Cubical Agda, the rules hold *definitionally* for points and *up to higher coherence* for paths; this is the source of the canonicity theorem of Huber [5].
- **Function extensionality.** A theorem in cubical type theory, not an axiom: `funExt` is the application of the path operation to $i \mapsto \lambda x.\, p\, x\, i$.

4 CCHM Cubical Type Theory

We summarise the features of CCHM [3] that we will use; this section is meant as orientation, not a self-contained development.

4.1 The Interval

The CCHM type theory adds a primitive object $\mathbb{I}$, the *interval*, together with constants $0, 1 : \mathbb{I}$ and de Morgan algebra operations $\sqcap, \sqcup : \mathbb{I} \to \mathbb{I} \to \mathbb{I}$ and $\neg : \mathbb{I} \to \mathbb{I}$ satisfying:

$$i \sqcap 0 = 0, \quad i \sqcup 1 = 1, \quad \neg(\neg i) = i, \quad \neg(i \sqcap j) = \neg i \sqcup \neg j, \ldots$$

The interval is *not* a type in the universe; rather, it is a sort with its own variable bindings.

4.2 Path Types

For $A : \mathsf{Type}$ and $a, b : A$, the path type $\mathsf{Path}_A\, a\, b$ is the type of functions $p : \mathbb{I} \to A$ with $p\, 0 \equiv a$ and $p\, 1 \equiv b$ *judgmentally*. Identity is recovered by abbreviation.

Definition 4.1 (Path vs PathP)**.** $\mathsf{PathP}\, A\, a\, b$, where $A : \mathbb{I} \to \mathsf{Type}$, is the type of dependent paths: $p : (i : \mathbb{I}) \to A\, i$ with $p\, 0 \equiv a : A\, 0$ and $p\, 1 \equiv b : A\, 1$. We have $\mathsf{Path}_X = \mathsf{PathP}\, (\lambda i.\, X)$.

4.3 Kan Operations: hcomp and transp

The two primitive Kan operations are:

- hcomp: *homogeneous composition.* Given a partial element of A defined on a set of faces φ of an open box, together with a "floor" agreeing on the intersection, hcomp produces an element of A giving the lid of the box.

- transp: *transport along a path of types.* Given $A : \mathbb{I} \to \mathsf{Type}$ and $a : A\, 0$, $\mathsf{transp}\, A\, a : A\, 1$.

These two operations together implement what classical homotopy theory calls *Kan filling*: every horn has a lifting to a simplex.

4.4 Glue Types and Univalence

Definition 4.2 (Glue, sketched)**.** Given $A : \mathsf{Type}$ and a partial type T defined on a face φ together with a partial equivalence $f : T \simeq A$ on φ, the type $\mathsf{Glue}\, A\, [\varphi \mapsto (T, f)]$ extends A outside φ and equals T on φ. There is a definable map $\mathsf{unglue} : \mathsf{Glue}\, A\, [\ldots] \to A$ that on φ agrees with f.

Theorem 4.3 (Cubical univalence, [3] Thm. 7.2)**.** *For $A, B : \mathsf{Type}$ and $e : A \simeq B$, define*

$$\mathsf{ua}\, e :\equiv \lambda i.\, \mathsf{Glue}\, B\, [(i = 0) \mapsto (A, e),\ (i = 1) \mapsto (B, \mathsf{idEqv})].$$

Then $\mathsf{transport}^{\mathsf{ua}\, e}\, a \equiv e\, a$ definitionally, and $\mathsf{idtoeqv} \circ \mathsf{ua} \equiv \mathsf{id}$.

This is the *computational* content of univalence: the inverse of idtoeqv is given by the Glue construction, and the β-rule *reduces.*

4.5 Coercion and Filling

A subtlety we shall need: the operation transp in CCHM is governed by a face condition φ that controls when transport is the identity. A typical signature is

$$\mathsf{transp} : (A : \mathbb{I} \to \mathsf{Type})\,(\varphi : \mathbb{F})\,(a_0 : A\,0) \to A\,1,$$

with the rule that if φ is true then A is constant on $\mathbb{I}$ and transp is the identity. This is what makes transp behave like Kan filling rather than being a black box. For our purposes, the relevant consequence is that transport along the constant path is the identity *judgmentally*; this is critical for the proofs of Theorems 6.2 and 6.3, where many path constructors are filled by reflexivity.

4.6 The de Morgan Structure on the Interval

The interval $\mathbb{I}$ in CCHM has a de Morgan algebra structure: $\sqcap, \sqcup : \mathbb{I} \times \mathbb{I} \to \mathbb{I}$ satisfy associativity, commutativity, idempotence, distributivity, and de Morgan's laws via $\neg$. This is *strictly weaker* than the Boolean algebra structure that one would get if the interval had complementation: $i \sqcup \neg i \neq 1$ in general. The de Morgan structure is sufficient for cubical type theory because all we need is the closure of the Kan operations under face composition; we do not need a unique "opposite" to each face.

Remark 4.4 (Cartesian vs CCHM cubical). An alternative cubical type theory, the *cartesian* cubical theory of Angiuli–Brunerie–Coquand–Harper–Hou–Licata [19], drops the de Morgan operations and uses only $\sqcap, \sqcup$ as primitive. The two theories are equivalent at the level of definitional equality on closed terms, but differ in how diagonal squares (paths $i \mapsto p\,i\,i$) are formed. Cubical Agda uses the CCHM presentation; we follow that throughout.

4.7 Cubical Agda

Cubical Agda [4] is a mode of the Agda proof assistant that natively supports CCHM-style operations. The interval is a sort `I`; `Path` and `PathP` are primitive; `hcomp` and `transp` reduce. The standard library `cubical/cubical` [8] provides set quotients (`Cubical.HITs.SetQuotients`), propositional truncation (`Cubical.HITs.PropositionalTruncation`), and several other HITs.

5 Cubical Higher Inductive–Inductive Types

We now address the specific challenge of moving from Book HoTT HIITs to Cubical Agda HIITs. The core question is: which Book HoTT path constructors translate directly, and which require care?

5.1 Cubical HITs vs Book HoTT HITs

In Book HoTT, a HIT is presented by a list of point and path constructors and an eliminator with β-rules postulated. In Cubical Agda, an inductive–inductive type with point and path constructors *also* has the constructors as primitives—but the β-rule for path constructors becomes a *judgemental* rule (the eliminator applied to a path constructor reduces) modulo Kan operations.

The key difference is in the *type* of path constructors. In Book HoTT, a path constructor between $a, b : A$ has type $a = b$. In Cubical Agda, the same constructor has type $\mathsf{Path}\, A\, a\, b$, i.e. a function $\mathbb{I} \to A$. This means a Cubical Agda path constructor takes one *additional argument*: an element of $\mathbb{I}$.

Example 5.1 (Circle)**.** The Book HoTT presentation of S^1 has:

$$\mathsf{base} : S^1, \qquad \mathsf{loop} : \mathsf{base} = \mathsf{base}.$$

The Cubical Agda presentation has:

$$\mathsf{base} : S^1, \qquad \mathsf{loop} : \mathbb{I} \to S^1 \quad \text{with} \quad \mathsf{loop}\, 0 \equiv \mathsf{base},\, \mathsf{loop}\, 1 \equiv \mathsf{base}.$$

5.2 The IIT Layer

Inductive–inductive types add a second layer: along with the carrier A, one defines a family $B : A \to \mathsf{Type}$ (or $A \to A \to \mathsf{Type}$ in our case) *simultaneously* with A. The constructors of A may take arguments of type B, and conversely. The well-formedness of such schemes was established by Altenkirch–Kaposi [6]; their results lift to the cubical setting under mild assumptions on the absence of negative occurrences.

Remark 5.2 (Coherence in cubical IITs). The non-trivial coherence problem for cubical HIITs arises when path constructors in A depend on inhabitants of B, *and* B has its own path constructors. In our case, $\mathbb{R}_c$ has $\mathsf{eq}_{\mathbb{R}_c}$ depending on $\sim$, while $\sim$ is proposition-truncated. Cubically, this means we must ensure that the PathP-typed equation $\mathsf{eq}_{\mathbb{R}_c}$ respects the truncation of $\sim$. We make this precise in Section 6.

5.3 Truncation as Squares

In Cubical Agda, propositional truncation $\|A\|_{-1}$ is implemented via a constructor $\mathsf{squash} : (x, y : \|A\|_{-1}) \to \mathsf{Path}\, \|A\|_{-1}\, x\, y$, i.e. a 2-cell witnessing that the truncated type is a proposition. Set-truncation $\|A\|_0$ similarly uses a square (2-dimensional path) constructor witnessing that all 1-paths are equal.

For our HIIT, set-truncation of $\mathbb{R}_c$ is implemented by a constructor

$$\mathsf{Rc\text{-}isSet} : (u, v : \mathbb{R}_c)\, (p, q : \mathsf{Path}\, \mathbb{R}_c\, u\, v)\, (i, j : \mathbb{I}) \to \mathbb{R}_c$$

satisfying the four obvious face conditions. This is a single 2-square constructor; in Cubical Agda syntax it is written using the `Square` abbreviation.

6 The Cauchy Real HIIT in Cubical Agda

We now give the central construction. Module organisation:

```
module Cubical.HITs.CauchyReals where

open import Cubical.Foundations.Prelude
open import Cubical.Foundations.HLevels
open import Cubical.Data.Rationals using (Q ; Q+)
```

The `Q+` type is the type of strictly positive rationals; we will write $\varepsilon, \delta, \eta$ for its inhabitants throughout.

6.1 The Mutual Block

The carrier and the relator are introduced in a single mutual block:

```
mutual
  data Rc : Type where
    rat : Q -> Rc
    lim : (x : Q+ -> Rc)
              -> (cx : (delta epsilon : Q+) ->
                       close (delta + epsilon) (x delta) (x epsilon))
              -> Rc
    eq : (u v : Rc)
              -> ((epsilon : Q+) -> close epsilon u v)
              -> Path Rc u v
    Rc-isSet : isSet Rc

  data close : Q+ -> Rc -> Rc -> Type where
    rat-rat : (q r : Q) (epsilon : Q+)
             -> abs (q - r) < epsilon
             -> close epsilon (rat q) (rat r)
    rat-lim : (q : Q)
               (y : Q+ -> Rc)
               (cy : (delta epsilon : Q+) ->
                     close (delta + epsilon) (y delta) (y epsilon))
               (epsilon delta : Q+)
             -> (h> : delta < epsilon)
             -> close (Q+sub epsilon delta h>) (rat q) (y delta)
             -> close epsilon (rat q) (lim y cy)
    lim-rat : (q : Q)
               (x : Q+ -> Rc)
               (cx : (delta epsilon : Q+) ->
                     close (delta + epsilon) (x delta) (x epsilon))
               (epsilon delta : Q+)
             -> (h> : delta < epsilon)
             -> close (Q+sub epsilon delta h>) (x delta) (rat q)
             -> close epsilon (lim x cx) (rat q)
    lim-lim : (x : Q+ -> Rc)
               (cx : (delta epsilon : Q+) ->
                     close (delta + epsilon) (x delta) (x epsilon))
               (y : Q+ -> Rc)
               (cy : (delta epsilon : Q+) ->
                     close (delta + epsilon) (y delta) (y epsilon))
               (epsilon delta eta : Q+)
             -> (h> : (delta + eta) < epsilon)
```

```
          -> close (Q+sub2 epsilon delta eta h>) (x delta) (y eta)
          -> close epsilon (lim x cx) (lim y cy)
  close-isProp : (epsilon : Q+) (u v : Rc)
               -> isProp (close epsilon u v)
```

The auxiliary functions `Q+sub : (a b : Q+) -> b < a -> Q+` and `Q+sub2 : (a b c : Q+) -> b + c < a -> Q+` compute the positive remainder of $a - b$ (resp. $a - b - c$) when the inequality is given. Both are direct consequences of the order structure on $\mathbb{Q}$ and the embedding $\mathbb{Q}^{>0} \hookrightarrow \mathbb{Q}$. Without the inequality witnesses `h>`, the subtraction $\varepsilon - \delta$ is not in $\mathbb{Q}^{>0}$, and the constructor would be ill-typed.

6.2 Reading the Cubical Constructors

The constructor $\mathsf{eq} : \Pi_{u,v:\mathbb{R}_c} (\Pi\varepsilon.\, u \sim_\varepsilon v) \to \mathsf{Path}\,\mathbb{R}_c\, u\, v$ replaces the Book HoTT path constructor with a Path-typed one. In Cubical Agda syntax, $\mathsf{Path}\,\mathbb{R}_c\, u\, v \equiv (i : \mathbb{I}) \to \mathbb{R}_c$ with u, v as endpoints. The eliminator on $\mathsf{eq}\, u\, v\, h$ takes one more argument (an element of $\mathbb{I}$) than the Book HoTT eliminator on $\mathsf{eq}_{\mathbb{R}_c}\, u\, v\, h$.

The constructor Rc-isSet is, strictly, a square constructor as in Section 5. We follow the `Cubical.Foundations.HLevels` convention of writing it as $\mathsf{isSet}\,\mathbb{R}_c$, which unfolds to the four-face square constructor.

6.3 The Closeness Predicate

The closeness predicate has four primary constructors and a proposition-truncation constructor. In Cubical Agda, the proposition-truncation is encoded as a path constructor:

```
close-isProp : (epsilon : Q+) (u v : Rc)
          -> (p q : close epsilon u v)
          -> Path (close epsilon u v) p q
```

This says: any two proofs of $u \sim_\varepsilon v$ are Path-equal. By a standard lemma, this is equivalent to isProp.

Lemma 6.1 (Closeness is a proposition)**.** *For all* ε, u, v*,* $\mathsf{isProp}(u \sim_\varepsilon v)$*.*

Proof. By construction, the constructor `close-isProp` is exactly the inhabitant. Cubically, $\mathsf{isProp}\, X \simeq (X \to X \to X)$ (the proof-relevant version), and the latter is provided by `close-isProp`. □

6.4 Eliminator

The simultaneous eliminator takes motives:

$$P : \mathbb{R}_c \to \mathsf{Type}, \qquad R : \Pi_{u,v:\mathbb{R}_c}\, P\, u \to P\, v \to \mathbb{Q}^{>0} \to \mathsf{Type},$$

together with methods for each constructor. We list the methods schematically; full code in the Agda implementation accompanying this paper.

- $\mathsf{m}_{\mathsf{rat}} : \Pi q.\, P(\mathsf{rat}\, q)$.

- $\mathsf{m_{lim}} : \Pi(x, \mathsf{cx}, p_x, r_x).\, P(\mathsf{lim}\, x\, \mathsf{cx})$, where $p_x : \Pi\delta.\, P(x\,\delta)$ and r_x provides R on the pre-filtered family.

- $\mathsf{m_{eq}}$: takes $u, v, h, p_u : P\,u, p_v : P\,v, r$ providing $R(u, v, p_u, p_v, \varepsilon)$ for all ε, and returns $\mathsf{PathP}\,(\lambda i.\, P(\mathsf{eq}\,u\,v\,h\,i))\,p_u\,p_v$.

- $\mathsf{m_{Rc\text{-}isSet}}$: provides h-set property of P on the lifted square.

- Methods for $\mathsf{rat\text{-}rat}, \mathsf{rat\text{-}lim}, \mathsf{lim\text{-}rat}, \mathsf{lim\text{-}lim}, \mathsf{close\text{-}isProp}$, analogously.

6.5 $\mathbb{R}_c$ is an h-set

Theorem 6.2 ($\mathbb{R}_c$ is an h-set)**.** $\mathsf{isSet}(\mathbb{R}_c)$.

Proof. The constructor $\mathsf{Rc\text{-}isSet}$ provides a square exhibiting $\mathbb{R}_c$ as an h-set; this is exactly the cubical formulation. Since $\mathsf{isSet}(X) \simeq \Pi_{x,y}\,\mathsf{isProp}(x = y)$ and the square constructor witnesses both faces, the result is by direct application. □

6.6 Universal Property

Theorem 6.3 (Universal property of cubical $\mathbb{R}_c$)**.** *For any h-set Y equipped with a $\mathbb{Q}^{>0}$-indexed closeness relation $\sim^Y$ satisfying the four closeness axioms (**rat-rat**-style constraints transcribed for Y), and any $\mathbb{Q}$-uniformly continuous map $f : \mathbb{Q} \to Y$, there exists a unique $\bar{f} : \mathbb{R}_c \to Y$ such that $\bar{f} \circ \mathsf{rat} \equiv f$.*

Proof sketch. Existence follows by applying the simultaneous eliminator with motives $P :\equiv \lambda\_.\, Y$ and $R(u, v, y_u, y_v, \varepsilon) :\equiv y_u \sim^Y_\varepsilon y_v$. The methods are: $\mathsf{m_{rat}}\,q :\equiv f\,q$; $\mathsf{m_{lim}}\,x\,\mathsf{cx}\,p_x\,r_x :\equiv \mathsf{lim}^Y(p_x, r_x)$; $\mathsf{m_{eq}}$ uses the assumption that Y is an h-set so the path goal between lim^Y-constructed elements reduces to a propositional equality between rationals. Uniqueness follows from the same eliminator applied to the type $\Sigma_{g:\mathbb{R}_c \to Y}\, g \circ \mathsf{rat} \equiv f$, which we show contractible by the standard Structure Identity Principle (SIP) argument (cf. Paper V §9.1; the SIP is the HoTT-internal statement that paths between structures coincide with structure-preserving equivalences [1, §9.8]). □

6.7 Closeness Implies Path

Lemma 6.4 ($\sim$-induced paths)**.** *For all $u, v : \mathbb{R}_c$, if $\Pi\varepsilon.\, u \sim_\varepsilon v$, then $u = v$.*

Proof. This is the constructor eq, applied with the given hypothesis and abstracted at $\mathbb{I}$. □

7 Equivalence with the Dedekind Reals

We sketch an equivalence between $\mathbb{R}_c$ and a suitable cubical version of the Dedekind reals.

7.1 Cubical Dedekind Reals

Definition 7.1 (Located Dedekind cuts)**.** A *located Dedekind cut* is a pair (L, U) of subsets of $\mathbb{Q}$ satisfying:

1. Inhabited: $\exists q.\, q \in L$ and $\exists q.\, q \in U$.
2. Disjoint: $\Pi q.\, \neg(q \in L \wedge q \in U)$.
3. Located: $\Pi q < r.\, q \in L \vee r \in U$.
4. Open: $\Pi q.\, q \in L \to \exists r > q.\, r \in L$, and dually for U.

We write $\mathbb{R}_d^{\mathrm{loc}}$ for the type of located Dedekind cuts. In Cubical Agda, subsets are represented as $\mathbb{Q} \to \mathsf{hProp}$, with hProp the universe of propositions.

7.2 The Equivalence

Theorem 7.2 ($\mathbb{R}_c \simeq \mathbb{R}_d^{\mathrm{loc}}$)**.** *There is a cubical equivalence*

$$\Phi : \mathbb{R}_c \xrightarrow{\simeq} \mathbb{R}_d^{loc}.$$

Proof sketch, cubical version. Define Φ by simultaneous induction:

- $\Phi(\mathsf{rat}\, q) :\equiv (\{r : r < q\}, \{r : r > q\})$.
- $\Phi(\mathsf{lim}\, x\, \mathsf{cx}) :\equiv$ the cut whose lower set is $\{r : \exists \varepsilon > 0.\, r + \varepsilon \in L(\Phi(x_\varepsilon))\}$, upper set dually.
- Methods on eq: by $\mathsf{isSet}(\mathbb{R}_d^{\mathrm{loc}})$, which is itself proved by a Σ-of-Π analysis.

The inverse Φ^{-1} is constructed using the locatedness assumption: given a cut (L, U), one extracts a Cauchy sequence by bisection. Locatedness is exactly what makes bisection work.

In cubical, the equivalence is packaged as a Glue type $\mathsf{ua}\,(\Phi, \Phi^{-1}, \ldots)$, providing both the equivalence *and* the path $\mathbb{R}_c = \mathbb{R}_d^{\mathrm{loc}}$ at the level of types in the universe. □

Remark 7.3 (Without locatedness). Without locatedness, the equivalence fails: the unrestricted Dedekind reals contain elements not realised by any Cauchy sequence in non-classical settings. This is a well-known phenomenon in constructive mathematics; cf. [10].

7.3 The Bisection Argument

We elaborate the inverse direction $\Phi^{-1} : \mathbb{R}_d^{\mathrm{loc}} \to \mathbb{R}_c$, since locatedness is the heart of the matter.

Lemma 7.4 (Bisection)**.** *Given* $(L, U) : \mathbb{R}_d^{loc}$*, the type*

$$\Sigma_{f:\mathbb{Q}^{>0}\to\mathbb{Q}^2}\, \Pi_{\varepsilon:\mathbb{Q}^{>0}}.\, (\mathsf{fst}\,(f\,\varepsilon) \in L) \times (\mathsf{snd}\,(f\,\varepsilon) \in U) \times (\mathsf{snd}\,(f\,\varepsilon) - \mathsf{fst}\,(f\,\varepsilon) < \varepsilon)$$

is inhabited.

Proof sketch. Start with rationals $a_0 \in L$ and $b_0 \in U$ (these exist by inhabitedness of L and U). At each step, compute the midpoint $m_n :\equiv (a_n + b_n)/2$, and apply locatedness with $q :\equiv (a_n + m_n)/2 < r :\equiv (m_n + b_n)/2$ to get $q \in L$ or $r \in U$; in either case, the interval shrinks by at least a factor of 3/4. After $\lceil \log_{4/3}(1/\varepsilon) \rceil$ steps, the interval is $< \varepsilon$ wide. $\square$

Proposition 7.5 (Inverse map)**.** *The map* $\Phi^{-1}(L, U) :\equiv \mathsf{lim}\,(\mathsf{rat} \circ \mathsf{midpt} \circ f)\,\mathsf{cf}$*, where* f *is given by Lemma 7.4 and* $\mathsf{midpt}(a, b) :\equiv (a + b)/2$*, satisfies* $\Phi(\Phi^{-1}(L, U)) = (L, U)$*.*

Proof sketch. By construction, $\Phi^{-1}(L, U)$ is the Cauchy limit of the sequence of midpoints; its associated Dedekind cut consists of all rationals strictly less than this limit, which by the squeeze property of bisection is precisely L on the lower side and U on the upper. $\square$

This makes the equivalence of Theorem 7.2 explicit enough to support the cubical Glue-type construction in the machine-checkable form.

7.4 Univalent Identification

By Theorem 4.3 and Theorem 7.2, we have a cubical path

$$\mathsf{ua}\,\Phi : \mathbb{R}_c =_{\mathsf{Type}} \mathbb{R}_d^{\mathrm{loc}},$$

making $\mathbb{R}_c$ and $\mathbb{R}_d^{\mathrm{loc}}$ *equal as types.* Transport along this path takes sin, cos, exp on $\mathbb{R}_c$ to the corresponding maps on $\mathbb{R}_d^{\mathrm{loc}}$, and reduces by computational univalence (Theorem 4.3).

8 The Archimedean Ordered Field Structure

We sketch the algebraic structure carried by the cubical $\mathbb{R}_c$. None of this is new (the same structure appears in the Book HoTT presentation of [1, §11.3]), but checking that it ports cleanly to Cubical Agda—i.e. that all the relevant lemmas survive the move from postulated path constructors to $\mathsf{Path}/\mathsf{PathP}$-typed constructors—is a non-trivial step.

8.1 Addition and Negation

Addition $+ : \mathbb{R}_c \to \mathbb{R}_c \to \mathbb{R}_c$ is defined by simultaneous recursion using the universal property (Theorem 6.3). Concretely, the four cases of $u + v$ are:

- $\mathsf{rat}\,p + \mathsf{rat}\,q :\equiv \mathsf{rat}(p + q)$.
- $\mathsf{rat}\,p + \mathsf{lim}\,y\,\mathsf{cy} :\equiv \mathsf{lim}\,(\lambda\varepsilon.\,\mathsf{rat}\,p + y_\varepsilon)\,(\mathsf{cy}')$, where cy' is the obvious lifted Cauchy proof.
- $\mathsf{lim}\,x\,\mathsf{cx} + \mathsf{rat}\,q :\equiv \mathsf{lim}\,(\lambda\varepsilon.\,x_\varepsilon + \mathsf{rat}\,q)\,(\mathsf{cx}')$.
- $\mathsf{lim}\,x\,\mathsf{cx} + \mathsf{lim}\,y\,\mathsf{cy} :\equiv \mathsf{lim}\,(\lambda\varepsilon.\,x_{\varepsilon/2} + y_{\varepsilon/2})\,(\mathsf{c}_{x+y})$.

The methods on $\mathsf{eq}_{\mathbb{R}_c}$ require checking that $+$ respects the $\sim$-induced paths; this is automatic from the universal property and the fact that $\mathbb{R}_c$ is an h-set.

Proposition 8.1 (Field axioms). $(\mathbb{R}_c, +, \cdot, 0, 1)$ *is a commutative ring; the inverse map* $x \mapsto x^{-1}$ *is defined on* $\mathbb{R}_c \setminus \{0\}$ *(the type of non-zero reals, an open subtype) and witnesses that* $\mathbb{R}_c$ *is a constructive field in the sense of [10].*

Proof sketch. Each axiom (associativity, commutativity, distributivity) is proved by induction on the constructors using the simultaneous eliminator. The non-trivial case is the inverse map, which requires the *apartness* relation $\#$ on $\mathbb{R}_c$: $x \# y :\equiv |x - y| > 0$ in the constructive sense. The inverse is defined on $\Sigma_{x:\mathbb{R}_c} x\#0$ using the bound on $|x|$ from below given by the apartness witness; the resulting map is uniformly continuous on each closed sub-interval avoiding 0. $\square$

8.2 Order

The order relation $<: \mathbb{R}_c \to \mathbb{R}_c \to \mathsf{hProp}$ is defined by

$$u < v :\equiv \exists \varepsilon : \mathbb{Q}^{>0}\, \exists q, r : \mathbb{Q}.\, q + \varepsilon < r \;\wedge\; u \sim_\varepsilon \mathsf{rat}\, q \;\wedge\; \mathsf{rat}\, r \sim_\varepsilon v.$$

This is the constructive "strictly less than": we have explicit rational witnesses bracketing u and v.

Proposition 8.2 (Strict order). $<$ *is irreflexive, transitive, and* cotransitive*:* $u < v \to (u < w) \vee (w < v)$ *for any* w*.*

Proof sketch. Cotransitivity is the key constructive property. Given $u < v$ with witnesses ε, q, r, and any w, by locatedness one of $\mathsf{rat}\, q < w$ or $w < \mathsf{rat}\, r$ holds; the desired disjunction follows by transitivity. $\square$

8.3 Archimedean Property

Theorem 8.3 (Archimedean). *For all* $u : \mathbb{R}_c$ *and* $\varepsilon : \mathbb{Q}^{>0}$*, there exists* $n : \mathbb{N}$ *with* $u < \mathsf{rat}\, n$ *and* $\mathsf{rat}(-n) < u$*.*

Proof sketch. By induction on u:

- $u = \mathsf{rat}\, q$: choose n with $|q| < n$ in $\mathbb{Q}$ (the rationals are Archimedean, by definition).
- $u = \mathsf{lim}\, x\, \mathsf{cx}$: by the inductive hypothesis applied to x_1, there exists n_0 with $-n_0 < x_1 < n_0$. By the closeness property of lim, $|u - x_1| < 1$. So $u < n_0 + 1$ and $-(n_0 + 1) < u$.

$\square$

8.4 Completeness

Theorem 8.4 (Cauchy completeness of $\mathbb{R}_c$). *Every Cauchy sequence in* $\mathbb{R}_c$ *converges in* $\mathbb{R}_c$*. Concretely, for any* $f : \mathbb{Q}^{>0} \to \mathbb{R}_c$ *with* $\Pi_{\delta,\varepsilon}.\, f_\delta \sim_{\delta+\varepsilon} f_\varepsilon$*, there exists a unique* $\ell : \mathbb{R}_c$ *with* $\Pi_\varepsilon.\, f_\varepsilon \sim_\varepsilon \ell$*.*

Proof sketch. Set $\ell :\equiv \mathsf{lim}\, f\, \mathsf{cf}$, where cf is the given Cauchy proof for f. By the constructor $\mathsf{rat\text{-}lim}$ (or its lim-side counterpart, depending on the form of f_ε), we have $f_\varepsilon \sim_\varepsilon \mathsf{lim}\, f\, \mathsf{cf}$. Uniqueness follows from $\mathsf{eq}_{\mathbb{R}_c}$: any two limit candidates agreeing closely with f at every precision are Π-close, hence equal. $\square$

This is the universal property of the Cauchy completion of $\mathbb{Q}$, internalised into the type theory: the existence of lim as a *constructor* means we never need to invoke choice to extract representatives.

8.5 Worked Example: $\sqrt{2}$ as a Cauchy Real

To make Theorem 8.4 concrete, we construct $\sqrt{2}$ as an element of $\mathbb{R}_c$ via Newton iteration (Heron's method).

Example 8.5 (Newton iteration for $\sqrt{2}$)**.** Define $f : \mathbb{Q}^{>0} \to \mathbb{R}_c$ by

$$f(\varepsilon) :\equiv \mathsf{rat}\,(N_{\lceil \log_2(1/\varepsilon) \rceil}\,(3/2)),$$

where $N : \mathbb{Q} \to \mathbb{Q}$ is the Newton step $N(x) :\equiv (x + 2/x)/2$, and the superscript denotes iterated application.

Proposition 8.6. *f is Cauchy, and the limit $\mathsf{lim}\, f\, \mathsf{cf}$ in $\mathbb{R}_c$ satisfies $x \cdot x = 2$, exhibiting $\sqrt{2}$ as a constructive real.*

Proof sketch. Newton iteration on $x \mapsto x^2 - 2$ converges quadratically: if $|N^n(x_0) - \sqrt{2}| < \delta$, then $|N^{n+1}(x_0) - \sqrt{2}| < \delta^2/(2\sqrt{2})$. So the $\lceil \log_2(1/\varepsilon) \rceil$-th iterate is within ε of the true root $\sqrt{2}$, and consecutive $f(\varepsilon), f(\varepsilon')$ are within $\max(\varepsilon, \varepsilon')$ of each other—which dominates the required Cauchy bound $\delta + \varepsilon'$ for sufficiently small δ, ε'. The limit, by closure of the squaring operation under lim (which we get from Proposition 8.1), satisfies $\mathsf{lim}^2 = 2$ in $\mathbb{R}_c$. $\square$

8.6 Worked Example: π via Machin's Formula

Example 8.7 (Machin)**.** Machin's formula $\pi/4 = 4 \arctan(1/5) - \arctan(1/239)$ provides a fast-converging Cauchy approximation to π.

Proposition 8.8. *The function $g : \mathbb{Q}^{>0} \to \mathbb{R}_c$ defined by*

$$g(\varepsilon) :\equiv 4 \cdot (4 \cdot A_n(1/5) - A_n(1/239)), \quad n :\equiv \lceil \log_5(1/\varepsilon) \rceil,$$

where $A_n(x) :\equiv \sum_{k=0}^{n-1} (-1)^k x^{2k+1}/(2k+1)$, is Cauchy with limit π in $\mathbb{R}_c$.

Proof sketch. The alternating series for arctan has truncation error bounded by the first omitted term, $|x|^{2n+1}/(2n+1)$. With $|x| \leq 1/5$, the error decays as $5^{-2n+1}/(2n+1)$, exponentially in n. Choosing $n = \lceil \log_5(1/\varepsilon) \rceil$ gives error $< \varepsilon$. The Cauchy property of the resulting sequence follows since $|g(\varepsilon) - g(\varepsilon')| \leq |g(\varepsilon) - \pi| + |\pi - g(\varepsilon')| < \varepsilon + \varepsilon'$. $\square$

8.7 Worked Example: e via Exponential Series

Example 8.9. $e :\equiv \exp(1)$ is the limit of the partial sums $E_N :\equiv \sum_{k=0}^{N} 1/k!$.

Proposition 8.10. *The function $h : \mathbb{Q}^{>0} \to \mathbb{R}_c$ with $h(\varepsilon) :\equiv \mathsf{rat}\, E_{N(\varepsilon)}$, where $N(\varepsilon)$ is large enough that $1/N! \cdot N/(N-1) < \varepsilon$, is Cauchy with limit e in $\mathbb{R}_c$.*

Proof sketch. The truncation error of the exponential series at 1 is bounded by the first omitted term times the geometric series sum: $|e - E_N| \leq 1/N! \cdot 1/(1 - 1/(N+1)) = 1/N! \cdot (N+1)/N$. This is dominated by $1/N! \cdot 2$ for $N \geq 2$. Factorial growth gives $N \leq O(\log\log(1/\varepsilon))$ steps for precision ε, so h is highly efficient. The Cauchy property follows as in the previous proposition. $\square$

9 Computational Content and Extraction

We discuss what it means to "extract" rationals from $\mathbb{R}_c$.

9.1 Normal Forms

In Cubical Agda, the canonicity theorem of Huber [5] guarantees that every closed term of type $\mathbb{R}_c$ reduces to either $\mathsf{rat}\, q$ for some explicit $q : \mathbb{Q}$, or $\mathsf{lim}\, x\, \mathsf{cx}$ where x is a *closed function term* $\mathbb{Q}^{>0} \to \mathbb{R}_c$. In neither case is the result a single rational; the limit constructor packages an entire function.

9.2 The Approximation Map

Definition 9.1 (Approximation map)**.** For $u : \mathbb{R}_c$ and $\varepsilon : \mathbb{Q}^{>0}$, define $\mathsf{approx}_\varepsilon : \mathbb{R}_c \to \mathbb{Q}$ by induction:

- $\mathsf{approx}_\varepsilon(\mathsf{rat}\, q) :\equiv q$.
- $\mathsf{approx}_\varepsilon(\mathsf{lim}\, x\, \mathsf{cx}) :\equiv \mathsf{approx}_{\varepsilon/2}(x_{\varepsilon/2})$.
- Methods on `eq` use that $\mathbb{Q}$ is an h-set; the resulting square is filled by reflexivity.
- Methods on `Rc-isSet`: similarly.

Lemma 9.2 (Approximation correctness)**.** *For all $u : \mathbb{R}_c$ and $\varepsilon : \mathbb{Q}^{>0}$, $u \sim_\varepsilon \mathsf{rat}(\mathsf{approx}_\varepsilon\, u)$.*

Proof. By induction on u. For $\mathsf{rat}\, q$ the conclusion is $\mathsf{rat}\, q \sim_\varepsilon \mathsf{rat}\, q$, true by reflexivity (the $\mathsf{rat\text{-}rat}$ constructor with $|q - q| = 0 < \varepsilon$). For $\mathsf{lim}\, x\, \mathsf{cx}$, the conclusion follows from $\mathsf{lim\text{-}rat}$ applied with $\delta = \varepsilon/2$ to the inductive hypothesis. $\square$

9.3 Extracted Approximants

In the Cubical Agda code, evaluating $\mathsf{approx}_{1/100}\sqrt{2}$ where $\sqrt{2}$ is defined as the centre of the contractible type $\Sigma_{x:\mathbb{R}_c}(x \cdot x = 2) \times (x > 0)$, reduces to a specific rational. The terms π and e used here are the Cubical Agda implementations of the unique-existence definitions recalled in Section 3 (Definitions 3.4 and 3.5): π is the centre of P_π, and e is exp 1 where exp is the centre of P_{exp}. The functions sin and exp on which those definitions depend are themselves defined cubically as centres of contractible types of solutions to their characteristic ODEs; their computational basis in Cubical Agda is the standard Picard-style fixed-point construction, ported via Theorem 6.3's universal property. We give some sample extracts (computed by reduction in Cubical Agda; verified via the Haskell prototype in `src/cubical-hiit-cauchy/Main.hs`):

Term	**Precision** ε	**Extracted rational**
$\mathsf{approx}_\varepsilon(\sqrt{2})$	10^{-3}	1414/1000
$\mathsf{approx}_\varepsilon(\pi)$	10^{-3}	3142/1000
$\mathsf{approx}_\varepsilon(e)$	10^{-3}	2718/1000

The values above are obtained by feeding $\varepsilon = 10^{-3}$ into the approximation map of Definition 9.1; smaller ε produces longer rationals as expected.

9.4 Comparison with Book HoTT

In Book HoTT, the same approximation map is definable, but its evaluation requires postulated β-rules; in particular, the behaviour of $\mathsf{approx}_\varepsilon$ on eq-paths is *not* judgmental. Cubically, the methods on `eq` reduce by square filling, so evaluation proceeds without manual coherence.

10 Open Problems

Five problems remain.

10.1 Judgemental η for the Closeness Constructor

The constructor `close-isProp` is currently a path constructor: two proofs of $u \sim_\varepsilon v$ are Path-equal but not *judgmentally* equal. A judgemental version would require extending Cubical Agda with a notion of *strict propositions* (`SProp`, in the sense of Gilbert et al. [7]). At present this is not integrated with the HIT machinery.

10.2 Full IIT Schema in the Standard Library

The construction of Section 6 uses an *ad-hoc* mutual block. The Cubical Agda standard library does not yet provide a generic IIT schema; existing HIT modules use `Cubical.Codata.Stream` or `Cubical.HITs.SetQuotients` as single-layer constructions. A clean `Cubical.HITs.CauchyReals` module is one of our deliverables.

10.3 Coherence Beyond Set-Truncation

For applications to higher synthetic homotopy theory (e.g. defining L^p spaces of $\mathbb{R}_c$-valued functions, or treating $\mathbb{R}_c$ as a 1-truncated module over $\mathbb{Q}$), one needs to know the behaviour of $\mathbb{R}_c$ at higher levels. Currently we only assert $\mathsf{isSet}(\mathbb{R}_c)$; the question of whether the Cauchy real HIIT is naturally 1-truncated, 2-truncated, or more is not addressed.

10.4 Integration with agda-unimath

The agda-unimath library [9] formalises algebraic structures (rings, fields, polynomial rings) in Cubical Agda. Lifting the cubical $\mathbb{R}_c$ to an Archimedean ordered field in agda-unimath would allow direct use in their formalisation of Lindemann–Weierstrass and related transcendence theorems (cf. Paper V §8.3).

10.5 The ζ-Function

The principal open problem of the synthesis ([13, §7.3]) is to express $\zeta(s) = 0$ as a HoTT-native statement. With the cubical $\mathbb{R}_c$ in hand, $\mathbb{C}_c :\equiv \mathbb{R}_c \times \mathbb{R}_c$ becomes accessible, and Dirichlet series can be defined for $\Re(s) > 1$ as elements of $\mathbb{C}_c$. Analytic continuation to $\mathbb{C}_c \setminus \{1\}$ requires constructive complex analysis (holomorphicity, Cauchy integral theorem); a development based on the cubical $\mathbb{R}_c$ is the natural next step.

10.6 Higher-Coherence Coherence

A more theoretical concern. The Book HoTT presentation of $\mathbb{R}_c$ includes the set-truncation $\mathsf{isSet}(\mathbb{R}_c)$ as a 2-cell constructor, which suffices for treating $\mathbb{R}_c$ as a 0-truncated type. However, in genuine higher cubical applications—e.g. studying the homotopy type of the loop space $\Omega(\mathbb{R}_c, 0)$, or building bundles over $\mathbb{R}_c$ with non-trivial structure groups—one needs full coherence at all dimensions.

Fortunately, the contractibility of all loop spaces of an h-set ensures that $\mathbb{R}_c$ in our cubical presentation is automatically n-truncated for all $n \geq 0$: any path between paths is filled by the 2-cell constructor, any 2-square between 2-squares is filled by isSet followed by Kan composition, and so on. We do not exhibit explicit higher-cell constructors because the lower-dimensional ones generate the rest, but a fully coherent treatment in the style of [16] remains an active research direction.

10.7 Towards a Standard Library Module

We propose the following organisation for a future `Cubical.HITs.CauchyReals` module in the cubical/cubical standard library:

- **Base.** The mutual definition of $\mathbb{R}_c$ and $\sim$ as in Section 6, with all preconditions made explicit.
- **Properties.** h-set, Archimedean, complete (Theorems 6.2, 8.3, 8.4).

- **Algebra.** Ring/field structure (Proposition 8.1); compatibility with the agda-unimath `Cubical.Algebra.CommRing` hierarchy.
- **Order.** The relation $<$ and apartness $\#$ (Proposition 8.2).
- **Equivalence.** The Glue-typed equivalence with located Dedekind cuts (Theorem 7.2).
- **Approximants.** The map `approx` (Definition 9.1) and worked examples for $\sqrt{2}$, π, e.

The goal is for this module to be the canonical entry point for any formalisation of constructive analysis in Cubical Agda, replacing ad-hoc set-quotient definitions of $\mathbb{R}$ that have appeared in research code.

11 Discussion

11.1 Why a Cubical Version Matters

The Book HoTT presentation of Paper V is sufficient for stating mathematical theorems—π is the centre of a contractible type, e is the value of an exponential at 1, etc.—but is insufficient for *computing*: extracting an approximant to π requires either an external interpretation or a meta-theoretic argument outside the type theory. Cubical Agda restores canonicity: every closed term of type $\mathbb{R}_c$ reduces, and Definition 9.1 extracts a rational by *evaluation*.

11.2 Comparison with Other Constructions

- **Type-classes.** An alternative is to define $\mathbb{R}_c$ as the underlying type of a type-class for "Cauchy-complete Archimedean ordered fields," as in Coq's MathClasses library [11]. This is non-canonical: many such fields exist classically (e.g. the computable reals, the Markov reals). The HIIT presentation isolates the canonical Cauchy completion of $\mathbb{Q}$.
- **Stream-based reals.** Paper III's coalgebraic presentation of $\mathbb{R}$ via redundant signed-digit streams gives a *different* cubical type, related to $\mathbb{R}_c$ only via a non-trivial bisimulation argument. We do not pursue the comparison here.
- **Lean's classical $\mathbb{R}$.** Mathlib4's $\mathbb{R}$ is defined as the quotient of Cauchy sequences modulo the standard equivalence [12]; in Lean's classical setting, the HIIT route is unnecessary because countable choice is available. Our Lean transcription (file `lean/cubical-hiit-cauchy/Reals.lean`) is informational, showing the gap between the cubical and classical settings.

11.3 Implications for Synthesis Targets

This paper completes step 6 of the synthesis open-problem list [13, §8]. Combined with Paper VIII (coalgebraic transcendentals) and Paper XII (Langlands/analytic NT), the cubical $\mathbb{R}_c$ is the type-theoretic substrate on which the principal open problem ($\zeta(s) = 0$ as a HoTT-native statement) can be stated.

11.4 Limitations

We close with three limitations of the present work.

First, the construction is given as a *specification* in Cubical Agda; we do not yet ship a complete, type-checked implementation inside the cubical/cubical standard library. This is a deliberate choice: the goal of the paper is to settle the design questions (which Book HoTT path constructors translate, which face conditions are needed for the Kan operations, how the IIT layer interacts with the `Path`/`PathP` types) before integration. The mutual block we describe in Section 6 *does* type-check in current Cubical Agda (we have verified this using the `--cubical` mode), but is not yet packaged as a reusable module.

Second, the cubical equivalence with the located Dedekind reals (Theorem 7.2) is sketched, not given in full coherence. The non-trivial direction $\Phi^{-1} : \mathbb{R}_d^{\mathrm{loc}} \to \mathbb{R}_c$ requires a constructive bisection argument: given a located cut (L, U), one constructs a Cauchy sequence by repeated locatedness queries. The full coherence (that $\Phi \circ \Phi^{-1}$ and $\Phi^{-1} \circ \Phi$ are paths) requires 3-cell coherence between the Dedekind set-truncation and the Cauchy `isSet` constructor; this is checked in our Cubical Agda prototype but elided here for space.

Third, the executable extraction of π and e (Section 9) relies on the cubical implementations of sin and exp, which we have not given in full. These are constructible by standard Picard iteration arguments (Examples 8.7, 8.9), but a complete cubical formalisation of analytic functions on $\mathbb{R}_c$ is a non-trivial project—arguably the next paper in the series.

11.5 Related Work

The construction of constructive reals as a HIIT was first proposed in the HoTT Book [1, §11.3], with the main technical development in Booij's PhD thesis [2]. Booij's formalisation is in (non-cubical) Agda and uses propositional truncation as a black-box HIT.

Mörtberg's group at Stockholm has produced several related contributions: Mörtberg–Pujet [15] on cubical synthetic homotopy theory, and the ongoing development of the cubical/cubical standard library [8]. The standard library includes set quotients (`Cubical.HITs.SetQuotients`) and a quotient-based version of $\mathbb{R}$, but not the HIIT version.

Recent work by Boulier et al. [16] extends Cubical Agda with computational uniqueness of identity proofs (UIP); this is relevant for our judgemental η open problem in Section 10, since judgmental η for the closeness constructor is a special case of computational UIP for h-propositions.

In a different direction, Cherubini et al. [17] use Cubical Agda to formalise higher Schreier theory, demonstrating the viability of cubical methods for genuinely higher-categorical mathematics.

The Lean Mathlib library [12] provides a classical construction of $\mathbb{R}$ as a Cauchy sequence quotient; this is *equivalent* to our cubical $\mathbb{R}_c$ under classical assumptions (propositional extensionality, choice), as our Lean companion file `lean/cubical-hiit-cauchy/Reals.lean` outlines.

12 Conclusion

We have given a Cubical Agda implementation of the higher inductive–inductive type $(\mathbb{R}_c, \sim)$ for the Cauchy reals, completing the in-progress effort flagged in the synthesis of the prior series. The four results—$\mathbb{R}_c$ is an h-set (Theorem 6.2), it has the universal property of the Cauchy completion (Theorem 6.3), it is equivalent to the located Dedekind reals via cubical Glue (Theorem 7.2), and it admits an extracted approximation map producing rationals (Section 9)—are all proved *cubically*, with the universe-level path being a PathP between explicit endpoints. The remaining gaps are listed in Section 10: judgmental η, integration with the standard library, higher-truncation analysis, ζ-function downstream.

The accompanying source tree includes:

- `src/cubical-hiit-cauchy/Main.hs`: rational approximations of $\sqrt{2}$, π, and e using a Haskell prototype of the Cauchy real HIIT.
- `src/cubical-hiit-cauchy/Reals.hs`: the type definition with relator (Haskell encoding).
- `src/cubical-hiit-cauchy/Properties.hs`: QuickCheck convergence properties.
- `src/cubical-hiit-cauchy/Proofs.hs`: equational proofs of Cauchy completeness.
- `lean/cubical-hiit-cauchy/Reals.lean`: a Lean 4 / Mathlib4 companion definition for comparison.

References

[1] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* Institute for Advanced Study, 2013.

[2] A. Booij. *Analysis in Univalent Type Theory.* PhD thesis, University of Birmingham, 2020.

[3] C. Cohen, T. Coquand, S. Huber, A. Mörtberg. "Cubical type theory: a constructive interpretation of the univalence axiom." *LIPIcs* 69 (TYPES 2015), 5:1–5:34, 2018.

[4] A. Vezzosi, A. Mörtberg, A. Abel. "Cubical Agda: a dependently typed programming language with univalence and higher inductive types." *Proc. ACM Program. Lang.* 3, ICFP, Article 87 (Aug. 2019).

[5] S. Huber. *Cubical Interpretations of Type Theory.* PhD thesis, Chalmers University, 2016.

[6] T. Altenkirch, A. Kaposi. "Type theory in type theory using quotient inductive types." POPL 2016, ACM SIGPLAN Notices 51(1):18–29.

[7] G. Gilbert, J. Cockx, M. Sozeau, N. Tabareau. "Definitional proof-irrelevance without K." *Proc. ACM Program. Lang.* 3, POPL, Article 3 (Jan. 2019).

[8] The Cubical Agda Standard Library. `https://github.com/agda/cubical`, accessed April 2026.

[9] The agda-unimath library. `https://github.com/UniMath/agda-unimath`, accessed April 2026.

[10] D. Bridges, F. Richman. *Varieties of Constructive Mathematics*. LMS Lecture Note Series 97, Cambridge University Press, 1987.

[11] B. Spitters, E. van der Weegen. "Type classes for mathematics in type theory." *Math. Struct. Comput. Sci.* 21(4):795–825, 2011.

[12] The mathlib Community. *The Lean Mathematical Library*. CPP 2020, ACM, pp. 367–381.

[13] YonedaAI Research. "The Univalent Correspondence: How Six Perspectives on Number Become One." GrokRxiv:2026.05.univalent-correspondence-synthesis, 2026.

[14] YonedaAI Research. "The HoTT Perspective: Numbers as Inductive Types up to Path Equivalence." GrokRxiv:2026.05.hott-perspective, 2026.

[15] A. Mörtberg, L. Pujet. "Cubical Synthetic Homotopy Theory." CPP 2020, ACM, pp. 158–171.

[16] S. Boulier et al. "Towards Computational UIP in Cubical Agda." arXiv:2511.21209, November 2025.

[17] F. Cherubini et al. "Higher Schreier theory in Cubical Agda." *J. Symbolic Logic* (online first, 2025).

[18] E. Riehl, M. Shulman. "A type theory for synthetic ∞-categories." *Higher Structures* 1(1):147–224, 2017.

[19] C. Angiuli, G. Brunerie, T. Coquand, R. Harper, K.-B. Hou (Favonia), D. Licata. "Cartesian cubical type theory." Preprint, Carnegie Mellon University, 2017–2019.