Final Coalgebras and Transcendental Numbers in HoTT: A Coinductive Characterisation of π and e

Matthew Long
The YonedaAI Collaboration
YonedaAI Research Collective
Chicago, IL
`research@yonedaai.com` · `https://yonedaai.com`

4 May 2026

Abstract

The univalent presentation of the real numbers admits two profoundly different formulations: an *inductive* one, in which $\mathbb{R}$ is built as a higher inductive–inductive type (HIIT) of Cauchy sequences modulo a path-constructor for the equivalence relation (Booij; HoTT Book §11.3); and a *coinductive* one, in which $\mathbb{R}$ (or $[0,1]$) is presented as a final coalgebra for an endofunctor on $\mathcal{U}$ encoding digit streams modulo carry-bisimulation. Paper III of our prior series [1] sketched the coalgebraic dual but left the transcendentals π and e entirely on the inductive side (Paper V §8). The present paper completes the coalgebraic side: we construct functors $F_b : \mathcal{U} \to \mathcal{U}$ and $G : \mathcal{U} \to \mathcal{U}$ whose final coalgebras present the digit-stream and Cauchy-stream models of $[0,1]$ and $\mathbb{R}$ respectively; we lift the standard analytic descriptions of π (the Bailey–Borwein–Plouffe series, Machin's formula, the Leibniz alternating series) and e (Euler's series) to *bisimulation-closed observable predicates* on these final coalgebras; and we prove that π and e arise as unique elements identified by these predicates without recourse to the HIIT path-constructors of Cauchy reals. Along the way we develop coinduction up-to-context, prove a constructive Lambek lemma in cubical HoTT via M-types (Ahrens–Capriotti–Spadotti), and supply Haskell and Lean 4 formalisations of the resulting stream coalgebras and their corecursive definitions of π and e. We close by relating the coalgebraic picture to the Riemann zeta function: a $\zeta(s) = 0$ statement entirely internal to the language of a final coalgebra remains the principal open problem and we frame it precisely.

Contents

1 Introduction

1.1 Two presentations of $\mathbb{R}$, two presentations of π

The synthesis paper of our series [3] identified six mutually equivalent presentations of the natural numbers (NNO, Peano, primitive recursion, set-theoretic encoding, Yoneda representable, HoTT inductive type). For the real numbers the analogue narrative produces two presentations rather than six, but each is itself a different *universal property*:

1. **Inductive (Cauchy/HIIT).** Booij's HoTT Cauchy reals $\mathbb{R}_c$ [4], presented as the initial algebra of a signature consisting of $\mathbb{Q}$-coercion, a limit operation taking $(\varepsilon \mapsto x_\varepsilon)$-Cauchy approximations to a point of $\mathbb{R}_c$, plus path constructors that quotient. This is the universal Cauchy-complete Archimedean ordered field.

2. **Coinductive (final coalgebra).** The cofree object on a digit-stream signature: $\mathbb{R} \cong \nu G$ for an endofunctor G which is roughly $G\,X = \mathbb{D} \times X$ (with $\mathbb{D}$ a digit alphabet) modulo a carry-bisimulation. Streams come with a *coinduction* principle (Lambek; Rutten [5]) rather than a recursion principle.

These two presentations are isomorphic as ordered fields by the standard uniqueness theorem for Dedekind-complete Archimedean fields (Paper III Theorem 4.1). They are however *not* isomorphic as effective structures: the Type II Turing computability literature [16] establishes that Cauchy realizers and stream realizers are interreducible but not equiprimitive. In the intensional setting of HoTT this distinction shows up in the *computation rules*: the inductive presentation supports recursors that compute on constructors, while the coinductive presentation supports corecursors that produce streams element by element with guarded productivity.

For the transcendentals π and e, the inductive side is well-understood: Paper V Definitions 8.1–8.2 present π as the centre of the contractible type $\Sigma_{p:\mathbb{R}_c}\,(\sin p = 0) \times (p > 0) \times (\forall x \in (0,p), \sin x > 0)$ and e as $\exp(1)$ via the universal property of exp as the unique smooth $f : \mathbb{R}_c \to \mathbb{R}_c$ with $f' = f$ and $f(0) = 1$. The coinductive side has, prior to this paper, been left implicit: the generic statement "π and e are computable hence have canonical digit streams" is true but does not constitute a HoTT-internal characterisation.

1.2 The principal contribution

The principal contribution of this paper is to write down such an internal characterisation. Concretely, we construct:

- A functor $F_b : \mathcal{U} \to \mathcal{U}$, $F_b\,X = \mathbb{D}_b \times X$, with $\mathbb{D}_b = \mathsf{Fin}(b)$ the alphabet on b digits, and prove that its final coalgebra νF_b in cubical HoTT is the M-type of streams $\mathbb{D}_b^{\mathbb{N}}$.

- A bisimulation $\sim_b$ on νF_b, the *carry-bisimulation*, which makes $\nu F_b / \sim_b$ an ordered Archimedean field equipped with a continuous map to $[0, 1]$.

- Two *bisimulation-closed observable predicates* P_π, P_e on νF_b, expressed entirely in the internal language of the coalgebra (head/tail/observation), which uniquely identify (up to bisimulation) the Bailey–Borwein–Plouffe digits of π in base 16 and the Euler-series digits of e in any base.

- A coinductive proof, via coinduction up-to-context, that the unique-existence statement is *contractible*: the type $\Sigma_{s:\nu F_b} P_\pi(s)$ is contractible.

This separates the question "what does π name?" into a foundational layer (the final coalgebra structure) and a content layer (the BBP series). The final-coalgebra layer is internal to HoTT and uses no HIIT path constructors. The content layer translates a classical analytic identity into a guarded corecursive equation, whose validity is checked by coinduction.

1.3 Why coinduction matters in HoTT

In Martin-Löf type theory the coinductive types of Coquand and the M-types of Ahrens–Capriotti–Spadotti [7] are treated by dualising the W-type construction. In cubical HoTT M-types satisfy a strong coinduction principle, namely that bisimulation implies path-equality. This is Theorem 5.3 of Paper III in our series, which we recall and refine here as Theorem 3.4.

Crucially, coinduction allows us to exhibit equalities between streams produced by *a priori* different corecursive schemes. For instance, the digit stream produced by Machin's formula

$$\pi = 16 \arctan(1/5) - 4 \arctan(1/239)$$

is not, as a sequence of base-10 digits, definitionally equal to the digit stream produced by the Leibniz series

$$\pi = 4 \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}.$$

Their equality must be proven, and coinduction up-to is the natural tool. Once equality is established, both schemes name the same $\pi : \nu F_{10}/\sim_{10}$. Our characterisation factors through this identification.

1.4 Outline

Section 2 fixes the type-theoretic framework: cubical HoTT, polynomial endofunctors, M-types, and the construction of final coalgebras. Section 3 states and proves a HoTT-internal Lambek lemma and the resulting coinduction principle. Section 4 develops the stream coalgebra in detail, including the carry-bisimulation. Section 5 introduces bisimulation-closed predicates and the up-to technique. Section 6 contains the main results: the coalgebraic characterisations of π and e. Section 7 sketches the cubical M-type realisation. Section 8 compares to the Cauchy HIIT of Paper V. Section 9 states the ζ-prerequisite chain and the $\zeta(s) = 0$-internal-to-coalgebra problem. Section 10 closes with open problems.

2 Mathematical framework

2.1 Cubical HoTT and polynomial endofunctors

We work in cubical type theory à la Cohen–Coquand–Huber–Mörtberg (CCHM) [8], with universes $\mathcal{U}_0 : \mathcal{U}_1 : \ldots$ closed under the usual type formers, and with the univalence axiom provable from the path-cube structure. We write $a =_A b$ for the path type, refl_a for the constant path, and Σ, Π for the dependent types. Truncation levels follow Voevodsky's conventions: a type is a *proposition* if it has at most one inhabitant up to path equality, a *set* if its identity types are propositions, and so on.

Definition 2.1 (Polynomial endofunctor)**.** A *polynomial endofunctor* on $\mathcal{U}$ is a functor $F : \mathcal{U} \to \mathcal{U}$ of the form

$$F\,X \;=\; \Sigma_{a:A}\, X^{B(a)}$$

for some $A : \mathcal{U}$ and $B : A \to \mathcal{U}$. The pair (A, B) is called a *container* or *signature*.

The action of F on a function $f : X \to Y$ is $F\,f\,(a, h) = (a, f \circ h)$.

Example 2.2. The *stream functor* on alphabet A is $F_A\,X = A \times X$. This is polynomial with A-many shapes (one per letter) and a single position per shape: $F_A\,X = \Sigma_{a:A}\, X^{\mathbf{1}}$, where $\mathbf{1}$ is the unit type.

Example 2.3. For $b \geq 2$, the *base-b digit functor* is $F_b\,X = \mathsf{Fin}(b) \times X$, the special case of Theorem 2.2 with $A = \mathsf{Fin}(b)$.

Definition 2.4 (Coalgebra)**.** An *F-coalgebra* is a pair (C, γ) with $\gamma : C \to F\,C$. A morphism of coalgebras $(C, \gamma) \to (D, \delta)$ is a function $f : C \to D$ such that $\delta \circ f = F\,f \circ \gamma$. The category of F-coalgebras is denoted $\mathbf{Coalg}(F)$.

Definition 2.5 (Final coalgebra)**.** $(\nu F, \mathsf{out})$ is a *final F-coalgebra* if for every F-coalgebra (C, γ) there is a unique morphism $\mathsf{unfold}_\gamma : (C, \gamma) \to (\nu F, \mathsf{out})$.

Remark 2.6 (Final = terminal in $\mathbf{Coalg}(F)$)*.* "Final" here is the standard terminology of Rutten [5]. In categorical language final coalgebras are terminal objects of $\mathbf{Coalg}(F)$, dual to initial algebras (Lambek).

2.2 M-types as final coalgebras

In MLTT the coinductive analogue of W-types is the M-type construction. The M-type for container (A, B) is

$$M\,A\,B \;=\; \Sigma_{u:\mathrm{Tree}^\infty(A,B)}\, \mathrm{Wf}(u),$$

where $\mathrm{Tree}^\infty(A, B)$ is the type of (potentially infinite) trees and Wf is the productivity (well-formedness) predicate. In cubical HoTT, Ahrens–Capriotti–Spadotti [7] prove:

Theorem 2.7 (ACS, M-type final coalgebra)**.** *For any container (A, B), the M-type $M\,A\,B$ carries a coalgebra structure $\mathsf{out} : M\,A\,B \to F_{(A,B)}(M\,A\,B)$ which is final.*

For our applications the relevant case is the stream functor $F_A\,X = A \times X$, where the M-type specialises to streams $\mathsf{Stream}\,A \cong A^{\mathbb{N}}$.

Definition 2.8 (Stream coalgebra)**.** $\mathsf{Stream}\,A := M\,A\,(\lambda -.\,\mathbf{1})$. The destructor

$$\mathsf{out} : \mathsf{Stream}\,A \to A \times \mathsf{Stream}\,A$$

decomposes into the two component maps

$$\mathsf{hd} : \mathsf{Stream}\,A \to A, \qquad \mathsf{tl} : \mathsf{Stream}\,A \to \mathsf{Stream}\,A.$$

Remark 2.9 (Cubical realisation). In cubical HoTT, $\mathsf{Stream}\,A$ admits a definition by guarded recursion using the modality $\triangleright$ of Birkedal et al. [9]: $\mathsf{Stream}\,A \cong A \times \triangleright\mathsf{Stream}\,A$. The fixed point exists because $\triangleright$ is a contractive functor and the guarded recursion theorem applies.

3 Lambek's lemma and coinduction

We recall Lambek's lemma in its dual coalgebraic form, which is the cornerstone of all coinductive characterisations.

Lemma 3.1 (Dual Lambek)**.** *If $(\nu F, \mathsf{out})$ is a final F-coalgebra, then $\mathsf{out} : \nu F \to F(\nu F)$ is a path-equivalence.*

Proof. Consider the coalgebra $(F(\nu F), F\mathsf{out})$. Finality yields a unique map $\theta : F(\nu F) \to \nu F$ with $\mathsf{out}\circ\theta = F(F\mathsf{out})\circ F\,\mathsf{out} = F(\mathsf{out}\circ\theta)$. Then $\theta\circ\mathsf{out} : \nu F \to \nu F$ is a coalgebra morphism, and so is $\mathrm{id}_{\nu F}$, so by uniqueness $\theta \circ \mathsf{out} = \mathrm{id}$. Symmetrically, $\mathsf{out} \circ \theta$ is a coalgebra morphism on $(F(\nu F), F\mathsf{out})$ to itself, equal to the identity. Hence out is a path-equivalence with inverse θ. □

Remark 3.2. Theorem 3.1 is conceptually crucial: it says the final coalgebra is a fixed point of F *up to path equivalence*, the coinductive dual of Lambek's algebraic "$\mu F \cong F(\mu F)$". In cubical HoTT this equivalence is *strict* (definitional after path-application) at base types, which is what makes hd, tl definable.

Definition 3.3 (Bisimulation)**.** A relation $R : C \times C \to \mathcal{U}$ on the carrier of an F-coalgebra (C, γ) is an F-bisimulation if there exists a coalgebra structure $\bar{\gamma} : \Sigma_{(x,y):C\times C} R(x, y) \to F(\Sigma \ldots R)$ making the two projections coalgebra morphisms. Concretely, for the stream functor $F_A\,X = A \times X$, the unfolded definition is: a relation R on $\mathsf{Stream}\,A$ is a bisimulation iff for all (x, y) with $R(x, y)$, we have $\mathsf{hd}(x) = \mathsf{hd}(y)$ and $R(\mathsf{tl}(x), \mathsf{tl}(y))$.

Theorem 3.4 (Coinduction principle)**.** *Let $(\nu F, \mathsf{out})$ be a final F-coalgebra. For any F-bisimulation R and any $x, y : \nu F$, if $R(x, y)$ holds, then $x = y$ in νF.*

Proof. By Theorem 3.3 the projections $\pi_1, \pi_2 : \Sigma_{(x,y)} R(x, y) \to \nu F$ are coalgebra morphisms into the final coalgebra. By uniqueness of mediating maps, $\pi_1 = \pi_2$ as coalgebra morphisms, hence as functions, hence pointwise. So if $(x, y, p) : \Sigma_{(x,y)} R(x, y)$ then $\pi_1(x, y, p) = \pi_2(x, y, p)$, i.e. $x = y$. □

This is the fundamental *coinductive proof principle*: to exhibit a path between two elements of a final coalgebra it suffices to exhibit a bisimulation containing them.

4 Stream coalgebras and digit streams

4.1 The stream functor and its final coalgebra

Theorem 4.1 (Streams as final coalgebra). *For any type A, the M-type* $\mathsf{Stream}\,A$ *together with* $\mathsf{out} = \langle \mathsf{hd}, \mathsf{tl} \rangle$ *is the final coalgebra of* $F_A = A \times (-)$.

Proof sketch. For coalgebra $(C, \gamma : C \to A \times C)$, define $\mathsf{unfold}_\gamma : C \to \mathsf{Stream}\,A$ by guarded corecursion:

$$\begin{aligned} \mathsf{hd}(\mathsf{unfold}_\gamma(c)) &:= \pi_1(\gamma(c)), \\ \mathsf{tl}(\mathsf{unfold}_\gamma(c)) &:= \mathsf{unfold}_\gamma(\pi_2(\gamma(c))). \end{aligned}$$

The morphism property $\mathsf{out} \circ \mathsf{unfold}_\gamma = F_A\, \mathsf{unfold}_\gamma \circ \gamma$ is judgmental. Uniqueness follows from coinduction (Theorem 3.4) applied to the relation $R(s,t) := \Sigma_{c:C}\,(s = \mathsf{unfold}_\gamma(c)) \times (t = \mathsf{unfold}_\gamma(c))$. □

Example 4.2 (Constant stream). For $a : A$, $\mathsf{const}_a := \mathsf{unfold}_\gamma$ where $C = \mathbf{1}$ and $\gamma(\star) = (a, \star)$, so $\mathsf{hd}(\mathsf{const}_a) = a$ and $\mathsf{tl}(\mathsf{const}_a) = \mathsf{const}_a$.

Example 4.3 (Naturals as stream). $\mathsf{nats} := \mathsf{unfold}_\gamma$ where $C = \mathbb{N}$ and $\gamma(n) = (n, n+1)$. Yields $\mathsf{nats} = 0, 1, 2, 3, \ldots$

4.2 Base-b digit streams and approximation maps

Fix $b \geq 2$. We work with digit streams of $\mathsf{Fin}(b)$.

Definition 4.4 (Approximation map). The base-b *approximation map* $\alpha_b : \mathsf{Stream}\,\mathsf{Fin}(b) \to \mathbb{R}_c$ takes a stream $d_0 d_1 d_2 \ldots$ to the Cauchy real

$$\alpha_b(d) := \lim_{n \to \infty} \sum_{k=0}^{n-1} \frac{d_k}{b^{k+1}},$$

which exists because the partial sums form a b^{-n}-Cauchy approximation in Booij's sense: for $m \geq n$ the difference $\left|\sum_{k=0}^{m-1} d_k / b^{k+1} - \sum_{k=0}^{n-1} d_k / b^{k+1}\right| = \sum_{k=n}^{m-1} d_k / b^{k+1} \leq \sum_{k=n}^{\infty} (b-1)/b^{k+1} = b^{-n}$, so the partial sums are uniformly Cauchy with modulus b^{-n} and Booij's lim-constructor applies.

Lemma 4.5 (Approximation surjects onto $[0,1]$). $\alpha_b : \mathsf{Stream}\,\mathsf{Fin}(b) \to \mathbb{R}_c$ *has image* $[0,1] \subset \mathbb{R}_c$.

Proof. Surjection onto $[0,1]$ is the standard digit-expansion: given $x \in [0,1] \cap \mathbb{R}_c$, define $d_k = \lfloor b^{k+1} x \rfloor \mod b$ using the floor function on Cauchy reals. The sum converges to x by the standard estimate. Existence of floor on $\mathbb{R}_c$ uses Booij's order. (Note that floor is not constant on bisimulation classes; this is the source of the carry equivalence.) □

Definition 4.6 (Carry-bisimulation)**.** The relation $\sim_b$ on $\mathsf{Stream}\,\mathsf{Fin}(b)$ is the smallest equivalence such that $\alpha_b(s) = \alpha_b(t)$ implies $s \sim_b t$. Concretely, for any n and any $d \in \mathsf{Fin}(b-1)$,

$$d_0 \cdots d_{n-1}\, d\, (b{-}1)(b{-}1) \cdots \;\sim_b\; d_0 \cdots d_{n-1}\, (d{+}1)\, 0\, 0 \cdots$$

This is exactly the "$0.999\ldots = 1.000\ldots$" identification.

Theorem 4.7 ($[0,1]$ as quotient final coalgebra)**.** *The set-quotient* $\mathsf{Stream}\,\mathsf{Fin}(b)/{\sim_b}$ *is in HoTT-equivalence with* $[0,1] \subset \mathbb{R}_c$ *via* α_b*.*

Proof. α_b factors through the quotient by definition of $\sim_b$. The induced map is injective by the kernel of α_b being $\sim_b$, and surjective by Theorem 4.5. Both maps are set-functions (between h-sets), hence the resulting bijection is an equivalence by univalence-for-sets. $\square$

Remark 4.8 ($\mathbb{R}$ from $[0,1]$). The whole real line $\mathbb{R}$ is recovered as $\mathbb{Z} \times [0,1]/\sim$ with the obvious carrying identifications, or equivalently as the final coalgebra of the functor $G\,X = \mathbb{Z} \times \mathsf{Stream}\,\mathsf{Fin}(b)\,X$ modulo the appropriate bisimulation.

5 Bisimulation-closed predicates and coinduction up-to

To characterise π and e *internally* to the final coalgebra, we need predicates that respect $\sim_b$. We formalise this notion and develop the up-to technique.

Definition 5.1 (Bisimulation-closed predicate)**.** A predicate $P : \mathsf{Stream}\,\mathsf{Fin}(b) \to \mathsf{Prop}$ is $\sim_b$*-closed* (or simply *closed*) if $s \sim_b t \to P(s) \leftrightarrow P(t)$. Equivalently, P factors through the quotient $\mathsf{Stream}\,\mathsf{Fin}(b)/{\sim_b}$.

Example 5.2. The predicate $P(s) := (\alpha_b(s) < 1/2)$ is closed; the predicate $P(s) := (\mathsf{hd}(s) = 4)$ in base 10 is *not* closed: the stream $0.4\,9\,9\,9\ldots$ has $\mathsf{hd} = 4$ but is bisimilar to $0.5\,0\,0\ldots$ which has $\mathsf{hd} = 5$.

Definition 5.3 (Observable predicate)**.** A predicate P is *observable* if it is determined by the function α_b, i.e. $P(s) = Q(\alpha_b(s))$ for some $Q : \mathbb{R}_c \to \mathsf{Prop}$. Every observable predicate is closed, but not conversely.

Lemma 5.4. *A predicate is closed iff it factors through* α_b*.*

Proof. ($\Rightarrow$) A closed predicate factors through $\mathsf{Stream}/{\sim_b}$, and α_b induces an equivalence between $\mathsf{Stream}/{\sim_b}$ and $\mathrm{im}(\alpha_b) \subseteq \mathbb{R}_c$ (Theorem 4.7). Composition with the inverse equivalence gives the factorisation. ($\Leftarrow$) Immediate: factoring through α_b implies $\alpha_b(s) = \alpha_b(t) \Rightarrow P(s) \leftrightarrow P(t)$, which is the definition of closure under the kernel of α_b. $\square$

5.1 Coinduction up-to-context

The plain coinduction principle (Theorem 3.4) requires exhibiting a strict bisimulation. In practice this is often inconvenient: one wants to prove $s = t$ by exhibiting a relation R such that $R(s,t)$ implies that the heads agree and the tails are related by R *after some closure operations*. Sangiorgi–Rutten formalised this via the *up-to* technique.

Definition 5.5 (Φ-up-to)**.** Let Φ be a monotone operator on relations on $\mathsf{Stream}\,A$. A relation R is a *bisimulation up-to* Φ if for every $(s,t) \in R$, $\mathsf{hd}(s) = \mathsf{hd}(t)$ and $(\mathsf{tl}(s), \mathsf{tl}(t)) \in \Phi(R)$.

Theorem 5.6 (Up-to-context soundness)**.** *If Φ is* compatible *(i.e. Φ preserves bisimulations in the sense of Pous–Sangiorgi [10]), then any bisimulation up-to Φ is contained in $\sim$.*

Proof sketch. Define $R^* := \bigcup_{n\geq 0} \Phi^n(R)$. Show by induction that R^* is a (plain) bisimulation: heads agree because Φ preserves head-equality, and tails of R^*-related pairs remain in R^* because $\Phi^n(R) \subseteq \Phi^{n+1}(R)$ by monotonicity and the bisimulation-up-to-Φ condition. □

Example 5.7 (Equivalence-closure up-to)**.** $\Phi(R) := \{(s,t) : \exists s', t'\, R(s', t') \wedge s = s' \wedge t = t'\}$ is compatible. So one may freely substitute path-equal streams inside coinductive proofs.

Example 5.8 (Arithmetic up-to)**.** For arithmetic operations on base-b digit streams (addition, multiplication, division, exponentiation by digit), the operators

$$\Phi_+(R) := \{(s+u, t+u) : R(s,t)\}, \qquad \Phi_\cdot(R) := \{(s \cdot u, t \cdot u) : R(s,t)\}$$

are compatible (their compatibility reduces to verifying that each operation is itself definable as coalgebra morphisms, which is [11]).

6 Coalgebraic characterisations of π and e

We now state the main results of the paper.

6.1 The case of e

Definition 6.1 (Series coalgebra for e)**.** Define a coalgebra (C_e, γ_e) on $C_e := \mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N}\times\mathbb{N}$ (state: $(n, k, \mathrm{num}, \mathrm{den}, \mathrm{outdig})$ representing partial summation of $\sum_k 1/k!$ in base b), with transition γ_e implementing the streaming factorial-series algorithm of Sale [12] (refined to spigot form by Rabinowitz–Wagon [13] and Gibbons [14]) for streaming the digits of e in base b. Concretely, $\gamma_e(s) = (d(s), s')$ where $d(s) \in \mathsf{Fin}(b)$ is the next emitted digit and s' is the updated state.

Theorem 6.2 (e as final-coalgebra unfold)**.** $\mathsf{unfold}_{\gamma_e}(s_0) = s_e \in \mathsf{Stream}\,\mathsf{Fin}(b)$ *where s_0 is the initial state and $\alpha_b(s_e) = e - 2 \in [0,1]$. Equivalently, the digit stream of e (after subtracting the integer part) is the unique solution of the corecursive equation*

$$s_e = \mathsf{streamingFraction}(\langle 1/k! \rangle_{k\geq 0}).$$

Proof. Existence is by guarded corecursion using the productivity of the streaming-factorial algorithm: the state-evolution is structurally decreasing on a measure (the rational interval bounding the unprocessed tail shrinks geometrically), so each output digit is produced after finitely many state transitions. The image under α_b equals $e-2$ by the algorithm's correctness theorem (a classical fact; see Sale [12] and the spigot-algorithm analysis in [13, 14]). □

Definition 6.3 (Predicate P_e)**.** $P_e : \mathsf{Stream\,Fin}(b) \to \mathsf{Prop}$ is the predicate $P_e(s) := \alpha_b(s) = e - 2$. It is observable (Theorem 5.3) hence closed (Theorem 5.4).

Theorem 6.4 (Coalgebraic characterisation of e)**.** *The type*

$$\Sigma_{s:\mathsf{Stream\,Fin}(b)}\, P_e(s)$$

quotiented by $\sim_b$ is contractible. Equivalently: $e-2$ is the unique element of $\nu F_b/{\sim_b}$ satisfying P_e.

Proof. Existence: Theorem 6.2 gives an inhabitant. Uniqueness: if s_1, s_2 both satisfy P_e then $\alpha_b(s_1) = e - 2 = \alpha_b(s_2)$, so $s_1 \sim_b s_2$ by definition of $\sim_b$. Hence after quotient they coincide. Since the fibres of the quotient map are propositions (a consequence of working with h-sets), the total space is a proposition. Combined with inhabitation, this proves contractibility. □

Remark 6.5. Theorem 6.4 is internal in the sense that P_e is expressed in terms of α_b, which is itself definable by guarded corecursion. The HIIT path constructors of Booij's $\mathbb{R}_c$ enter only *inside* α_b. If one is willing to take α_b as the basic "observation", the characterisation is purely coalgebraic. We discuss this point further in Section 8.

6.2 The case of π

For π the situation is more subtle because no *finite-state* streaming algorithm is known. We use Bailey–Borwein–Plouffe (BBP) in base 16, which is streamable digit-by-digit although the state grows linearly.

Definition 6.6 (BBP coalgebra)**.** The Bailey–Borwein–Plouffe identity (1995):

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right).$$

This induces a coalgebra (C_π, γ_π) with C_π encoding partial summation state, and γ_π implementing the BBP digit extraction algorithm in base 16.

Theorem 6.7 (π as final-coalgebra unfold)**.** $\mathsf{unfold}_{\gamma_\pi}(s_0) = s_\pi \in \mathsf{Stream\,Fin}(16)$ *satisfies* $\alpha_{16}(s_\pi) = \pi - 3$.

Proof. By the BBP correctness theorem (which is constructively valid; see [15]), the partial sums $\sum_{k=0}^{n-1}$ approximate π to within $O(16^{-n})$. The hexadecimal digit-extraction state evolves productively, so guarded corecursion yields a stream whose image under α_{16} equals $\pi - 3$. □

Definition 6.8 (Predicate P_π)**.** $P_\pi : \mathsf{Stream\,Fin}(16) \to \mathsf{Prop}$ is $P_\pi(s) := \alpha_{16}(s) = \pi - 3$. Closed by Theorem 5.4.

Theorem 6.9 (Coalgebraic characterisation of π)**.** *The type $\Sigma_{s:\mathsf{Stream\,Fin}(16)}\, P_\pi(s)$ quotiented by $\sim_{16}$ is contractible.*

Proof. Same structure as Theorem 6.4, using Theorem 6.7. □

6.3 Internalisation: removing the reference to $\mathbb{R}_c$

Theorem 6.4 and Theorem 6.9 reference $e-2$ and $\pi-3$ as elements of $\mathbb{R}_c$, hence implicitly the HIIT. We now upgrade these to characterisations expressed *purely* in terms of bisimulations between streams.

Theorem 6.10 (Internal characterisation of e)**.** *There is a stream $s^* \in \mathsf{Stream\,Fin}(b)$ defined by guarded corecursion (no use of $\mathbb{R}_c$ or HIIT) such that:*

1. *$s_e \sim_b s^*$, where s_e is from Theorem 6.2.*
2. *For any stream t, if t satisfies the corecursive equation $\mathsf{out}(t) = \gamma_e(\mathrm{state}(t))$ for the (no-$\mathbb{R}_c$) state projection* state : $\mathsf{Stream} \to \mathbb{N}^5$ *(a partial function defined on the image of $\mathsf{unfold}_{\gamma_e}$, see Section B), then $s^* \sim_b t$.*

The unique-existence is proved by coinduction up-to-context.

Proof. Take $s^* := \mathsf{unfold}_{\gamma_e}(s_0)$ where γ_e from Theorem 6.1 is purely combinatorial (state in $\mathbb{N}^5$, no real numbers; the explicit transition rules are spelled out in Section B). Existence: Theorem 6.2.

Uniqueness is the substantive part. Suppose $t \in \mathsf{Stream\,Fin}(b)$ satisfies the equation $\mathsf{out}(t) = \gamma_e(\mathrm{state}(t))$ for some state-projection function state : $\mathsf{Stream} \to \mathbb{N}^5$ that agrees with s_0 at t. Define the relation

$$R(u_1, u_2) \ := \ \Sigma_{c:\mathbb{N}^5} \big(u_1 = \mathsf{unfold}_{\gamma_e}(c)\big) \times \big(u_2 = \mathsf{unfold}_{\gamma_e}(c)\big).$$

This is reflexive on the image of $\mathsf{unfold}_{\gamma_e}$. We claim R is a bisimulation *up-to the closure* Φ defined by $\Phi(R)(v_1, v_2) := \Sigma_c\, R(\mathsf{unfold}_{\gamma_e}(c), \mathsf{unfold}_{\gamma_e}(c))$ (equivalently, Φ is the equivalence-closure under $\mathsf{unfold}_{\gamma_e}$-substitution; this is compatible by the example following Theorem 5.6). To verify the bisimulation-up-to property: given $R(u_1, u_2)$ with witness c, both heads equal $\pi_1(\gamma_e(c))$ and both tails equal $\mathsf{unfold}_{\gamma_e}(\pi_2(\gamma_e(c)))$, so heads agree and tails are again related (with the new witness $\pi_2(\gamma_e(c))$). By Theorem 5.6 $R \subseteq\sim$, and by Theorem 3.4 $R(s^*, t)$ implies $s^* = t$.

Now apply this with $u_1 := s^* = \mathsf{unfold}_{\gamma_e}(s_0)$ and $u_2 := t$: we need to exhibit a c with $t = \mathsf{unfold}_{\gamma_e}(c)$. Such a c is provided by $\mathrm{state}(t) = s_0$ together with the corecursive equation, by another application of coinduction (using $R'(v_1, v_2) := \exists c.\, v_2 = \mathsf{unfold}_{\gamma_e}(c) \wedge \mathrm{state}(v_1) = c \wedge v_1$ satisfies the equation). Hence $s^* = t$.

Therefore s^* is the unique stream satisfying the corecursive equation, and the bisimulation class $[s^*] \in \mathsf{Stream}/\sim_b$ is uniquely identified by an internal predicate. □

Theorem 6.11 (Internal characterisation of π)**.** *There is a stream $s^*_\pi \in \mathsf{Stream\,Fin}(16)$ defined by guarded corecursion (no use of $\mathbb{R}_c$) such that $s^*_\pi \sim_{16} s_\pi$ uniquely up to bisimulation, characterised by the recursion:*

$$\mathsf{out}(s^*_\pi) = \gamma_\pi^{combinatorial}(\mathrm{state}_0).$$

Proof. The argument is structurally identical to that of Theorem 6.10. Concretely, replace γ_e by the combinatorial BBP transition function $\gamma_\pi^{\mathrm{combinatorial}}$: its state space is $\mathbb{N}^k$ (with

$k = 4$ corresponding to the four sub-series in BBP, plus a position counter; see [15]) holding the integer numerators and denominators of the modular partial sums. The relation

$$R(u_1, u_2) \ := \ \Sigma_{c:\mathbb{N}^k} \left(u_1 = \mathsf{unfold}_{\gamma_\pi^{\text{combinatorial}}}(c)\right) \times \left(u_2 = \mathsf{unfold}_{\gamma_\pi^{\text{combinatorial}}}(c)\right)$$

is a bisimulation up-to the substitution-closure Φ, by the same head/tail computation as in Theorem 6.10, and the BBP modular-arithmetic update of c ensures the up-to-context operator is compatible (as in the arithmetic example following Theorem 5.6). By Theorems 3.4 and 5.6 s_π^* is the unique stream up to $\sim_{16}$ satisfying the BBP corecursive equation. □

Remark 6.12 (What Theorem 6.10 and Theorem 6.11 do and do not say). The key advance of Theorem 6.10 and Theorem 6.11 over Theorem 6.4 and Theorem 6.9 is that the characterising predicate is now of the form "satisfies the corecursive equation $\mathsf{out}(s) = \gamma(\text{state}(s))$", expressed purely in terms of out, hd, tl, and combinatorial operations on $\mathbb{N}$. No reference to $\mathbb{R}_c$ or its path constructors is needed. This realises programmes (a)–(c) of [1, §9.2] with $F = F_b$ and $P = P_\pi$ resp. P_e.

We are however candid about a limitation: the predicate "s satisfies the corecursive equation $\mathsf{out}(s) = \gamma(\text{state}(s))$" is, in a sense, exactly the description of being the unfold of γ, and the state projection state is canonically defined only on the image of unfold_γ (see Section B). Read *externally*, the theorem says: "the unique stream satisfying the corecursive equation for algorithm X is the stream generated by algorithm X." This is not yet a characterisation of π or e as classical reals *without reference to any external structure*. Such a fully external characterisation would require either (a) an internal proof that two distinct combinatorial algorithms (e.g. Leibniz and BBP) are bisimilar, or (b) an internal axiomatisation of which streams are "transcendental".

What Theorems 6.10 and 6.11 *do* establish is that the bisimulation classes

$$[s_e]_{\sim_b} \in \mathsf{Stream}/{\sim_b} \qquad \text{and} \qquad [s_\pi]_{\sim_{16}} \in \mathsf{Stream}/{\sim_{16}}$$

are the unique fixed points of combinatorial corecursive equations, with no path-constructor data (HIIT-data) needed to express the predicate. This is a strict logical advance over the inductive characterisation of Paper V Definition 8.1, which uses sin-zero data internal to $\mathbb{R}_c$. The remaining "gap" is exactly Theorem 10.1 below.

6.4 Comparison: Leibniz, Machin, BBP all name the same π

To illustrate the strength of bisimulation as an equality principle in HoTT, we show how three distinct corecursive definitions of π-as-stream are proved equal by coinduction.

Definition 6.13 (Three coalgebras for π)**.** Let γ_L (Leibniz), γ_M (Machin), γ_B (BBP) be three coalgebras whose unfolds in base 10 (resp. 10, 16) yield streams s_L, s_M, s_B. Each algorithm is a corecursive streaming version of the corresponding analytic identity for π.

Proposition 6.14. *$s_L \sim_{10} s_M$, and $s_L \sim_{10}$ base-conversion(s_B).*

Proof. $\alpha_{10}(s_L) = \alpha_{10}(s_M) = \pi - 3 = \alpha_{10}(\text{conv}(s_B))$ by classical analytic identities (Leibniz, Machin, BBP all sum to π). Bisimulation follows from Theorem 5.4 and the definition of $\sim_b$ as the kernel of α_b. □

Remark 6.15 (Limitation: external vs. internal equivalence). Theorem 6.14 *factors through* the approximation map α_{10}, hence through $\mathbb{R}_c$. This is therefore not yet a fully *internal* proof of bisimilarity: the equality of streams is established by mapping to the inductive $\mathbb{R}_c$, computing the values, and pulling back along the kernel of α_{10}.

A truly internal proof would establish a direct stream-level bisimulation between the Leibniz, Machin, and BBP state machines (after base conversion). Schematically, one might construct a *product coalgebra* on $C_L \times C_M \times C'_B$ (running all three algorithms in parallel, with C'_B the base-converted BBP state), together with an invariant $I \subseteq C_L \times C_M \times C'_B$ asserting that the three running partial sums agree to within a shrinking 10^{-n} window. The bisimulation $R(u_L, u_M, u_B) := \exists(c_L, c_M, c_B) \in I.\ \bigwedge u_X = \mathsf{unfold}_{\gamma_X}(c_X)$ would then be *coinductively* a bisimulation, and Theorem 3.4 would conclude. The technical obstacle is constructing the invariant I in HoTT without circular reference to the value of π. This is precisely the form Theorem 10.1 would take and we do not claim a solution.

The conceptual picture is striking even in the weaker "factor-through-$\mathbb{R}_c$" form: in HoTT, *algorithm equivalence* on streams becomes *path-equality* on the final coalgebra, and coinduction is the proof method.

7 Cubical M-types and stream realisation

We sketch the cubical-HoTT realisation of the M-type used above, following ACS [7].

7.1 The M-type construction

Definition 7.1 (M-type)**.** For container (A, B), define

$$M\,A\,B \;:=\; \lim_{n} W_n$$

where $W_0 = \mathbf{1}$, $W_{n+1} = \Sigma_{a:A}(B(a) \to W_n)$, and the limit is taken along the projections induced by the unique map $W_1 \to W_0 = \mathbf{1}$.

Theorem 7.2 (Cubical M-type, ACS)**.** *In cubical HoTT (CCHM), the limit Theorem 7.1 exists and $M\,A\,B$ is the carrier of the final $F_{(A,B)}$-coalgebra. The destructor* $\mathsf{out} : M\,A\,B \to F_{(A,B)}(M\,A\,B)$ *is the projection onto the first level.*

Remark 7.3 (Why cubical, not just MLTT). In plain MLTT one cannot prove that out is an equivalence (Theorem 3.1) without function extensionality and η-rules. Cubical HoTT provides function extensionality *judgmentally* (via path-application) and treats M-types coherently with the path structure, which is what we need for the coinduction principle to compute.

7.2 Productivity and guarded recursion

In cubical HoTT, the topos of trees [9] provides a guarded modality $\triangleright$ (read "later"). Streams are then a fixed point $\mathsf{Stream}\,A \cong A \times \triangleright\mathsf{Stream}\,A$, and corecursion is just guarded recursion: the recursive call is hidden under $\triangleright$, ensuring productivity at the type level. This is what makes our unfold_γ judgmentally well-defined.

8 Comparison with the Cauchy HIIT

8.1 Side-by-side

Inductive (Booij/HoTT Cauchy)	**Coinductive (final coalgebra)**
HIIT $\mathbb{R}_c$ with rat, lim, path-constr	M-type $\mathsf{Stream}\,\mathsf{Fin}(b) + \sim_b$
Recursor: $f(\mathrm{rat}\,q)$, $f(\lim x)$, $f(\mathrm{path}\ \ldots)$	Corecursor: $\mathsf{hd}(\mathsf{unfold}(c))$, $\mathsf{tl}(\mathsf{unfold}(c))$
$\pi :=$ centre of contractible sin-zero type	$\pi :=$ centre of contractible BBP unfold
(Paper V Def. 8.1)	type (Theorem 6.11)
Equality by computation rule + truncation	Equality by bisimulation
Computational content: Cauchy realiser	Computational content: digit stream
$\mathbb{R}_c$ has join-semilattice + lim-axioms	$\mathsf{Stream}/{\sim_b}$ has digit-shift action

8.2 Equivalence theorem

Theorem 8.1 (Inductive–Coinductive equivalence)**.** *There is a HoTT equivalence*

$$\beta_b : \mathsf{Stream}\,\mathsf{Fin}(b)/{\sim_b} \;\simeq\; [0,1] \subset \mathbb{R}_c$$

sending the bisimulation class of s to $\alpha_b(s)$.

Proof. Forward direction by Theorem 4.7. Inverse: given $x \in [0,1] \cap \mathbb{R}_c$, produce its base-$b$ digit stream by the floor algorithm of Theorem 4.5; the resulting class $[s] \in \mathsf{Stream}/{\sim_b}$ is well-defined because two different digit-expansion choices for the same x differ only at carry boundaries (the $0.999\cdots = 1$ identifications), which is exactly $\sim_b$. $\square$

Corollary 8.2. *Under β_b, the inductive π (Paper V Definition 8.1) and the coinductive π (Theorem 6.11) coincide.*

8.3 Why the duality matters

Theorem 8.1 says the two presentations describe the same mathematical object. But *computationally* they differ:

- The inductive presentation supports *recursion*: to define $f : \mathbb{R}_c \to X$ specify $f(\mathrm{rat}\,q)$, $f(\lim x)$, and verify path coherence.
- The coinductive presentation supports *corecursion*: to define $g : X \to \mathsf{Stream}$ specify $\mathsf{hd}(g(c))$ and $\mathsf{tl}(g(c))$ in terms of head/tail of an evolving state.

For the transcendentals, the natural definitions are *corecursive*: e and π are typically given by streaming-digit algorithms (Sale, BBP, spigot algorithms in general). The coinductive presentation matches these algorithms *judgmentally*; the inductive presentation requires a layer of approximation-and-quotient.

9 Connection to the ζ-prerequisite chain

9.1 The principal open problem

The synthesis paper [3, §7.3, §8 item 2] identifies "$\zeta(s) = 0$ as a HoTT-native statement" as the principal open problem. Loeffler–Stoll [17] formalised ζ in classical Lean/Mathlib (3300 lines for the analytic continuation alone), but no HoTT-native formalisation of the Riemann zeta function is known.

The coalgebraic toolkit developed here suggests an attack route:

Conjecture 9.1 (Coalgebraic ζ). There is a polynomial endofunctor $H : \mathcal{U} \to \mathcal{U}$ with νH modelling the Riemann zeta function as a stream of digits in some base parametrised by the complex argument $s \in \mathbb{C}_c$, such that:

1. For s with $\Re(s) > 1$, $\mathsf{unfold}_{\gamma_\zeta(s)}$ produces the stream of digits of $\zeta(s)$ via the Dirichlet series.
2. For s in the analytic continuation domain $\mathbb{C}_c \setminus \{1\}$, $\mathsf{unfold}_{\gamma'_\zeta(s)}$ produces digits via Hurwitz formula.
3. There is a bisimulation-closed predicate $P_{\zeta=0} : \nu H \to \mathsf{Prop}$ such that $\zeta(s) = 0$ in $\mathbb{C}_c$ iff the unique unfold satisfies $P_{\zeta=0}$.

This conjecture is open. We make some progress in Theorem 9.2 below.

9.2 Dirichlet series as stream

Proposition 9.2 (Dirichlet partial-sum stream). *For $s \in \mathbb{C}_c$ with $\Re(s) > 1$, define a coalgebra (C_s, γ_s) on $C_s := \mathbb{N} \times \mathbb{C}_c$ with $\gamma_s(n, z) = (\mathrm{digit}(z'), (n + 1, z'))$ where $z' = z + n^{-s}$ (the next partial-sum). Then $\alpha_b(\mathsf{unfold}_{\gamma_s}(0, 0)) = \zeta(s)$, modulo the carry-bisimulation.*

Proof sketch. The partial sums $\sum_{n=1}^{N} n^{-s}$ form a Cauchy approximation to $\zeta(s)$ for $\Re(s) > 1$ (standard fact, constructive proof in Booij's framework). Streaming digits is then Theorem 4.4 applied to that Cauchy approximation. □

Theorem 9.2 only handles the convergence half-plane. The hard part is analytic continuation (steps 2 and 3 of Theorem 9.1). The classical Mellin-transform approach (Loeffler–Stoll) does not obviously translate to the coalgebraic setting because contour integration along non-trivial paths in $\mathbb{C}_c$ is itself a non-trivial constructive issue. We mark this as the principal open problem of the present series.

9.3 Toward analytic continuation coalgebraically

A possible attack: use the functional equation of ζ,

$$\zeta(s) = 2^s \pi^{s-1} \sin\left(\frac{\pi s}{2}\right) \Gamma(1 - s)\, \zeta(1 - s),$$

to define ζ on $\Re(s) < 0$ by reflection. This is a *recursive* definition (refer $\zeta(s)$ to $\zeta(1 - s)$), which corresponds in our framework to a *morphism between two coalgebras* for the same functor H, with the morphism implementing the $s \leftrightarrow 1 - s$ symmetry. A construction of such a morphism in HoTT requires:

- A coalgebraic sin and Γ in $\mathbb{C}_c$ (the former is approachable via the universal property of exp from Paper III §4.3; the latter is significantly harder).
- A coalgebraic z^s (complex exponentiation), again via exp and log.

This is a substantial programme; we leave it as a long-term goal of the YonedaAI HoTT mathematics initiative.

10 Open problems and future work

10.1 The internalisation gap

Theorem 6.10 and Theorem 6.11 characterise π, e via combinatorial corecursive equations. But the correctness theorems (that the unfold equals $\pi - 3$ resp. $e - 2$) still reference $\mathbb{R}_c$ via α_b.

Open Problem 10.1 (Fully internal correctness). Find a HoTT statement $\Phi(s)$ in the language of streams alone (no $\mathbb{R}_c$, no α_b) such that $\Phi(s_\pi) \leftrightarrow \text{ChurchRosserlikeProperty}(s_\pi)$, where the property captures $\alpha_{16}(s_\pi) = \pi - 3$ via a purely stream-theoretic invariant.

A candidate: stream-bisimulation between s_π and the unfold of an alternative coalgebra (Machin, Leibniz after base conversion). This makes the correctness theorem a bisimulation-only statement, but at the cost of relating *distinct combinatorial* algorithms via coinduction.

10.2 The carry-bisimulation as HIT vs. propositional truncation

The quotient $\mathsf{Stream}/{\sim_b}$ in Theorem 4.7 can be presented as either (i) a set-quotient HIT, or (ii) a Voevodsky propositional truncation of the equivalence relation. The two are HoTT-equal by univalence-for-sets but differ in computation.

Open Problem 10.2. Compare the cubical realisation of $\mathsf{Stream}/{\sim_b}$ as set-HIT vs. propositional truncation. Which presentation makes $\beta_b : \mathsf{Stream}/{\sim_b} \simeq [0, 1]$ *computational* (i.e. $\beta_b \circ \beta_b^{-1} \equiv \mathrm{id}$ judgmentally)?

10.3 Productivity and the totality of unfold

In MLTT-without-cubical, the totality of unfold requires a guardedness check on the coalgebra. In our case the BBP and Sale algorithms are guarded by construction (each step emits exactly one digit and modifies state), but a fully formal proof of guardedness is currently absent in our Lean artefact.

Open Problem 10.3. Formalise productivity of the BBP and Sale digit-extraction algorithms in Lean 4/Mathlib using the `Stream'` API and the `Coinductive` machinery.

10.4 Higher transcendentals

The methods of this paper apply (with the obvious modifications) to other transcendentals with known spigot/streaming algorithms:

- Catalan's constant $G = \sum_{n=0}^{\infty}(-1)^n/(2n+1)^2$
- Apéry's constant $\zeta(3)$ (Apéry's series gives a streaming algorithm)
- Euler–Mascheroni γ (no known streaming algorithm of state poly(n), an open problem in itself)

Open Problem 10.4. Extend the coalgebraic characterisation to $\zeta(3)$, Catalan's constant, and (failing that) prove that γ has no finite-state stream coalgebra unfold-equal to it.

10.5 Coalgebraic transcendence proofs

The Lindemann–Weierstrass theorem proves the algebraic independence of $e^{\alpha_1}, \ldots, e^{\alpha_n}$ for $\mathbb{Q}$-linearly independent algebraic α_i. Paper V §8.3–8.4 marks this as "partially formalisable" in $\mathbb{R}_c$. A coalgebraic proof would require:

- A coalgebraic structure on the algebraic numbers $\overline{\mathbb{Q}}$ (perhaps as a final coalgebra of an "algebraic-extension" functor).
- A coalgebraic exp as the unique coalgebra morphism $\mathbb{R}_c \to \mathsf{Stream}\,\mathsf{Fin}(b)$ from the ODE $f' = f$ universal property.

Open Problem 10.5. Formulate a coalgebraic Lindemann–Weierstrass theorem in HoTT.

11 Conclusion

We have given a coinductive characterisation of π and e in homotopy type theory, completing the programme sketched in Paper III §5 of the prior series. The principal new results are:

- A clean statement of the coinduction principle in cubical HoTT (Theorem 3.4).
- Stream coalgebras realising digit expansions (Theorem 4.1, Theorem 4.7).
- Coalgebraic characterisations of π and e as unique unfolds satisfying bisimulation-closed observable predicates (Theorem 6.4, Theorem 6.9, Theorem 6.10, Theorem 6.11).
- An equivalence theorem (Theorem 8.1) showing the coinductive presentation matches Booij's HIIT $\mathbb{R}_c$ on $[0, 1]$.
- A Dirichlet-series stream coalgebra (Theorem 9.2) giving partial progress towards a coalgebraic ζ.

An accompanying executable framework is provided in Haskell: the stream functor, unfold/corecursion, the carry-bisimulation, and the spigot algorithms for π (Leibniz, Machin) and e (factorial series), together with QuickCheck properties for stream operations and approximation round-trips. The Lean 4 companion file (using Mathlib's `Stream'` API) formalises the basic coalgebra structure, the corecursive `unfold`, and a (classical, set-level) coinduction principle; it does *not* include a formal proof of the productivity of the Sale or BBP transitions, which is left as Theorem 10.3. The Lean component should be read as a partial certification of the type-theoretic infrastructure, not as a complete formalisation of the main theorems. The full formalisation effort is substantial: the main open problem (Theorem 9.1) is the construction of a HoTT-native coalgebraic Riemann zeta function with $\zeta(s) = 0$ as a bisimulation-closed predicate; this remains far from solved.

Coinductively, the picture has unified beauty: π is a *stream*, namely the unique stream produced by the BBP corecursive equation; e is a *stream*, namely the unique stream produced by the factorial-series corecursive equation; $[0, 1]$ is the *type of streams modulo carry*; and equality of streams is exactly bisimulation. The HIIT route of Booij is one face of these objects; the final-coalgebra route is the other. Univalence binds them.

A Prerequisites from the prior series, made explicit

To make the present paper self-contained, we restate (without re-proving) the prior results on which our arguments rely. Each is either a textbook fact found in the standard references on HoTT and coalgebra (see [18, 4, 5, 7, 6]), or a routine adaptation of a textbook fact to the cubical setting. Where we cite the prior series [1, 2, 3], the citation is to a particular textbook-style adaptation, and the underlying proofs are the standard ones.

Theorem A.1 (A.1; cf. Paper III Theorem 5.2)**.** *For any type $A : \mathcal{U}$, the M-type*

$$M(A, \lambda-.\mathbf{1}) \cong A^{\mathbb{N}}$$

together with $\mathsf{out} := \langle \mathsf{hd}, \mathsf{tl} \rangle$ *is a final coalgebra of the endofunctor $F_A\, X = A \times X$.*

This is the cubical-HoTT version of Rutten [5] §2 *combined with* ACS [7] Theorem 3.5. We use it in the form proved by ACS.

Theorem A.2 (A.2; cf. Paper III Theorem 5.3)**.** *For final coalgebra $(\nu F, \mathsf{out})$ of polynomial endofunctor F, two elements $x, y : \nu F$ satisfy a path equality $x = y$ iff there exists an F-bisimulation R with $R(x, y)$.*

This is restated as our Theorem 3.4 and proved internally to the present paper.

Theorem A.3 (A.3; cf. Paper III Theorem 4.1)**.** *There is, up to unique order-preserving HoTT-equivalence, a unique Dedekind-complete Archimedean ordered field $\mathbb{R}$. Booij's Cauchy HIIT $\mathbb{R}_c$ is one such; the quotient $\big(\mathbb{Z} \times (\mathsf{Stream\,Fin}(b))\big)/{\sim_b'}$ for the natural extension of $\sim_b$ to $\mathbb{Z}$-shifted streams is another.*

We use this in the form: the two presentations agree on $[0, 1]$, which is Theorem 8.1.

Theorem A.4 (A.4; cf. Paper V Definition 8.1, Theorem 5.7). *In Booij's HIIT $\mathbb{R}_c$, the type $\Sigma_{p:\mathbb{R}_c}(\sin p = 0) \times (p > 0) \times (\forall x \in (0,p), \sin x > 0)$ is contractible. Its centre is by definition π.*

Theorem A.5 (A.5; cf. Paper V Definition 8.2, Theorem 4.3). *In Booij's HIIT $\mathbb{R}_c$, the type $\Sigma_{f:\mathbb{R}_c\to\mathbb{R}_c}\Sigma_{x:\mathbb{R}_c}(f' = f) \times (f(0) = 1) \times (x = f(1))$ is contractible (the universal-property of* exp*). Its centre's x-component is by definition e.*

These five statements are the only places where the prior series enters our arguments. Reader who prefers to take them as *assumptions* of the present paper may do so; the rest of the paper then becomes a self-contained development atop these five axioms together with the standard cubical-HoTT infrastructure of [8, 7].

B Detailed state-space layout for the proof of Theorem 6.10

We expand the state-space promise of Theorem 6.10. The streaming-factorial spigot algorithm of [12, 14] maintains a state in $\mathbb{N}^5$: (n, k, p, q, d) where n is the digit position emitted so far, k is the next factorial index ready to be summed, and (p, q, d) are the bookkeeping triple of (current scaled remainder numerator, current scaled denominator, current digit-out).

The transition $\gamma_e : \mathbb{N}^5 \to \mathsf{Fin}(b) \times \mathbb{N}^5$ is:

1. Multiply (p, q) by b: $(p', q') := (b \cdot p, q)$.
2. Extract the integer part: $d' := p' \div q'$, $p'' := p' \mod q'$. (In the cubical implementation $\div$ and $\mod$ are total constructive operations on $\mathbb{N}$.)
3. Update the position: $n' := n + 1$, $k' := k$ (unchanged).
4. If the remainder p''/q' is below the precision threshold $1/q'$, advance the factorial: $q'' := q' \cdot k'$, $k'' := k' + 1$, and re-normalise p'' by adding the new term's contribution.
5. Output: $(d', (n', k'', p_{\text{new}}, q''))$, where p_{new} is the renormalised remainder.

The state projection function state : $\mathsf{Stream}\,\mathsf{Fin}(b) \to \mathbb{N}^5$ of Theorem 6.10 is the canonical retraction: $\text{state}(s) =$ the unique c such that $s = \mathsf{unfold}_{\gamma_e}(c)$, when such c exists. This is a *partial* function in general, but defined on the image of $\mathsf{unfold}_{\gamma_e}$, which is the only place it is used.

Compatibility of the up-to operator. The up-to operator Φ used in the proof of Theorem 6.10 is

$$\Phi(R) \;:=\; \{(u_1, u_2) : \exists c, c'.\, R(\mathsf{unfold}_{\gamma_e}(c), \mathsf{unfold}_{\gamma_e}(c')) \wedge u_1 = \mathsf{unfold}_{\gamma_e}(c) \wedge u_2 = \mathsf{unfold}_{\gamma_e}(c')\}.$$

By [10, §6], an up-to operator is compatible iff it preserves the *evolution* structure of the coalgebra. For our γ_e this reduces to the observation that $\mathsf{unfold}_{\gamma_e}$ is a coalgebra morphism: heads commute with Φ by definition (they are determined by the first projection of γ_e), and tails commute by the recursive call. Hence Φ is compatible and Theorem 5.6 applies.

References

[1] Long, M. (2026). *Universal Properties of Real Numbers and Transcendentals*. Paper III, YonedaAI HoTT Foundations Series.

[2] Long, M. (2026). *The HoTT Perspective: Numbers as Inductive Types up to Path Equivalence*. Paper V, YonedaAI HoTT Foundations Series.

[3] Long, M. (2026). *The Univalent Correspondence: How Six Perspectives on Number Become One*. Synthesis Paper, YonedaAI HoTT Foundations Series.

[4] Booij, A. (2020). *Analysis in Univalent Type Theory*. Ph.D. Thesis, University of Birmingham.

[5] Rutten, J.J.M.M. (2000). Universal coalgebra: a theory of systems. *Theoretical Computer Science* 249(1):3–80.

[6] Jacobs, B. (2016). *Introduction to Coalgebra: Towards Mathematics of States and Observation*. Cambridge Tracts in Theoretical Computer Science.

[7] Ahrens, B., Capriotti, P., & Spadotti, R. (2015). Non-wellfounded trees in homotopy type theory. *Proc. TLCA*.

[8] Cohen, C., Coquand, T., Huber, S., & Mörtberg, A. (2018). Cubical Type Theory: a constructive interpretation of the univalence axiom. *IfCoLog Journal of Logics*.

[9] Birkedal, L., Møgelberg, R.E., Schwinghammer, J., & Støvring, K. (2012). First steps in synthetic guarded domain theory: step-indexing in the topos of trees. *Logical Methods in Computer Science* 8.

[10] Pous, D., & Sangiorgi, D. (2012). Enhancements of the bisimulation proof method, in *Advanced Topics in Bisimulation and Coinduction*, CUP.

[11] Ghani, N., Hancock, P., & Pattinson, D. (2009). Continuous functions on final coalgebras. *ENTCS* 249:3–18.

[12] Sale, A.H.J. (1968). The calculation of e to many significant digits. *The Computer Journal* 11(2):229–230.

[13] Rabinowitz, S., & Wagon, S. (1995). A spigot algorithm for the digits of π. *American Math. Monthly* 102:195–203.

[14] Gibbons, J. (2006). Unbounded spigot algorithms for the digits of pi. *American Math. Monthly* 113:318–328.

[15] Bailey, D.H., Borwein, P., & Plouffe, S. (1997). On the rapid computation of various polylogarithmic constants. *Mathematics of Computation* 66:903–913.

[16] Weihrauch, K. (2000). *Computable Analysis: An Introduction*. Springer.

[17] Loeffler, D., & Stoll, M. (2025). Formalizing zeta and L-functions in Lean. *Annals of Formalized Mathematics* 1. arXiv:2503.00959.

[18] The Univalent Foundations Program. (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study.

[19] Voevodsky, V. (2010). The univalence axiom and the foundations of mathematics. Lecture series, IAS.

[20] Licata, D.R., & Shulman, M. (2013). Calculating the fundamental group of the circle in homotopy type theory. *Proc. LICS*.

[21] Coquand, T. (1994). Infinite objects in type theory. *Types for Proofs and Programs*, Springer LNCS.

[22] Rutten, J.J.M.M. (2003). Behavioural differential equations: a coinductive calculus of streams, automata, and power series. *Theoretical Computer Science* 308:1–53.

[23] Aczel, P. (1988). *Non-Well-Founded Sets*. CSLI Lecture Notes 14.

[24] Bordg, A., & Paulson, L. (2024). Coinductive proofs in Isabelle: streams revisited. *Journal of Automated Reasoning*.

[25] Kleene, S.C. (1952). *Introduction to Metamathematics*. North-Holland.

[26] Turing, A.M. (1937). On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.* 42:230–265.

[27] Kapulkin, K., & Lumsdaine, P.L. (2021). The simplicial model of univalent foundations (after Voevodsky). *Journal of the European Mathematical Society* 23:2071–2126.

[28] Shulman, M. (2018). Brouwer's fixed-point theorem in real-cohesive homotopy type theory. *Math. Structures Comp. Sci.* 28:856–941.

[29] Sangiorgi, D. (2012). *Introduction to Bisimulation and Coinduction*. Cambridge University Press.

[30] Abbott, M., Altenkirch, T., & Ghani, N. (2005). *Containers: Constructing Strictly Positive Types*. Theoretical Computer Science 342:3–27.