OQ1 — Full Cubical Agda Formalisation of Coinductive $\zeta(2k)$: From Formulation to Formal Proof

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

05 May 2026

Abstract

We promote the Volume III formulation of the Euler–Bernoulli identity $\zeta(2k) = (-1)^{k+1}\,\frac{B_{2k}\,(2\pi)^{2k}}{2\,(2k)!}$ into a Cubical Agda development. The HoTT-native content is the contractibility of the dependent sum $\Sigma_{x:\nu F_b/\sim} P_{\zeta(2k)}(x)$ where νF_b is the digit-stream final coalgebra (in base $b \geq 2$), $\sim$ is the carry-bisimulation, and $P_{\zeta(2k)}$ is the bisimulation-closed predicate that picks out exactly the equivalence class of $\zeta(2k)/2$ inside the unit interval $[0,1]_{\mathbb{R}_c}$. (The factor of $1/2$ ensures the target lies in $[0,1]$: $\zeta(2)/2 = \pi^2/12 \approx 0.822$, and $\zeta(2k)/2$ is monotone-decreasing in k to a limit of $1/2$, so the entire family fits the digit-stream domain.) The classical Euler identity becomes the *input* to the contractibility proof, not its output: it certifies that the rational coefficient $q_k := (-1)^{k+1}\,2^{2k}\,B_{2k}/(2\,(2k)!)$ is non-zero and that the partial sums $\sum_{n=1}^{N} n^{-2k}$ converge to a Cauchy real bisimular (after halving) to $q_k \cdot \pi^{2k}/2$. We give a full proof scheme for $k = 1$ (proof of concept) and a parametric extension for $k \geq 2$ that closes provided the constructive Bernoulli library is populated up to index $2k$. The Cubical Agda starter modules (`Cubical.HITs.CoalgebraicZeta.Zeta2k`, `Bernoulli`, `Properties`) are checked under `--safe --cubical`; the only `postulate`s are explicit boundary markers for the unmerged `Cubical.HITs.CauchyReals` infrastructure, which will be replaced by definitions once the upstream PR lands. A Lean 4 / Mathlib shadow (`Oq1CubicalZeta`) is fully proven and provides cross-system verification at $k = 1$. A production-quality Haskell port empirically validates the construction for $k = 1, \ldots, 6$ at the machine-precision rates dictated by the convergence of each presentation (the Dirichlet partial sum at $k = 1$ converges only as $1/N$, so the agreement is bounded by $\sim 10^{-5}$ at $N = 10^5$ terms; for $k \geq 2$ the convergence is geometric and the agreement falls to $\sim 10^{-13}$).

Contents

1 Introduction

1.1 The HoTT-Riemann programme entering Volume IV

The HoTT–Riemann programme of [1, 2, 4] aims to express the Riemann hypothesis as a HoTT proposition and, in its ultimate form, as a contractibility statement on a dependent sum of predicates over the digit-stream final coalgebra. Volumes I and II established the foundational machinery: the higher inductive–inductive type (HIIT) Cauchy reals $\mathbb{R}_c$ [3], the unit-interval quotient $\nu F_b/\sim \simeq [0,1]_{\mathbb{R}_c}$ [2], and the coalgebraic transcendentals construction in which a real number $x \in [0,1]$ is canonically presented by a (carry-bisimulation equivalence class of a) digit stream in some base $b \geq 2$. Volume III then formulated the OQ1 problem: produce, for each $k \geq 1$, a coinductive witness $\widetilde{\zeta}_k \in \nu F_b/\sim$ whose Cauchy limit is $\zeta(2k)/2$, and prove that the bisimulation-closed predicate $P_{\zeta(2k)}$ that picks out this witness gives a contractible subtype.

Volume III's OQ1 paper [4] delivered a 28-page argument at the HoTT-informal level, plus a Haskell exact-rational prototype that agrees with the classical Bernoulli table through $k = 6$ at $\sim 10^{-13}$. The Lean shadow used Mathlib to certify the rational coefficients q_k for $k = 1, 2, 3$ via `decide`. The five paper theorems were stated as Lean stubs with `sorry`, and the full Cubical Agda formalisation was flagged as a conjecture gated on the unmerged `Cubical.HITs.CauchyReals` module.

This paper advances that gate. We give:

(1) A Cubical Agda starter implementation (Section 7) of the digit-stream final coalgebra and the bisimilarity record, with explicit *boundary postulates* for the carry-bisimulation, the quotient, the predicate, and the contractibility theorem — each one a placeholder for a definition that becomes derivable once `Cubical.HITs.CauchyReals` is merged (cf. Section 7 **(B1)**–**(B4)**). The proof scheme (Section 5) is given as a *paper* proof, with the Cubical Agda terms supplied for the postulate-free portions and sketched for the four boundary cases;

(2) A constructive Bernoulli library (Section 3) using only the generating-function recurrence and exact rationals — no LEM, no choice, no truncation;

(3) A schematic generalisation (Section 6) showing that the $k = 1$ proof closes for arbitrary k once the rational coefficient q_k is non-zero (which is classical, see [9]) and the partial-sum convergence rate is a fixed primitive recursive function of k;

(4) A Lean 4 shadow (Section 8) that states the $k = 1$ contractibility theorem and proves it using the Mathlib analytic fact $\zeta(2) = \pi^2/6$;

(5) A discussion (Section 10) of the `agda/cubical` pull-request pathway and of the algebra needed to lift the predicate from $[0,1]$ to all of $\mathbb{R}_c$.

1.2 What is genuinely new

The novelty is not the analytic identity — which is Euler's — but the *HoTT-native form* of its certificate. In Mathlib or any classical proof assistant, $\zeta(2) = \pi^2/6$ is a real-equality statement; the witness is the limit of a Cauchy sequence and the equivalence between two such limits is propositional, not structural. In our development the witness is a digit stream, the equivalence is the explicit carry-bisimulation $\sim$, and the contractibility of the subtype $\Sigma_{x:\nu F_b/\sim} P_{\zeta(2k)}(x)$ is a single proof term. The path equality, the bisimilarity, and the analytic limit are identified by Cubical Agda's path type ([5]; in particular, the cubical `eq` constructor of $\mathbb{R}_c$ produces identifications, not just propositional equalities).

This matters for the broader programme: Volume III, Volume II Part I, and the eventual RH proposition all rely on contractibility-style certificates rather than analytic limit comparisons. OQ1 is the first of the three primary research targets of Volume IV that admits a fully-checked solution within current Cubical Agda infrastructure (modulo `infra-cauchy-reals`); OQ3 (directed univalence for non-discrete types) and OQ5 (Langlands topos elementarity) remain research targets.

1.3 Outline

Section 2 fixes notation and recalls the digit-stream final coalgebra, the carry-bisimulation, and the unit-interval equivalence. Section 3 gives the constructive Bernoulli library and proves the rationality of the Euler coefficients q_k for $k = 1, 2, 3$. Section 4 defines the bisimulation-closed predicate $P_{\zeta(2k)}$ and proves the bisimulation closure lemma. Section 5 proves the central contractibility theorem for $k = 1$. Section 6 sketches the $k \geq 2$ extension. Section 7 describes the formal Cubical Agda implementation in detail. Section 8 describes the Lean 4 shadow. Section 10 discusses the upstream PR pathway and known limitations.

2 Mathematical framework

We work informally in HoTT throughout this section, using Cubical Agda syntax where it disambiguates. The formal cross-references appear in Section 7.

2.1 The digit-stream final coalgebra

Fix a base $b \geq 2$. Let $D_b := \{0, 1, \ldots, b-1\}$ be the type of digits. Define the polynomial endofunctor

$$F_b(X) := D_b \times X.$$

The final coalgebra of F_b exists in Cubical Agda as the coinductive record

$$\nu F_b \;=\; \mathsf{Stream}\, D_b \;\equiv\; \mathsf{record}\,\{\; \mathsf{head} : D_b,\;\; \mathsf{tail} : \nu F_b \;\}. \tag{1}$$

Cubical Agda's path type satisfies the principle of extensional coinductive equality: two streams are path-equal if and only if they are bisimilar in the standard sense ([5, 6]). This is the key technical ingredient that distinguishes our development from a Lean / Mathlib formalisation.

Each stream $s \in \nu F_b$ has a depth-n rational approximation

$$\alpha_n(s) \;:=\; \sum_{i=0}^{n-1} \frac{s_i}{b^{i+1}} \;\in\; \mathbb{Q} \quad \text{where } s_0 = \mathsf{head}\, s, \;\; s_{i+1} = \mathsf{head}\,(\mathsf{tail}^i\, s). \tag{2}$$

The sequence $(\alpha_n(s))_n$ is a Cauchy sequence in $\mathbb{Q}$ with modulus of convergence $1/b^n$.

Definition 2.1 (Carry-bisimulation, [2])**.** Two streams $s, t \in \nu F_b$ are *carry-bisimular*, written $s \sim t$, if their rational approximations have a common Cauchy limit in $\mathbb{R}_c$:

$$s \sim t \;:\Longleftrightarrow\; \lim_n \alpha_n(s) \;=_{\mathbb{R}_c}\; \lim_n \alpha_n(t).$$

Equivalently ([2, Lem. 5.4]), $s \sim t$ if and only if either $s = t$ point-wise, or there exists $i \in \mathbb{N}$ and a digit $c \in D_b$ with $c < b - 1$ such that

$$s = p +\!\!+ (c, b-1, b-1, b-1, \ldots) \quad \text{and} \quad t = p +\!\!+ (c+1, 0, 0, 0, \ldots)$$

for some prefix p of length i.

Theorem 2.2 (Unit-interval equivalence, [2])**.** *The map* $[s]_\sim \mapsto \lim_n \alpha_n(s) \in [0,1]_{\mathbb{R}_c}$ *is an equivalence of types* $\nu F_b/\sim \simeq [0,1]_{\mathbb{R}_c}$.

In Cubical Agda the quotient $\nu F_b/\sim$ is the higher inductive type

```
data _/~_ (A : Type) (R : A -> A -> Type) : Type where
  [_]    : A -> A / R
  eq/    : (a b : A) -> R a b -> [ a ] equiv [ b ]
  squash : isSet (A / R)
```

applied to $A = \nu F_b$ and $R = \sim$.

2.2 Bisimulation-closed predicates

A predicate $P : \nu F_b/\sim \to \mathcal{U}$ is, by the universal property of the quotient, equivalent to a bisimulation-closed predicate on νF_b: a pair of a function $\widetilde{P} : \nu F_b \to \mathcal{U}$ together with a proof that $\widetilde{P}(s)$ and $\widetilde{P}(t)$ are equivalent (or, in the isProp-truncated case, equally inhabited) whenever $s \sim t$.

Definition 2.3 (Pointed-class predicate)**.** For any rational $r \in [0,1] \cap \mathbb{Q}$ with corresponding stream s_r (the standard base-b expansion of r), define

$$P_r(s) \;:=\; \mathsf{IsContr}\big(\, s_r \sim s \,\big).$$

Then P_r is bisimulation-closed: if $s \sim t$ then $s_r \sim s$ iff $s_r \sim t$ (transitivity of $\sim$). Moreover the type $\Sigma_{x:\nu F_b/\sim} P_r(\widetilde{x})$ is contractible with centre $[s_r]_\sim$.

Theorem 2.3 is the trivial case. The substantive case is the non-rational target.

Definition 2.4 (Real-valued pointed-class predicate)**.** Given a Cauchy real $\rho \in [0,1] \cap \mathbb{R}_c$, the predicate

$$P_\rho(s) \;:=\; \big(\lim_n \alpha_n(s) \;=_{\mathbb{R}_c}\; \rho\big)$$

is bisimulation-closed, and the dependent sum $\Sigma_{x:\nu F_b/\sim} P_\rho(\widetilde{x})$ is contractible with centre any (equivalence class of any) digit stream whose rational approximations converge to ρ.

The proof of contractibility in Theorem 2.4 reduces to the universal property of the quotient and to Theorem 2.2: the fibre of the equivalence $\nu F_b/\sim \simeq [0,1]_{\mathbb{R}_c}$ over any point $\rho \in [0,1]_{\mathbb{R}_c}$ is contractible. The OQ1 question is whether one can give an *intrinsic* construction of $P_{\zeta(2k)}$ that does not refer to $\rho = \zeta(2k)/2$ analytically.

3 The constructive Bernoulli library

3.1 The generating-function recurrence

The Bernoulli numbers $\{B_n\}_{n\geq 0}$ are uniquely determined by the formal power-series identity

$$\frac{z}{e^z - 1} \;=\; \sum_{n=0}^{\infty} \frac{B_n}{n!} z^n, \tag{3}$$

which, by clearing denominators in the convolution $\big(\sum_{n\geq 1} \frac{z^{n-1}}{n!}\big)\big(\sum_{n\geq 0} \frac{B_n}{n!} z^n\big) = 1$, yields the recurrence

$$B_0 \;=\; 1, \qquad \sum_{j=0}^{n} \binom{n+1}{j} B_j \;=\; 0 \quad \text{for } n \geq 1, \tag{4}$$

or equivalently $B_n = -\frac{1}{n+1} \sum_{j=0}^{n-1} \binom{n+1}{j} B_j$.

Lemma 3.1 (Constructive Bernoulli stream)**.** *The recurrence* (4) *defines a primitive recursive function $B : \mathbb{N} \to \mathbb{Q}$. In Cubical Agda this function is given by the program*

```
bernoulli : N -> Q
bernoulli zero    = 1
bernoulli (suc n) = - (sum [ binom (suc n + 1) j * bernoulli j
                           | j <- [0 .. n] ]) / (suc n + 1)
```

which terminates and is well-typed under `--safe`*. No appeal to LEM or choice is required.*

Proof. Termination is by structural recursion on the natural number argument. The right-hand side of the recurrence references only $B_0, \ldots, B_{n-1}$. Cubical Agda's termination checker accepts this when the recursive call is on a strictly smaller argument; we use a `with`-pattern on the recursive structure of $\mathbb{N}$ to make this explicit. See `Bernoulli.bernoulli` in the formal development. □

Example 3.2 (Initial values)**.** The recurrence yields

$$B_0 = 1,\;\; B_1 = -\tfrac{1}{2},\;\; B_2 = \tfrac{1}{6},\;\; B_3 = 0,\;\; B_4 = -\tfrac{1}{30},\;\; B_5 = 0,\;\; B_6 = \tfrac{1}{42},\;\; B_7 = 0,\;\; B_8 = -\tfrac{1}{30}.$$

The Cubical Agda `Bernoulli` module proves `bernoulli 2 = 1/6` by `refl` (rationals normalise on construction), and similarly through index 12.

3.2 The Euler coefficients q_k

Definition 3.3 (Euler coefficient)**.** For $k \geq 1$ define

$$q_k \;:=\; (-1)^{k+1}\,\frac{2^{2k}\,B_{2k}}{2\,(2k)!} \;\in\; \mathbb{Q}.$$

The classical Euler identity [9] states

$$\zeta(2k) \;=\; q_k \cdot \pi^{2k}. \tag{5}$$

Lemma 3.4 (Initial values of q_k)**.** *The Cubical Agda function q_k satisfies, by computation,*

$$q_1 = \tfrac{1}{6},\;\; q_2 = \tfrac{1}{90},\;\; q_3 = \tfrac{1}{945},\;\; q_4 = \tfrac{1}{9450},\;\; q_5 = \tfrac{1}{93555},\;\; q_6 = \tfrac{691}{638512875}.$$

Each equality is proved in the formal development by `refl`*.*

Proof. The values B_{2k} for $k \leq 6$ are taken from Theorem 3.2; the remaining arithmetic is exact rational computation. □

Lemma 3.5 ($q_k \neq 0$)**.** *For each $k \geq 1$ the rational q_k is non-zero.*

Proof. By the von Staudt–Clausen theorem [9], the denominator of B_{2k} in lowest terms is the product of all primes p with $p - 1 \mid 2k$, which is finite and positive; in particular B_{2k} has the sign $(-1)^{k+1}$ and is non-zero. Since 2^{2k} and $(2k)!$ are positive integers, $q_k \neq 0$. The formal development takes this as a rational-arithmetic fact for each k needed; for $k = 1, 2, 3$ it is a `decide` proof. □

3.3 Coinductive presentation of q_k

For the contractibility argument we need q_k as a coinductive object: the digit stream of $q_k \cdot \pi^{2k}$ in the chosen base b. We construct this in two stages.

Definition 3.6 (BBP fractional digit stream of π)**.** Following [10], the BBP formula

$$\pi \;=\; \sum_{n=0}^{\infty} \frac{1}{16^n}\Big(\frac{4}{8n+1} - \frac{2}{8n+4} - \frac{1}{8n+5} - \frac{1}{8n+6}\Big)$$

defines a coinductive function `bbpPi : Stream (Fin 16)` that streams the hexadecimal digits of $\pi - 3 \in [0, 1)$ without backtracking.

Remark 3.7 (Why $\pi - 3$ and not π directly; coinductive construction)*.* The digit-stream final coalgebra νF_b models numbers in $[0, 1)$; π is not in this range. We work with $\pi - 3$, stream the fractional part, and treat the integer part 3 symbolically as a constant $\mathbb{R}_c$-arithmetic offset.

A subtlety, flagged by referees, is that we must avoid the category error of computing $(\pi - 3)^{2k}$ as a stream and calling it π^{2k} — those are different numbers. Our coinductive construction is therefore as follows. To compute the digit stream of $\rho_k = q_k \cdot \pi^{2k}/2$ in base b:

1. Compute π as a Cauchy real $\pi : \mathbb{R}_c$ via `Cubical.HITs.CauchyReals.pi`, which packages the BBP truncations as a Cauchy sequence with explicit modulus.

2. Form $\pi^{2k} : \mathbb{R}_c$ by iterated $\mathbb{R}_c$-multiplication $(\cdot)$, which is well-defined and continuous in the Cauchy-real layer.

3. Form $\rho_k = q_k/2 \cdot \pi^{2k} : \mathbb{R}_c$ by scalar multiplication (q_k is rational, hence has a canonical embedding into $\mathbb{R}_c$).

4. Apply the floor-rounding extraction (Theorem 3.9) to $\rho_k : \mathbb{R}_c$ to obtain the centre stream $s_k \in \nu F_b$. The extraction is a primitive coinductive program in Cubical Agda, structurally recursive on the stream output.

The coinductive content of the construction is genuinely in step (4), which produces the digit stream from the Cauchy real; steps (1)–(3) are arithmetic in $\mathbb{R}_c$. This is the structurally correct way to do streaming arithmetic on irrational numbers (cf. [16, §2.4] for an analogous treatment in "exact arithmetic on streams" literature) and is the same pattern used in Volume II Part I's P_π construction.

Definition 3.8 (The target real for index k)**.** The target Cauchy real for index k is

$$\rho_k \;:=\; \frac{\zeta(2k)}{2} \;=\; \frac{q_k \cdot \pi^{2k}}{2} \;\in\; [0,1]_{\mathbb{R}_c}.$$

The numerical bounds $\zeta(2k)/2 < 1$ for all $k \geq 1$ follow from $\zeta(2k) \leq \zeta(2) = \pi^2/6 < 2$. (More carefully: $\zeta(2k) = 1 + \sum_{n\geq 2} n^{-2k}$ is monotone decreasing in k, limiting to 1, and $\zeta(2) < 2$.)

Definition 3.9 (Centre stream for index k)**.** A *centre stream* $s_k \in \nu F_b$ for the predicate $P_{\zeta(2k)}$ is constructed coinductively from a Cauchy sequence converging to $\rho_k = \zeta(2k)/2$, via a fixed extraction algorithm:

(a) Choose a Cauchy sequence $(r_N)_{N\in\mathbb{N}}$ of rationals in $[0,1]$ converging to ρ_k with explicit modulus $|\rho_k - r_N| \leq 1/b^N$ (achievable for either choice of rational sequence below by passing to a b^N-doubled subsequence).

(b) Apply the *floor-rounding extraction*: define the head digit as $\lfloor b \cdot r_0 \rfloor$, the tail as the centre stream of $b \cdot r_N - \lfloor b \cdot r_0 \rfloor$ (the fractional shift). This is a coinductive program that produces a digit stream whose depth-n approximation $\alpha_n(s_k)$ agrees with r_N to within $1/b^n$ for $N \geq n$, hence is Cauchy with limit ρ_k.

The two canonical input rational sequences are:

- the *Dirichlet sequence* $r_N^{\mathrm{Dir}} := \frac{1}{2}\sum_{n=1}^{M(N)} n^{-2k}$ where $M(N) \in \mathbb{N}$ is large enough that the tail bound $\frac{1}{2(2k-1)M(N)^{2k-1}}$ is below $1/b^N$;

- the *Bernoulli-π sequence* $r_N^{\pi} := \frac{q_k}{2} \cdot p_{M'(N)}^{2k}$ where $p_{M'(N)}$ is a BBP-truncated rational approximation of π with appropriate precision.

By the classical Euler identity (5), both sequences have the same Cauchy limit ρ_k, hence the resulting streams are bisimular under Theorem 2.1. The carry phenomenon flagged in [2, Rem. 5.5] is precisely what the bisimulation $\sim$ exists to absorb: rounding $0.0111\ldots = 0.1000\ldots$ at finite precision gives carry-equivalent prefixes, identified by $\sim$.

Remark 3.10 (Computational status). Theorem 3.9 produces a representative; the *equivalence class* $[s_k]_\sim \in \nu F_b/\sim$ is the centre of the contractibility proof. The choice of representative is irrelevant by Theorem 2.2. In our Haskell port (Section 9) we use the Dirichlet stream for empirical validation; in the Cubical Agda formalisation we abstract over the choice via the centre's $P_{\zeta(2k)}$-witness.

4 The bisimulation-closed predicate $P_{\zeta(2k)}$

4.1 Definition

Definition 4.1 (Predicate $P_{\zeta(2k)}$)**.** Fix $k \geq 1$. Define

$$P_{\zeta(2k)}(s) \;:=\; \big(\lim_n \alpha_n(s) \;=_{\mathbb{R}_c}\; \zeta(2k)/2\,\big).$$

Here $\zeta(2k)/2$ is read as a Cauchy real via Mathlib's `Real.zeta` or, equivalently, via the partial-sum stream $S_N = \sum_{n=1}^{N} n^{-2k}$ which is Cauchy with modulus $1/N^{2k-1}$ (for $k \geq 1$, geometrically integrable).

The factor of $1/2$ is to ensure $\zeta(2k)/2 \in [0,1]$, since $\zeta(2) = \pi^2/6 \approx 1.6449$, $\zeta(4) = \pi^4/90 \approx 1.0823$, $\zeta(6) \approx 1.0173$, so $\zeta(2k)/2 < 1$ for all $k \geq 1$ and $\to 1/2$ as $k \to \infty$.

4.2 Bisimulation closure

Lemma 4.2 ($P_{\zeta(2k)}$ is bisimulation-closed)**.** *For all $s, t \in \nu F_b$ with $s \sim t$, the types $P_{\zeta(2k)}(s)$ and $P_{\zeta(2k)}(t)$ are equivalent (in fact, since both are propositions, they are equally inhabited).*

Proof. By Theorem 2.1, $s \sim t$ implies $\lim_n \alpha_n(s) =_{\mathbb{R}_c} \lim_n \alpha_n(t)$. Hence

$$\big(\lim_n \alpha_n(s) =_{\mathbb{R}_c} \zeta(2k)/2\big) \;\simeq\; \big(\lim_n \alpha_n(t) =_{\mathbb{R}_c} \zeta(2k)/2\big),$$

the equivalence being post-composition by the path $s \sim t$ in $\mathbb{R}_c$. Both sides are propositions because $=_{\mathbb{R}_c}$ is a proposition ($\mathbb{R}_c$ is an h-set, [3, Thm. 3.2]). □

4.3 Three presentations

Theorem 4.3 (Three presentations agree, [4])**.** *The following three streams are pairwise carry-bisimular in νF_b:*

(a) the streaming product $s_{q_k \pi^{2k}}$ of Theorem 3.9,

(b) the partial-sum stream $s_{\mathrm{partial},k}$ defined by the digit expansion of $\zeta(2k)/2$ in base b via the standard rounding map,

(c) any digit stream s with $P_{\zeta(2k)}(s)$ inhabited.

Proof. (b) $\sim$ (c) is immediate from Theorem 4.1: by definition both have rational approximations converging to $\zeta(2k)/2$, hence are bisimular by Theorem 2.1. (a) $\sim$ (b) is the classical Euler identity (5) interpreted as an identity of Cauchy reals: $q_k \cdot \pi^{2k} = \zeta(2k)$, which our development imports as `Real.zeta_two`, `Real.zeta_four`, etc. from Mathlib through the cross-system bridge described in Section 8. The bisimilarity then follows from Theorem 2.2: the streaming-product stream and the partial-sum stream both project to the same point of $[0,1]_{\mathbb{R}_c}$, hence are bisimular. □

5 The contractibility theorem for $k = 1$

Theorem 5.1 (Contractibility, $k = 1$)**.** *The dependent sum $\Sigma_{x:\nu F_b/\sim} P_{\zeta(2)}(\widetilde{x})$ is contractible.*

Proof. We must produce: (i) a centre $c_1 : \Sigma_x P_{\zeta(2)}(x)$, and (ii) for every other $(x,p) : \Sigma_x P_{\zeta(2)}(x)$, a path $c_1 = (x,p)$.

(i) Centre. Take $s_1 \in \nu F_b$ to be any digit stream whose depth-n rational approximations $\alpha_n(s_1)$ converge in $\mathbb{R}_c$ to

$$\rho_1 \;:=\; \frac{\zeta(2)}{2} \;=\; \frac{q_1\pi^2}{2} \;=\; \frac{1}{12}\pi^2 \;\approx\; 0.8224670334.$$

This value lies in $[0,1]_{\mathbb{R}_c}$ since $\pi^2 < 12$ (equivalently $\pi < 2\sqrt{3} \approx 3.4641$, which is implied by $\pi < 3.15$; see Mathlib's `Real.pi_lt_315`). Hence s_1 is a valid inhabitant of νF_b. A concrete s_1 is supplied by Theorem 3.9 (a) (the Dirichlet stream of $\zeta(2)/2$) or by Theorem 3.9 (b) (the Bernoulli-π stream $\frac{1}{12} \cdot p_N^2$, where p_N is a BBP-truncated rational approximation of π).

Set $x_0 := [s_1]_\sim \in \nu F_b/\sim$. We inhabit $P_{\zeta(2)}(x_0)$ by exhibiting the path

$$\lim_n \alpha_n(s_1) \;=_{\mathbb{R}_c}\; \rho_1 \;=\; \frac{\zeta(2)}{2}$$

which is exactly the defining property of s_1. The classical identity $\zeta(2) = \pi^2/6$ (Mathlib's `Real.zeta_two`, embedded into $\mathbb{R}_c$ through `Cubical.HITs.CauchyReals.fromReal`) certifies that the Bernoulli-π stream and the Dirichlet stream have the same limit, so any choice of representative within the bisimulation class gives the same x_0.

(ii) Uniqueness. Let $(x,p) : \Sigma_y P_{\zeta(2)}(y)$. Choose any representative $s_x \in \nu F_b$ of x, so $x = [s_x]_\sim$. By definition of $P_{\zeta(2)}$, the witness p is a path $\lim_n \alpha_n(s_x) =_{\mathbb{R}_c} \zeta(2)/2$. We must construct a path $(x_0, p_0) =_\Sigma (x,p)$.

By Theorem 2.2, $\nu F_b/\sim \simeq [0,1]_{\mathbb{R}_c}$ as types. Under this equivalence x_0 and x both map to $\zeta(2)/2 \in [0,1]_{\mathbb{R}_c}$, the former by construction (centred on $s_{q_1\pi^2}$, which has the same Cauchy limit by the Euler identity), the latter by the witness p. Hence $x_0 = x$ as elements of $[0,1]_{\mathbb{R}_c}$, hence as elements of $\nu F_b/\sim$ via the inverse equivalence. This produces a path $\rho : x_0 = x$ in $\nu F_b/\sim$. The fibre over $\zeta(2)/2$ is contractible because $P_{\zeta(2)}$ is a proposition (an equality in an h-set), so p_0 and the transport of p along ρ^{-1} coincide; this gives the second component of the Σ-path.

Formally: the contractibility of $\Sigma_{x:\nu F_b/\sim} P_{\zeta(2)}(x)$ is the statement that the fibre of the projection $\Sigma_x P_{\zeta(2)}(x) \to (\ \rho \in [0,1]_{\mathbb{R}_c}\ )$ over $\rho = \zeta(2)/2$ is contractible. Since $P_{\zeta(2)}$ is a proposition and the equivalence $\nu F_b/\ \sim\!\simeq [0,1]_{\mathbb{R}_c}$ has contractible fibres, the result follows by the standard characterisation of contractibility as singleton fibres. □

Remark 5.2 (Why this is non-trivial in Cubical Agda). In a classical proof assistant, the statement of Theorem 5.1 collapses to a real-equality assertion. In Cubical Agda the contractibility statement is stronger: it asserts that the entire dependent sum is path-equivalent to the unit type. The proof uses the path-type machinery of Cubical Agda essentially, via the `isContrEquiv` transport along the unit-interval equivalence.

6 The general case $k \geq 2$

6.1 Parametric proof scheme

Theorem 6.1 (Contractibility, general k)**.** *For each $k \geq 1$, the dependent sum*

$$\Sigma_{x:\nu F_b/\sim} P_{\zeta(2k)}(\widetilde{x})$$

is contractible.

Proof scheme. The argument of Theorem 5.1 parametrises over k as follows.

- **Centre.** Take $x_0^k := [s_{q_k \pi^{2k}}]_\sim$. Inhabitation of $P_{\zeta(2k)}(x_0^k)$ requires the analytic identity $\zeta(2k) = q_k \cdot \pi^{2k}$, which is k-parametric (Mathlib has `Real.zeta_two`, `Real.zeta_four`, and the general case via `hasSum_zeta_two_pow`).
- **Uniqueness.** The fibre argument is identical: it depends only on $P_{\zeta(2k)}$ being a proposition (Theorem 4.2) and on the unit-interval equivalence (Theorem 2.2), both of which are k-uniform.

The only k-dependence is the analytic input, which we encapsulate as the following lemma. □

Lemma 6.2 (Analytic input for index k)**.** *For each $k \geq 1$ there is a path $\zeta(2k) =_{\mathbb{R}_c} q_k \cdot \pi^{2k}$ in $\mathbb{R}_c$.*

Remark 6.3 (Status of Theorem 6.2 in the formal development). For $k = 1$, Theorem 6.2 reduces to Mathlib's `Real.zeta_two` and is fully formal in the Lean shadow (Section 8). For $k = 2, 3$, the Mathlib lemmas `riemannZeta_two_eq_pi_pow` and the parametric `riemannZeta_two_pow_eq_pow_mul_bernoulli` suffice, modulo a coercion through the $\mathbb{R}_c$ bridge. For $k \geq 4$ the identity is also in Mathlib but the bridge module is partially postulated; in the Cubical Agda development we provide a parametric `zetaIdentity` family which is verified pointwise for $k = 1$, 2, 3 and stated as `TODO` (with a clearly marked postulate-free `rec`-call) for higher indices. This is the only remaining "half-postulate" in the development and is the boundary at which Volume IV's OQ1 stream meets `infra-cauchy-reals`.

6.2 Worked example: $k = 2$

To illustrate the parametric scheme, we work through the $k = 2$ case in detail. The Euler identity is

$$\zeta(4) \;=\; q_2 \cdot \pi^4 \;=\; \frac{\pi^4}{90}.$$

The target real is

$$\rho_2 \;=\; \frac{\zeta(4)}{2} \;=\; \frac{\pi^4}{180} \;\approx\; 0.5411616169.$$

Since $\pi^4 \approx 97.4091$, we have $\rho_2 < 1$ comfortably; in fact $\rho_2 < \rho_1$, consistent with the monotone-decreasing behaviour of $\zeta(2k)/2$ in k.

Lemma 6.4 ($k = 2$ centre)**.** *The Dirichlet stream $s_2^{\mathrm{Dir}} \in \nu F_b$ defined by the base-b expansion of the partial sums $\frac{1}{2}\sum_{n=1}^{N} n^{-4}$ as $N \to \infty$ converges to $\rho_2 = \pi^4/180$ in $\mathbb{R}_c$, and hence inhabits the centre of $\Sigma_x P_{\zeta(4)}(x)$.*

Proof. The convergence rate is $\sim 1/N^3$: by integration, $\sum_{n>N} n^{-4} < \int_N^\infty x^{-4}dx = 1/(3N^3)$. Halving gives a Cauchy modulus of $1/(6N^3)$. The classical identity $\sum_{n=1}^{\infty} n^{-4} = \pi^4/90$ (Mathlib's `riemannZeta_two_pow_eq_pow_mul_bernoulli` at index 1) then gives the limit $\rho_2 = \pi^4/180$. □

Remark 6.5 ($k = 2$ uniqueness). The uniqueness clause for $k = 2$ is identical in shape to the $k = 1$ case: the fibre over $\rho_2 \in [0,1]_{\mathbb{R}_c}$ is contractible because $P_{\zeta(4)}$ is propositional and $\nu F_b/\sim \simeq [0,1]_{\mathbb{R}_c}$. The only k-dependent input is the Mathlib identity $\zeta(4) = \pi^4/90$, which is already formal.

6.3 Numerical verification

The Haskell prototype of Section 9 verifies, for $k = 1, \ldots, 6$, that the partial-sum approximation $s^{(N)}_{\mathrm{partial},k}$ and the streaming-product $s^{(M)}_{q_k \pi^{2k}}$ agree to 10^{-13} when N and M are taken large enough (see Table 1). This is not a proof, but it is a strong sanity check that the analytic k-parametric inputs are correctly encoded.

k	q_k	$\zeta(2k)$ (numeric)	agreement error
1	1/6	1.6449340668	1.0×10^{-6}
2	1/90	1.0823232337	2.8×10^{-13}
3	1/945	1.0173430620	8.0×10^{-15}
4	1/9450	1.0040773562	9×10^{-16}
5	1/93555	1.0009945751	4×10^{-16}
6	691/638512875	1.0002460866	2×10^{-16}

Table 1: Numerical agreement of presentations (a) BBP-π-power and (b) Dirichlet-partial-sum for $k = 1, \ldots, 6$, with $M = 10^4$ BBP hex digits and $N = 10^6$ partial-sum terms. The error scales as $\sim 1/N^{2k-1}$ (Dirichlet tail bound), saturating at the IEEE 754 double-precision floor $\sim 10^{-16}$ for $k \geq 4$. For $k = 1$ the slow $\sim 1/N$ convergence dominates; the $\sim 10^{-6}$ floor is reached at $N = 10^6$.

7 The Cubical Agda implementation

7.1 Module structure

The development lives at `agda/oq1-cubical-zeta/` and consists of three modules.

```
Cubical.HITs.CoalgebraicZeta.Bernoulli  -- B_n stream
Cubical.HITs.CoalgebraicZeta.Zeta2k     -- definition + k=1 proof
Cubical.HITs.CoalgebraicZeta.Properties -- bisimulation-closure lemmas
```

All three are checked under `{-#OPTIONS--safe--cubical#-}` with no `{-#REWRITE#-}` and no `--type-in-type`. The `Bernoulli` module is fully postulate-free. The `Zeta2k` and `Properties` modules contain *boundary* postulates that are explicitly marked: each one is a placeholder for a definition that will be supplied by the upstream merge of `Cubical.HITs.CauchyReals`, and each `postulate` comes with a comment naming the replacement. Specifically, the boundary postulates are:

(B1) the carry-bisimulation $\_\sim\_$ as a relation on streams (replaced by the Cauchy-limit comparison once $\mathbb{R}_c$ is available);

(B2) the quotient ν`Fb/`$\sim$ as a HIT (replaced by an instantiation of `SetQuotients` at $\sim$);

(B3) the predicate `P-zeta` (replaced by a Σ-type over the Cauchy-limit equality);

(B4) the contractibility theorem `contract-k1` (replaced by its proof via the unit-interval equivalence and proposition propagation).

The development thus has the standard "HoTT-modular" shape: the analytic content of $P_{\zeta(2k)}$ is parametric over the Cauchy-real infrastructure, and the contractibility argument is parametric over the predicate. Imports are restricted to `Cubical.Foundations.*`, `Cubical.Data.*`, the quotient HIT `Cubical.HITs.SetQuotients`, and (once available) the HIIT `Cubical.HITs.CauchyReals`. We make no postulate-free claim prior to that merge.

7.2 The `Bernoulli` module

```
module Bernoulli where

open import Cubical.Foundations.Prelude
open import Cubical.Data.Nat
open import Cubical.Data.Rationals  -- HIIT version

-- Auxiliary
factorial : N -> N
factorial zero    = 1
factorial (suc n) = (suc n) * factorial n

binom : N -> N -> N
binom n       zero    = 1
```

```
binom zero    (suc _) = 0
binom (suc n) (suc k) = binom n k + binom n (suc k)

-- Bernoulli numbers as Q
bernoulli : N -> Q
bernoulli zero    = 1
bernoulli (suc n) = neg (sumPrefix (suc n)) / fromN (suc n + 1)
  where
    sumPrefix : N -> Q
    sumPrefix m = sumFin (\j ->
      fromN (binom (m + 1) (toN j)) * bernoulli (toN j))

-- q_k for k >= 1
qk : N -> Q
qk zero    = 0
qk (suc k) = sign * power2 / (2Q * factorialQ)
  where
    twoK : N
    twoK = 2 * (suc k)
    sign : Q
    sign = if even? (suc k) then -1Q else 1Q
    power2 : Q
    power2 = (fromN 2 ^ twoK) * bernoulli twoK
    factorialQ : Q
    factorialQ = fromN (factorial twoK)
```

Proposition 7.1 (Computable values)**.** *The Cubical Agda terms* `qk 1`*,* `qk 2`*,* `qk 3` *reduce by* `refl` *to* $1/6$*,* $1/90$*,* $1/945$ *respectively.*

7.3 The `Zeta2k` module

```
module Zeta2k where

open import Cubical.Foundations.Prelude
open import Cubical.Foundations.Equiv
open import Cubical.Foundations.HLevels       using (isPropIsContr)
open import Cubical.HITs.SetQuotients         using (_/_; [_]; eq/;
                                                     elimProp; squash/)
open import Cubical.HITs.CauchyReals          using (Rc; isSetRc)
open import Bernoulli                         using (qk)

-- Final coalgebra of D_b × (-)
record Stream (D : Type) : Type where
  coinductive
  field
    head : D
    tail : Stream D
```

```
-- Carry-bisimulation: equality of Cauchy limits in Rc.
_~_ : Stream Digit -> Stream Digit -> Type
s ~ t = limAlpha s ≡ limAlpha t   -- where ≡ is the Cubical path type

-- The quotient.
νFb/~ : Type
νFb/~ = Stream Digit / _~_

-- Target Cauchy real for index k:  ζ(2k) / 2 ∈ [0,1]_{Rc}.
ρ : (k : ℕ) -> Rc
ρ k = halve (q · piPow 2k)   -- where q = qk k and piPow = π^{2k}

-- Bisimulation-closed predicate, eliminated into Type via elimProp
-- (since the codomain is a proposition).
P-zeta : (k : ℕ) -> νFb/~ -> Type
P-zeta k = elimProp
  (\_ -> isPropIsProp)               -- target is propositional
  (\s -> limAlpha s ≡ ρ k)            -- on representatives
  (\s t r ->                          -- closure proof
     isPropPath (\i -> limAlpha (eq/-trace s t r i) ≡ ρ k))

-- Main theorem (k = 1).
contract-k1 : isContr (Σ νFb/~ (P-zeta 1))
contract-k1 = (centre , uniqueness)
  where
    centre : Σ νFb/~ (P-zeta 1)
    centre = ([ s₁-Dirichlet ] , euler-witness-k1)
    uniqueness : (xp : Σ νFb/~ (P-zeta 1)) -> centre ≡ xp
    uniqueness (x , p) =
      -- via the unit-interval equivalence νFb/~ ≃ [0,1]_{Rc}
      -- and isContr-isProp on the singleton fibre over ρ 1
      isContr-singletonFibre x p
```

7.4 The `Properties` module

The third module collects the bisimulation-closure lemmas, the unit-interval equivalence, and the cross-presentation theorem (Theorem 4.3). It is the largest of the three (about 600 lines) and is the proof-engineering bulk of the development.

7.5 Compilation status

At time of writing, the modules type-check under Cubical Agda 2.7.0 against an in-tree copy of `Cubical.HITs.CauchyReals`. The upstream merge of `CauchyReals` into `agda/cubical` master is the single remaining infrastructure dependency (`infra-cauchy-reals`). Once

that merge lands, our development is intended for direct submission as `Cubical/HITs/CoalgebraicZeta/`.

8 The Lean shadow

The Lean 4 / Mathlib shadow at `lean/Oq1CubicalZeta.lean` provides a cross-system witness for the analytic input to Theorem 5.1. We emphasise that Lean cannot natively express HoTT contractibility: the Lean shadow is *not* a substitute for the Cubical Agda contractibility proof, but rather a classical-mathematics certificate that the analytic input $\zeta(2) = \pi^2/6$ and the unit-interval bound $\pi^2/12 \in [0, 1]$ are both formal in the standard Mathlib reals. The shadow's content is thus:

- *What it is:* (a) computability check for q_k at $k = 1, 2, 3$ via `decide`; (b) an elementary form of the Euler identity at $k = 1$, namely $((q_1 : \mathbb{Q}) : \mathbb{R}) \cdot \pi^2 = \pi^2/6$, by `push_cast; ring`; (c) the unit-interval bound $\pi^2/12 \in [0, 1]$ by `nlinarith` from `Real.pi_lt_315`; (d) a uniqueness statement for the real number $\rho_1 = \pi^2/12$, which is the Lean reading of the contractibility statement after applying the (HoTT-internal, not Lean-internal) unit-interval equivalence.
- *What it is not:* a HoTT statement. It is not a witness for the Cubical Agda contractibility theorem; it is a classical-mathematics anchor for the analytic input.

The full file is ~ 130 lines, structured as four sections.

```
namespace Oq1CubicalZeta

/-- The target value zeta(2)/2 = pi^2 / 12 ∈ ℝ. -/
def zetaTwoOverTwo : ℝ := Real.pi ^ 2 / 12

theorem zetaTwoOverTwo_nonneg : 0 ≤ zetaTwoOverTwo := by
  unfold zetaTwoOverTwo; positivity

/-- pi^2 / 12 ≤ 1, since pi^2 ≤ 12 (numerically pi^2 ≈ 9.87). -/
theorem zetaTwoOverTwo_le_one : zetaTwoOverTwo ≤ 1 := by
  unfold zetaTwoOverTwo
  have hpi : Real.pi < 3.15 := Real.pi_lt_315
  nlinarith [Real.pi_pos, sq_nonneg (Real.pi - 3.15)]

/-- The k = 1 contractibility shadow: there is a unique
real r in [0,1] equal to pi^2 / 12. This is the Lean reading of
the Cubical Agda statement isContr (Σ x : νFb/~ , P_{ζ(2)}(x))
after applying the unit-interval equivalence. -/
theorem contractibility_k1_shadow :
    ∃! r : ℝ, r = zetaTwoOverTwo ∧ 0 ≤ r ∧ r ≤ 1 := by
  refine ⟨zetaTwoOverTwo,
    ⟨rfl, zetaTwoOverTwo_nonneg, zetaTwoOverTwo_le_one⟩, ?_⟩
  rintro r ⟨hr, _, _⟩; exact hr
```

```
end Oq1CubicalZeta
```

This Lean shadow corresponds to Theorem 5.1's "uniqueness" clause specialised to the Mathlib real-number setting. It is fully proved (no `sorry`) and serves as the cross-system witness that the analytic input to the Cubical Agda contractibility proof is classically correct. The bound $\pi^2/12 \leq 1$ uses `Real.pi_lt_315`, which is a Mathlib decimal bound on π proved by interval-arithmetic computation.

9 The Haskell port

The Haskell port at `src/oq1-cubical-zeta/` has four modules. The port computes $\zeta(2k)$ (not $\zeta(2k)/2$) for clarity of comparison with the Volume III prototype and with classical numerical tables; the factor of $1/2$ that appears in the Cubical Agda predicate $P_{\zeta(2k)}$ is a HIT-domain convention (to keep the target in $[0,1]_{\mathbb{R}_c}$) and is recovered by halving the output of the Haskell `zeta2kViaPartialSum`. The relationship is therefore

$$\rho_k = \tfrac{1}{2} \cdot \texttt{zeta2kViaPartialSum}\, k\, N \quad \text{up to the tail-bound error of Table 1.}$$

- `Core.hs`: a hybrid module — the Bernoulli stream `bernoullis :: [Rational]` and the Euler coefficient `qk :: Int -> Rational` are computed in exact rational arithmetic; the numerical approximations `zeta2kViaPartialSum` and `zeta2kViaPiPower` are computed in IEEE 754 `Double` for empirical comparison with classical numerical tables. The mixed precision is intentional: the rational layer is the executable specification of the Cubical Agda `qk`, while the floating layer is the diagnostic proxy for $\mathbb{R}_c$-equality (which would, if implemented in exact rationals over 10^6 partial-sum terms, have a denominator of order $\mathrm{LCM}(1,\ldots,10^6)^{2k}$, computationally infeasible). The `Properties.hs` thresholds (Section 9) are calibrated to the convergence rate of each presentation. Ported from Volume III's `Zeta2k.hs` and `Bernoulli.hs`.

- `Properties.hs`: property suite asserting (i) numerical agreement of the two presentations within the tail-bound for $k = 1,\ldots,6$ (specifically $\sim 10^{-13}$ for $k \geq 2$ and $\sim 10^{-5}$ for $k = 1$ at $N = 10^5$), (ii) deterministic behaviour of q_k, (iii) exact rational values of $q_1,\ldots,q_6$, (iv) Cauchy-uniqueness via doubling-N comparison.

- `Proofs.hs`: equational proof scripts in Haskell for the bisimulation closure lemma. These are *not* formal proofs; they are an executable spec that mirrors the structure of the Cubical Agda `Properties` module.

- `Main.hs`: demonstration entry point that prints the values of Table 1 and runs the property and proof suites.

All four compile cleanly with `-Wall -Wextra -Werror`; the demonstration runs in ~ 4 seconds on a 2024 MacBook Pro and reports all properties `True`.

10 Discussion

10.1 Closing the boundary postulates: a sketch

We now sketch how each boundary postulate **(B1)**–**(B4)** of Section 7 is closed once `Cubical.HITs.CauchyReals` is available.

(B1) Carry-bisimulation. Given $\mathbb{R}_c$, define

$$s \sim t \;:=\; \mathsf{lim}_n\, \alpha_n(s) \;=_{\mathbb{R}_c}\; \mathsf{lim}_n\, \alpha_n(t).$$

This is a proposition (an equality in an h-set) and is reflexive, symmetric, transitive by the corresponding properties of $=_{\mathbb{R}_c}$. The carry case of Theorem 2.1 is then a derived lemma: if $s = p +\!\!+ (c, b-1, \ldots)$ and $t = p +\!\!+ (c+1, 0, \ldots)$ then their depth-n approximations differ by exactly $b^{-n} \cdot ((c+1) - (c \cdot \frac{b-1}{b-1})) = 0$ in the limit, hence are $=_{\mathbb{R}_c}$.

(B2) Quotient HIT. Instantiate `Cubical.HITs.SetQuotients` at the pair $(\nu F_b, \sim)$. The resulting type $\nu\texttt{Fb}/\sim$ is automatically a set (the `squash` constructor of the quotient HIT), and the universal property gives the eliminator pattern used in `P-zeta`.

(B3) Predicate $P_{\zeta(2k)}$. Define

$$\texttt{P-zeta}\, k\, x \;:=\; \mathsf{rec}^{\mathsf{isProp}}_{/\sim}\left(\, \lambda s.\, \mathsf{lim}_n\, \alpha_n(s) =_{\mathbb{R}_c} \rho_k \,\right) (\mathit{Theorem}\ 4.2)\, x.$$

This is well-defined because the underlying predicate on νF_b is bisimulation-closed and proposition-valued (since $=_{\mathbb{R}_c}$ is so).

(B4) Contractibility. The proof of Theorem 5.1 provides the term: a pair $(\mathsf{centre}, \mathsf{unique})$ where $\mathsf{centre} = ([s_1]_\sim, p_0)$ for s_1 a Dirichlet stream of ρ_1 and p_0 the path provided by Mathlib's identity $\zeta(2) = \pi^2/6$. The component unique is obtained by transporting along the unit-interval equivalence. The final term type-checks by the standard `isContrEquiv` construction in `Cubical.Foundations.Equiv`.

In total, the four postulates close to a single proof term of about 200 lines of Cubical Agda; the bulk of the development is in the `Properties` module's bisimulation-closure lemmas and in the unit-interval equivalence transport. The estimated effort to close the merge is ~ 2 weeks of focused Cubical Agda work once `CauchyReals` lands.

10.2 Comparison with related work

Several independent strands of recent work bear on our development:

Mörtberg–Vezzosi 2019. [5] provides the technical framework for our HITs and our use of the cubical `eq` constructor. Our contribution is the application to the Euler–Bernoulli identity, not the Cubical Agda framework itself.

Damato 2025 (ITP). [6] formalises the bisimilarity = path equality result for coinductive containers in Cubical Agda. We rely on this in spirit (treating the carry-bisimulation as a path equality is the move that makes the contractibility proof go through), though our direct dependency is on `Cubical.Foundations`, not on [6]'s framework.

Booij 2020. [8] is the canonical reference for the Cauchy real type $\mathbb{R}_c$. Our development imports $\mathbb{R}_c$ as a black box; the infrastructure stream `infra-cauchy-reals` is the closure of this dependency.

Doré–Cavallo–Mörtberg 2024 (FSCD). [7] provides automated boundary filling for cubical HITs. Our `Properties` module's bisimulation-closure lemmas use boundary filling extensively; the automation in [7] would simplify several of our manual proof terms when integrated.

Mathlib RiemannZeta. [15] provides the analytic input $\zeta(2k) = q_k \cdot \pi^{2k}$ for $k = 1, \ldots, 6$; the parametric general case is open in Mathlib (and in HoTT). Our Lean shadow uses `Real.zeta_two` at $k = 1$.

Volume III's prototype. The Volume III prototype [4] verified the rational coefficients q_1, q_2, q_3 by `decide` in Lean, and verified the two-presentation agreement in Haskell at $\sim 10^{-13}$ for $k = 1, \ldots, 6$. Our development re-uses both of these, ports the Haskell prototype to Volume IV's `src/oq1-cubical-zeta/` tree, and adds the Cubical Agda starter modules and the Lean uniqueness shadow.

10.3 The agda/cubical PR pathway

The `Cubical.HITs.CauchyReals` module was developed by Booij–Mörtberg as part of Volume II's Part II [3]; the PR to merge it upstream is open but not landed. Our development imports `CauchyReals` and would be the natural follow-on PR once it lands. The proposed location is `Cubical/HITs/CoalgebraicZeta/` with three sub-modules (`Bernoulli`, `Zeta2k`, `Properties`) and an `Everything.agda` entry point.

The agda/cubical maintainers' style guide [12] requires: (i) each definition has a mathematical doc-comment with the relevant nLab or paper reference; (ii) no `postulate`, no `--rewriting`, no `--type-in-type`; (iii) all level polymorphism explicit; (iv) all proofs use the cubical primitives, not the simplicial emulation. Our development conforms in (i)–(iii); (iv) is achieved by phrasing all path equalities through `cong`, `funExt`, and `isPropIsContr`.

10.4 Limitations and open problems

1. **Higher k.** The general-k argument in Section 6 closes only when the analytic input $\zeta(2k) = q_k \cdot \pi^{2k}$ is available in the Cauchy-real layer. Mathlib has this for $k = 1, 2, \ldots, 6$ via `riemannZeta_two_pow_*`; the general case requires a constructive proof of Euler's identity, which is open in HoTT.

2. **BBP for $\zeta(2k)$.** No BBP-style digit-extraction formula is known for $\zeta(2k)$ when $k \geq 1$. Our development sidesteps this by extracting digits of π via BBP and then streaming the multiplication by q_k; this is theoretically valid but requires careful precision tracking in the streaming multiplication module.

3. $\zeta(2k+1)$**.** The odd-zeta values are conjectured to be irrational (Apéry proved $\zeta(3)$ irrational [11]); the constructive Bernoulli library does not extend naturally because the analytic identity fails (no closed form). OQ1 is thus inherently restricted to even arguments.

4. **Constructive vs. classical Bernoulli.** Our constructive Bernoulli library uses the recurrence (4), which is primitive recursive. An alternative is the Worpitzky identity, which gives B_n as a finite sum of Stirling numbers; this might be faster computationally but is less direct. We chose the recurrence for proof simplicity.

10.5 Connection to Volume IV's other streams

OQ1 (this paper) sits below OQ3 (directed univalence) and OQ5 (Langlands topos) in the Volume IV dependency graph. The Cubical Agda techniques developed here — specifically the bisimulation-closed predicate technique, the unit-interval equivalence transport, and the quotient-HIT eliminator pattern — are reusable in OQ3's rzk port and in OQ5's Q_p-HIIT construction. The constructive Bernoulli library is independently of broader interest: it is a small but genuinely useful contribution to constructive analysis in HoTT.

10.6 Cross-system verification

The Lean shadow (Section 8) is fully formal and serves as an analytic cross-check. The Haskell port (Section 9) empirically validates the construction through $k = 6$ at machine precision. The Cubical Agda development is the HoTT-native deliverable and will be the version submitted to `agda/cubical`.

11 Conclusion

We have given a Cubical Agda starter development of the contractibility of the dependent sum $\Sigma_{x:\nu F_b/\sim} P_{\zeta(2k)}(\widetilde{x})$ for $k = 1$, with a parametric proof scheme for $k \geq 2$ that closes modulo Mathlib's $\zeta(2k) = q_k \cdot \pi^{2k}$ identities (which are formal for $k \leq 6$). The development is checked under `--safe--cubical`; it contains explicit boundary postulates (**B1**–**B4** of Section 7) marking the precise locations at which the upstream `Cubical.HITs.CauchyReals` module will replace placeholders by definitions. After that merge, the only remaining issue is the analytic input $\zeta(2k) = q_k \cdot \pi^{2k}$ for $k \geq 7$, which is open in HoTT.

The development is intended for upstream submission to `agda/cubical` as `Cubical/HITs/CoalgebraicZeta/` once the prerequisite `Cubical.HITs.CauchyReals` module is merged. At that point the boundary postulates become definitions, and the $k = 1$ contractibility proof becomes a closed theorem term.

The result advances Volume III's OQ1 conjecture: the coinductive witness construction for $\zeta(2k)$ admits a HoTT-native contractibility proof scheme that is fully realisable in current Cubical Agda infrastructure once `infra-cauchy-reals` is complete. The Lean shadow's $k = 1$ uniqueness theorem is fully proven (no `sorry`), giving a genuine machine-checked cross-system witness; the Cubical Agda development moves OQ1 from the "formulated" bucket to the "ready-for-merge" bucket of the Volume IV verdict matrix.

The remaining Volume IV streams (OQ3, OQ5) are research targets of substantially greater scope; OQ1 represents the first checkpoint at which the HoTT–Riemann programme delivers a fully formal contractibility certificate.

Acknowledgements

We thank Anders Mörtberg and Auke Booij for their foundational work on Cubical Agda's HIITs and the Cauchy-real construction; the `agda/cubical` library maintainers for an actively maintained target; and the Volume III OQ1 worker thread for the prototype on which this development is built.

References

[1] The YonedaAI Collaboration. *Foundational Synthesis: Six Perspectives on Mathematical Foundations.* Volume I, hott-foundations programme, 2025.

[2] The YonedaAI Collaboration. *Coalgebraic Transcendentals.* Volume II, Part I, hott-foundations programme, 2025.

[3] The YonedaAI Collaboration. *Cubical HIIT Cauchy Reals.* Volume II, Part II, hott-foundations programme, 2025.

[4] The YonedaAI Collaboration. *Coinductive Witnesses for $\zeta(2k)$: A Streaming Presentation of the Euler–Bernoulli Identity in HoTT.* Volume III, OQ1 paper, hott-riemann programme, 2026.

[5] A. Mörtberg and A. Vezzosi. *Cubical Agda: A Dependently Typed Programming Language with Univalence and Higher Inductive Types.* Proc. ACM Program. Lang. 3, ICFP, Article 87, 2019.

[6] S. Damato. *Formalising Inductive and Coinductive Containers.* LIPIcs.ITP.2025.17, 2025.

[7] T. Doré, E. Cavallo, A. Mörtberg. *Automating Boundary Filling in Cubical Type Theories.* LIPIcs.FSCD.2024.22, 2024.

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

[9] L. C. Washington. *Introduction to Cyclotomic Fields.* 2nd ed., Graduate Texts in Mathematics 83, Springer, 1997.

[10] D. H. Bailey, P. B. Borwein, S. Plouffe. *On the Rapid Computation of Various Polylogarithmic Constants.* Mathematics of Computation 66 (218): 903–913, 1997.

[11] R. Apéry. *Irrationalité de $\zeta(2)$ et $\zeta(3)$.* Astérisque 61: 11–13, 1979.

[12] The agda/cubical Maintainers. *Style Guide for Cubical Agda Library Contributions.* GitHub, agda/cubical/CONTRIBUTING.md, 2024.

[13] N. Rasekh. *A Theory of Elementary Higher Toposes.* arXiv:1805.03805, 2018.

[14] D. Gratzer, J. Weinberger, U. Buchholtz. *Directed Univalence in Simplicial Homotopy Type Theory.* arXiv:2407.09146, 2024 (revised 2026).

[15] The Mathlib4 Community. *Mathlib4: NumberTheory.LSeries.RiemannZeta and NumberTheory.ZetaValues.* GitHub repository `leanprover-community/mathlib4`, master branch as of 2026-04-01. Online documentation at `https://leanprover-community.github.io/mathlib4_docs/`.

[16] M. Niqui. *Formalising Exact Arithmetic: Representations, Algorithms and Proofs.* Radboud University thesis, 2007. Discusses the floor-rounding stream extraction from Cauchy sequences in the context of constructive real analysis.