Provenance-Preserving Arithmetic

Toni Milovan

Independent Researcher, Pula, Croatia

tmilovan@fwd.hr

Abstract: Provenance-Preserving Arithmetic

Standard arithmetic is already reversible — except for zero operations. Given `a × b = c`, we can recover `a = c / b`. Given `a + b = c`, we can recover `a = c - b`. **The only gap:** `a × 0 = 0` loses `a` forever.

This system fills exactly that gap. We present a formal system where **zero is an infinitesimal unit (** $|1|_{-1}$ **)**, making zero operations reversible while maintaining distributivity and associativity.

Algebraic foundation: The composite structure is modeled on Laurent polynomial arithmetic (multiplication and same-dimension addition) over : $\mathbb{C}[z, z^{-1}]$, but with the additive identity excluded. [Lang 2002, Jacobson 1985]. The algebra itself is not novel.

Two contributions:

1. **Simple infinitesimal algebra** — arithmetic on infinitesimals/infinities with bidirectional translation to limits and hyperreals
2. **One-step provenance for ×0 and ÷0** — the only operations standard math loses information on

Key tradeoff: We sacrifice the universal additive identity ($0 + 0 = |2|_{-1}$, not `0`) to gain reversibility.

Primary application: Universal calculus machine — dimensional structure naturally encodes Taylor series coefficients for exact symbolic differentiation, integration, and multivariate calculus.

Key insight: `0/0` is not indeterminate here. Division of zeroes yields provenance-dependent results: $|2|_{-1} / |1|_{-1} = 2$. This is a feature, not a bug.

Applications: Automatic differentiation, reversible computing, silent ×0 corruption detection (potential), and educational tool for limits/hyperreals.

Algebraic Foundation

> 🏗️ **The algebra is standard ring theory. This section establishes notation.**

Composites share the **multiplication and addition rules of the Laurent polynomial ring $\mathbb{C}[z, z^{-1}]$** [Lang 2002, Jacobson 1985]. The departure: we reinterpret which element acts as semantic zero, deliberately sacrificing the additive identity to gain provenance.

> 💡 **Note:** This system **borrows multiplication and same-dimension addition from** $\mathbb{C}[z, z^{-1}]$, but **deliberately excludes the zero polynomial** (the additive identity of the Laurent ring).

> Instead, z^{-1} is interpreted as "zero with provenance" — a structural zero that accumulates rather than annihilates. This makes the system a **commutative monoid under addition** (associative, commutative, with no additive identity) paired with standard ring multiplication. Note: a *rng* ("ring without i") refers to rings lacking *multiplicative* identity; our system has multiplicative identity ($|1|$) but lacks an *additive* identity by design

Note on Naming: While formally related to **Graded Algebra** and **Laurent polynomial rings**, we refer to this system as an **Arithmetic** because our primary focus is on the operational semantics of calculation rather than the study of algebraic structures.

Our Notation	Laurent Polynomial	Meaning
$\|a\|_n$	az^n	Coefficient a at dimension n
$\|a\|_m \times \|b\|_n$	$az^m \cdot bz^n = abz^{m+n}$	Multiplication (dims add)
$\|a\|_m / \|b\|_n$	$az^m / bz^n = (a/b)z^{m-n}$	Division (dims subtract)
$0 = \|1\|_{-1}$	z^{-1}	Structural zero
$\infty = \|1\|_1$	z	Structural infinity
$\|a\|_n + \|b\|_n$	$az^n + bz^n = (a+b)z^n$	Addition (same dim only)

Notation convention: $|a|$ and $|a|_0$ are equivalent — both denote coefficient a at dimension 0 (the rational dimension). The subscript-free form $|a|$ is preferred for simplicity when the dimension is zero.

Cross-dimension addition: When adding terms at *different* dimensions, they remain as separate terms (like polynomial addition). For example: $|3|_0 + |2|_{-1}$ does not simplify further — it represents a composite with coefficient 3 at dimension 0 and coefficient 2 at dimension -1. Only terms at the *same* dimension combine: $|3|_{-1} + |2|_{-1} = |5|_{-1}$.

Dimension Scale

$$\begin{array}{ccccccccc} \cdots & |\_|_3 & |\_|_2 & |\_|_1 & |r| & |\_|_{-1} & |\_|_{-2} & |\_|_{-3} & \cdots \\ & \infty^3 & \infty^2 & \infty & R & 0 & 0^2 & 0^3 & \end{array}$$

Inherited Properties (Standard Ring Theory)

The following hold because multiplication and same-dimension addition follow the Laurent polynomial rules:

- **Associativity**: $(a \times b) \times c = a \times (b \times c)$ — convolution is associative
- **Commutativity**: $a \times b = b \times a$ — $\mathbb{C}$ is commutative, $\mathbb{Z}$ is abelian
- **Distributivity**: $a \times (b + c) = (a \times b) + (a \times c)$ — convolution distributes
- **Multiplicative identity**: $|1|$ (coefficient 1 at dimension 0)

Proofs omitted — see Lang (2002) or Jacobson (1985) for standard ring theory.

Matrix Representation (Linear Algebra View)

Since the system forms a vector space, operations can be represented as linear transformations. If we truncate the system to a finite window of dimensions (e.g., infinity, rational, zero), the state vector is $\vec{v} = [c_1, c_0, c_{-1}]^T$.

Multiplication by zero acts as a **Shift Matrix**:

$$M_{\times 0} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Applying this matrix shifts components "down": the infinity component becomes rational, and the rational component becomes a zero.

Multiplication by infinity acts as the inverse shift:

$$M_{\times \infty} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

This demonstrates that the "dimensional shift" is a well-defined linear operator, equivalent to the shift operator in polynomial spaces. This allows implementation using standard linear algebra libraries.

Computational complexity: Each arithmetic operation is **O(n)** where n is the number of active dimensions (non-zero coefficients). For truncated systems (fixed dimension window), operations are O(1). See demos for working implementations.

What's Novel: The Provenance Interpretation

> 🎯 **This is our actual contribution.**

> 📖 *For the Reduction Theorem showing how calculus reduces to algebra, see the companion note:* https://github.com/tmilovan/composite-machine/blob/main/papers/Composite_Machine__A_Unified_Framework-companion_note.pdf

Scope of this paper: This paper defines the arithmetic and its properties; the companion note focuses on the computational "calculus machine" viewpoint and executable validation.

Clarification: What Standard Math Already Provides

Standard arithmetic already offers **one-step provenance for all operations except zero**:

- `a × b = c` → recoverable: `a = c / b` ✅
- `a + b = c` → recoverable: `a = c - b` ✅
- `a - b = c` → recoverable: `a = c + b` ✅
- `a / b = c` → recoverable: `a = c × b` ✅
- `a × 0 = 0` → **not recoverable** ❌ ← *This is the only gap we fill*

Our contribution is narrow but complete: We make zero operations reversible. Nothing more, nothing less.

---

1. Reframing Zero as Information-Preserving

Standard arithmetic: $5 \times 0 = 0$ — the 5 is gone forever.

Our interpretation: $5 \times 0 = |5|_{-1}$ — the coefficient 5 is preserved as provenance.

The algebraic operation is just $5 \cdot z^{-1} = 5z^{-1}$ in Laurent polynomials. The novelty is *interpreting* z^{-1} as "a zero that remembers what created it."

2. Modified Additive Semantics

Standard ring theory: The Laurent polynomial ring has an additive identity (the zero polynomial).

Our deliberate modification: We interpret $0 = |1|_{-1}$ as a *structural zero* that accumulates:

- $|1|_{-1} + |1|_{-1} = |2|_{-1}$ (one zero + one zero = two zeroes)

This is **intentional**. We sacrifice the additive identity to enable provenance tracking. If $0 + 0 = 0$, we'd lose count of how many zero-operations occurred.

3. Zero-Infinity Duality

Interpretation: Zero and infinity are dimensional opposites that perfectly cancel:

- $0 \times \infty = |1|_{-1} \times |1|_{1} = |1|$ (dimensions: -1 + 1 = 0)

This follows from Laurent polynomial multiplication, but the *interpretation* as a duality principle is novel.

4. Self-Division Identity (No Exceptions)

Standard math: $a/a = 1$ for all $a \neq 0$. Exception required for zero.

Our system: $a/a = 1$ for ALL a, including zero:

- $|1|_{-1} / |1|_{-1} = |1|$ (dimensions: -1 - (-1) = 0)

$0/0 = 1$ with no exception needed.

⚠️ 0/0 Is Not Indeterminate Here

In standard math: $0/0$ is indeterminate because infinitely many values satisfy $0 \times x = 0$.
In this system: Division of zeroes yields **provenance-dependent** results:

- $|1|_{-1} / |1|_{-1} = |1|_0 = 1$ (one zero divided by one zero)
- $|2|_{-1} / |1|_{-1} = |2|_0 = 2$ (two zeroes divided by one zero)
- $|5|_{-1} / |1|_{-1} = |5|_0 = 5$ (five zeroes divided by one zero)

This is the crux of the system. Zeroes carry provenance (their coefficient), and that provenance determines the result of division. Without this property, reversibility breaks.

System Tradeoffs

Gains	Sacrifices
✅ Reversible multiplication by zero	❌ No universal additive identity
✅ Division by zero is defined	❌ Zeroes accumulate ($0+0 \neq 0$)

Gains	Sacrifices
✅ Perfect information preservation	❌ More complex equality semantics
✅ $0/0 = 1$ (no exception needed)	❌ Not a drop-in replacement
✅ Provenance tracking through all operations	

Novel Theorems

The following theorems arise from the provenance interpretation and are not direct consequences of standard Laurent polynomial arithmetic:

Theorem 1: Information Preservation (Non-Collapse)

Operations with zero preserve the operand's identity.

- $\forall a: a \times 0 = |a|_{-1}$ where the coefficient a is preserved
- $\forall a: a / 0 = |a|_1$ where the coefficient a is preserved

Proof:

By convolution: $|a|_0 \times |1|_{-1} = |a \cdot 1|_{0+(-1)} = |a|_{-1}$

The coefficient a appears unchanged. Multiplication by zero is a **bijection** (one-to-one mapping), not a collapse. ∎

Theorem 2: Zero-Infinity Duality

Zero and infinity are multiplicative inverses.

- $0 \times \infty = 1$
- $1/0 = \infty$
- $1/\infty = 0$

Proof:

$|1|_{-1} \times |1|_1 = |1 \cdot 1|_{(-1)+1} = |1|_0 = 1$ ∎

Theorem 3: Provenance (Non-Uniqueness of Zero)

Zero is a family of values, not a singular point.

- $|a|_{-1} \neq |b|_{-1}$ for $a \neq b$

Proof:

Composed numbers are functions $f: \mathbb{Z} \to \mathbb{C}$. Two functions are equal iff they agree at all points. Since $|a|_{-1}$ has $f(-1) = a$ and $|b|_{-1}$ has $g(-1) = b$, if $a \neq b$ then $f \neq g$. ∎

Consequence: $5(0)$ and $2(0)$ are qualitatively different zeroes.

Theorem 4: Reversibility

Operations with zero and infinity are reversible.

- $\forall a: (a \times 0) / 0 = a$
- $\forall a: (a / 0) \times 0 = a$

Proof (both directions):

Direction 1: $(|a|_0 \times |1|_{-1}) / |1|_{-1} = |a|_{-1} / |1|_{-1} = |a|_{(-1)-(-1)} = |a|_0 = a$ ✓

Direction 2: $(|a|_0 / |1|_{-1}) \times |1|_{-1} = |a|_1 \times |1|_{-1} = |a \cdot 1|_{1+(-1)} = |a|_0 = a$ ✓ ∎

Theorem 5: Coefficient Cancellation

When symbolic zeroes and infinities multiply, orders cancel and coefficients multiply.

- $|a|_{-1} \times |b|_1 = |ab|$
- $|a|_{-n} \times |b|_n = |ab|$ (for all n)

Proof:

Dimensions add: $-n + n = 0$. Coefficients multiply: $a \times b = ab$. ∎

Theorem 6: Identity Elements

Multiplicative identity exists. No universal additive identity.

- $|1|$ is the multiplicative identity
- Structural zero $|1|_{-1}$ does NOT act as additive identity: $|1|_{-1} + |1|_{-1} = |2|_{-1} \neq |1|_{-1}$

Proof:

For any $|a|_n$: $|a|_n \times |1|_0 = |a \cdot 1|_{n+0} = |a|_n$ ✓

For structural zeroes: $|1|_{-1} + |1|_{-1} = |1+1|_{-1} = |2|_{-1}$ (coefficients add at same dimension).

This is **intentional** — zeroes accumulate to preserve provenance. ∎

Theorem 7: Fractional Orders (Roots)

Roots create fractional-order values.

- $\sqrt{|a|_{-2}} = |\sqrt{a}|_{-1}$
- $(|a|_n)\hat{}k = |a\hat{}k|_{nk}$

Domain restriction: For real coefficients, $a \geq 0$ is required for even roots. For complex coefficients, branch cuts apply — the principal root is used by convention.

Proof:

Define $(|a|_n)\hat{}(1/k) := |a\hat{}(1/k)|_{n/k}$

Verification: $(|a\hat{}(1/k)|_{n/k})\hat{}k = |(a\hat{}(1/k))\hat{}k|_{(n/k) \cdot k} = |a|_n$ ✓ ∎

Theorem 8: Total Ordering (Real Coefficients Only)

The system admits a total ordering on two orthogonal axes:

1. **Dimension axis (primary):** Higher dimension > Lower dimension
2. **Coefficient axis (secondary):** Standard ℝ ordering within each dimension

Ordering algorithm for composites:

1. Find the highest dimension with a non-zero coefficient in either composite
2. Compare coefficients at that dimension using standard ℝ ordering (treat missing coefficients as 0)

3. If equal, proceed to the next lower dimension that has a non-zero coefficient in either composite
4. Repeat until a difference is found (or composites are equal)

Edge case handling:

- Sparse composites: Compare only at dimensions where at least one composite has a non-zero coefficient
- Example: $|3|_0 + |2|_{-1}$ vs $|3|_0 + |5|_{-2}$ → At dim 0: equal. At dim -1: 2 > 0 (implicit). **First wins.**
- Infinite-dimensional composites: This ordering extends to any finite comparison, but infinite composites require truncation for practical computation

Examples:

- $|5|_1$ vs $|1000000|_0$ → $|5|_1$ **wins** — Dimension 1 > Dimension 0
- $|-1|_1$ vs $|1000000|_0$ → $|1000000|_0$ **wins** — At dim 1: −1 < 0 (implicit)
- $|2|_1 + |-10|_0$ vs $|-5|_1 + |3|_0$ → **First wins** — At dim 1: 2 > −5
- $|3|_1 + |5|_0$ vs $|3|_1 + |2|_0$ → **First wins** — Dim 1 equal; at dim 0: 5 > 2
- $|1|_{-1}$ vs $|1|_{-2}$ → $|1|_{-1}$ **wins** — Dimension −1 > Dimension −2

Intuition: Think of dimension and coefficient as orthogonal axes. Infinity ($\backslash|1\backslash|_1$) dominates all rationals. Negative infinity ($\backslash|-1\backslash|_1$) is less than all positive rationals. Infinitesimals ($\backslash|a\backslash|_{-1}$) are smaller than any positive rational but ordered among themselves by coefficient.

Note: This ordering is only defined for real coefficients. Complex coefficients do not admit a total ordering.

Proof:

Lexicographic ordering on (dimension, coefficient) pairs. Since $\mathbb{Z}$ is totally ordered and $\mathbb{R}$ is totally ordered at each dimension, the product order is total. ∎

Applications

1. Composed Calculus Machine (Primary)

The dimensional structure naturally encodes **Taylor series coefficients**, making this system a **universal calculus machine** for exact symbolic differentiation, integration, and multivariate calculus — equivalent to Taylor-mode automatic differentiation [Griewank & Walther 2008].

Works universally on all operations.

How it works:

- Represent a value at point x as a comp: $|f(x)|_0 + |f'(x)|_{-1} + |f''(x)/2!|_{-2} + \ldots$
- Dimensions encode derivative orders (dimension -n = nth derivative coefficient)
- Arithmetic operations automatically propagate derivatives

Example: f(x) = x^4 at x = 2

```
Input:  |2|₀ + |1|₋₁  (value 2 with derivative seed 1)
Result: |16|₀ + |32|₋₁ + |24|₋₂ + |8|₋₃ + |1|₋₄

Reading off: f(2)=16, f'(2)=32, f''(2)/2!=24, f'''(2)/3!=8, f''''(2)/4!=1
Verification: d/dx(x⁴) = 4x³ → 4(8) = 32 ✓
```

Definition: Standard Part

The **standard part** operation `st(x)` extracts the coefficient at dimension 0:

- `st(|a|₀ + |b|₋₁ + |c|₋₂ + ...) = a`

This is a **ring homomorphism** from composites to ℝ: `st(x + y) = st(x) + st(y)` and `st(x × y) = st(x) × st(y)`. It maps infinitesimals to 0 and infinities to ±∞ (or undefined for finite truncation).

Working demo: See: https://github.com/tmilovan/composite-machine/tree/main/tests for runnable Python code verifying these derivatives.

Connection to existing work:

- Equivalent to **Taylor-mode automatic differentiation** (forward AD)
- Related to TPSA (Truncated Power Series Algebra) used in particle physics
- JAX's `jet` module uses similar principles

What's novel: The interpretation of z^{-1} as "zero with provenance" — existing systems don't frame the infinitesimal dimension this way.

Applications:

- Automatic differentiation in machine learning
- Sensitivity analysis (financial Greeks: delta, gamma, vega)
- Scientific computing with exact error propagation

Secondary Applications

The following applications demonstrate provenance preservation in specific domains:

- **Silent ×0 Corruption Detection** *(exploratory)* — Potential application for detecting when multiplication by zero silently destroys data in numerical pipelines.
- **Database-Level Provenance** — Track zero-operations at the database level.
- **Reversible Computing** — The Multiplication Chain Protocol enables reversible computation via dimensional shifts.
- **Educational Value** — Simpler path to limits and hyperreals. See dedicated section below.

Educational Value: Simpler Path to Limits and Hyperreals

> 🎓 **This system may serve as a pedagogical bridge to advanced calculus concepts.**

The Problem with Teaching Limits

Students struggle with limits because:

- ε-δ proofs require nested quantifiers and indirect reasoning
- "Approaching" is philosophically confusing
- Hyperreals require ultrafilters and transfer principles

This System as a Teaching Tool

Our arithmetic provides a **concrete, algebraic model** where:

1. **Infinitesimals are first-class objects** — $|1|_{-1}$ is "zero" you can compute with
2. **No limits needed** — just algebraic manipulation
3. **Standard part is simple** — project to dimension 0
4. **Bidirectional translation** — results match classical limits exactly

Example: Derivative of x^2

Classical: $\lim(h \to 0)\ [(x+h)^2 - x^2]/h = \lim(h \to 0)\ [2x + h] = 2x$

This system: Let $h = |1|_{-1}$

$$[(|x|_0 + |1|_{-1})^2 - |x^2|_0] / |1|_{-1}$$
$$= [|x^2|_0 + |2x|_{-1} + |1|_{-2} - |x^2|_0] / |1|_{-1}$$
$$= [|2x|_{-1} + |1|_{-2}] / |1|_{-1}$$
$$= |2x|_0 + |1|_{-1}$$

Standard part: 2x ✓

The same result, achieved algebraically. Students can verify limits by computation before learning the formal ε-δ machinery.

Comparison with Other Infinitesimal Systems

System	Higher Orders	Complexity	Educational Use
Dual Numbers ($\varepsilon^2=0$)	❌ Truncated	Simple	Good for derivatives only
Hyperreals	✅ Preserved	High (ultrafilters)	Difficult to teach
Sergeyev's Grossone	✅ Preserved	Medium	Novel notation
This System	✅ **Preserved**	**Low (ring theory)**	**Explicit dimensions**

Comparison with Existing Approaches

Approach	Handles ÷0?	Preserves Distributivity?	Preserves Information?	Reversible?
Standard Arithmetic	❌	✅	❌	❌
Wheel Theory	✅	❌	❌	❌
IEEE 754	✅ (→NaN/∞)	✅	❌	❌
This System	✅	✅	✅	✅

The tradeoff we accept: No universal additive identity.

Limitations

This system has inherent boundaries that users should understand:

Practical Limitations

1. **Drop-in replacement** — Code expecting `0 + 0 = 0` will break. Migration requires explicit handling of accumulating zeroes.

Numerical Considerations

1. **Coefficient explosion** — High-order derivatives can produce very large or very small coefficients, causing floating-point precision issues in implementations.
2. **Truncation decisions** — Practical implementations must choose a dimension window; this introduces approximation.
3. **Multi-term division** — Division by composites with multiple non-zero dimensions is not uniquely defined (analogous to polynomial long division with remainder).

Theoretical Boundaries

1. **Complex ordering** — Theorem 8 (Total Ordering) only applies to real coefficients. Complex coefficients require a different approach.
2. **Infinite composites** — The ordering and equality semantics for composites with infinitely many non-zero dimensions require careful treatment.
3. **Not a field** — The system lacks multiplicative inverses for multi-term composites, so it's a ring, not a field.

Extensions: Beyond Basic Differentiation

The following capabilities extend the core system to handle cases initially thought to be limitations:

Extension 1: Symbolic Integration via Positive Dimensions

Key insight: If negative dimensions encode derivatives, **positive dimensions can encode antiderivatives**.

How it works:

- Dimension -n encodes $f^{(n)}(x)/n!$ (nth derivative coefficient)
- Dimension +n encodes $\int^n f(x)$ (nth antiderivative coefficient)
- Integration shifts dimensions *up* (opposite of differentiation)

Example: $\int x^2\,dx$

```
Input:  |x²|₀ (function x² at rational dimension)
Integrate: shift up one dimension, divide by new power
Result: |x³/3|₁ + |C|₀
```

```
The |C|₀ term is the constant of integration —
provenance-preserving arithmetic naturally leaves a "slot" for it.
```

The integration operator:

$$\int : |a|_n \mapsto |a/(n+1)|_{n+1}$$

Note: This handles *polynomial* integration directly. Transcendental functions require their Taylor series representation, then term-by-term integration (which this system supports).

Constant of integration: The system naturally accommodates C as a free coefficient at dimension 0. Definite integrals eliminate C by boundary conditions.

Extension 2: PDEs via Multi-Index Dimensions

Key insight: Replace scalar dimensions with **multi-index dimensions** for multivariate calculus.

Structure:

- Single variable: dimension $n \in \mathbb{Z}$
- Two variables: dimension $(n, m) \in \mathbb{Z}^2$ for $\partial^{n+m}f/\partial x^n \partial y^m$
- General: dimension $\mathbf{n} \in \mathbb{Z}^d$ for d variables

Example: $f(x,y) = x^2y$

```
Input: |x²y|₍₀,₀₎ + |2xy|₍₋₁,₀₎ + |x²|₍₀,₋₁₎ + |2y|₍₋₂,₀₎ + |2x|₍₋₁,₋₁₎ + ...

Reading off:
  f(x,y)    = x²y    (dim 0,0)
  ∂f/∂x     = 2xy    (dim -1,0)
  ∂f/∂y     = x²     (dim 0,-1)
  ∂²f/∂x²   = 2y     (dim -2,0)
  ∂²f/∂x∂y  = 2x     (dim -1,-1)
```

Algebraic structure: This is the **multivariate Laurent polynomial ring** $\mathbb{C}[z_1, z_1^{-1}, z_2, z_2^{-1}, \ldots]$, which inherits all the same properties.

Connection to existing work: This is equivalent to **jet bundles** in differential geometry and **multivariate Taylor-mode AD**. The provenance interpretation extends naturally — each partial derivative direction has its own "zero with provenance."

Laplacian, gradient, etc.: Standard differential operators become linear combinations of dimension-shift operations.

Extension 3: Non-Analytic Functions via Signed Infinitesimals

Key insight: Introduce **signed zeroes** to capture directional limits.

The problem: Functions like $|x|$ have different left and right derivatives at $x=0$:

- $\lim(h \to 0^+)\ (|h| - 0)/h = 1$
- $\lim(h \to 0^-)\ (|h| - 0)/h = -1$

Solution: Signed infinitesimals

- $|1|_{-1}^{+}$ = positive infinitesimal (approaching from right)
- $|1|_{-1}^{-}$ = negative infinitesimal (approaching from left)

Example: Derivative of |x| at x=0

```
Using h = |1|₋₁⁺ (right approach):
  (|h| - 0) / h = h / h = |1|₀ = 1 ✓

Using h = |1|₋₁⁻ (left approach):
  (|-h| - 0) / h = h / (-h) = |-1|₀ = -1 ✓

The system correctly produces different results based on
approach direction — this IS the derivative information for |x|.
```

Heaviside step function: H(x) = 0 for x<0, 1 for x≥0

```
H(0⁺) = |1|₀
H(0⁻) = |0|₀
H'(x) = δ(x) encoded as |1|₋₁⁺ - |1|₋₁⁻ (a "jump" at zero)
```

Distributional derivatives: The Dirac delta δ(x) emerges naturally as the derivative of the Heaviside function. In this system, it's represented as a signed infinitesimal difference — provenance that remembers "there was a jump here."

Algebraic structure: Signed infinitesimals extend the dimension axis from $\mathbb{Z}$ to $\mathbb{Z} \times \{^-, ^o, ^+\}$, where the second component tracks approach direction. This remains a well-defined ring.

💡 **Summary: The System Extends Naturally**

Extension	Mechanism	Algebraic Basis
Integration	Positive dimensions	Same Laurent ring, opposite shift
PDEs	Multi-index dimensions	Multivariate Laurent ring
Non-analytic	Signed infinitesimals	Extended dimension structure

The core provenance principle holds in all cases: operations preserve information through dimensional encoding.

Related Work

This section situates our contribution within the landscape of existing approaches to infinitesimals, automatic differentiation, and extended arithmetic systems.

Taylor-Mode Automatic Differentiation

What it is: Taylor-mode AD (also called "jet transport" or "high-order forward-mode AD") propagates truncated Taylor series through computation. Given a function f and a point a, it computes f(a), f'(a), f''(a)/2!, ... up to some fixed order k.

Key implementations:

- **JAX** `jet` — Google's implementation propagates "jets" (truncated Taylor polynomials) through XLA-compiled functions [Bradbury et al. 2018]
- **TaylorDiff.jl** — Julia package for high-order derivatives via Taylor propagation [Tan 2022]
- **FADBAD++** — C++ library using templated Taylor arithmetic [Bendtsen & Stauning 1996]

Relationship to this work:

Taylor-mode AD uses the *same underlying mathematics* — polynomial coefficient propagation. The key differences:

Aspect	Taylor-Mode AD	This System
Truncation	Fixed order k	Unbounded (Laurent series)
Zero handling	Standard (0 annihilates)	Provenance-preserving
Division by zero	Undefined	Defined via dimensional shift
Negative dimensions	Not used	Encode infinitesimal orders
Graph construction	Typically required	Direct evaluation f(a + h)

What we add: The provenance interpretation and bidirectional Laurent structure. Existing Taylor-mode AD cannot reverse zero operations or define 0/0. Our framing also eliminates the need for explicit graph construction — you simply evaluate f(a + h) where h is the structural zero.

Dual Numbers and Jet Spaces

What they are: Dual numbers extend ℝ with an element ε satisfying $\varepsilon^2 = 0$. The algebra $\mathbb{R}[\varepsilon]/(\varepsilon^2)$ captures first derivatives: f(a + ε) = f(a) + f'(a)ε. Higher-order jets use ε with $\varepsilon^{k+1} = 0$.

Mathematical foundation: Jet spaces $J^k(M, N)$ formalize the notion of "k-th order contact" between manifolds. They are the geometric counterpart to truncated Taylor polynomials [Saunders 1989].

Relationship to this work:

Aspect	Dual Numbers / Jets	This System
Infinitesimal	ε with $\varepsilon^{k+1} = 0$ (nilpotent)	z^{-1} (non-nilpotent)
Derivative orders	Truncated at order k	All orders preserved
Information loss	Orders > k discarded	No information lost
Semantic interpretation	Formal algebraic element	"Zero with provenance"

What we add: The non-nilpotent structure preserves all orders indefinitely. Dual numbers lose information at order k+1; our system loses nothing. The provenance interpretation also gives semantic meaning to the infinitesimal dimension that jets lack.

Wheel Theory

What it is: Wheel theory [Carlström 2004] extends fields to make division by zero well-defined. A wheel adds a "bottom" element ⊥ and defines 0/0 = ⊥, x/0 = ⊥ for x ≠ 0.

Key properties:

- Division is total (defined for all inputs)
- 0/0 = ⊥ (a special "undefined" value)
- ⊥ propagates: ⊥ + x = ⊥, ⊥ × x = ⊥

Relationship to this work:

Aspect	Wheel Theory	This System
0/0 result	⊥ (absorbing element)	$1\|_0 = 1$ (usable value)
x/0 result	⊥ (absorbing)	$\|x\|_1$ (infinity with provenance)
Propagation	⊥ contaminates all results	Results remain computable
Distributivity	Modified (not standard)	Preserved (standard ring)
Reversibility	No (⊥ destroys information)	Yes (provenance preserved)

What we add: Wheel theory makes division *total* but produces unusable results. Our system makes division *meaningful* — the result carries provenance and can participate in further computation. $0/0 = 1$ is not arbitrary; it follows from $|1|_{-1} / |1|_{-1} = |1|_0$ by standard Laurent polynomial division.

Non-Standard Analysis (Hyperreals)

What it is: Non-standard analysis [Robinson 1966] constructs a rigorous extension *ℝ of the reals containing actual infinitesimals and infinite numbers. The transfer principle guarantees that first-order statements true in ℝ remain true in* ℝ.

Construction: Hyperreals are typically constructed as ultrapowers: $^*\mathbb{R} = \mathbb{R}^{\mathbb{N}}/U$ where U is a non-principal ultrafilter on ℕ. This requires the Axiom of Choice and produces non-constructive infinitesimals.

Relationship to this work:

Aspect	Hyperreals	This System
Construction	Ultrapower (non-constructive)	Laurent polynomials (constructive)
Requires	Axiom of Choice	Standard algebra only
Transfer principle	Yes (first-order statements)	No
Computability	Not directly implementable	Fully computable (finite polynomial ops)
Infinitesimal type	Abstract (equivalence classes)	Explicit (dimensional coefficients)
Standard part	Via shadow map	Projection to dimension 0

What we add: A *constructive* infinitesimal system. Hyperreals are powerful for proofs but not for computation — you cannot implement an ultrafilter. Our system is fully computable: every operation is a finite polynomial operation. We sacrifice the transfer principle but gain implementability.

Summary: Positioning This Work

We are not claiming to replace these systems. Each has its strengths:

- **Taylor-mode AD** — Production-ready, optimized implementations
- **Dual numbers** — Simple, widely taught, sufficient for first derivatives

- **Wheel theory** — Elegant algebraic treatment of totality
- **Hyperreals** — Foundationally rigorous, powerful transfer principle

Our contribution is a specific combination:

1. **Constructive infinitesimals** (unlike hyperreals)
2. **Unbounded order** (unlike dual numbers)
3. **Usable 0/0** (unlike wheel theory)
4. **No computational graph** (unlike Taylor-mode AD implementations)
5. **Provenance preservation** as a unifying semantic principle

The core insight — that changing `a × 0 = 0` to `a × 0 = |a|₋₁` converts algorithmic calculus into algebraic calculus — is the novel framing. The mathematics is Laurent polynomials; the interpretation is ours.

References

Algebraic foundations:

- Lang, S. (2002). *Algebra* (Revised 3rd ed.). Springer.
- Jacobson, N. (1985). *Basic Algebra I* (2nd ed.). W.H. Freeman.

Automatic differentiation:

- Griewank, A. & Walther, A. (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation* (2nd ed.). SIAM.

Related work:

- Carlström, J. (2004). "Wheels — On Division by Zero." *Mathematical Structures in Computer Science*, 14(1), 143-184.
- Robinson, A. (1966). *Non-standard Analysis*. North-Holland Publishing Company.
- Bennett, C.H. (1973). "Logical Reversibility of Computation." *IBM Journal of Research and Development*, 17(6), 525-532.

Automatic differentiation implementations:

- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S. & Zhang, Q. (2018). *JAX: Composable transformations of Python+NumPy programs.* Software available at github.com/google/jax.
- Bendtsen, C. & Stauning, O. (1996). *FADBAD, a Flexible C++ Package for Automatic Differentiation.* Technical Report IMM-REP-1996-17, Department of Mathematical Modelling, Technical University of Denmark.
- Tan, S. (2022). *TaylorDiff.jl: Fast Higher-order Automatic Differentiation in Julia.* Software available at github.com/JuliaDiff/TaylorDiff.jl.

Differential geometry:

- Saunders, D.J. (1989). *The Geometry of Jet Bundles.* London Mathematical Society Lecture Note Series 142. Cambridge University Press.

Pedagogical context:

- Tall, D. & Vinner, S. (1981). "Concept Image and Concept Definition in Mathematics with Particular Reference to Limits and Continuity." *Educational Studies in Mathematics*, 12(2), 151-169.

Main software repository containing implementation, tests, documentation and examples

- https://github.com/tmilovan/composite-machine