Arithmetic within the VOID Granularity Framework

Mathematics Without Infinity

Konrad Wojnowski
Probabilistic Minds Consortium

December 5, 2024

Contents

1 Introduction

In this chapter, we develop Arithmetic within the **VOID Granularity Framework** (VGF), building upon the Number Theory established in Chapter 1. Arithmetic operations form the foundational tools for mathematical computation and reasoning, and adapting them to the VGF ensures consistency across the framework. We aim to define arithmetic operations that respect the minimal meaningful distinction, δ_{VOID}, and integrate seamlessly with the VOID numbers previously defined. By doing so, we bridge Number Theory and Algebra within the VGF, providing a cohesive mathematical structure.

2 Referencing the Existing Framework

We recall several key axioms and definitions established in earlier parts of the VGF:

- **Axiom 1.1 (Universal VOID Granularity):** δ_{VOID} sets the minimal scale at which distinctions are meaningful.
- **Axiom 1.2 (Onto-Epistemological Granularity):** Links ontological (what exists) and epistemological (what can be known) limitations, indicating that the inability to distinguish elements below δ_{VOID} is inherent both to their existence and our perception.
- **Definition 1.4 (Distinguishability Function $\mu_S(x, y)$):** Initially defined as a probability measure that determines whether two elements x and y are distinguishable, this was previously considered in a simpler, more static form.

We also rely on VOID Natural Numbers, VOID Integers, and VOID Real Numbers from Chapter 3, where numbers represent quantifiable changes exceeding δ_{VOID}.

3 Establishing Continuity and Incorporating Distinguishability Functions

Arithmetic operations must align with the VOID numbers and the constraints imposed by δ_{VOID}. They should produce results that remain meaningful within the framework, whether we consider a stable context (static DF) or an evolving context (dynamic DF), where the observer's capacity $V(t)$ influences distinguishability.

3.1 Static vs. Dynamic Distinguishability

- **Static Distinguishability (Baseline):** Under a static approach, two values x and y are distinguishable if $|x - y| \geq \delta_{\text{VOID}}$. This simple threshold-based rule does not change over time or depend on observer parameters. It ensures a fixed level of granularity: if the difference is smaller than δ_{VOID}, the values are effectively identical within the framework's resolution.
- **Dynamic Distinguishability (Observer-Dependent):** In a dynamic scenario, distinguishability depends on an observer's processing capacity $V(t)$, which can change over time. We can represent the dynamic Distinguishability Function $\mu^{\text{dyn}}_{\text{VOID}}(x, y, t)$

as a sigmoid-like function sensitive to both δ_{VOID} and a similarity measure $S(x, y)$. For example:

$$\mu_{\text{VOID}}^{\text{dyn}}(x, y, t) = \frac{1}{1 + e^{-k[V(t)(1-S(x,y))-\theta]}}$$

where:

- $S(x, y)$ is a discrete similarity measure between x and y.
- $V(t)$ is the observer's processing capacity at time t.
- k and θ are parameters controlling the steepness and threshold of distinguishability.

In this dynamic view, as $V(t)$ increases, the observer can resolve finer differences, effectively making δ_{VOID} more "powerful" because smaller distinctions become noticeable. Conversely, with low $V(t)$, even differences near δ_{VOID} might not be reliably distinguished.

4 Essential Definitions in VGF Arithmetic

4.1 VOID Rounding Function

Definition 4.1. ***VOID Rounding Function*** *void_round*(x)*: For any* $x \in \mathbb{R}_{VOID}$*:*

$$void\_round(x) = \delta_{VOID} \times \left\lfloor \frac{x}{\delta_{VOID}} + \frac{1}{2} \right\rfloor.$$

This function ensures that every arithmetic result snaps to the discrete grid defined by δ_{VOID}. Under static conditions, this grid is fixed. Under dynamic conditions, while δ_{VOID} remains the fundamental increment, the observer's ability to distinguish results may shift how we interpret these rounded values.

4.2 Distinguishability Criterion

- **Static Criterion:**

$$|x - y| \geq \delta_{\text{VOID}} \quad \text{implies distinguishability, otherwise not.}$$

- **Dynamic Criterion:** In a dynamic setting, we might say:

$$\mu_{\text{VOID}}^{\text{dyn}}(x, y, t) \geq \mu_{\text{threshold}}$$

for some $\mu_{\text{threshold}}$ (e.g., 0.5) to declare x and y distinguishable. As $V(t)$ evolves, the capacity to distinguish similar values changes, adding a layer of complexity to arithmetic comparisons.

4.3 Similarity Measure for Dynamic Distinguishability

Definition 4.2. ***Discrete Similarity Measure*** $S(x, y)$*: If needed for the dynamic DF, we define:*

$$S(x, y) = \frac{\#(\textit{Shared Features})}{\#(\textit{Total Features})},$$

yielding a discrete value in $\left\{0, \frac{1}{M}, \ldots, 1\right\}$.

For pure numeric comparisons, we could adapt similarity measures based on how close the ratio $\frac{x}{y}$ *or difference* $(x - y)$ *is to an integer multiple of* δ_{VOID}.

5 VOID Arithmetic Operations

We now define arithmetic operations so they remain consistent with δ_{VOID}, regardless of whether we use a static or dynamic DF.

5.1 VOID Addition ($\oplus_{\textbf{VOID}}$)

Definition 5.1. ***VOID Addition*** $\oplus_{VOID}$*:*

$$x \oplus_{VOID} y = void\_round(x + y).$$

Regardless of static or dynamic conditions, the sum is rounded to adhere to δ_{VOID}. Under dynamic conditions, while the operation itself remains the same, the interpretation of whether two results $x \oplus_{\text{VOID}} y$ and z are distinguishable may depend on $V(t)$ and the dynamic DF.

5.2 VOID Subtraction ($\ominus_{\textbf{VOID}}$)

Definition 5.2. ***VOID Subtraction*** $\ominus_{VOID}$*:*

$$x \ominus_{VOID} y = void\_round(x - y).$$

This ensures that differences land on the δ_{VOID}-grid. Under a dynamic DF, how fine differences are perceived may evolve as $V(t)$ changes.

5.3 VOID Multiplication ($\otimes_{\textbf{VOID}}$)

Definition 5.3. ***VOID Multiplication*** $\otimes_{VOID}$*:*

$$x \otimes_{VOID} y = void\_round(x \times y).$$

The product is discretized. If two products are very close, a dynamic DF might let an observer distinguish them at higher capacity $V(t)$, where previously they could not.

5.4 VOID Division ($\oslash_{\textbf{VOID}}$)

Definition 5.4. ***VOID Division*** $\oslash_{VOID}$*: For* $y \neq 0$*:*

$$x \oslash_{VOID} y = void\_round\left(\frac{x}{y}\right).$$

Division by zero remains undefined. Division results, like all others, are snapped to a δ_{VOID}-based lattice. A dynamic DF could allow finer discernment of quotients as capacity evolves.

6 Properties and Challenges

6.1 Algebraic Properties Under Static vs. Dynamic DF

- **Commutativity:** VOID addition and multiplication remain commutative under both static and dynamic DF scenarios.
- **Associativity and Distributivity:** Due to rounding, these properties may not hold strictly. The dynamic DF does not restore classical properties but may influence when small discrepancies become noticeable. As $\delta_{\text{VOID}} \to 0$, classical properties are better approximated. Similarly, as $V(t) \to \infty$ (high capacity), the dynamic DF allows for finer granularity, making arithmetic appear more classical.

6.2 Interplay of Dynamic DF and Arithmetic

Under dynamic DF:

- If $V(t)$ is small, the observer cannot distinguish values that differ slightly above δ_{VOID}. Arithmetic might feel more "coarse."
- As $V(t)$ increases, previously indistinguishable results become distinguishable, effectively refining the observer's experience of the arithmetic space. The arithmetic definitions remain the same, but the interpretation and comparison of results change.

This creates a layered understanding:

- **Static Layer:** The arithmetic rules themselves, fixed by δ_{VOID}.
- **Dynamic Layer:** How the observer perceives and compares results evolves over time, influenced by $V(t)$.

7 Practical Considerations and Examples

7.1 Algorithms

The core algorithms for VOID arithmetic remain:

Listing 1: VOID Arithmetic Algorithms

```
def void_round(x, delta_VOID):
    return round(x / delta_VOID) * delta_VOID

def void_add(x, y, delta_VOID):
    return void_round(x + y, delta_VOID)

def void_subtract(x, y, delta_VOID):
    return void_round(x - y, delta_VOID)

def void_multiply(x, y, delta_VOID):
    return void_round(x * y, delta_VOID)

def void_divide(x, y, delta_VOID):
```

```
    if y == 0:
        raise ValueError("Division by zero is undefined.")
    return void_round(x / y, delta_VOID)
```

The difference under dynamic DF is not in the arithmetic itself, but in how we interpret the results $x \oplus_{\text{VOID}} y$ vs. z. We might compute $\mu^{\text{dyn}}_{\text{VOID}}(x, y, t)$ post-hoc to decide if results are distinct enough to matter.

7.2 Examples

7.2.1 Static Example (No Capacity Involved)

Let $\delta_{\text{VOID}} = 0.05$, $x = 2.3$, $y = 1.7$:

- $x \oplus_{\text{VOID}} y = \text{void\_round}(4.0) = 4.0$.

No dynamic factor influences whether we consider 4.0 distinguishable from, say, 4.05. By static DF alone, $|4.0 - 4.05| = 0.05 \geq \delta_{\text{VOID}}$ means distinguishable.

7.2.2 Dynamic Example (Capacity Involved)

Let $\delta_{\text{VOID}} = 0.05$, $V(t) = 10$, and consider two results 4.0 and 4.05:

$$\mu^{\text{dyn}}_{\text{VOID}}(4.0, 4.05, t) = \frac{1}{1 + e^{-k[10(1-S(4.0,4.05))-\theta]}}$$

If $S(4.0, 4.05)$ is high (close to 1), then $1 - S(4.0, 4.05)$ is small, making the argument in the exponent possibly small as well. Depending on k and θ, the observer might or might not distinguish them. Increasing $V(t)$ further would increase the chance they are distinguished.

8 Potential Paths Forward

- **Formalizing Dynamic DF:** Provide rigorous definitions and proofs for properties of arithmetic under changing $V(t)$.
- **Adapting Data Structures and Computations:** Explore how data structures or numerical algorithms might be optimized when the ability to distinguish results is not static but observer-dependent.
- **Beyond Real Numbers:** Extend to VOID complex numbers or vector spaces, where dynamic DF might enrich geometric interpretations, eventually leading into probabilistic geometry.

9 Conclusion

By introducing both static and dynamic Distinguishability Functions into the arithmetic layer of the VGF, we reconcile the need for a stable, δ_{VOID}-based arithmetic with the evolving perceptual capacities of an observer. The static DF provides a baseline framework for discrete arithmetic operations, while the dynamic DF adds a layer of adaptive interpretation, making the distinction between values capacity-dependent.

This integrated approach lays a robust foundation for future work in algebraic structures, advanced number systems, and eventually probabilistic geometry, where both fixed and evolving conditions matter. We have thus created a versatile arithmetic framework that acknowledges the finite, discrete nature of VOID mathematics and the potentially dynamic nature of observation and perception within the VGF.