Geometry of Arithmetic Expressions: I. Basic Concepts and Unsolved Problems (Draft)

A preprint

Mingli Yuan
Swarma Research
Beijing, 100083
mingli.yuan@gmail.com

May 5, 2025

Abstract

This paper introduces a novel geometric framework for studying arithmetic expressions, establishing a rigorous connection between algebraic operations and hyperbolic geometry. We formalize arithmetic expressions as syntactic structures and demonstrate how they can be embedded into continuous geometric spaces where addition and multiplication correspond to movements along orthogonal directions. Central to our approach is a flow equation that governs how expression values propagate through this geometric space. We construct the first kind arithmetic expression space $\mathfrak{E}_1$ on the upper half-plane with a hyperbolic metric, where the assignment function satisfies the flow equation and serves as an eigenfunction of the Laplacian. This construction reveals that arithmetic torsion—the non-commutativity of addition and multiplication—directly corresponds to geometric area, analogous to how curvature measures deviation from flatness. The paper establishes arithmetic expressions as geometric objects with intrinsic invariants, opening new avenues for exploring the interplay between computation and geometry.

***K*eywords** arithmetic expressions, hyperbolic geometry

Contents

1 Introduction

From the syntactic trees of arithmetic expressions to the hyperbolic geometry of modular forms, the interplay between discrete algebra and continuous geometry has long fascinated mathematicians. This work addresses a fundamental question: Can the evaluation dynamics of arithmetic expressions themselves form a geometric space? We demonstrate that threadlike arithmetic expressions naturally embed into hyperbolic surfaces through a novel flow equation.

1.1 Key Definitions

We formalize arithmetic expressions $a \in \mathbb{E}[\mathbb{Q}]$ using production rules that generate terms through addition, subtraction, multiplication, and division operations. The evaluation $\nu(a)$ of these expressions can be viewed from multiple perspectives:

- *Syntactically* as tree structures with branch nodes (operators) and leaf nodes (constants)
- *Algebraically* as compositions of elementary operations
- *Geometrically* as paths through a continuous space

Of particular importance are threadlike expressions, where all left nodes are leaf nodes. These expressions, analogous to paths in homotopy theory, provide a natural bridge between algebraic and geometric perspectives.

Through currying and path notation, we establish a formal framework for representing threadlike expressions as sequences of elementary operations:

$$xa_1a_2 \cdots a_n \coloneqq a_n(a_{n-1}(\cdots a_2(a_1(x))\cdots)) \tag{1}$$

This representation reveals that the commutator of addition and multiplication operations exhibits a non-trivial torsion:

$$\tau = x \oplus_\mu \otimes_\lambda - x \otimes_\lambda \oplus_\mu = \mu(e^\lambda - 1) \tag{2}$$

1.2 Foundational Results

The central insight of our approach is that arithmetic operations—specifically addition and multiplication—can be interpreted as movements along orthogonal directions in a properly constructed geometric space. This interpretation transforms arithmetic evaluation into geometric propagation, with expression values corresponding to points in a hyperbolic manifold.

The embedding of arithmetic expressions into geometry is governed by a flow equation:

$$\frac{da}{ds} = \mu \cos\theta + a\lambda \sin\theta \tag{3}$$

This partial differential equation describes how assignment values propagate through space along directions with angle θ. In its coordinate-free form:

$$\|\nabla a\| = \sqrt{\mu^2 + a^2\lambda^2} \tag{4}$$

This is an Eikonal equation equivalent to a special Hamilton-Jacobi equation, connecting our construction to fundamental concepts in analytical mechanics and differential geometry.

We establish a specific realization of this framework in the first kind arithmetic expression space $\mathfrak{E}_1$, defined on the upper half-plane $\mathcal{B} = \{(x, y) \mid y > 0\}$ with a hyperbolic metric:

$$ds^2 = \frac{1}{y^2}\left(\frac{dx^2}{\mu^2} + \frac{dy^2}{\lambda^2}\right) \tag{5}$$

In this space, the assignment function $a = -\frac{x}{y}$ satisfies the flow equation with parameters μ and λ. Moreover, a is an eigenfunction of the Laplacian with eigenvalue 2, reinforcing the intrinsic geometric nature of this construction.

1.3 Implications

1.4 Roadmap

Our main results establish: (1) A flow equation governing arithmetic propagation in curved spaces; (2) The $\mathfrak{E}_1$ space as a universal geometric framework for expression evaluation;

Through this work, we demonstrate that arithmetic expressions do indeed form a geometric space—one with rich structure and deep connections to hyperbolic geometry, differential equations, and algebraic systems.

2 Basic concepts

Historically, researchers such as Post[10] and Chomsky[1] linked formal grammars to automata and rewriting systems, providing the classic recipe for generating well-formed strings via production rules. Meanwhile, foundational work by Russell[13], Church[2][3], and later Martin-Löf[7][8] introduced type-theoretic frameworks to avoid paradoxes and to systematically handle partial operations, especially critical when dealing with real numbers and division by zero. These two trajectories—rewriting rules versus type discipline—ultimately converge on the need for both syntactic generativity and semantic rigor[4][14].

In this paper, we begin by defining arithmetic expressions through basic production rules for addition, subtraction, multiplication, and division over $\mathbb{Q}$. We keep this first step lean, acknowledging that purely rewriting-based definitions can clash with hidden subtleties when extended to $\mathbb{R}$. The result is a fertile middle ground: a tree-structured syntax and partial evaluation that highlight geometric properties (e.g., "threadlike" paths), but also carry unresolved issues around singularities, infinite expansions, and semantic gaps. We postpone those deeper type-theoretic and topological remedies to the final sections, where we address how these complexities reveal new facets of "arithmetic torsion" and other structural phenomena in the geometry of expressions.

2.1 Arithmetic expression

In order to define arithmetic expressions involving real numbers $\mathbb{R}$ in a rigorous way, we need to use a sophisticated type theory. However, in order to keep things simple and maintain clarity, we will start by using only production rules, but with certain semantic restrictions. We will also begin with rational numbers $\mathbb{Q}$ to avoid the difficulties inside real numbers $\mathbb{R}$.

Definition 2.1. *An arithmetic expression a over $\mathbb{Q}$ is a structure given by the following production rules:*

$$\begin{aligned} a &\longleftarrow x \\ a &\longleftarrow (a+a) \\ a &\longleftarrow (a-a) \\ a &\longleftarrow (a\times a) \\ a &\longleftarrow (a\div a) \end{aligned} \tag{6}$$

where $x \in \mathbb{Q}$, and we denote this as $a \in \mathbb{E}[\mathbb{Q}]$.

During the production process, we can obtain both a string representation and a tree representation of arithmetic expression a, where the two representations are equivalent. For instance, the string representation of a might be:

$$(((((1\times 2)\times 2)-1)\times (2+1))-6) \tag{7}$$

and the parsed syntax tree is depicted in Figure 1.

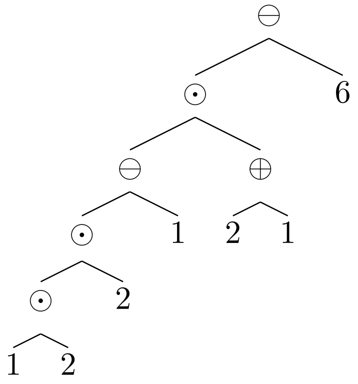

Figure 1: a tree representation of an arithmetic expression

If we interpret the target as a string and the building processes in production rule (6) as string building, we get the *string representation*. On the other hand, if the target is a tree, tree building leads to the *tree representation*. We can easily obtain the string representation of a from its tree representation by performing a pre-order traversal.

The concept of a *sub-expression* can also be derived from the concept of a subtree. The branch nodes are all labeled with operators: $+$, $-$, $\times$, $\div$. The leaf nodes are all labeled with numbers.

Evaluation ν is a partial function that operates on arithmetic expression $a \in \mathbb{E}[\mathbb{Q}]$. It is undefined only if division by zero occurs during the recursive evaluation process.

We can define evaluation $\nu(a)$ of a recursively as follows:

- Constant leaf: for any $x \in \mathbb{Q}$, $\nu(x) = x$.
- Compositional node by $+$: For any $(a + b)$, $\nu((a + b)) = \nu(a) + \nu(b)$.
- Compositional node by $-$: For any $(a - b)$, $\nu((a - b)) = \nu(a) - \nu(b)$.
- Compositional node by $\times$: For any $(a \times b)$, $\nu((a \times b)) = \nu(a)\nu(b)$.
- Compositional node by $\div$: For any $(a \div b)$, if $\nu(b) \neq 0$, then $\nu((a \div b)) = \nu(a)/\nu(b)$.

We say that an arithmetic expression a is *evaluable* if $\nu(a)$ is defined. In the rest of this article, we will only consider evaluable arithmetic expressions unless stated otherwise.

Given an arithmetic expression a, whatever evaluable or not, we can obtain its tree representation. If a node l is a leaf node, its corresponding subexpression s is a number, so we consider it to be already "evaluated". If a node b is a branch node, its corresponding subexpression s is an expression, and we can apply ν to it to obtain a number $\nu(s)$. During the recursive evaluation process, starting from the leaves and moving towards the root, the subexpressions are evaluated one after another. However, the order of evaluations is generally not unique.

Definition 2.2. *The evaluation order of an arithmetic expression a is an ordering of branch nodes in the tree representation of a such that every node (sub-expression) is evaluated before its parent.*

For example, the possible evaluation orders of the arithmetic expression in Figure 1 are:

- $1 \times 2 \to \underline{2}; \underline{2} \times 2 \to \underline{4}; \underline{4} - 1 \to \underline{3}; 2 + 1 \to \underline{3}; \underline{3} \times \underline{3} \to \underline{9}; \underline{9} - 6 \to 3$
- $1 \times 2 \to \underline{2}; \underline{2} \times 2 \to \underline{4}; 2 + 1 \to \underline{3}; \underline{4} - 1 \to \underline{3}; \underline{3} \times \underline{3} \to \underline{9}; \underline{9} - 6 \to 3$
- $1 \times 2 \to \underline{2}; 2 + 1 \to \underline{3}; \underline{2} \times 2 \to \underline{4}; \underline{4} - 1 \to \underline{3}; \underline{3} \times \underline{3} \to \underline{9}; \underline{9} - 6 \to 3$
- $2 + 1 \to \underline{3}; 1 \times 2 \to \underline{2}; \underline{2} \times 2 \to \underline{4}; \underline{4} - 1 \to \underline{3}; \underline{3} \times \underline{3} \to \underline{9}; \underline{9} - 6 \to 3$

The underlined numbers are the numbers that are evaluated during the evaluation process.

Below are examples of expressions that have a unique evaluation order. These include right-expanded, left-expanded, and combinations of them, as shown in Figure 2 and Figure 3.

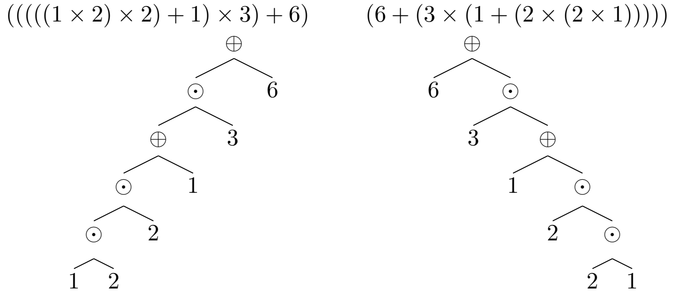

Figure 2: right-expanded and left-expanded expressions

The evaluation order of an arithmetic expression is related to the topological order of its tree representation, but they are not the same. The topological order of a tree is an ordering of nodes such that every node is visited before its parent[5]. However, we are only interested in the ordering of branch nodes, as leaf nodes have already been evaluated and can be ignored. Additionally, the topological order goes from parent to children, while the evaluation order goes from children to parent.

Definition 2.3. *A threadlike expression is an arithmetic expression that all the left nodes in its tree representation are leaf nodes.*

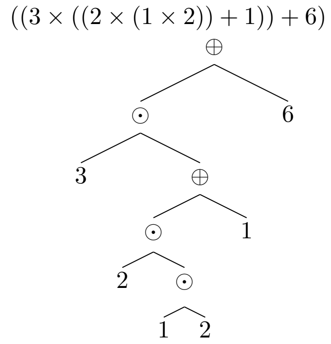

Figure 3: combinations of right-expanded and left-expanded expressions

So a threadlike expression is right-expanded and its evaluation order is unique. One example of threadlike expressions is shown on the left side of Figure 2.

Threadlike expressions are significant here because they are analogous to the concept of paths in homotopy theory in geometry. In a more general context, certain special types of threadlike expressions are also interesting: for example, *alternating threadlike expressions* are expressions in which the additional and multiplicative operators appear in an alternating manner. In the field of computing, a hardware component called *multiplier-accumulator* (MAC) unit has been implemented [11], which is a special case of an alternating threadlike expression. As a result, some numerical algorithms based on MAC units have been studied [6].

2.2 A scalar field and a mesh grid

Consider the upper half plane $\{\mathcal{H} : (x, y)|y > 0\}$ equipped with an inner product and metrics defined as follows:

$$\mathbf{a} \cdot \mathbf{b} = [a_x \quad a_y] \begin{bmatrix} \frac{1}{y^2} & 0 \\ 0 & \frac{1}{y^2 \ln^2 2} \end{bmatrix} \begin{bmatrix} b_x \\ b_y \end{bmatrix}$$

and

$$ds^2 = \frac{1}{y^2}(dx^2 + \frac{dy^2}{\ln^2 2})$$

We consider a scalar field satisfying

$$a = -\frac{x}{y} \tag{8}$$

We call this field an *assignment*.

Proper assignments allow us to establish a connection between paths in homotopy and threadlike arithmetic expressions, and to incorporate function theory into the study of arithmetic expression geometry.

We can draw a grid on the scalar field A and underlying upper half plane $\mathcal{H}$ as shown in Figure 4. The blue lines encode a $+1$ relationship, the green lines encode a $\times 2$ relationship, and they are line families that are perpendicular to each other. The length of the line segments between two neighboring crossing points are unit length(calculations in lemma **??**). The red value at the crossing points is the value of the scalar field A at that point. Based on the relationships encoded by the lines, we can encode threadlike arithmetic expressions, which will be introduced in the subsection 2.3.

The addition-multiplication grid is also scale-invariant under the transformation

$$\begin{cases} x' = \alpha x \\ y' = \alpha y \end{cases}$$

where $\alpha = 2^k, k \in \mathbb{Z}$.

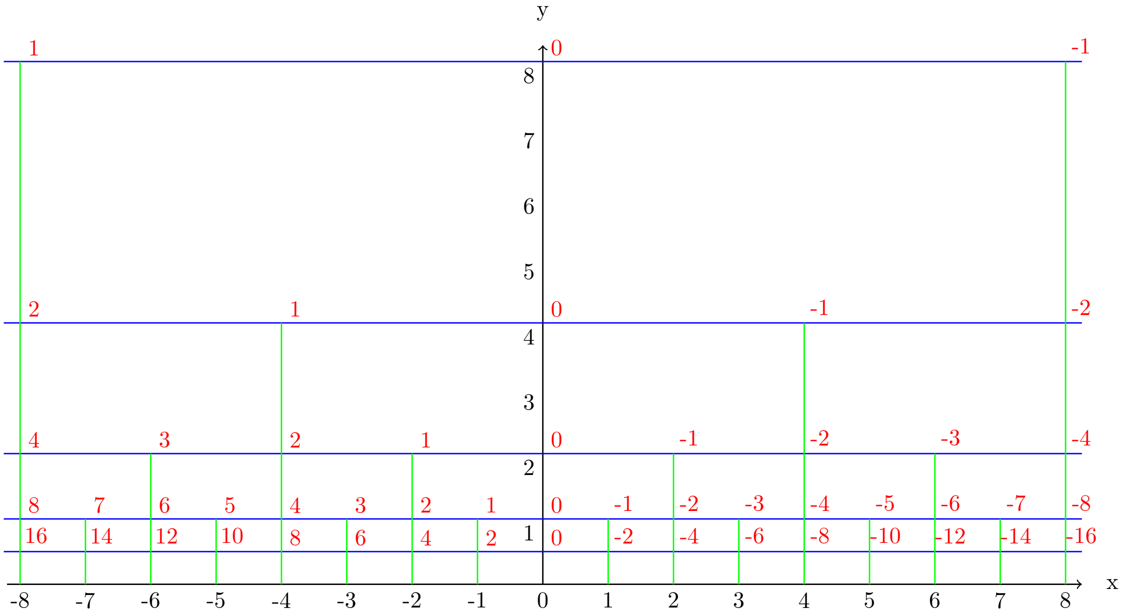

Figure 4: An addition-multiplication grid by generators with $\mu = 1$ and $\lambda = \ln 2$

We can imagine if we make the grid finer and finer, the grid will become a continuous space. This leads to a rigorous treatment of arithmetic expressions as a geometric space in section **??**.

2.3 Encoding threadlike expressions on the addition-multiplication grid

If we interpret the horizontal blue lines as $+1$ and the vertical green lines as $\times 2$ in Figure 4, we can encode threadlike expressions on the addition-multiplication grid. For example, in Figure 5 we encode $((((1 \times 4) - 1) \times 2) - 3)$ as the bold black lines.

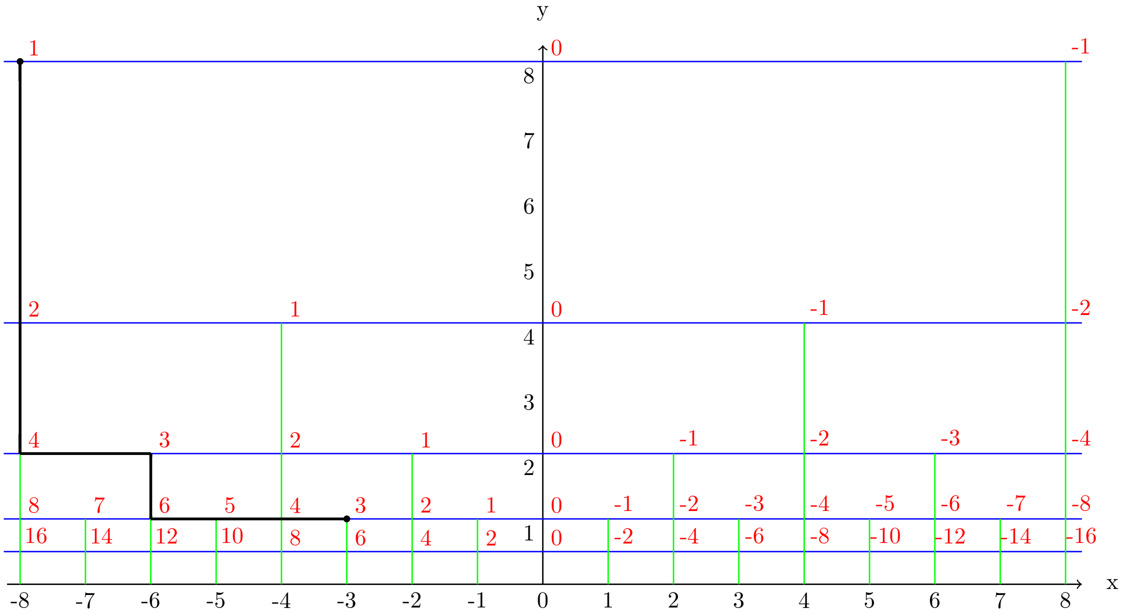

Figure 5: encoding threadlike expression

The zigzag lines in Figure 5 can be divided into four parts:

- the vertical line from 1 to 4: encoded as multiplication by 4
- the horizontal line from 4 to 3: encoded as subtraction by 1
- the vertical line from 3 to 6: encoded as multiplication by 2
- the horizontal line from 6 to 3: encoded as subtraction by 3

2.4 From a scalar field to a space of threadlike expressions

As shown in Figure 6, we have the following paths and expressions:

- the black path: $((1 \times 8) - 5) = 3$
- the purple path: $((1 - \frac{5}{8}) \times 8) = 3$
- the brown path: $((((((1 - \frac{1}{8}) \times 2) - \frac{1}{2}) \times 2) - 1) \times 2) = 3$
- the orange path: infinite many addition-multiplication terms accumulated together, a special kind of integration

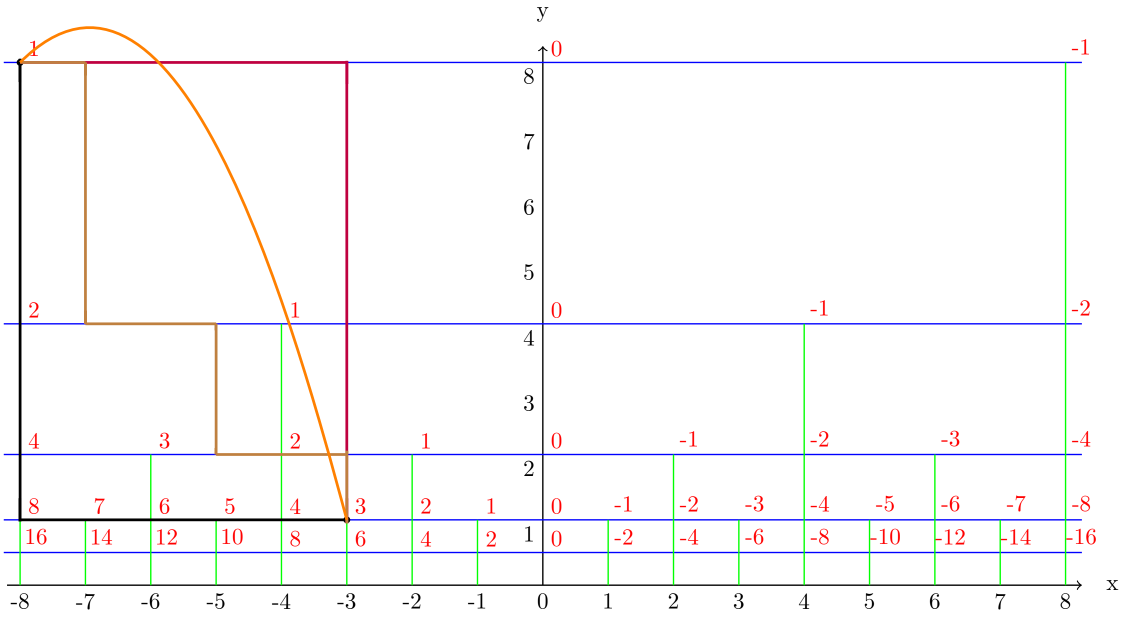

Figure 6: different encodings and their canonical form

All of the paths in Figure 6 have the same source 1 and same target 3. We will discuss a canonical form for these paths.

It is easy to see that the expressions can be transformed into each other by using the multiplication distributive law and by combining and decomposing terms.

Conversion form brown path to black path

$$3 = ((((((1 - \frac{1}{8}) \times 2) - \frac{1}{2}) \times 2) - 1) \times 2) \tag{9}$$

$$= 1 \times 8 - \frac{1}{8} \times 8 - \frac{1}{2} \times 4 - 1 \times 2 \tag{10}$$

$$= ((1 \times 8) - 5) \tag{11}$$

Conversion form brown path to purple path

$$3 = ((((((1 - \frac{1}{8}) \times 2) - \frac{1}{2}) \times 2) - 1) \times 2) \tag{12}$$

$$= (1 - \frac{1}{8}) \times 8 - \frac{1}{2} \times 4 - 1 \times 2 \tag{13}$$

$$= (1 - \frac{1}{8}) \times 8 - \frac{1}{4} \times 8 - \frac{1}{4} \times 8 \tag{14}$$

$$= (1 - \frac{1}{8} - \frac{1}{4} - \frac{1}{4}) \times 8 \tag{15}$$

$$= ((1 - \frac{5}{8}) \times 8) \tag{16}$$

Therefore, we can define the black and purple paths in Figure 6 as a pair of canonical paths, which represent all threadlike expressions connecting the source 1 and the target 3.

Once we have such canonical paths, we can determine the canonical form of the whole space relative to an arbitrary source point O and any other target point P. This allows us to define the space as a space of threadlike expressions.

2.5 Currying and path notation

Currying is a basic technique in functional programming[12], which is used to transform a function with multiple arguments into a sequence of functions with one argument. By currying a threadlike arithmetic expression, we can obtain a sequence of functions that operate on an operand, which is the leftmost leaf node.

We introduce the following notation for currying a threadlike arithmetic expression:

- initial operand: the leftmost leaf node
- operator: $\oplus_y : x \mapsto x + y$
- operator: $\ominus_y : x \mapsto x - y$
- operator: $\otimes_y : x \mapsto x \cdot e^y$
- operator: $\oslash_y : x \mapsto x \cdot e^{-y}$

For example, the threadlike arithmetic expression $(((((1 \times 2) \times 2) + 1) \times 3) + 6)$ can be curried as

$$\oplus_6(\otimes_{\ln 3}(\oplus_1(\otimes_{\ln 2}(\otimes_{\ln 2}(1)))))$$

Suppose we have a series of operators $a_1, a_2, \cdots a_{n-1}, a_n$, we introduce a *path notation*.

$$xa_1a_2 \cdots a_{n-1}a_n \coloneqq a\_n(a_{n-1}(\cdots a_2(a_1(x)) \cdots))$$

So, the above example can be written as

$$1 \otimes_{\ln 2} \otimes_{\ln 2} \oplus_1 \otimes_{\ln 3} \oplus_6$$

If a path begins with a number, we refer to it as a *bounded path*. If it does not, we refer to it as a *free path*, similar to the concept of vectors from the origin versus vectors at arbitrary points. a bounded path results in a number, while a free path results in a function.

Now we will verify that the operators within a path are associative.

Lemma 2.1. *The operators within a path are associative, i.e. we have*

$$a[bc] = [ab]c$$

Proof. We use normal typeface to express the path notation, and bold typeface to express the function notation.

For a free path, follow the definition, we have

$$a[bc] = [bc](\mathbf{a}) = \mathbf{c}(\mathbf{b}(\mathbf{a}))$$

$$[ab]c = \mathbf{c}([ab]) = \mathbf{c}(\mathbf{b}(\mathbf{a}))$$

hence, we have

$$a[bc] = [ab]c$$

is hold for a free path.

For a bounded path, we have

$$xa[bc] = [bc](\mathbf{a}(x)) = \mathbf{c}(\mathbf{b}(\mathbf{a}(x)))$$
$$x[ab]c = \mathbf{c}([ab](x)) = \mathbf{c}(\mathbf{b}(\mathbf{a}(x)))$$

hence, we have

$$a[bc] = [ab]c$$

is hold for a bounded path.

□

Definition 2.4. *The concatenation of paths $p_1 \cdot p_2$ is defined as the composite of functions:*

$$p_1 \cdot p_2 \coloneqq p_2 \circ p_1$$

When a sequence of paths is concatenated, and only the first path can be bounded. If the first path is bounded, the concatenated result is a bounded path. Otherwise, the concatenated result is a free path.

2.6 Alternating threadlike expressions

Now we can define alternating threadlike expressions, which were mentioned in Section **??**, using the path notion.

$$\alpha = a_1 b_1 a_2 b_2 \cdots a_l b_l, a_i = \otimes_{\lambda_i}, b_i = \oplus_{\mu_i}, \lambda_i, \mu_i \in \mathbb{R} \tag{17}$$

where $\bigoplus$ and $\bigotimes$ denote addition and multiplication, respectively, and the expression is a zigzag of alternating addition and multiplication operations. α is a free path, and we can bind a number to it.

Since 0 is the identity element for addition and 1 is the identity element for multiplication, it is straightforward to see that any arithmetic expression can be converted into an alternating threadlike expression by introducing more 0 and 1 into the original expression. So alternating threadlike expression is a kind of canonical form.

We can derive a formula for perturbations in alternating threadlike expressions.

Let us define the left-to-right accumulated sum of λ_i as $\check{\lambda}_i$, such that:

$$\check{\lambda}_i = \sum_{j=1}^{i} \lambda_j, \check{\lambda}_0 = 0 \tag{18}$$

Then we also have right-to-left accumulated sum of λ_i

$$\hat{\lambda}_i = \check{\lambda}_l - \check{\lambda}_{l-i}, \hat{\lambda}_0 = 0 \tag{19}$$

Expanding equation (17) using the distributive law and the above notion at point μ_0, we obtain:

$$\alpha(\mu_0) = e^{\lambda_l}(\cdots(e^{\lambda_2}(e^{\lambda_1}\mu_0 + \mu_1) + \mu_2)\cdots) + \mu_l \tag{20}$$
$$= e^{\hat{\lambda}_l}\mu_0 + e^{\hat{\lambda}_{l-1}}\mu_1 + e^{\hat{\lambda}_{l-2}}\mu_2 + \cdots + e^{\hat{\lambda}_1}\mu_{l-1} + e^{\hat{\lambda}_0}\mu_l \tag{21}$$

Next, at the starting point μ_0, we introduce a perturbation $\tilde{\mu}_0 = e^{\eta_0}\mu_0 + \epsilon_0$, where η_0 and ϵ_0 are the disturbance terms added by the summation and multiplication operations, respectively. Then, we have:

$$\alpha(\tilde{\mu}_0) = e^{\hat{\lambda}_l}(\tilde{\mu}_0) + e^{\hat{\lambda}_{l-1}}\mu_1 + e^{\hat{\lambda}_{l-2}}\mu_2 + \cdots + e^{\hat{\lambda}_1}\mu_{l-1} + e^{\hat{\lambda}_0}\mu_l \tag{22}$$
$$= \alpha(\mu_0) + e^{\hat{\lambda}_l}(\tilde{\mu}_0 - \mu_0) \tag{23}$$

As a result, purely from an arithmetic perspective, without the need for limits, we can derive the following meaningful ratio:

$$\frac{\alpha(\tilde{\mu}_0) - \alpha(\mu_0)}{\tilde{\mu}_0 - \mu_0} = e^{\hat{\lambda}_l} = e^{\check{\lambda}_l} \tag{24}$$

Now we extend this relationship from the starting point μ_0 to the entire process, we define the recursive formula

$$w_i = e^{\lambda_i} w_{i-1} + \mu_i, w_0 = 0$$

and then we have

$$\frac{\tilde{w}_i - w_i}{\tilde{\mu}_0 - \mu_0} = e^{\check{\lambda}_i}, i \in \{1, ..., l\} \tag{25}$$

So, we have

$$\tilde{w}_i - w_i = e^{\check{\lambda}_i}(\tilde{\mu}_0 - \mu_0)$$

and hence

$$\tilde{w}_i - w_i = e^{\lambda_i}(\tilde{w}_{i-1} - w_{i-1}) \tag{26}$$

That means the perturbation along the path is controlled by the multiplication terms of e^{λ_i}.

2.7 Generated structure, commutator and arithmetic torsion

In order to study mesh grids like the one described in subsection 2.2, we need to investigate the algebraic structure of the threadlike arithmetic expressions that are generated.

For real number $\mathbb{R}$ and elements $\mu, \lambda \in \mathbb{R}$, we consider all the arithmetical expressions that are freely generated from

- initial operand: 0
- operator: $\oplus_\mu : x \mapsto x + \mu$
- operator: $\ominus_\mu : x \mapsto x - \mu$
- operator: $\otimes_\lambda : x \mapsto x \cdot e^\lambda$
- operator: $\oslash_\lambda : x \mapsto x \cdot e^{-\lambda}$

We denote these expressions as $E(\mu, \lambda)$, where μ is the additional generator and e^λ is the multiplicative generator. In cases where the context is clear, we may omit μ and λ from the index. Our goal is not to study only a single $E(\mu, \lambda)$, but rather to use a family of $E(\mu, \lambda)$ to approach a continuous space.

Since $\oplus_\mu$ and $\ominus_\mu$ are mutually inverse operations, it follows that $\otimes_\lambda$ and $\oslash_\lambda$ are also mutually inverse. This means that $E(\mu, \lambda)$ forms a group. An observation is that the commutator of this group is not equal to identity generally, especially the commutator of the generators.

$$x \oplus_\mu \otimes_\lambda \ominus_\mu \oslash_\lambda - x = \mu(1 - e^{-\lambda}) \tag{27}$$
$$x \otimes_\lambda \oplus_\mu \oslash_\lambda \ominus_\mu - x = -\mu(1 - e^{-\lambda}) \tag{28}$$

Formula 27 obey the right-hand rule, and formula 28 obey the left-hand rule.

Or equivlently[1], we define below difference τ obey the right-hand rule:

$$\tau = x \oplus_\mu \otimes_\lambda - x \otimes_\lambda \oplus_\mu = \mu(e^\lambda - 1) \tag{29}$$

These differences are constant, indicating a type of torsion in the generated group. And torsion τ is specifically referred to as the arithmetic torsion.

We will reveal that τ is related to the curvature of the surface in later sections.

[1] Please reference section 3.4, the equivlence here is refered to the same order of the infinitesimal

2.8 Levels of Equality, Singularity, and Symmetry Problems

From theoretical and computational perspectives, it is useful to consider different levels of equality within structures generated freely from arithmetic operations. These levels reveal distinct stages of abstraction, from the most basic sequential representation to the final numerical or geometric realization. We can distinguish at least the following key levels:

1. **Literal Equality**:
   This is the finest level of equality, judged directly by the sequence of operations (or its string representation). Two threadlike expressions are literally equal only if their sequences are identical. While this level might be too strict to reflect computational outcomes, it provides the richest *base textures* without any imposed algebraic constraints, forming the foundation upon which more complex structures and spaces can be woven.

2. **Operational/Syntactic Equality**:
   This level is based on fundamental rules governing the composition of operations, chief among them being **associativity** (as proven in Lemma **??**). Associativity ensures that the result of composing a sequence of operations is well-defined, independent of the grouping (parenthesization). This endows our arithmetic expressions with at least a foundational **monoid** or **groupoid** structure, enabling the consistent definition of "paths" and their composition, which is essential for geometrization. It is crucial to note that the sequential path operations central to this framework (e.g., compositions of addition $\oplus_\mu$ and multiplication $\otimes_\lambda$) generally do **not** satisfy the standard **distributive law** at this level. This is evidenced by the non-zero arithmetic torsion τ, a key feature indicating the non-commutative and non-distributive nature of these combined operations.

3. **Relational/Algebraic Equality**:
   Building upon the operational structure above, specific contexts often introduce **additional, concrete algebraic relations**. These relations might stem from:

   - The existence of inverses for the operations, leading to a group structure. * Geometric or topological constraints, such as identities that must be satisfied by relators in a knot group. * Arithmetic identities satisfied by parameters involved in the operations, such as a minimal polynomial $P(\lambda) = 0$.

   This level of equality is defined by the **specific presentation (generators and relations)** of the resulting algebraic structure, shaping the particular algebraic object of interest (e.g., a specific group). Formal tools from combinatorial group theory, such as **HNN extensions** and **amalgamated free products**, provide a precise framework for describing these algebraic structures arising from geometric, topological, or arithmetic constraints. Results like Britton's Lemma ensure that these relationally-defined structures are typically non-trivial.

4. **Semantic Equality**:
   This is the coarsest level of equality, determined by the final **numerical evaluation** of the expression or by the **value of the assignment function** a in a geometric model. Two expressions might be distinct at the Relational/Algebraic level (representing different group elements) but still yield the same semantic value. This level directly corresponds to the computational output or the state within the geometric realization.

Hierarchy and Challenges:

We can view these levels of equality as forming a lattice or hierarchy based on refinement, ordered by implication from finer to coarser levels:

$$E_{\text{Literal}} \implies E_{\text{Op/Syn}} \implies E_{\text{Rel/Alg}} \implies E_{\text{Semantic}}$$

where E_X denotes "equality at level X", and $\implies$ means "implies". A finer equality necessarily satisfies any coarser equality.

It is important, however, to distinguish this lattice of *equality relations* (ordered by refinement) from the relationship between the corresponding *algebraic structures*. For instance, in the context of HNN extensions relevant to fibered knots (corresponding to the Relational/Algebraic level, e.g., the knot group $\pi_1(K)$), the structure typically contains the base group (corresponding to the finer Operational/Syntactic level, e.g., $\pi_1(F)$) as a (normal) **subgroup**. That is, the algebraic structure defined by the coarser ("higher" in the lattice) equality relation embeds the structure defined by the finer ("lower") relation. This is not a contradiction; it highlights that a coarser equivalence relation (identifying more pairs) can define a larger, more complex algebraic structure which contains the structure defined by the finer relation as a component.

Significant theoretical challenges arise from the "distance" between these different levels, particularly between the Relational/Algebraic structure and its Semantic/Geometric realization. For example:

- **Role of Background Algebra**: While the AEG path operations themselves might be non-distributive, the discussion of Relational/Algebraic equality (e.g., involving $P(\lambda) = 0$) relies on the parameters (like λ) residing within standard algebraic systems (fields, rings) where distributivity holds, allowing polynomials and minimal polynomials to be defined. These relations from standard algebra are then imposed as constraints on the non-distributive AEG operational structure.
- **Singularity Problem**: Certain literally or syntactically valid expressions might be semantically invalid (e.g., involving division by zero), especially when extending definitions from $\mathbb{Q}$ to $\mathbb{R}$ or $\mathbb{C}$. How do these semantic singularities manifest in the geometric model? Do they correspond to singularities in the space or specific behaviors of the assignment function a?
- **Symmetry Problem**: What is the relationship between the symmetries of the algebraic structure (e.g., automorphisms of the HNN group) and the symmetries of the final geometric space (e.g., isometries of the $\mathfrak{E}_1$ space)? How does the process of constructing the geometric space ("weaving the textures into a space") selectively realize or break the symmetries present at the algebraic level? Different levels of equality correspond to different "symmetry resources".

Exploring these problems is central to our research, aiming to develop a unified theoretical framework capable of accommodating these different structural levels and revealing the deep connections between arithmetic, algebra, geometry, and topology.

Foundational problem: A careful reader may have noticed that the definition 2.1 is based on rational numbers $\mathbb{Q}$. Why can't we use real numbers $\mathbb{R}$ instead? The answer is that syntactically valid expressions may not be semantically valid. Dividing by zero can lead to invalid expressions, and the evaluation of the expression cannot be defined in this situation. Therefore, in real numbers, an expression may be syntactically valid but semantically not valid, and there is no algorithm that can decide whether an expression is semantically valid or not. How can we bridge this gap and provide a continuous geometry space? We will attempt to partially solve this problem in some special cases in section **??**.

Singular point problem: We have a very strong intuition that semantically invalid expressions lead to singular points. The way we discussed in complex analysis may be borrowed here: essential singularities and poles.

Symmetry and classification problem: We conjecture that the equality lattice may not only play a role in the construction of a space, but also determine the symmetry of that space. We can imagine that, at certain levels of the lattice, we weave syntactically generated substructures into points to form a space, and the weaving process uses up some symmetrical resources, leaving the rest to form a symmetry on the space. The structure within the total symmetry may provide us with a systematic way of constructing spaces, and allow us to classify spaces based on their symmetries.

3 Flow equation

This section derives the central flow equation governing how assignment values propagate through geometric space: $\frac{da}{ds} = \mu \cos\theta + a\lambda \sin\theta$, where θ represents the angle of movement. The flow equation can be reformulated in various ways, including its coordinate-free form ($||\nabla a|| = \sqrt{\mu^2 + a^2\lambda^2}$), which is an Eikonal equation equivalent to a special Hamilton-Jacobi equation.

The section demonstrates that the flow equation is consistent with discrete generating processes and can be expressed in contour-gradient form. It establishes a relationship between arithmetic torsion and geometric area ($d\tau = \mu\lambda dudv$), linking non-commutativity in arithmetic to measurable geometric properties. The flow equation is shown to have a geometric propagation interpretation, where assignment values correspond to wavefront evolution in hyperbolic space. The section concludes with a discussion of the existence of metrics that make functions satisfy the flow equation, presenting a local morphing process while noting that the global case remains unsettled.

3.1 Derivation of the flow equation

Consider an infinitesimal generating process on a Riemannian surface M using two generators: one for an additional action μ and the other for a multiplicative action e^{λ}. These two generators are perpendicular. This generation process produces an assignment $A : M \to R$ over the surface.

For any point with an assignment a_0, if we consider a movement of distance ϵ in a direction with angle θ over a time period of δ, we can establish the following:

$$a_\delta = (a_0 + \mu\epsilon \cos\theta)e^{\lambda\epsilon \sin\theta}$$

or

$$a_\delta = a_0 e^{\lambda\epsilon \sin\theta} + \mu\epsilon \cos\theta$$

Both formula can be simplified to the same result:

$$a_\delta = a_0 + \epsilon(a_0\lambda \sin\theta + \mu \cos\theta)$$

Then, we have the following equation:

$$\frac{1}{\delta}(a_\delta - a_0) = \frac{\epsilon}{\delta}(\mu \cos\theta + x_0\lambda \sin\theta)$$

When both δ and ϵ are towards zero, we get da/dt, and hence

$$\frac{da}{dt} = u(\mu \cos\theta + a\lambda \sin\theta)$$

Or, we can change it to another form

$$\frac{da}{ds} = \mu \cos\theta + a\lambda \sin\theta \tag{30}$$

We name this equation (30) as the flow equation.

The left side of this equation is governed by the distance structure, while the right side is governed by the angle structure. So that the isometrics of the surface keep the flow equation (30).

We can also get a direct formal solution of the flow equation (30)(details in Appendix A).

$$a = (a_0 + \frac{\mu}{\lambda} \cot\theta)e^{\lambda s \sin\theta} - \frac{\mu}{\lambda} \cot\theta \tag{31}$$

3.2 Discrete generating

In section 2.7, we have discussed a discrete generating process. Since flow equation governs an infinitesimal generating process, we will show the above discrete generating process can be emerged from the solution of the flow equation (31) naturally. We expand the formula by the Taylor series:

$$a = a_0 e^{\lambda s \sin\theta} + \frac{\mu}{\lambda}[1 + \lambda s \sin\theta + \frac{1}{2!}(\lambda s \sin\theta)^2 + \frac{1}{3!}(\lambda s \sin\theta)^3 + \cdots - 1]\cot\theta$$

Change the formula slightly:

$$a = a_0 e^{\lambda s \sin\theta} + \mu s \cos\theta + \frac{\mu}{\lambda}\sin\theta\cos\theta(\frac{\lambda^2 s^2}{2!} + \frac{\lambda^3 s^3}{3!}\sin\theta + \frac{\lambda^4 s^4}{4!}\sin^2\theta + \cdots)$$

By the formula of double angle, we have

$$a = a_0 e^{\lambda s \sin\theta} + \mu s \cos\theta + \frac{\mu}{2\lambda}\sin 2\theta(\frac{\lambda^2 s^2}{2!} + \frac{\lambda^3 s^3}{3!}\sin\theta + \frac{\lambda^4 s^4}{4!}\sin^2\theta + \cdots)$$

We denote

$$\Psi(s) = \frac{1}{2!} + \frac{\lambda s}{3!}\sin\theta + \frac{\lambda^2 s^2}{4!}\sin^2\theta + \cdots \tag{32}$$

Then we have

$$a = a_0 e^{\lambda s \sin\theta} + \mu s \cos\theta + \frac{\mu\lambda}{2} s^2 \Psi(s) \sin 2\theta \tag{33}$$

This formula gives the discrete generating process, when $\theta = \frac{k\pi}{2}, k = 0, 1, 2, 3\cdots, s = 0, 1, 2, 3\cdots$, we have

$$a = a_0 e^{\lambda s \sin\theta} + \mu s \cos\theta \tag{34}$$

Especially, we have the following four cases:

- $\theta = 0$: $a_s = a_0 + \mu s$
- $\theta = \frac{\pi}{2}$: $a_s = a_0 e^{\lambda s}$
- $\theta = \pi$: $a_s = a_0 - \mu s$
- $\theta = \frac{3\pi}{2}$: $a_s = a_0 e^{-\lambda s}$

This result is straightforward, but it demonstrates that the infinitesimal generating process is consistent with the discrete generating process. And this expands our toolset, enabling us to explore the interplay between discrete and infinitesimal generating processes.

3.3 The contour-gradient form of flow equation

It is easy to derive the contour equation in the local coordinate

$$\mu\cos\theta_c + a\lambda\sin\theta_c = 0 \tag{35}$$

then we have

$$\theta_c = -\arctan\frac{\mu}{a\lambda} \tag{36}$$

the contour and the gradient are perpendicular to each other

$$\theta_g = \pm\frac{\pi}{2} - \arctan\frac{\mu}{a\lambda} \tag{37}$$

then along θ_g we have

$$\frac{da}{ds} = \mu \cos(\pm\frac{\pi}{2} - \arctan\frac{\mu}{a\lambda}) + a\lambda \sin(\pm\frac{\pi}{2} - \arctan\frac{\mu}{a\lambda}) \tag{38}$$

$$\frac{da}{ds} = \pm\sqrt{\mu^2 + \lambda^2 a^2} \tag{39}$$

By introducing the right-hand rotation angle ϕ along the gradient direction, we can establish a local polar coordinate system based on the gradient and contour lines. Then the growth rate of a along the angle ϕ is

$$\frac{da}{ds} = \mu \cos(\frac{\pi}{2} - \arctan\frac{\mu}{a\lambda} + \phi) + a\lambda \sin(\frac{\pi}{2} - \arctan\frac{\mu}{a\lambda} + \phi) \tag{40}$$

And the simplified equation is

$$\frac{da}{ds} = \sqrt{\mu^2 + a^2\lambda^2} \cos\phi \tag{41}$$

or

$$\frac{da_\phi}{ds_\phi} = \sqrt{\mu^2 + a^2\lambda^2} \cos\phi \tag{42}$$

if we want to emphasize the path is along the angle ϕ.

The equation (41) is the flow equation in the contour-gradient coordinate system.

Equation (41) is solvable, and we get the relation between a and s:

$$\tanh(\lambda s \cos\phi - c) = \frac{\lambda a}{\sqrt{\mu^2 + \lambda^2 a^2}} \tag{43}$$

we can further simplify the equation to

$$a = \pm\frac{\mu}{\lambda} \sinh(\lambda s \cos\phi - c) \tag{44}$$

Under the initial condition $a = a_0$ when $s = 0$, we can get the following equation:

$$a = \frac{\mu}{\lambda} \sinh(\lambda s \cos\phi + \operatorname{arcsinh}\frac{a_0\lambda}{\mu}) \tag{45}$$

or

$$a = -\frac{\mu}{\lambda} \sinh(\lambda s \cos\phi - \operatorname{arcsinh}\frac{a_0\lambda}{\mu}) \tag{46}$$

In this coordinate system, the additional line and the multiplicative line are:

$$\phi = \arccos\frac{\mu}{\sqrt{\mu^2 + a^2\lambda^2}} \tag{47}$$

$$\phi = \arcsin\frac{\mu}{\sqrt{\mu^2 + a^2\lambda^2}} \tag{48}$$

3.4 Arithmetic coordinate and area formula

We begin our exploration by examining the flow equation (30) within the framework of a local polar coordinate system:

$$\frac{da}{ds} = \mu \cos\theta + a\lambda \sin\theta \tag{49}$$

In an effort to re-contextualize this equation, we set $du = \cos\theta ds$ and $dv = \sin\theta ds$, where du and dv are perpendicular infinitesimal movements. We can use these movements to construct a local Descartes coordinate system, and the first fundamental form of this system is:

$$ds^2 = A^2 du^2 + B^2 dv^2 \tag{50}$$

Thereby this enables us to express the flow equation in a different light:

$$da = \mu du + a\lambda dv \tag{51}$$

Our attention now turns to the concept of arithmetic torsion, particularly at an infinitesimal level. Delving into the interplay between two infinitesimal generating processes, we observe that:

$$d\tau = (a_0 + \mu du)e^{\lambda dv} - (a_0 e^{\lambda dv} + \mu du) \tag{52}$$

From this relationship, we deduce:

$$d\tau = \mu du(e^{\lambda dv} - 1) \tag{53}$$

This leads us to an area formula, capturing the essence of this interaction:

$$d\tau = \mu\lambda du dv \tag{54}$$

and because the area element have a form

$$dS = |AB| du dv \tag{55}$$

Then we have

$$\frac{d\tau}{\mu\lambda} = \frac{dS}{|AB|} \tag{56}$$

This formula is compelling as it establishes a link between area elements and arithmetic torsion. Such formulations find parallels in the realms of classic analysis and differential geometry. For instance, they resonate with concepts akin to Stokes' theorem or the Gauss-Bonnet theorem. We intend to expand upon this formula in the ensuing section**??**, aiming to forge a connection with curvature and delve into the intricacies of the Gauss-Bonnet theorem.

It's noteworthy to emphasize the distinctiveness of the local Descartes coordinate system. This system, by integrating the assignment, lays the foundation for a theoretical framework. We refer to this as the *arithmetic coordinate system*, given its unique properties and alignment with arithmetic principles.

3.5 The coordinate-free form of flow equation

From the contour-gradient form of the flow equation (41), we can derive a coordinate-free form of the flow equation. Let's consider the direction of $\phi = 0$ in the contour-gradient coordinate system, and we have

$$\frac{da}{ds}|_{\phi=0} = \sqrt{\mu^2 + a^2\lambda^2}\cos 0$$

Notice the gradient of a is not dependent on the coordinate system, and we have the coordinate-free form of the flow equation:

$$||\nabla a|| = \sqrt{\mu^2 + a^2\lambda^2} \tag{57}$$

It should be noted that the coordinate-free form of the flow equation (57) is an Eikonal equation, and can be viewed as a special Hamilton–Jacobi equation

$$H(x, a, \nabla a) = 0$$

where the Hamiltonian is

$$H(x, a, p) = ||p|| - \sqrt{\mu^2 + a^2\lambda^2} \tag{58}$$

3.6 Propagation method

Starting from Equations (45) and (46), we can derive a geometric propagation interpretation of the flow equation. By rewriting them in a unified form, we obtain

$$a = \pm\frac{\mu}{\lambda}\sinh\bigl(\lambda s\cos\phi + \operatorname{arcsinh}\bigl(\tfrac{a_0\lambda}{\mu}\bigr)\bigr). \tag{59}$$

If we set $a_0 = 0$ and choose the gradient direction $\phi = 0$, the expression simplifies to

$$a = \pm\frac{\mu}{\lambda}\sinh(\lambda s). \tag{60}$$

Comparing Equation (60) with the circumference of a circle of radius s in a hyperbolic space with curvature k:

$$C = \frac{2\pi}{\sqrt{-k}}\sinh\bigl(\sqrt{-k}s\bigr), \tag{61}$$

we see that the assignment a can be interpreted as a propagating circumference along an expanding circle of radius s. With the centroids forming a zero line, each point on this line generates a wavefront described by concentric circles. Hence, the propagation of a corresponds to the wavefront evolution in this geometric sense. A more detailed and rigorous geometric interpretation of the flow equation will be provided in Section 4.

3.7 Flow and function

In this section, we aim to present novel insight into functions. Namely, the treatment of functions as flows will be discussed.

Definition 3.1. *Given a function k on the real domain R, we can introduce a mapping l on the arithmetic expression space H such that the following diagram commutes.*

$$\begin{tikzcd}
H \arrow[r, "l"] \arrow[d, "\nu"'] & H \arrow[d, "\nu"] \\
R \arrow[r, "k"'] & R
\end{tikzcd}$$

where ν is the evaluation function of the expression. Then we call the mapping l is the promotion of the function k, or function k is the projection of the mapping l.

Giving an arithmetic expression space as definition at the beginning of the section2.2, we will show examples of flows as functions in the following Section 4.

3.8 The existence theorem

There are two existence problems related to the flow equation (30). The first existence problem concerns the existence of a function a on a Riemannian surface S that satisfies the flow equation (30). The second existence problem concerns the existence of a metric g on a Riemannian surface S that makes a function a satisfy the flow equation (30).

We can proof there is a local morphing process over metric g to make a function a satisfy the flow equation (30) locally. But the global morphing process is more complicated, and we need to consider the global structure of the surface S, which is still not settled.

Lemma 3.1. *(By Le Zhang) Given an oriented compact Riemannian surface S, and a smooth function a over S, there exists a metric g on S that makes a satisfying the flow equation* (30).

Proof. Local perspective:

Consider a point p on the surface S, and there is a neighborhood U around p. In this area, we can find a local isothermal coordinate system in which the metric takes the form:

$$ds^2 = e^{2\rho}(du^2 + dv^2),$$

where u and v are the coordinates of U, and ρ is a function of u, v in U. The gradient of a in this local isothermal coordinate system is expressed as:

$$\nabla a = \frac{\partial a}{\partial u} du + \frac{\partial a}{\partial v} dv.$$

Using the definition of the directional derivative, we obtain:

$$\frac{da_\psi}{ds_\psi} = ||\nabla a|| \cos \psi,$$

where $||\nabla a||$ is the norm of ∇a, and ψ is the angle between ∇a and the direction of movement.

Now, considering the flow equation 42 in the gradient-contour coordinate system, we have:

$$\frac{da_\phi}{ds_\phi} = \sqrt{\mu^2 + a^2\lambda^2} \cos \phi.$$

Note that $||\nabla a||$ is fixed for the given function a and the local coordinate system, and $\sqrt{\mu^2 + a^2\lambda^2}$ is also fixed for the given function a. We can scale $e^{2\rho}$ with a linear factor α to make $||\nabla a||$ match the fixed value of $\sqrt{\mu^2 + a^2\lambda^2}$, thus we have a morphing process controlled by α that

$$\begin{aligned} ds^2 &= \alpha e^{2\rho}(du^2 + dv^2) && (62) \\ &= e^{2\rho + \ln \alpha}(du^2 + dv^2). && (63) \end{aligned}$$

Under the morphing ratio α, we have:

$$||\nabla_\alpha a|| = \alpha^{-1} ||\nabla a||, \tag{64}$$

and when α is set to the value of:

$$||\nabla_\alpha a|| = \sqrt{\mu^2 + a^2\lambda^2},$$

the flow equation (30) is satisfied in the local coordinate system.

The morphing ratio α is calculated as follows:

$$\alpha = \frac{||\nabla a||}{\sqrt{\mu^2 + a^2\lambda^2}}. \tag{65}$$

□

When we consider the broader scope of the surface S, it's possible to extend the morphing process to every point, ensuring that the flow equation (30) is satisfied on a global scale. However, this expansion necessitates a harmonious integration of the morphing process across neighboring locales. Specifically, this means that the morphing should not only preserve the circles centered at point p within its immediate local chart but also maintain the integrity of these circles within the adjacent charts of point p. In essence, the morphing process must be seamlessly coordinated across the various local regions to achieve a unified global transformation. How to achieve this harmonious integration remains an open question, and further exploration is needed to address this challenge.

4 The first kind space $\mathfrak{E}_1$

This section introduces the first kind arithmetic expression space ($\mathfrak{E}_1$), providing a geometric framework for analyzing arithmetic expressions. The space is constructed on the upper half-plane with a hyperbolic metric: $ds^2 = \frac{1}{y^2}(dx^2+dy^2)$, where the assignment function $a = -\frac{x}{y}$ satisfies the flow equation and serves as an eigenfunction of the Laplacian with eigenvalue 2.

Two equivalent examples of $\mathfrak{E}_1$ are presented: the upper half-plane model and a horocycle-based coordinate system, connected through Möbius transformation. The section explores geometric propagation mechanisms, showing how the assignment value propagates like expanding concentric circles in hyperbolic space. It examines grid structures in $\mathfrak{E}_1$, revealing dual grids reflecting the geometric structure of the Baumslag-Solitar group, and demonstrates how arithmetic torsion corresponds precisely to hyperbolic areas enclosed between evaluation paths. The section concludes by introducing tube structures, which extend $\mathfrak{E}_1$ to parameterized families, enabling analysis of how expressions evolve across parameter variations.

4.1 Foundational exemplars

We present two analytically equivalent examples that belong to the class of spaces designated as the first kind arithmetic expression space $\mathfrak{E}_1$.

4.1.1 Example 1: Upper Half Plane Model

Consider the upper half plane $\mathcal{H} : (x, y) \mid y > 0$ equipped with the following inner product and metric tensor:

$$\mathbf{a} \cdot \mathbf{b} = \begin{bmatrix} a_x & a_y \end{bmatrix} \begin{bmatrix} \frac{1}{y^2} & 0 \\ 0 & \frac{1}{y^2} \end{bmatrix} \begin{bmatrix} b_x \\ b_y \end{bmatrix}$$

$$ds^2 = \frac{1}{y^2}(dx^2 + dy^2)$$

On this manifold, we define an assignment field a as follows:

$$a = -\frac{x}{y} \tag{66}$$

Theorem 4.1. *The assignment a defined by formula* (66) *satisfies the flow equation* (30).

Proof. We initiate with the differential of the assignment:

$$da = d\left(-\frac{x}{y}\right) = \frac{xdy - ydx}{y^2} = -\frac{dx + ady}{y}$$

The differential of arc length is given by:

$$ds = \frac{\sqrt{dx^2 + dy^2}}{y}$$

Therefore:

$$\frac{da}{ds} = -\frac{dx + ady}{y} \cdot \frac{y}{\sqrt{dx^2 + dy^2}} = -\frac{dx + ady}{\sqrt{dx^2 + dy^2}}$$

In the local coordinate system determined by $(-1, 0)$ and $(0, -1)$ under the right-hand rule, we have:

$$\cos\theta = \frac{-dx}{\sqrt{dx^2 + dy^2}} \quad \text{and} \quad \sin\theta = \frac{-dy}{\sqrt{dx^2 + dy^2}}$$

Substituting these values:

$$\frac{da}{ds} = \cos\theta + a\sin\theta$$

This precisely corresponds to the flow equation (30) with $\mu = 1$ and $\lambda = 1$. □

We can verify that a constitutes an eigenfunction of the Laplacian operator:

$$\Delta a = -y^2 \left(\frac{\partial^2 a}{\partial x^2} + \frac{\partial^2 a}{\partial y^2} \right) = y^2 \left(\frac{\partial}{\partial y} \left(\frac{\partial}{\partial y} \frac{x}{y} \right) \right) = 2a$$

4.1.2 Example 2: Horocycle-Based Coordinate System

For our second exemplar, we introduce a horocycle-based coordinate system for hyperbolic surfaces. This global coordinate system comprises two orthogonal families of curves: horocycles sharing the same ideal point, and geodesics perpendicular to these horocycles.

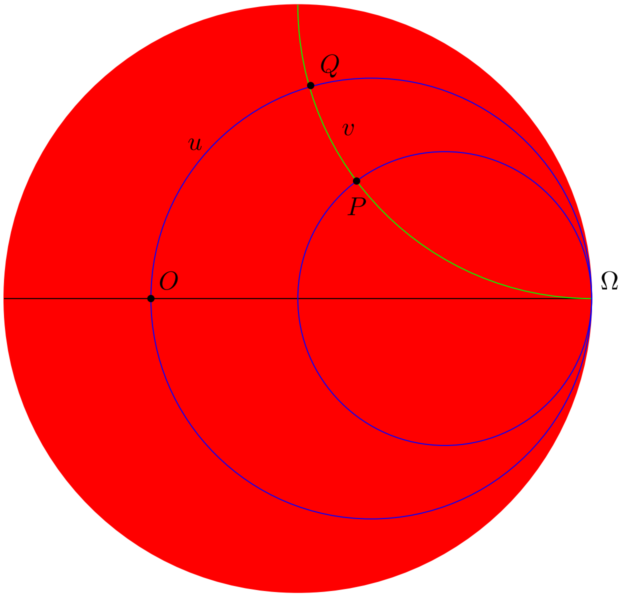

Figure 7: A horocycle-based coordinate system on the Poincaré disc. Blue curves represent horocycles tangent at ideal point Ω, green lines depict perpendicular geodesics.

On the Poincaré disc $\mathcal{P}$, the coordinates of a point P are denoted by (u, v), where:

- u represents the signed length of OQ
- v represents the signed length of QP
- The sign conventions adhere to the right-hand rule and orientation relative to the ideal point Ω

We equip this coordinate system with the inner product:

$$\mathbf{a} \cdot \mathbf{b} = \begin{bmatrix} a_u & a_v \end{bmatrix} \begin{bmatrix} e^{-2v} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} b_u \\ b_v \end{bmatrix}$$

And the corresponding metric tensor:

$$ds^2 = e^{-2v} du^2 + dv^2$$

The Laplacian operator in this coordinate system is expressed as:

$$\Delta = e^{2v} \frac{\partial^2}{\partial u^2} + \frac{\partial^2}{\partial v^2} - \frac{\partial}{\partial v}$$

In this coordinate framework, we define an assignment:

$$a = ue^{-v} \tag{67}$$

Theorem 4.2. *The assignment a defined by formula* (67) *satisfies the flow equation* (30).

Proof. We establish this result by demonstrating that examples 1 and 2 are equivalent through a Möbius transformation. Consider the complex representation of the upper half plane:

$$z = x + yi$$

The Möbius transformation mapping the upper half plane to the Poincaré disc is given by:

$$z \mapsto \frac{z - i}{z + i}$$

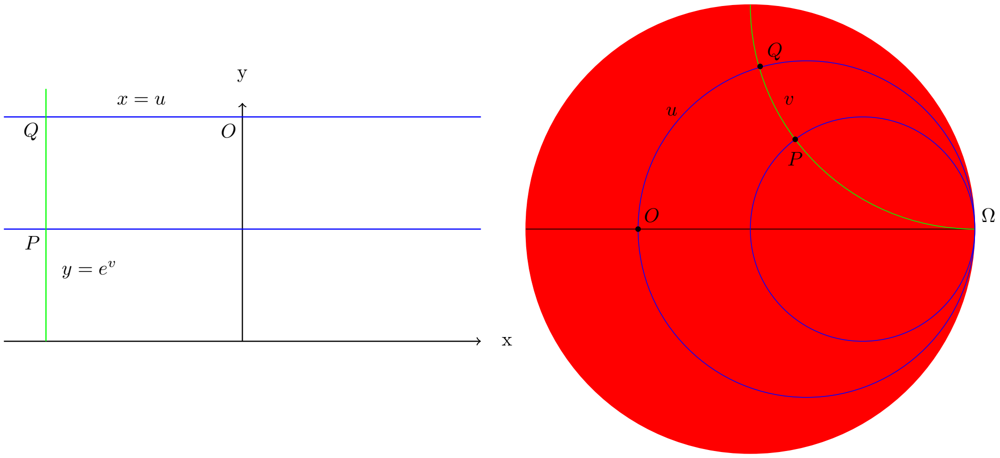

Figure 8: Mapping between the upper half plane and Poincaré disc models

This conformal transformation maps horizontal lines in $\mathcal{H}$ to horocycles sharing the ideal point $\Omega = 1$ in $\mathcal{P}$, and vertical geodesics in $\mathcal{H}$ to perpendicular geodesics in $\mathcal{P}$.

Expressed in the target coordinate system, this transformation yields:

$$\begin{cases} x = u \\ y = e^v \end{cases}$$

Substituting into the assignment from Example 1:

$$a = -\frac{x}{y} = -\frac{u}{e^v} = -ue^{-v}$$

Since the Möbius transformation is conformal and preserves the flow equation, and accounting for the orientation change, we obtain $a = ue^{-v}$ satisfying the flow equation. □

As in Example 1, we can verify that a constitutes an eigenfunction of the Laplacian:

$$\Delta a = e^{2v} \frac{\partial^2 (ue^{-v})}{\partial u^2} + \frac{\partial^2 (ue^{-v})}{\partial v^2} - \frac{\partial (ue^{-v})}{\partial v} = 2a$$

These two examples, emerging from the same geometric foundation but expressed in different coordinate systems, demonstrate the fundamental properties of the first kind arithmetic expression space.

4.2 Theoretical framework of $\mathfrak{E}_1$ space

Building upon the foundational exemplars, we now establish a comprehensive theoretical framework for the first kind arithmetic expression space $\mathfrak{E}_1$.

Consider the upper half plane $\mathcal{B}$:

$$\{\mathcal{B} : (x, y) | y > 0\}$$

equipped with an inner product and metric tensor parameterized by constants μ and λ:

$$\mathbf{a}\cdot\mathbf{b} = [a_x \quad a_y] \begin{bmatrix} \frac{1}{\mu^2 y^2} & 0 \\ 0 & \frac{1}{\lambda^2 y^2} \end{bmatrix} \begin{bmatrix} b_x \\ b_y \end{bmatrix}$$

$$ds^2 = \frac{1}{y^2}\left(\frac{dx^2}{\mu^2} + \frac{dy^2}{\lambda^2}\right)$$

The assignment function in this generalized framework maintains the form:

$$a = -\frac{x}{y} \tag{68}$$

This defines the first kind arithmetic expression space $\mathfrak{E}_1$, characterized by the following theorem:

Theorem 4.3. *The assignment a given by* (68) *satisfies the flow equation* (30) *with parameters μ and λ, independent of the specific values of these generators.*

Proof. The differential of the assignment is given by:

$$da = d\left(-\frac{x}{y}\right) = \frac{xdy - ydx}{y^2} = -\frac{dx + ady}{y}$$

The differential of arc length is expressed as:

$$ds = \frac{1}{y}\sqrt{\frac{dx^2}{\mu^2} + \frac{dy^2}{\lambda^2}}$$

Therefore:

$$\frac{da}{ds} = -\frac{dx + ady}{y} \cdot \frac{y}{\sqrt{\frac{dx^2}{\mu^2} + \frac{dy^2}{\lambda^2}}} = -\frac{dx + ady}{\sqrt{\frac{dx^2}{\mu^2} + \frac{dy^2}{\lambda^2}}}$$

In the local coordinate system determined by $(-1, 0)$ and $(0, -1)$ according to the right-hand rule:

$$\cos\theta = \frac{-\frac{dx}{\mu}}{\sqrt{\frac{dx^2}{\mu^2} + \frac{dy^2}{\lambda^2}}} \quad \text{and} \quad \sin\theta = \frac{-\frac{dy}{\lambda}}{\sqrt{\frac{dx^2}{\mu^2} + \frac{dy^2}{\lambda^2}}}$$

Substituting these values:

$$\frac{da}{ds} = \mu\cos\theta + a\lambda\sin\theta$$

This precisely corresponds to the flow equation (30) with the given parameters μ and λ. □

The $\mathfrak{E}_1$ space is distinguished by its intrinsic connection to hyperbolic geometry and the property that the assignment function $a = -x/y$ constitutes an eigenfunction of the Laplacian operator with eigenvalue 2. This space provides a natural geometric framework for analyzing arithmetic expressions, particularly those involving addition and multiplication operations.

4.3 Geometric propagation mechanisms

The flow equation (30) in the $\mathfrak{E}_1$ space, $da/ds = \mu\cos\theta + a\lambda\sin\theta$, provides a dynamic interpretation of arithmetic expression evaluation as a propagation process. This perspective extends the propagation method discussed in Section 3.6.

We can visualize this process using the concept of wavefronts. Considering the locus where the assignment $a = 0$ (the y-axis, $x = 0$) as the initial source or baseline, the "information" or "value" propagates outwards into the upper half-plane $\mathcal{H}$. The flow equation governs how the assignment value a changes as this propagation occurs.

Points with the same assignment value form equipotential lines, or contours, defined by $a = -x/y = a_0$ (constant). In the (x, y) coordinate chart of $\mathcal{H}$, these contours are rays emanating from the origin, described by the equation $x = -a_0 y$.

A key insight arises when observing how these equipotential lines behave as the propagation proceeds. Propagation, fundamentally occurring along directions related to the gradient of a (which are orthogonal to the contours), leads to an increase in the magnitude $|a|$ as points move further from the initial $a = 0$ line.

Now, consider the orientation of the contour ray $x = -a_0 y$. Its slope in the (x, y) plane is $dy/dx = -1/a_0$.

- When $|a_0|$ is very small (i.e., close to the initial $a = 0$ state on the y-axis), the slope $-1/a_0$ is very large in magnitude, meaning the ray is nearly vertical, aligned closely with the y-axis.

- As the wavefront propagates outwards, the magnitude $|a_0|$ increases. Consequently, the magnitude of the slope $|-1/a_0|$ decreases, approaching zero as $|a_0| \to \infty$. This means the ray becomes increasingly horizontal, aligning more closely with the x-axis.

Therefore, the geometric propagation driven by the flow equation manifests as a dynamic **sweeping or change in orientation of the equipotential rays** $a = a_0$. As the magnitude $|a|$ increases due to propagation away from the zero line, the corresponding ray appears to **rotate** from a near-vertical orientation (along the y-axis for $a = 0$) towards a near-horizontal orientation (along the positive x-axis if $a \to -\infty$, or the negative x-axis if $a \to +\infty$).

This increase in magnitude $|a|$ with the propagation distance s along the gradient is quantified by the relationship $|a| = (\mu/\lambda)\sinh(\lambda s)$ (derived from (60) for propagation from $a_0 = 0$), confirming that further propagation indeed leads to larger $|a|$ values and thus more horizontally oriented contour lines.

This dynamic picture of equipotential rays sweeping from vertical towards horizontal alignment provides a core geometric interpretation of the propagation mechanism inherent in the flow equation within the $\mathfrak{E}_1$ space.

4.4 Grid structures

A significant geometric characteristic of the first kind arithmetic expression space is the presence of two distinct yet interrelated grid structures, each encoding addition and multiplication operations in different ways. These dual grids reflect the geometric structure of the Baumslag–Solitar group, whose Cayley graph exhibits an anisotropic, hierarchical lattice with a natural correspondence to mixed additive-multiplicative expressions.

Both grid structures are constructed within the upper half-plane model. The first grid is rectilinear, consisting of horizontal lines encoding addition operations and vertical lines encoding multiplication operations. Specifically, this grid is constructed through iterative applications of these two operations to generate a lattice in the (x, y) coordinates. Each horizontal displacement from (x, y) to $(x + 1, y)$ represents an addition, and each vertical displacement from (x, y) to $(x, 2y)$ represents a multiplication by 2. The grid vertices correspond to values of expressions constructed from repeated applications of addition and multiplication operations, typically originating from a rational base point, often designated as 1.

Notably, in this first grid, the scalar field a remains invariant under variations of the metric parameters μ and λ. That is, while the geometric properties of the space (lengths and angles) vary with these parameters, the locus where $a = 0$—specifically, the vertical line $x = 0$—remains structurally invariant and unaffected by changes in μ and λ.

The second grid emerges through a conformal transformation, specifically the Möbius transformation acting within the upper half-plane. This transformation maps horizontal lines to semicircles centered on the real axis and vertical rays to orthogonal semicircles. Under this transformation, the roles of addition and multiplication are effectively interchanged. The addition-multiplication structure of the first grid transforms into a new system of curved geodesics: the images of addition steps now follow curved trajectories around the origin, while the multiplicative steps exhibit contraction and inversion properties.

Although the visual structure of the second grid exhibits greater complexity, it retains a profound arithmetic coherence. The grid vertices continue to correspond to expressions generated through repeated applications of addition and multiplication operations, albeit composed under the transformed geometry. This grid demonstrates enhanced flexibility: under the conformal transformation, the images of zero lines such as $x = 0$ are no longer fixed but deform into dynamic arcs. Consequently, the second grid accommodates a richer family of zero structures, allowing for the possibility of curved, branching, or nested nodal sets that vary with μ, λ, or the choice of conformal framing.

Each of these grid structures can be interpreted as a geometric realization of a Cayley graph:

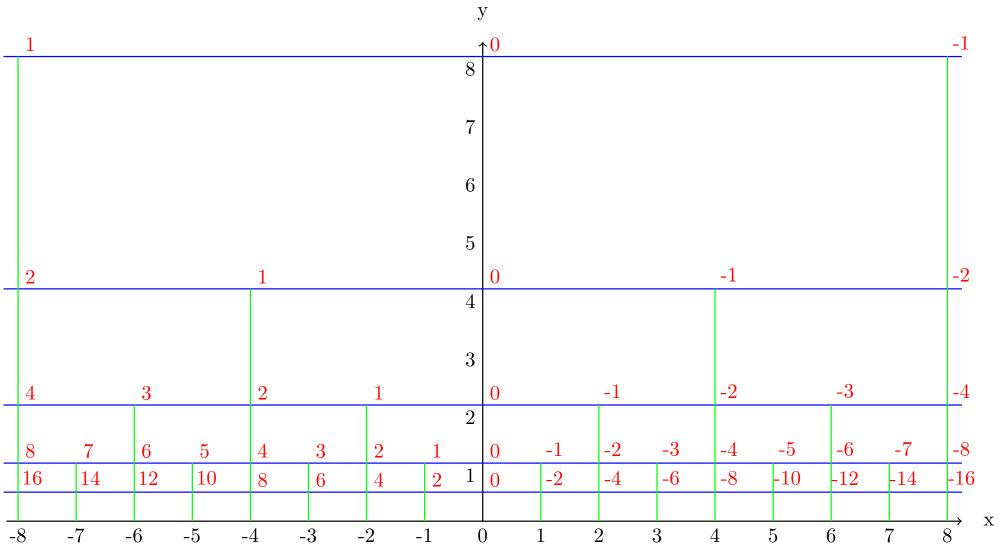

Figure 9: Rectilinear grid structure in the first kind arithmetic expression space $\mathfrak{E}_1$

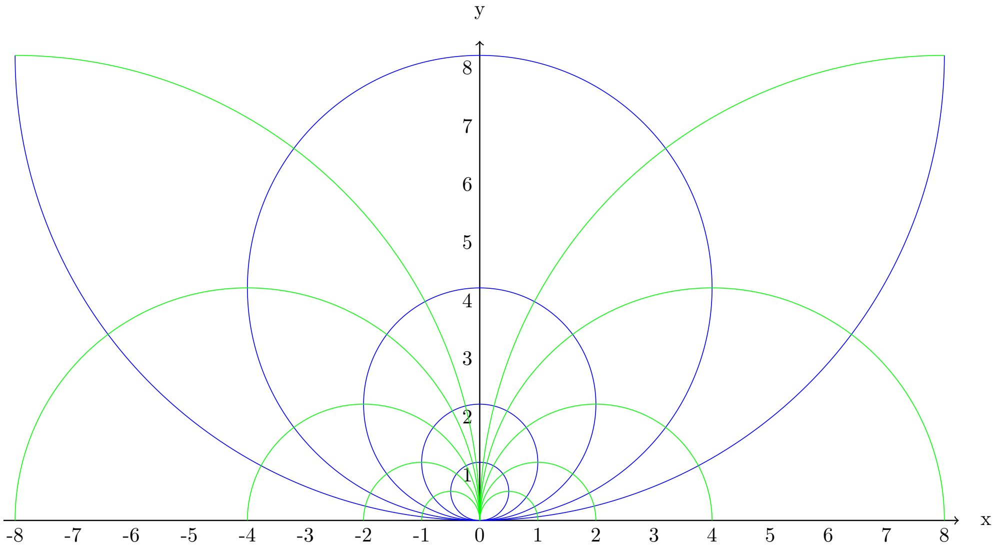

Figure 10: Transformed grid structure in the first kind arithmetic expression space $\mathfrak{E}_1$

- The rectilinear grid corresponds to the Cayley graph of the Baumslag–Solitar group $BS(1,2)$, where multiplication by 2 followed by addition by 1 follows the relation $b^{-1}ab = a^2$.
- The transformed (dual) grid corresponds to the Cayley graph of the dual Baumslag–Solitar group, where the roles of addition and multiplication are inverted.

Thus, the two grid systems exhibit duality not only in geometric terms but also in group-theoretic structure. The Möbius transformation, acting within the upper half-plane, connects these dual groups by exchanging generators and inverting flow directions, reflecting an intrinsic duality in the geometric composition of arithmetic operations.

The coexistence of these dual grid structures—one linear, one curved—linked via conformal symmetry, suggests a profound underlying symmetry in the geometry of arithmetic expressions. This duality reflects how different evaluation paths or expression embeddings can be interpreted as projections from a common, more complex geometric source. Understanding this symmetry may provide a pathway to expressing arithmetic relations via modular or automorphic structures, particularly in the context of recursive expressions and iterative identities, where Baumslag–Solitar-like behavior naturally emerges.

We conjecture that this grid duality corresponds to a deeper expression-theoretic equivalence, and that the Möbius transformation connecting them manifests an underlying arithmetic symmetry in expression geometry.

4.5 Torsion under scale transformation

The addition-multiplication grid introduced in Section 2.2 has a natural embedding in the arithmetic expression space $\mathfrak{E}_1$. This grid consists of two orthogonal families of curves:

1. **Addition curves** (blue lines): horizontal geodesics along which y remains constant, representing iterated additions.
2. **Multiplication curves** (green lines): vertical or logarithmically scaled geodesics where the ratio x/y remains constant, representing multiplicative transformations.

This grid structure facilitates the geometric analysis of *arithmetic torsion*—a quantity arising from the non-commutativity of certain additive and multiplicative expression sequences. Specifically, torsion quantifies the discrepancy between two seemingly equivalent but differently ordered expressions.

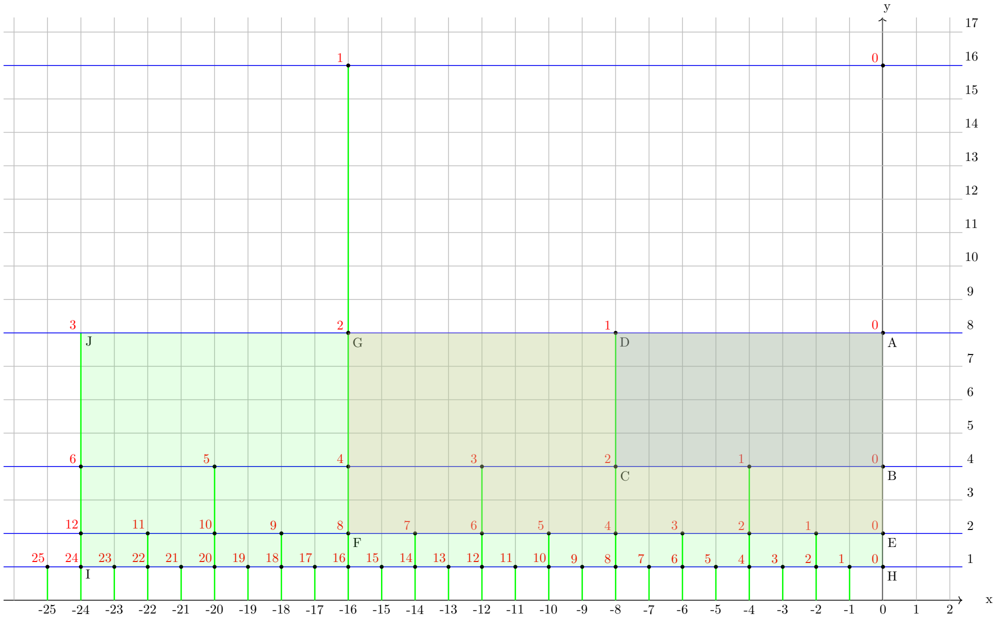

Figure 11: Illustration of the correspondence between hyperbolic area and arithmetic torsion

Figure 11 illustrates the relationship between the area enclosed by expression paths in the grid and the resulting torsion. Consider the following expression identity comparisons:

- One-step case:
$$x \times 2 + 1 - (x + 1) \times 2 = -1 \tag{69}$$
- Two-step case:
$$x \times 4 + 2 - (x + 2) \times 4 = -6 \tag{70}$$
- Three-step case:
$$x \times 8 + 3 - (x + 3) \times 8 = -21 \tag{71}$$

These differences correspond precisely to the hyperbolic areas enclosed between alternative evaluation paths within these specific grid examples:

- The region $ABCD$ encompasses 1 unit cell.
- The region $AEFG$ encompasses 6 unit cells.
- The region $AHIJ$ encompasses 21 unit cells.

These examples compellingly suggest that arithmetic torsion accumulates in proportion to the area enclosed by the grid paths, indicating a potential connection between algebraic non-commutativity and geometric surface area.

Further supporting this connection, we established a differential formulation:

$$d\tau = \mu\lambda \, du \, dv \tag{72}$$

where $d\tau$ represents the infinitesimal arithmetic torsion, and $du \, dv$ denotes the area element in the (u, v) coordinate system adapted to the grid. This equation provides the precise microscopic law linking infinitesimal torsion to the infinitesimal area element.

However, while we possess both macroscopic observations suggesting a torsion-area relationship (from the scaled grid examples) and the exact microscopic differential law (72), the explicit integral theorem rigorously bridging these scales is currently underdeveloped. Formulating how the microscopic torsion density $d\tau$ integrates over a finite region to yield the total accumulated torsion—potentially involving boundary terms in analogy with Stokes' theorem, or relating the total torsion to curvature and topology similarly to the Gauss-Bonnet theorem—remains a key objective for future work.

The analogy with curvature in differential geometry therefore becomes particularly pertinent: just as Gaussian curvature encodes deviation from flatness, arithmetic torsion quantifies deviation from commutativity in arithmetic flow. In this sense, torsion constitutes a measure of the operational significance of evaluation order.

The $\mathfrak{E}_1$ space thus provides a mathematical framework where algebraic non-commutativity manifests as measurable geometric distortion—establishing a novel interpretation of arithmetic structure as a form of discrete curvature. This opens avenues for investigating further geometric invariants such as torsion density, torsion-induced flow bifurcation, and **realizing** a Gauss–Bonnet-type integral identity for arithmetic surfaces.

4.6 Tube structure

In preceding sections, we introduced the first kind arithmetic expression space $\mathfrak{E}_1$ as a geometric realization of arithmetic flow under **fixed** generator parameters μ and λ. However, more complex structures emerge when we consider the entire family of spaces indexed by the parameter λ (or potentially both μ and λ) and analyze how expression behavior evolves across this family. This naturally leads to the concept of a *tube structure*.

4.6.1 From Slices to Parameterized Families

Each individual $\mathfrak{E}_1$ space, denoted $\mathfrak{E}_1^{(\lambda)}$, can be conceptualized as a single *slice* or *fiber* (in the sense of fiber bundles) within the family of expression spaces indexed by the parameter λ. Within each slice, the evaluation of arithmetic expressions is realized through traversal along (geodesic) paths, the result is governed by the scalar field a, and the flow is determined by the metric tensor corresponding to that slice.

Consider a fixed algebraic structure—for instance, an alternating path (with a fixed internal multiplier) corresponding to a polynomial $P(x)$—and examine how its evaluation result $P(e^\lambda)$ evolves as the tube structure parameter λ varies. For each value of λ, the evaluation $P(e^\lambda)$ corresponds to a point (or more accurately, the assignment value a at that point) within the λ-slice $\mathfrak{E}_1^{(\lambda)}$. As λ varies continuously, these points trace out a continuous trajectory through the family of spaces. We refer to such a trajectory generated by P as a *section* or a λ*-trajectory*. The collection of all slices corresponding to the allowed λ values, along with these structures upon them, together form a new, higher-dimensional entity: the tube structure.

4.6.2 Tube Structure as Total Space

We define a **tube structure** $\mathcal{T}$ as the *total space* formed by the family of $\mathfrak{E}_1$ spaces indexed by a continuous parameter λ (typically $\lambda > 0$), which can be formally written as the disjoint union:

$$\mathcal{T} = \bigsqcup_{\lambda > 0} \mathfrak{E}_1^{(\lambda)} \tag{73}$$

This total space needs to be endowed with an appropriate topology (and possibly a differential or fiber bundle structure) to support coherent analysis along the λ-direction.

In this structure:

- The *base space* is the parameter domain Λ for λ (e.g., $\mathbb{R}^+$).
- The *fiber* over each point λ in the base space is the geometric expression space $\mathfrak{E}_1^{(\lambda)}$.
- Fixed algebraic expression structures (especially those corresponding to polynomials $P(x)$, via the evaluation $P(e^\lambda)$) trace *canonical sections* or λ-trajectories through $\mathcal{T}$. These sections connect the fibers for different λ.

4.6.3 Zero Loci and Nodal Evolution

A primary motivation for studying tube structures is to investigate how *zero loci*—the sets of points where an expression evaluates to zero ($a = 0$)—evolve with the parameter λ.

- **In the Tube Structure $\mathcal{T}_1$ based on** $\mathfrak{E}_1$: For the $\mathfrak{E}_1$ space ($a = -x/y$) that we have discussed in detail, the zero locus within **each slice** $\mathfrak{E}_1^{(\lambda)}$ is always the **same simple** line: the y-axis ($x = 0$). Consequently, in the tube structure $\mathcal{T}_1 = \bigsqcup \mathfrak{E}_1^{(\lambda)}$, the overall zero locus is the trivial hyperplane $x = 0$.
- **Outlook for Non-Trivial Spaces**: However, as our research suggests, the simplicity of the zero locus in $\mathfrak{E}_1$ might limit its capacity to explain more complex phenomena (like those observed in knot theory examples). Therefore, there is strong motivation to seek and construct **"non-trivial" arithmetic expression spaces** $\mathfrak{E}_{NT}$, where a single slice $\mathfrak{E}_{NT}^{(\lambda)}$ might possess **multiple or morphologically more complex zero lines**. In the tube structures $\mathcal{T}_{NT}$ built from such non-trivial spaces, the zero locus itself could evolve with λ, potentially exhibiting various interesting phenomena, such as:
  - **Bifurcation**: New zero lines might emerge or merge with existing ones as λ varies.
  - **Branching**: The zero locus of certain expressions might exhibit multi-valued behavior along the λ-direction.
  - **Topology change**: The overall zero surface might develop handles (genus), singularities, or undergo other changes in its topological structure.

The analysis of such complex zero loci and their evolution (potentially within $\mathcal{T}_{NT}$) constitutes a core direction for studying expression dynamics, particularly when considering families of expressions or differential equations involving λ.

4.6.4 Investigative Approaches and Outlook

The formalism of tube structures opens up multiple avenues for research in arithmetic expression geometry:

- Examining the global properties of zero surfaces (in the general case where they might be non-trivial), such as their genus, regions of curvature concentration, and dependence on the parameter λ.
- Studying the geometric properties of sections γ_P corresponding to polynomials $P(e^\lambda)$ within the tube structure, and investigating whether imposing geometric continuity conditions leads to algebraic rigidity.
- Exploring the possibility of establishing flow equations across the λ-family, perhaps defining a notion of *connection* or *parallel transport* between different λ-slices.
- Defining *moduli spaces* of expression geometries as structured fiber bundles over parameter spaces.

Ultimately, tube structures provide a mathematical framework wherein the dynamics of arithmetic expressions can be analyzed analogously to field theory. In this analogy, expressions (or their underlying algebraic structures) act as structured sections, while quantities like arithmetic torsion, curvature, and zero loci serve as local or global invariants.

5 Arithmetic Gauss-Bonnet theorem in $\mathfrak{E}_1$ space

5.1 Discrete total arithmetic torsion calculations

The foundational operations defining threadlike arithmetic expressions, namely addition (denoted $\oplus_\mu$) and multiplication (denoted $\otimes_\lambda$, representing multiplication by e^λ), exhibit a fundamental non-commutativity. This non-commutativity is the origin of *arithmetic torsion*, a measure of the discrepancy arising from different sequential orderings of these operations. Our goal in this subsection is to develop a systematic, purely discrete method for quantifying the accumulated effect of this non-commutativity along a given threadlike expression path, and to relate this accumulated torsion directly to differences in the final numerical evaluation outcomes.

We begin by considering the simplest instance of non-commutativity: swapping a single additive step $\oplus_\mu$ and a single multiplicative step $\otimes_\lambda$. The difference in outcome, often referred to as the *local torsion* or the commutator effect (relative to a starting value x), is given by:

$$\tau_{\text{local}}(x;\mu,\lambda) = (x \oplus_\mu \otimes_\lambda) - (x \otimes_\lambda \oplus_\mu) = ((x+\mu)e^\lambda) - (xe^\lambda + \mu) = \mu(e^\lambda - 1). \tag{74}$$

This quantity $\mu(e^\lambda - 1)$ represents the elementary "error" or difference generated by exchanging the order of one adjacent pair of operations.

For longer expression paths involving multiple additions and multiplications, the total accumulated effect of non-commutativity is not merely the sum of local torsions from adjacent swaps. Crucially, the effect of swapping an operation $\oplus_{\mu_i}$ with a *subsequent* operation $\otimes_{\lambda_j}$ $(j > i)$ is influenced by all the intermediate multiplicative operations $\otimes_{\lambda_k}$ $(i < k < j)$. These intermediate multiplications effectively *amplify* the scale at which the non-commutativity between $\oplus_{\mu_i}$ and $\otimes_{\lambda_j}$ manifests. This amplification phenomenon is consistent with, and can be understood through, the analysis of perturbation propagation along such expression paths (as discussed in Section 2.6 and subsequent analyses).

To systematically quantify the total effect, we calculate the contribution of each additive step $\oplus_{\mu_i}$ to the overall discrepancy between two canonical evaluation orders. Let γ represent a specific evaluation order (e.g., a stepwise multiply-then-add sequence like $\alpha = (\otimes_{\lambda_1}\oplus_{\mu_1})\dots(\otimes_{\lambda_l}\oplus_{\mu_l})$) and let $\bar{\gamma}$ represent a reference order (e.g., the global add-then-multiply sequence corresponding to $(x + \sum \mu_k)\exp(\sum \lambda_k)$, or the global multiply-then-add sequence $x\exp(\sum \lambda_k) + \sum \mu_k$). The contribution τ_i of a single step $\oplus_{\mu_i}$ to the total difference $E(\gamma) - E(\bar{\gamma})$ arises from its interaction with all subsequent multiplicative steps $\otimes_{\lambda_k}$ for $k > i$. As derived through algebraic manipulation involving telescoping sums (detailed, for instance, in the discussions leading to this summary), this contribution is precisely:

$$\tau_i = \mu_i\left(\exp\left(\sum_{k=i+1}^{l} \lambda_k\right) - 1\right) = \mu_i(e^{\Lambda_{>i}} - 1), \tag{75}$$

where we define the sum of subsequent logarithmic scaling factors as $\Lambda_{>i} := \sum_{k=i+1}^{l} \lambda_k$. The term $e^{\Lambda_{>i}}$ represents the total scaling factor applied to μ_i due to all subsequent multiplications in one evaluation order (γ), while the -1 term reflects its baseline contribution (scaled by $e^0 = 1$) in the reference order ($\bar{\gamma}$) where it is added without prior scaling by subsequent λ_k.

The *total discrete accumulated torsion* τ_{sum} for the path α is then defined as the sum of the contributions from all additive steps:

$$\tau_{sum}(\alpha) = \sum_{i=1}^{l} \tau_i = \sum_{i=1}^{l} \mu_i(e^{\Lambda_{>i}} - 1). \tag{76}$$

While this formula was initially derived considering paths with a specific structure (e.g., $(\otimes\oplus)^l$), numerical verification on examples (such as $x \oplus_1^3 \otimes_{\ln 2}^3$) suggests that the formula and its underlying logic remain valid for other structures (like $\oplus^n\otimes^m$) provided $\Lambda_{>i}$ is correctly interpreted as the sum of exponents of *all* multiplicative steps occurring after the i-th additive step along the path.

The principal result of this discrete analysis is the direct identification of this algebraically computed sum τ_{sum} with the difference in the final numerical outcomes between the chosen evaluation orders γ and $\bar{\gamma}$:

$$\nu(\gamma) - \nu(\bar{\gamma}) = \tau_{sum}(\alpha) = \sum_{i=1}^{l} \mu_i(e^{\Lambda_{>i}} - 1) \tag{77}$$

(The sign depends on the precise definition of γ and $\bar{\gamma}$; the magnitude of the difference is given by $|\tau_{sum}|$). This equality has been consistently verified in concrete examples, such as comparing the evaluation of $(x + 3) \times 8$ (as $E(\gamma)$) and $x \times 8 + 3$ (as $E(\bar{\gamma})$), where the difference 21 exactly matches the value computed by the τ_{sum} formula.

Example 1: Additions followed by Multiplications

Consider the expression $E_1 = ((x+1)+1+1) \times 2 \times 2 \times 2$, corresponding to the path $\alpha_1 : x \oplus_1 \oplus_1 \oplus_1 \otimes_{\ln 2} \otimes_{\ln 2} \otimes_{\ln 2}$.

- Additive steps: $\mu_1 = 1, \mu_2 = 1, \mu_3 = 1$.
- Multiplicative steps: $\lambda_1 = \ln 2, \lambda_2 = \ln 2, \lambda_3 = \ln 2$.
- For $\mu_1 = 1$ (first addition): Subsequent multiplications are $\lambda_1, \lambda_2, \lambda_3$. $\Lambda_{>1} = \ln 2 + \ln 2 + \ln 2 = 3\ln 2 = \ln 8$. Contribution $\tau_1 = 1(e^{\ln 8} - 1) = 7$.
- For $\mu_2 = 1$ (second addition): Subsequent multiplications are $\lambda_1, \lambda_2, \lambda_3$. $\Lambda_{>2} = \ln 8$. Contribution $\tau_2 = 1(e^{\ln 8} - 1) = 7$.
- For $\mu_3 = 1$ (third addition): Subsequent multiplications are $\lambda_1, \lambda_2, \lambda_3$. $\Lambda_{>3} = \ln 8$. Contribution $\tau_3 = 1(e^{\ln 8} - 1) = 7$.
- Total discrete torsion: $\tau_{sum}(\alpha_1) = \tau_1 + \tau_2 + \tau_3 = 7 + 7 + 7 = 21$.
- Verification: Let γ be the evaluation order corresponding to $(x + 1 + 1 + 1) \times 8$, so $\nu(\gamma) = (x + 3) \times 8 = 8x + 24$. Let $\bar{\gamma}$ be the order corresponding to $(x \times 8) + 1 + 1 + 1$, so $\nu(\bar{\gamma}) = 8x + 3$. The difference is $\nu(\gamma) - \nu(\bar{\gamma}) = (8x + 24) - (8x + 3) = 21$. This matches $\tau_{sum}(\alpha_1)$.

Example 2: Alternating Operations

Consider the expression $E_2 = (((x \times 2) + 3) \times 4) + 5$, corresponding to the path $\alpha_2 : x \otimes_{\ln 2} \oplus_3 \otimes_{\ln 4} \oplus_5$.

- Additive steps: $\mu_1 = 3$ (occurring after λ_1), $\mu_2 = 5$ (occurring after λ_2).
- Multiplicative steps: $\lambda_1 = \ln 2, \lambda_2 = \ln 4$.
- For $\mu_1 = 3$: The subsequent multiplication is $\lambda_2 = \ln 4$. $\Lambda_{>1} = \ln 4$. Contribution $\tau_1 = 3(e^{\ln 4} - 1) = 3(4 - 1) = 9$.
- For $\mu_2 = 5$: There are no subsequent multiplications. $\Lambda_{>2} = 0$. Contribution $\tau_2 = 5(e^0 - 1) = 5(1 - 1) = 0$.
- Total discrete torsion: $\tau_{sum}(\alpha_2) = \tau_1 + \tau_2 = 9 + 0 = 9$.
- Verification: Let γ be the natural evaluation order $\nu(\gamma) = 8x + 17$ (as calculated previously). Let $\bar{\gamma}$ be the order where all multiplications happen first, then all additions: $\nu(\bar{\gamma}) = (x \times 2 \times 4) + 3 + 5 = 8x + 8$. The difference is $\nu(\gamma) - \nu(\bar{\gamma}) = (8x + 17) - (8x + 8) = 9$. This matches $\tau_{sum}(\alpha_2)$.

These examples illustrate the application of the discrete sum formula (76) and confirm its consistency with the evaluation difference (77) for different path structures. This establishes a solid, purely discrete framework for quantifying accumulated arithmetic torsion, independent of continuous geometric interpretations or integration techniques, and serves as a basis for further investigation.

This establishes a robust, purely discrete framework for quantifying accumulated arithmetic torsion based on the operational sequence of additions and multiplications. It provides a solid foundation, independent of continuous geometric interpretations or integration techniques, upon which further investigations into the geometric implications (like area correspondence via $\iint e^v dudv$) and the structure of a full Arithmetic Gauss–Bonnet theorem can be built.

5.2 From discrete to continuous

References

[1] Noam Chomsky. Three models for the description of language. *IRE Trans. Inf. Theory*, 2:113–124, 1956.

[2] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56 – 68, 1940.

[3] Alonzo Church. The calculi of lambda-conversion. 1941.

[4] William A. Howard. The formulae-as-types notion of construction. 1969.

[5] Donald Ervin Knuth. The art of computer programming, volume i: Fundamental algorithms, 2nd edition. 1997.

[6] Peter W. Markstein. Software division and square root using goldschmidt's algorithms. 2004.

[7] Per Martin-Löf. An intuitionistic theory of types: Predicative part. *Studies in logic and the foundations of mathematics*, 80:73–118, 1975.

[8] Per Martin-Löf. Intuitionistic type theory. In *Studies in proof theory*, 1984.

[9] Tristan Needham. Visual differential geometry and forms. 2021.

[10] Emil L. Post. Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*, 65:197, 1943.

[11] Eric Quinnell, Earl E. Swartzlander, and Carl Lemonds. Floating-point fused multiply-add architectures. *2007 Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers*, pages 331–337, 2007.

[12] John C. Reynolds. Definitional interpreters for higher-order programming languages. *Higher-Order and Symbolic Computation*, 11:363–397, 1972.

[13] Bertrand Russell. Mathematical logic as based on the theory of types. *American Journal of Mathematics*, 30:222, 1908.

[14] Jaap van Oosten. The univalent foundations program. homotopy type theory: Univalent foundations of mathematics. http: //homotopytypetheory.org/book, institute for advanced study, 2013, vii + 583 pp. *Bull. Symb. Log.*, 20:497–500, 2014.

A Solution of the flow equation

We can also get a direct formal solution of the flow equation ((30)) step by step:

$$\frac{da}{\mu\cos\theta + a\lambda\sin\theta} = ds$$

$$\frac{1}{\lambda\sin\theta}\frac{d(\mu\cos\theta + a\lambda\sin\theta)}{\mu\cos\theta + a\lambda\sin\theta} = ds$$

$$\frac{1}{\lambda\sin\theta}ln(\mu\cos\theta + a\lambda\sin\theta) = s + C$$

$$\mu\cos\theta + a\lambda\sin\theta = e^{\lambda s\sin\theta}e^{C\lambda\sin\theta}$$

Considering the initial condition

$$\mu\cos\theta + a_0\lambda\sin\theta = e^{C\lambda\sin\theta}$$

We have

$$\mu\cos\theta + a\lambda\sin\theta = e^{\lambda s\sin\theta}(\mu\cos\theta + a_0\lambda\sin\theta)$$

$$a = \frac{\mu\cos\theta + a_0\lambda\sin\theta}{\lambda\sin\theta}e^{\lambda s\sin\theta} - \frac{\mu}{\lambda}\cot\theta$$

$$a = (a_0 + \frac{\mu}{\lambda}\cot\theta)e^{\lambda s\sin\theta} - \frac{\mu}{\lambda}\cot\theta$$

$$a = a_0 e^{\lambda s\sin\theta} + \frac{\mu}{\lambda}(e^{\lambda s\sin\theta} - 1)\cot\theta \tag{78}$$

$$a = a_0 e^{\lambda s\sin\theta} + \frac{\mu}{\lambda}(e^{\lambda s\sin\theta} - 1)\cot\theta \tag{79}$$

B Geometry calculation

Giving

$$ds^2 = \frac{1}{y^2}(\frac{dx^2}{\mu^2} + \frac{dy^2}{\lambda^2})$$

we calculate the geometric quantities. We follow the notion in the text book[9], and the first fundamental form is given by

$$ds^2 = A^2dx^2 + B^2dy^2$$

where

$$A = \frac{1}{\mu y}, \quad B = \frac{1}{\lambda y}$$

B.1 Line element

The line element is already given by above equation.

B.2 Area element

The area element is given by

$$dS = ABdxdy$$

hence we have

$$dS = \frac{1}{\mu\lambda y^2}dxdy$$

B.3 Gauss curvature

Gauss curvature K is given by

$$K = -\frac{1}{AB}\left(\partial_y\left(\frac{\partial_y A}{B}\right) + \partial_x\left(\frac{\partial_x B}{A}\right)\right)$$

so we have

$$K = -\mu\lambda y^2\left(\partial_y\left(\lambda y\partial_y\left(\frac{1}{\mu y}\right)\right) + \partial_x\left(\mu y\partial_x\left(\frac{1}{\lambda y}\right)\right)\right)$$

$$K = -\lambda^2y^2\left(\partial_y\left(y\partial_y\left(\frac{1}{y}\right)\right)\right)$$

$$K = -\lambda^2y^2\frac{1}{y^2}$$

$$K = -\lambda^2$$

B.4 Laplacian

Given a metric tensor

$$g = \begin{bmatrix} A^2 & 0 \\ 0 & B^2 \end{bmatrix},$$

where A and B are functions of the coordinates (typically x and y), the Laplacian of a function $f(x, y)$ can be derived from the general expression of the Laplace-Beltrami operator for a Riemannian manifold. The formula for the Laplacian Δf in such a setting, using the metric components g_{ij}, is given by:

$$\Delta f = \frac{1}{\sqrt{|g|}} \partial_i \left(\sqrt{|g|} g^{ij} \partial_j f \right),$$

where $|g|$ is the determinant of the metric tensor g_{ij}, g^{ij} are the components of the inverse metric tensor, and ∂_i denotes partial differentiation with respect to the ith coordinate.

Given the metric tensor, the determinant $|g|$ is A^2B^2. The inverse metric tensor g^{ij} is simply:

$$g^{ij} = \begin{bmatrix} \frac{1}{A^2} & 0 \\ 0 & \frac{1}{B^2} \end{bmatrix}.$$

Plugging these into the formula for the Laplacian, we get:

$$\Delta f = \frac{1}{AB} \left[\partial_x \left(BA^{-1} \partial_x f \right) + \partial_y \left(AB^{-1} \partial_y f \right) \right],$$

In our setting, $A = \frac{1}{\mu y}$ and $B = \frac{1}{\lambda y}$:

$$\Delta f = y^2 \left(\mu^2 \frac{\partial^2 f}{\partial x^2} + \lambda^2 \frac{\partial^2 f}{\partial y^2} \right)$$

And for the function $f = -\frac{x}{y}$, we have

$$\Delta f = -\frac{2\lambda^2 x}{y} = 2\lambda^2 f$$

So, we reach the conclusion that the function $f = -\frac{x}{y}$ is a eigenfunction of the Laplacian with eigenvalue $2\lambda^2$.