

The Attralucian Essays:
Exploring the Finite



First Edition

Copyright © 2026 by Kevin R. Haylett. All rights reserved.

This work is shared under the Creative Commons Licence.

Creative Commons CC BY-ND 4.0 License.

<https://creativecommons.org/licenses/by-nd/4.0/>

This work is intended for academic and research use. Any unauthorized distribution, modification, or commercial use beyond the creative use license is strictly prohibited. Typeset in

L^AT_EX

The Attralucian Essays



Finite Process Unfolding
A Method for Recovering Temporal Structure
from Static Symbolic Forms

Kevin R. Haylett

Overview

Mathematics is typically presented as a static system of symbols, equations, and identities. Yet every such expression is produced through finite, ordered operations carried out under constraints of measurement and distinction. This paper introduces *Finite Process Unfolding* (FPU), a methodology for reconstructing the sequential structure compressed within symbolic forms.

FPU operates entirely within the symbolic frame: no appeal is made to entities or structures beyond the boundary at which symbols are formed. Each symbol is treated as the result of a finite act of distinction—a generonic event—embedded within a history of measurement. The aim is to recover admissible generating procedures for symbolic expressions, making explicit their intermediate states, ordering constraints, and process costs.

We present the method, situate it historically, and demonstrate its application to non-commutativity. The broader claim is that mathematics may be treated as a nonlinear dynamical system of symbolic trajectories, in which equations are stable compressions of finite processes.

Introduction

Mathematical expressions appear static, but their production, interpretation, and verification are inherently

sequential. Finite Process Unfolding (FPU) provides a structured method for reversing symbolic compression, recovering the ordered processes hidden within expressions.

FPU is not presented as a unique method. Rather, it is one admissible procedure for reconstructing process from symbolic form, with the expectation that alternative unfoldings may exist and be compared.

The Symbolic Frame and Generonic Boundary

Finite Process Unfolding operates entirely within the symbolic frame. No appeal is made to entities beyond the boundary at which symbols are formed.

Each symbol is treated as the outcome of a finite act of distinction, referred to here as a *generonic event*. Such events occur at the boundary of distinguishability, producing symbols that carry measurable structure and historical provenance.

Meaning does not arise from correspondence to a pre-symbolic domain. Instead, it emerges through the accumulation of relational structure within a finite symbolic container. The task of analysis is therefore to reconstruct the admissible processes by which symbolic forms are generated, stabilised, and transformed.

Historical Context

Early mathematics was procedural. Euclidean constructions, for example, consisted of explicit sequences of operations. The rise of algebra introduced symbolic compression, allowing procedures to be stabilised as expressions and manipulated through substitution and equivalence.

Modern mathematics emphasises structure. Operations are treated as relations between abstract objects, often independent of any explicit generating procedure. This shift enables powerful generalisation but suppresses the sequential dynamics underlying symbolic forms.

FPU reintroduces these dynamics, not by rejecting structure, but by treating it as a compression of process.

Problem Statement

Several recurring phenomena suggest that symbolic compression obscures essential structure:

- Non-commutativity appears as a structural property rather than a consequence of ordered operations.
- Limits and derivatives collapse extended approximation procedures into instantaneous values.
- Statistical measures compress experimental histories into scalar summaries.

- Representational failures (e.g., gimbal lock) indicate insufficient capacity to preserve distinguishability.

These observations motivate the central question:

What finite, sequential process must occur for a symbolic expression to be admissible?

Finite Process Unfolding Method

Let S be a symbolic expression. FPU constructs one or more admissible procedures $\mathcal{P}(S)$ that generate or verify S under finite constraints.

Step 1: Identify the Compression Boundary

Select the symbolic object S to be analysed. Treat it as a boundary beyond which process has been compressed.

Step 2: Classify the Compression Type

Determine whether S represents:

- Composition (operators, matrices)
- Aggregation (sums, integrals)
- Limiting processes (derivatives, limits)

- Statistical compression (distributions, test statistics)

Step 3: Reconstruct a Sequential Procedure

Construct a finite sequence of operations such that:

- Each step is ordered and finite
- Each step produces a distinguishable state
- Each distinction incurs a minimum cost $\Delta M > 0$

Step 4: Identify State Requirements

Determine what information must be retained:

- Intermediate states
- Ordering information
- Measurement results

Step 5: Estimate Process Cost

Estimate a lower bound:

$$\text{Cost} \geq N_{\text{steps}} \cdot \Delta M$$

Step 6: Test Admissibility

Assess whether the process can be represented within a finite symbolic container:

- Are states distinguishable?
- Is storage sufficient?

Step 7: Restate the Expression

Rewrite S as a compressed trace of the reconstructed process.

Example: Non-Commutativity

Consider:

$$AB \neq BA.$$

Unfolding yields:

1. Apply A to a state
2. Store the intermediate result
3. Apply B
4. Repeat in reverse order
5. Compare final states

The commutator

$$[A, B] = AB - BA$$

is thus interpreted as the measurable residue of two ordered procedures.

Discussion

FPU reframes symbolic expressions as compressions of process. Commutativity corresponds to order-indifference, while non-commutativity implies preserved sequence and memory.

Mathematics may therefore be viewed as a nonlinear dynamical system of symbolic trajectories, where stable expressions act as attractors.

Conclusion

Finite Process Unfolding provides one method for recovering temporal structure from static symbolic forms. By reconstructing admissible generating procedures, it reveals the sequential dynamics underlying mathematical expressions.

The symbolic form is not primary. It is a compression of process.

Appendix A: LLM Prompt for Finite Process Unfolding

Prompt Template:

Given a symbolic expression S , perform a Finite Process Unfolding analysis as follows:

1. Identify the type of symbolic compression present in S .
2. Construct a finite sequential procedure that generates or verifies S .
3. Explicitly list all intermediate distinguishable states.
4. Determine whether the outcome depends on order.
5. Estimate the minimum process cost in terms of ΔM .
6. Evaluate whether the representation is admissible under finite constraints.
7. Restate S as a compressed trace of the reconstructed process.

Extension Task:

After completing the unfolding:

- Propose at least one alternative unfolding procedure $\mathcal{P}'(S)$.

Finite Process Unfolding

- Compare the process costs and memory requirements of $\mathcal{P}(S)$ and $\mathcal{P}'(S)$.
- Identify any differences in admissibility or representational efficiency.

Objective:

Treat the symbolic expression not as a static object, but as a compressed representation of one or more admissible finite processes.