Technical Appendix: UNNS Simulation Showcase v1.0

Temporal Recursion Engine, Klein Surface Mapper, and Theoretical Coupling

UNNS Research Initiative, 2025

Abstract

This document provides the technical and theoretical foundation for the *UNNS Simulation Showcase v1.0*, an interactive visualization suite that demonstrates recursive temporal dynamics and their geometric realization on non-orientable manifolds. The prototype integrates three conceptual domains:

1. **Temporal Recursion Engine (TRE)** — simulates forward and reverse UNNS recursion.
2. **Klein Surface Mapper (KSM)** — embeds recursion trajectories onto a dynamic Klein surface.
3. **Confluence Architecture** — couples numeric recursion and geometric deformation as dual projections of the same process.

Each subsystem is implemented in JavaScript (Canvas and WebGL/Three.js) with modular interconnectivity and educational instrumentation.

1 System Overview

The UNNS Simulation Showcase visualizes recursion as a computational substrate where *depth* replaces continuous time. Each iteration represents a discrete operator transformation:

$$a_{n+1} = \alpha a_n + \beta \tanh(a_{n-1}) + \delta n + \sigma \varepsilon_n,$$

where:

- α — damping or persistence parameter,
- β — coupling coefficient controlling nonlinearity,
- δ — drift term (time-dependent forcing),

- $\sigma\varepsilon_n$ — stochastic perturbation (Gaussian noise).

The forward mapping defines the *temporal cone* (information expansion), while the reverse operator:

$$a_{n-1} = \tanh^{-1}\left(\frac{a_{n+1} - \alpha a_n - \delta n}{\beta}\right)$$

exists only when the transformation $F(a_n, a_{n-1})$ is locally invertible.

2 Module I — Temporal Recursion Engine (TRE)

2.1 Purpose

TRE visualizes the progression of recursion depth as a function of stability and energy evolution. It allows:

- Parameter scanning across $(\alpha, \beta, \delta, \sigma)$,
- Visualization of state trajectories a_n,
- Real-time heatmap of Jacobian magnitudes,
- Bidirectional time traversal (forward and reverse recursion).

2.2 Algorithmic Core

Forward recursion follows the iterative rule:

$$a_{n+1} = F(a_n, a_{n-1}, n),$$

computed sequentially across N steps. For reversibility testing, the inverse operator F^{-1} is approximated symbolically and checked for domain validity:

$$|(a_{n+1} - \alpha a_n - \delta n)/\beta| < 1.$$

When violated, the mapping becomes non-invertible, representing loss of information (entropy increase).

2.3 Stability Metric

The local Jacobian norm is given by:

$$J_n = \sqrt{\left(\frac{\partial F}{\partial a_n}\right)^2 + \left(\frac{\partial F}{\partial a_{n-1}}\right)^2},$$

which informs a color-coded stability plot:

$$\text{bright regions: } J_n < 1 \quad \text{(stable)}, \quad \text{dark regions: } J_n > 1 \quad \text{(divergent)}.$$

This provides an empirical sense of attractor behavior.

3 Module II — Klein Surface Mapper (KSM)

3.1 Topology

The KSM represents a non-orientable manifold defined by the identification:

$$(\theta, 0) \sim (\theta, 1), \qquad (0, y) \sim (1, 1 - y).$$

The resulting surface (Klein bottle) exhibits $w_1 \neq 0$, signifying global non-orientability, though locally Euclidean.

3.2 Parametric Embedding

The surface is parametrized by $(u, v) \in [0, 2\pi]^2$:

$$\begin{cases} x = (r + a\cos v)\cos u, \\ y = (r + a\cos v)\sin u, \\ z = a\sin v\cos(u/2), \end{cases}$$

rendered via `THREE.ParametricGeometry`. The surface dynamically *breathes* and rotates, with emissive glow proportional to recursion energy.

3.3 Flow Field Visualization

Flow particles are advected along the surface according to:

$$\dot{u} = \omega_u, \quad \dot{v} = 0,$$

where ω_u varies sinusoidally with recursion energy, simulating transport along the A-cycle of the Klein surface.

4 Coupling: TRE $\rightarrow$ KSM

The TRE output is used as an input driver for the KSM:

$$\begin{aligned} \text{scale factor} &= 1 + 0.15\sin(n \cdot 0.05), \\ \text{color hue} &= f(\alpha, E_n), \\ \text{emissive intensity} &= g(|\sin n|, E_n). \end{aligned}$$

This coupling demonstrates that recursive numeric evolution can deform a geometric topology in time-dependent ways, representing the duality between recursion and geometry.

5 Data Structures

- **TRE data model:**

```
{
  alpha, beta, delta, noise, steps,
  sequence[], energies[], stabilities[]
}
```

- **KSM dynamic parameters:**

```
{
  rotSpeed, flowDensity, twist, energyCoupling
}
```

6 Theoretical Correspondence

1. **Temporal recursion ↔ local flow:** Each recursive depth increment corresponds to infinitesimal displacement on the surface.
2. **Invertibility ↔ orientability:** Non-invertible recursion corresponds to non-orientable phase identification on the Klein surface.
3. **Energy decay ↔ metric contraction:** Damping reduces local curvature, compressing the surface volume.
4. **Stability cone ↔ recursion cone:** The Jacobian norm visualized as cone opening angle illustrates local causality spread.

7 Implementation Summary

7.1 Technologies

- JavaScript (ES6), Canvas 2D API, WebGL (Three.js)
- CSS grid and variable-driven theming
- Event-driven architecture for modular simulation control
- JSON export for reproducibility and analysis

7.2 Performance

Frame synchronization achieved via `requestAnimationFrame`, with adaptive rendering frequency for smooth visuals at ∼60 FPS.

8 Interpretation and Future Work

The prototype reveals how recursive numeric processes can encode and visualize reversible (and non-reversible) temporal structures. Next steps include:

1. Introducing a **UNNS–Manifold Tensor Engine** (symbolic mapping of recursion operators to metric tensors),
2. Adding **energy–entropy cross-spectra** to analyze information dissipation,
3. Implementing a **Klein Flow Integrator** for topological charge preservation.

References

- I. Chomko, *On the Possibility of Temporal Recursion in the UNNS Substrate*, 2025.
- I. Chomko, *Temporal Recursion and Klein Surface Realization*, 2025.
- UNNS Research Notes: `https://ukbbi.github.io/UNNS/`