UNNS Inletting and Inlaying

Formal Definitions, Properties, and Algorithms

UNNS Research Notes

September 21, 2025

Abstract

This document formalizes two complementary operations on the UNNS substrate: *Inletting* (injection of external data into the substrate) and *Inlaying* (internal embedding of motifs within the substrate). Each notion is presented with definitions, examples, a basic stability lemma, an algorithmic recipe, and an illustrative diagram.

1 Overview

UNNS (Unbounded Nested Number Sequences) supports two dual coupling primitives:

- **Inletting:** coupling external finite data to UNNS (boundary input, source terms).
- **Inlaying:** re-embedding internal motifs into other parts of the substrate (self-coupling).

Both are essential for modeling interaction, repair, and engineered topology in the substrate.

2 UNNS Inletting

Definition 1 (UNNS Substrate). *Let $\mathcal{U}$ denote a UNNS substrate: a recursive medium capable of sustaining sequences, mesh labelings, and echo residues generated by finite-order recurrences or local stencils.*

Definition 2 (UNNS Inletting). *A* UNNS inletting *is a morphism*

$$\iota : D \longrightarrow \mathcal{U},$$

where D is a finite dataset (numerical samples, boundary assignments, or source coefficients), such that:

(i) ***Compatibility:*** *$\iota(D)$ aligns with the recurrence/order and data types supported by $\mathcal{U}$.*

(ii) ***Stability:*** *the mapped data either respect the substrate's stability thresholds (e.g. UPI bounds), or the inletting is followed by repair/renormalization.*

(iii) ***Continuity:*** *echoes generated from $\iota(D)$ propagate without producing unacceptable discontinuities.*

Example 1 (Sequence seeding). *Injecting $(u_0, u_1) = (2, 5)$ into a Fibonacci-type UNNS by ι overwrites initial seeds and produces a new trajectory.*

Example 2 (Mesh boundary inletting)**.** *Prescribing edge coefficients $c(e)$ on the boundary of a UNNS-labeled mesh from external measurements; the recurrence extends inward.*

Lemma 1 (Bounded inletting)**.** *Let $\mathcal{U}$ be generated by a linear recurrence with companion matrix C and spectral radius $\rho(C) < 1$. If ι injects bounded data $|d| \le M$, then the resulting UNNS trajectory remains bounded by a constant depending on M and $\rho(C)$.*

Sketch. The recurrence acts as a stable linear filter: the bounded input is propagated with geometric decay controlled by $\rho(C)$, giving an output bound of order $M/(1 - \rho(C))$ up to constant factors depending on the recurrence order. □

Algorithmic recipe (Inletting)

1. **Prepare:** collect external data D and determine whether they are seeds, coefficients, or boundary assignments.
2. **Compatibility test:** check type/order compatibility; if necessary, project to admissible algebraic rings.
3. **Apply ι:** write the data into $\mathcal{U}$ at the prescribed locations.
4. **Stabilize:** if diagnostics (UPI, growth factors) exceed thresholds, apply repair operators (proofreading, excision, renorm).
5. **Monitor:** evolve forward, log echoes/residues, and revert if unacceptable side-effects occur.

Figure 1: Schematic of UNNS inletting.

3 UNNS Inlaying

Definition 3 (UNNS Inlaying)**.** *A* UNNS inlaying *is an internal embedding operator*

$$\mathcal{I}_{P \to L} : \mathcal{U} \longrightarrow \mathcal{U},$$

which selects a motif P already present in $\mathcal{U}$ (a subsequence, local stencil, or small labeled subgraph), produces a transformed copy $\widetilde{P}$ (optionally rescaled, phase-shifted or projected), and inserts $\widetilde{P}$ into a target location L so that UNNS recursive dynamics extend over $\widetilde{P}$ under the same ruleset.

Example 3 (Sequence inlay)**.** *Copy subsequence $P = (u_k, \ldots, u_{k+m-1})$ and insert a transformed copy at index j to influence local recurrence behavior.*

Example 4 (Mesh inlay)**.** *Copy a triangular patch of edge coefficients and paste a projected copy at a distant patch to produce controlled echo variations.*

Lemma 2 (Local stabilization by contractive inlay)**.** *If a motif P has a companion matrix with spectral radius $\rho(C_P) < 1$, then inlaying P into a region R whose local effective companion had $\rho_R > 1$ makes R locally contractive (modulo boundary coupling effects).*

Sketch. Replacing the local update rule by a contractive local companion matrix reduces the local propagation spectral radius; with sufficiently weak boundary coupling, the interior contractive dynamics dominate and local modes decay. □

Algorithmic recipe (Inlaying)

1. **Select motif** P from a stable area of $\mathcal{U}$.
2. **Transform** to $\widetilde{P}$ (optional projection or scaling).
3. **Simulate insertion** at target L in a local model; compute local spectral diagnostics.
4. **Accept/reject** based on thresholds; commit if acceptable.
5. **Monitor** after commit and apply repairs if needed.

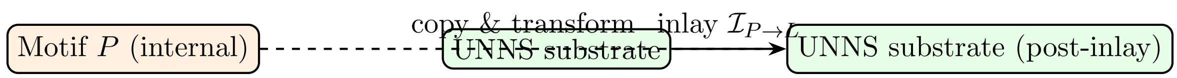

Figure 2: Schematic of UNNS inlaying (copying an internal motif and embedding it elsewhere).

4 Duality and remarks

- **Duality:** Inletting couples an external boundary to the substrate; inlaying couples the substrate to itself.
- **Composability:** Inletting followed by inlaying (or vice versa) produces a rich set of possible dynamics; one can inlay motifs derived from inletted data.
- **Topology:** Inlaying acts like attaching maps in combinatorial topology (a way to locally alter cell structure), while inletting plays the role of boundary conditions.