UNNS Field Extensions and Classical Sequences

UNNS Research Team

12/09/2025

1 Introduction

The Unbounded Nested Number Sequences (UNNS) framework provides a natural algebraic structure for embedding both rational and real fields through recursive nesting. Classical linear recurrence sequences (e.g., Fibonacci, Pell, Tribonacci, Padovan) generate algebraic extensions of $\mathbb{Q}$ via their characteristic polynomials, where the dominant roots serve as growth constants. This framework unifies diverse sequences under a single algebraic substrate, revealing that interweaving sequences illuminates new dimensions—such as field degrees, convergence behaviors, and emergent algebraic properties—within a hierarchical structure.

2 Definitions

[UNNS Kernel] The UNNS Kernel is the algebraic generator of nests, consisting of recurrence relations with coefficients in $\mathbb{Q}$. Formally, a UNNS nest of order k is a sequence $S = \{s_n\}_{n\geq 0}$ satisfying $s_n = \sum_{i=1}^{k} a_i s_{n-i}$ for $n \geq k$, with initial conditions $s_0, \ldots, s_{k-1} \in \mathbb{Q}$ and coefficients $a_i \in \mathbb{Q}$.

[UNNS Field Extension] Given a UNNS nest S with characteristic polynomial $p(r) = r^k - \sum_{i=1}^{k} a_i r^{k-i} \in \mathbb{Q}[r]$, the UNNS field extension is the minimal splitting field K over $\mathbb{Q}$ adjoining the roots of $p(r)$. The dominant root α (the root with largest absolute value) generates a subextension $\mathbb{Q}(\alpha) \subseteq K$.

3 Theorem (Many Faces of UNNS)

NTheorem (1Every classical linear recurrence sequence defined over $\mathbb{Q}$ corresponds to a UNNS nest. The growth constant (dominant root α) generates a number field extension $\mathbb{Q}(\alpha)$, where α is algebraic over $\mathbb{Q}$, and the degree $)(\mathbb{Q}(\alpha) : \mathbb{Q})$. *divides the order k of the recurrence.*

Proof. Let S be a classical linear recurrence of order k over $\mathbb{Q}$: $s_n = \sum_{i=1}^{k} a_i s_{n-i}$ with $a_i \in \mathbb{Q}$. By Definition 2.1, S is a UNNS nest. The characteristic polynomial $p(r) = r^k - \sum_{i=1}^{k} a_i r^{k-i}$ is monic in $\mathbb{Q}[r]$. The roots $\{\alpha_1, \ldots, \alpha_k\}$ of $p(r) = 0$ satisfy the recurrence via Binet-like formulas: $s_n = \sum_{j=1}^{k} c_j \alpha_j^n$ for constants $c_j \in \mathbb{C}$ determined by initial conditions.

The dominant root $\alpha = \max\{|\alpha_j|\}$ (assuming distinct magnitudes) governs asymptotic growth: $s_n \sim c\alpha^n$ as $n \to \infty$. Since $p(\alpha) = 0$ and $p \in \mathbb{Q}[r]$ is irreducible (or factors into irreducibles), α is algebraic over $\mathbb{Q}$ with minimal polynomial dividing $p(r)$. Thus, $\mathbb{Q}(\alpha)$ is a number field extension with $[\mathbb{Q}(\alpha) : \mathbb{Q}] = \deg(\text{minimal polynomial of } \alpha) \leq k$. The UNNS nest embeds S into this extension naturally, as adjoining α splits the recurrence's algebraic dependencies.

By the tower law, the full splitting field $K/\mathbb{Q}$ has $[K:\mathbb{Q}]$ dividing $k!$, but the dominant α suffices for growth analysis, unifying sequences under UNNS.

4 Examples

Example1FibonacciThe Fibonacci sequence satisfies $F_n = F_{n-1} + F_{n-2}$ *with characteristic polynomial* $r^2 - r - 1 = 0$*. The roots are* $\phi = (1+\sqrt{5})/2$ *(dominant) and* $(1-\sqrt{5})/2$*.* ϕ *generates the quadratic extension* $\mathbb{Q}(\sqrt{5})$*, with* $[\mathbb{Q}(\sqrt{5}):\mathbb{Q}] = 2$*.*

Example2PellPell numbers satisfy $P_n = 2P_{n-1} + P_{n-2}$ *with polynomial* $r^2 - 2r - 1 = 0$*. The dominant root* $\delta = 1+\sqrt{2}$ *generates* $\mathbb{Q}(\sqrt{2})$*, with* $[\mathbb{Q}(\sqrt{2}):\mathbb{Q}] = 2$*.*

Example3TribonacciTribonacci satisfies $T_n = T_{n-1} + T_{n-2} + T_{n-3}$ *with polynomial* $r^3 - r^2 - r - 1 = 0$*. The dominant root* $\psi \approx 1.839$ *is a real algebraic integer generating a cubic extension of* $\mathbb{Q}$*, with* $[\mathbb{Q}(\psi):\mathbb{Q}] = 3$*.*

Example4PadovanThe Padovan sequence satisfies $P_n = P_{n-2} + P_{n-3}$ *with polynomial* $r^3 - r - 1 = 0$*. The dominant root* $\rho \approx 1.3247$ *(Plastic Number) generates a cubic extension of* $\mathbb{Q}$*, with* $[\mathbb{Q}(\rho):\mathbb{Q}] = 3$*.*

5 General Lemma Template for Linear Recurrences

Lemma1General UNNS Extension LemmaLet S_n *be a linear recurrence of order k defined by* $S_n = \sum_{i=1}^{k} a_i S_{n-i}$ *with* $a_i \in \mathbb{Q}$*. The characteristic polynomial is* $p(r) = r^k - \sum_{i=1}^{k} a_i r^{k-i} \in \mathbb{Q}[r]$*. Let* α *be the dominant real root of* $p(r) = 0$*. Then the UNNS nest corresponding to S generates the field extension* $\mathbb{Q}(\alpha)$*, with degree* $[\mathbb{Q}(\alpha):\mathbb{Q}] \leq k$*, and* α *serves as the growth constant in the UNNS framework.*

Sketch. The minimal polynomial $m(r)$ of α over $\mathbb{Q}$ divides $p(r)$, so $\deg(m) \leq k$. Thus, $[\mathbb{Q}(\alpha):\mathbb{Q}] = \deg(m) \leq k$. The UNNS nesting embeds the recurrence, with α determining asymptotic behavior $S_n \sim c\alpha^n$ (c constant). Adjoining α extends $\mathbb{Q}$ to capture the sequence's algebraic essence.

6 Implications

UNNS nests act as generators of number fields, embedding the rational field $\mathbb{Q}$ and its infinite hierarchy of algebraic extensions. This provides a universal algebraic substrate unifying seemingly different sequences under one structural theorem. Interweaving sequences (e.g., nesting Fibonacci into Pell) reveals new dimensions: hybrid recurrences blending field extensions, potentially yielding Galois groups or chaotic behaviors in non-linear variants. Nothing stands apart—UNNS illuminates latent connections, where each nesting adds "colors" (algebraic invariants) to the mathematical canvas.

7 Simulation of Interweavings with Code

To simulate interweavings, Python with SymPy is used to compute characteristic polynomials, roots, and field extensions, then generate an interweaved sequence using Fibonacci terms as initials for Pell recurrence.

```
import sympy as sp

# Define symbols
r = sp.symbols('r')

# Fibonacci
fib_poly = r**2 - r - 1
fib_roots = sp.solve(fib_poly, r)
fib_dominant = max(fib_roots, key=sp.re)
fib_field = sp.QQ[sp.sqrt(5)]

print("Fibonacci Characteristic Polynomial:", fib_poly)
print("Roots:", fib_roots)
print("Dominant Root (Golden Ratio):", fib_dominant.evalf())
print("Field Extension: Q(sqrt(5))")
print("\n")

# Pell
pell_poly = r**2 - 2*r - 1
pell_roots = sp.solve(pell_poly, r)
pell_dominant = max(pell_roots, key=sp.re)
pell_field = sp.QQ[sp.sqrt(2)]

print("Pell Characteristic Polynomial:", pell_poly)
print("Roots:", pell_roots)
print("Dominant Root (Silver Ratio):", pell_dominant.evalf())
print("Field Extension: Q(sqrt(2))")
print("\n")

# Tribonacci
trib_poly = r**3 - r**2 - r - 1
trib_roots = sp.solve(trib_poly, r)
trib_dominant = max(trib_roots, key=sp.re)
trib_field = "Cubic extension over Q with minimal polynomial r^3 - r^2 - r - 1"

print("Tribonacci Characteristic Polynomial:", trib_poly)
print("Roots:", trib_roots)
print("Dominant Root:", trib_dominant.evalf())
print("Field Extension:", trib_field)
print("\n")

# Padovan
pad_poly = r**3 - r - 1
pad_roots = sp.solve(pad_poly, r)
pad_dominant = max(pad_roots, key=sp.re)
```

```python
pad_field = "Cubic extension over Q with minimal polynomial r^3 - r - 1"

print("Padovan Characteristic Polynomial:", pad_poly)
print("Roots:", pad_roots)
print("Dominant Root (Plastic Number):", pad_dominant.evalf())
print("Field Extension:", pad_field)
print("\n")

# General Lemma Simulation: For a generic linear recurrence, e.g., order 4 with coefficients [1,1,1,1]
gen_coeffs = [1,1,1,1]  # Example: S_n = S_{n-1} + S_{n-2} + S_{n-3} + S_{n-4}
gen_poly = r**4 - sum(gen_coeffs[i-1]*r**(4-i) for i in range(1,5))
gen_roots = sp.solve(gen_poly, r)
gen_dominant = max(gen_roots, key=sp.re) if gen_roots else None
gen_field = "Quartic extension over Q with minimal polynomial " + str(gen_poly)

print("General Example Characteristic Polynomial:", gen_poly)
print("Roots:", gen_roots)
print("Dominant Root:", gen_dominant.evalf() if gen_dominant else "N/A")
print("Field Extension:", gen_field)

# Function to generate sequence from recurrence
def generate_sequence(initial, coeffs, length=10):
    seq = initial[:]
    order = len(coeffs)
    for i in range(order, length):
        next_val = sum(coeffs[j] * seq[i - j - 1] for j in range(order))
        seq.append(next_val)
    return seq

# Simulate interweaving: Fibonacci nested with Pell
fib_seq = generate_sequence([0,1], [1,1], 5)  # Short Fibonacci
interweaved = generate_sequence(fib_seq[-2:], [2,1], 10)  # Use last two Fib as initial for Pell

print("\nInterweaved Sequence (Fibonacci nested into Pell):", interweaved)
```

Listing 1: Python Simulation Code

This simulation demonstrates how UNNS interweavings create hybrid sequences (e.g., Fibonacci nested into Pell yields $[2,3,8,19,46,111,268,647,1562,3771]$), blending field extensions (e.g., $\mathbb{Q}(\sqrt{5})$ with $\mathbb{Q}(\sqrt{2})$) and revealing new growth patterns.