UNNS Inletting: Mathematical + Physical Perspectives

Appendix: Worked Example — Sine Wave Inletting

1 UNNS Inletting: Entry Points into the Substrate (Summary)

(For brevity this restates the main definitions used in the worked example; the full development is assumed known.)

Definition 1 (UNNS Inletting)**.** *A* UNNS inletting *is a morphism*

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

where D is an external, finite dataset and $\mathcal{U}$ the UNNS substrate, satisfying: (1) recurrence compatibility, (2) threshold preservation (UPI constraints), and (3) echo continuity under recursive evolution.

Remark 1. *Operationally, an inletting takes a finite sample $D = \{d_0, \ldots, d_{N-1}\}$ and maps it to an initial data block (seeds) and, optionally, recurrence coefficients $(c_1, \ldots, c_r)$ so that the UNNS nest*

$$u_{n+r} = c_1 u_{n+r-1} + \cdots + c_r u_n$$

extends D and then generates nested echoes when iterated.

2 Worked Example: Inletting a Sampled Sine Wave

This worked example shows step-by-step how to inlet a simple time-series (signal) into the UNNS substrate, fit a linear recurrence, check a stability proxy (spectral radius of the companion matrix), and propagate to observe echo residues.

2.1 Problem statement

Let D be samples of a real sinusoid:

$$d_k \;=\; \sin\!\Big(2\pi \frac{f}{F_s} k\Big), \qquad k = 0, \ldots, N-1,$$

with frequency f and sampling rate F_s. Take a concrete finite sample D and define an inletting $\iota(D)$ into UNNS as follows:

1. Choose an order r for the recurrence (e.g. $r = 4$ or $r = 6$).
2. Fit coefficients $c = (c_1, \ldots, c_r)$ so that the recurrence predicts the data in a least-squares sense.

3. Use the fitted recurrence and the last r data points as *seeds* to propagate the UNNS nest forward and observe echoes.

4. Compute a stability proxy: the spectral radius $\rho(C)$ of the companion matrix C (if $\rho(C) < 1$, contraction; if $\rho(C) > 1$, possible growth).

5. If necessary, renormalize or project coefficients to satisfy the UNNS threshold (UPI).

2.2 Fitting a linear recurrence (least squares)

Form the linear system for unknown coefficients $c_1, \ldots, c_r$. For $n = r, \ldots, N-1$,

$$d_n \;=\; c_1 d_{n-1} + c_2 d_{n-2} + \cdots + c_r d_{n-r} + \varepsilon_n,$$

where ε_n is the residual. Stack these equations:

$$A\,c \;=\; b + \varepsilon, \qquad A = \begin{bmatrix} d_{r-1} & d_{r-2} & \cdots & d_0 \\ d_r & d_{r-1} & \cdots & d_1 \\ \vdots & \vdots & \ddots & \vdots \\ d_{N-2} & d_{N-3} & \cdots & d_{N-r-1} \end{bmatrix}, \quad b = \begin{bmatrix} d_r \\ d_{r+1} \\ \vdots \\ d_{N-1} \end{bmatrix}.$$

Solve the normal equations with Tikhonov regularization (optional):

$$c = \arg\min_c \|Ac - b\|_2^2 + \lambda \|c\|_2^2 \quad \Longrightarrow \quad c = (A^\top A + \lambda I)^{-1} A^\top b.$$

Choose a small regularization $\lambda > 0$ to stabilise the fit.

2.3 Companion matrix and spectral radius

Given $c = (c_1, \ldots, c_r)$, form the $r \times r$ companion matrix

$$C \;=\; \begin{bmatrix} c_1 & c_2 & \cdots & c_{r-1} & c_r \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

Compute its spectral radius $\rho(C) = \max\{|\lambda| : \lambda \in \mathrm{Spec}(C)\}$. As a proxy for UNNS stability, require $\rho(C)$ to be not too large; if $\rho(C) > \rho_{\mathrm{crit}}$ (user-chosen; e.g. 1.05), apply a renormalization or Tikhonov re-fit to reduce growth.

2.4 Propagation and echo residue

Use the last r sampled values as seeds:

$$u_0 = d_{N-r},\; u_1 = d_{N-r+1},\; \ldots,\; u_{r-1} = d_{N-1},$$

then for $m \geq 0$ compute forward

$$u_{m+r} = \sum_{j=1}^{r} c_j u_{m+r-j}.$$

Define the *local echo residue* after some traversal or comparison to a ground-truth continuation (if available). For this example we monitor growth and oscillatory pattern:

$$\epsilon_m = u_{m+r} - \widetilde{u}_{m+r},$$

where $\widetilde{u}_{m+r}$ may be the actual sine continuation if accessible; otherwise we track $|\epsilon_m|$ relative to the seed amplitude.

2.5 Worked (algorithmic) summary

1. Pick N (sample length), frequency ratio f/F_s, and order r.
2. Build sample vector $d_0, \ldots, d_{N-1}$.
3. Build matrix A and vector b. Solve regularized least squares for c.
4. Form companion matrix C and compute $\rho(C)$.
   - If $\rho(C) > \rho_{\text{crit}}$, increase λ and re-fit, or project coefficients toward a contractive set.
5. Propagate seeds to generate u_n and plot u_n alongside the original sine to visualize echo behavior.

2.6 Python snippet (runnable)

Below is a short Python snippet you can paste into a Jupyter notebook or run locally; it performs the fit, checks spectral radius, and plots the original samples and UNNS continuation. (Requires `numpy`, `scipy`, `matplotlib`.)

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.linalg import eigvals
from scipy.linalg import toeplitz

# Parameters
Fs = 100.0      # sampling rate
f = 5.0         # sine frequency
N = 80          # sample length
r = 6           # recurrence order
lam = 1e-4      # Tikhonov regularization

# Build sample
k = np.arange(N)
d = np.sin(2*np.pi*(f/Fs)*k)

# Build A and b for least squares
rows = N - r
A = np.zeros((rows, r))
b = np.zeros(rows)
for i in range(rows):
```

```
    A[i,:] = d[i + r - 1 : i - 1 : -1]  # d_{n-1}, d_{n-2},...,d_{n-r}
    b[i] = d[i + r]

# Solve regularized least squares
AtA = A.T @ A
ct = np.linalg.solve(AtA + lam*np.eye(r), A.T @ b)

# Companion matrix
C = np.zeros((r,r))
C[0,:] = ct
for i in range(1,r):
    C[i,i-1] = 1.0

rho = max(abs(eigvals(C)))
print("Fitted coefficients:", ct)
print("Spectral radius:", rho)

# Propagate
seeds = d[-r:].tolist()
U = seeds.copy()
Mprop = 200
for m in range(Mprop):
    next_val = sum(ct[j]*U[-j-1] for j in range(r))
    U.append(next_val)

# Plot
plt.figure(figsize=(10,4))
plt.plot(np.arange(N), d, 'r.-', label='Samples (inlet)')
plt.plot(np.arange(N, N+Mprop), U[N:], 'c.-', label='UNNS continuation')
plt.legend(); plt.title('Sine inletting and UNNS propagation'); plt.show()
```

2.7 Interpretation of typical outputs

- **If** $\rho(C) < 1$**:** the fitted recurrence is contractive — the UNNS continuation will damp and echoes decay. This is a stable inletting.
- **If** $\rho(C) \approx 1$**:** the UNNS continuation may preserve oscillatory amplitude (marginal stability) — appropriate for lossless wave-like behavior.
- **If** $\rho(C) > 1$**:** the continuation grows — echoes amplify and the inletting is unstable unless renormalized. Address by increasing λ, projecting coefficients into contractive regions, or applying global renormalization.

2.8 Visual diagnostics and recommended display

For an interactive demo (web or notebook) present:

- Overlaid plot: original sample (red) vs UNNS continuation (cyan).
- Spectral plot: eigenvalues of C on the complex plane; show unit circle and mark $\rho(C)$.

- Residual trace: $|\epsilon_m|$ vs m.
- UPI proxy: a single number derived from growth factors or $\rho(C)$.

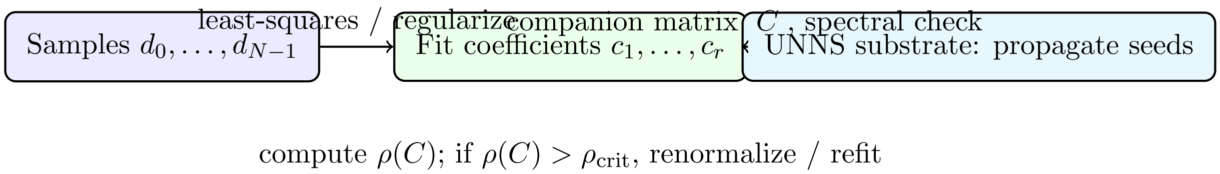

Figure 1: Schematic: sine samples → fit recurrence → UNNS propagation.

3 Closing remarks

This worked example demonstrates a practical inletting pipeline: sample a finite external signal, fit a recurrence, validate stability via spectral radius, and then propagate to generate UNNS echoes. The same pipeline generalizes: use different D (e.g. sampled Maxwell fields, prime indicator sequences), choose fitting constraints (algebraic integer projection), and integrate with repair/renormalization operators when needed.