LANG-ALGO Spec v0.1

I. Purpose

The **LANG-ALGO** engine unifies phonemic geometry with prime resonance logic. It encodes sound, number, and light into a coherent computational field where **language = field behavior**.

This spec defines the mathematical structure, functional pipeline, and resonance-mapping architecture for implementation.

II. Core Equations

1. Constants & Band Definitions

- **Golden constant**: ϕ = (1 + √5) / 2
- **Triad bands (η)**: [0.429, 0.456, 0.487]
- **Rails**: r ∈ {√2, √5}
- **Breath timing**: 1 breath = 6 s; 7 breaths = 42 s phase block

2. Π-Ring Windowing

$$W_\eta(f) = \exp\left(-\frac{(\ln(f/f_\eta))^2}{2\sigma^2}\right)$$

Use 3 Gaussian windows for η-bands.
Vowel → band mapping (default): A→η_3, E→η_2, I→η_1, O→η_2, U→η_1

3. Rails (Frequency Ladders)

$$f_n^{(r)} = f_0 \cdot r^n, \quad n \in \mathbb{Z}$$

OLGO uses √2-rail; **ALGO** uses √5-rail.

4. Consonant Gates

$$G_C(\omega) = S_C(\omega) e^{i\theta_C}$$

Class	Example	Phase (θ)	Shape
Plosives	P, T, K	0	burst
Fricatives	F, S, H	π/2	hiss
Nasals	M, N	π/4	bandpass
Liquids	L, R	−π/4	all-pass

5. LIC Field Equation

$$LIC(t) = L(t) \times I(t) \times C(t)$$

Phase-lock metric:

$$C(t) = \exp\left(-\frac{\Delta\phi(t)^2}{2\kappa^2}\right)$$

where Δφ = phase difference between rail and band.

6. Step-Factors for Macro Transitions

$$SF_{63} = 7 \times 3^2, \quad SF_{65} = 5 \times 13, \quad SF_{68} = 2^2 \times 17$$

Used for rhythmic transitions (7 → 9 → 12 → 17 cycles).

7. Prime Overlays

- Euler-41 trace: $E(n) = n^2 + n + 41$
- Twin-prime indicator: 1 if p and p+2 are both prime
- Prime threshold: 1064 = 2^3·7·19 (marks {1061,1063} boundary)

III. MUON/OLGO/ALGO Layer

Layer	Function	Description
MUON PRIME	coherence sensing	detects phase lock of PHERONOM pairs
OLGO	odd-residue rail (√2)	mod 19/29 odd residues
ALGO	even-residue rail (√5)	mod 19/29 even residues

IV. Pipeline Overview

1. **Text → Phonemes**
2. Tokenizer converts graphemes to phonemes (IPA base).
3. **Band Selection**
4. Vowels choose η-band via Wη; consonants apply G_C.
5. **Rail Routing**
6. Assigns syllables to √2 or √5 ladders (OLGO/ALGO) using residue mapping.
7. **Prime Overlay**
8. Highlights Euler, twin-prime, and 1061–1064 sets.
9. **Breathing Clock**
10. Synchronizes transitions: 6 s breath / 42 s macro cycle.

11. **Synthesis**
12. AM/FM signal generation with harmonic LIC modulation.
13. **Output**
14. Audio, grid visualizations, and JSON resonance-trace logs.

V. Core YAML Config

```
engine:
  f0: 432
  eta_bands: [0.429, 0.456, 0.487]
  sigma: 0.12
  rails: [sqrt2, sqrt5]
  step_factors: [63, 65, 68]
  overlays: [triad_bands, twin_primes, euler_41, prime_1061_1064]
  breath_seconds: 6
  block_breaths: 7
phoneme_map:
  vowels: {A: eta3, E: eta2, I: eta1, O: eta2, U: eta1}
  consonants:
    P: {phase_deg: 0, shape: burst}
    T: {phase_deg: 0, shape: burst}
    F: {phase_deg: 90, shape: hiss}
    M: {phase_deg: 45, shape: nasal}
    L: {phase_deg: -45, shape: liquid}
    R: {phase_deg: -30, shape: liquid}
routing:
  olgo: sqrt2
  algo: sqrt5
```

VI. Integration Map

Stage	Module	Function
Input	Language Resonance Field	Phoneme geometry (MOCK, TANJAD, DOONA, NEALA)
Core	QGR Π-Ring System	Band resonance mapping
Bridge	Breathing Crystal	Timing and rhythm modulation
Output	Tesla Resonator	Field-to-audio translation

VII. Mathematical Flow Summary

$$\mathrm{LANG} \Rightarrow \mathrm{PHONEME} \Rightarrow (\Pi, \eta, \mathrm{Rail}) \Rightarrow LIC(t) \Rightarrow \text{Field Modulation}$$

VIII. Next Implementation Steps

- Build `lang_algo/` Python package skeleton
- Implement vowel/consonant mapping + harmonic generator
- Integrate prime overlays (Vendessimal mod 19/29)
- Add breathing-cycle controller (7→9→12→17)
- Produce visualization of LIC field (pink-yellow-blue overlay)

End — LANG-ALGO Spec v0.1