NIV Engine v6 — Code-to-Formula Mapping and Verification Summary

Diren Kumaratilleke

2025

Purpose

This document provides a full one-to-one mapping between the mathematical formulation of the **National Impact Velocity (NIV) Engine v6** and its corresponding Python implementation. It verifies that each symbol, transformation, and calibration parameter is reproducibly encoded in the source scripts (`niv_make.py`, `niv_visualization_v6.py`, and `fred_fetch.py`).

—

1. Core Equation Mapping

Concept	Mathematical Definition	Python Implementation
National Impact Velocity	$NIV_t = \frac{u_t P_t^2}{(X_t + F_t)^\eta}$	`NIV = (u * (P**2)) / ((X + F)**eta)`
Activation Intensity	$u_t = \tanh(\alpha_1 dG_t + \alpha_2 dA_t - \alpha_3 dr_t)$	`u = np.tanh(a1*dG + a2*dA - a3*dr)`
Regeneration Share	$P_t = \frac{I_t + R_t + E_t}{GDP_t}$	`P = (I + R + E) / GDP`
Idle Capacity	$X_t = 1 - \frac{TCU_t}{100}$	`X = 1 - (TCU / 100)`
Aggregate Friction	$F_t = \beta_1 s_t + \beta_2(r_t - \pi_t) + \beta_3 \sigma_{r,t}$	`F = b1*s + b2*(r - pi) + b3*sigma_r`
Term Spread	$s_t = DGS10_t - TB3MS_t$	`s = DGS10 - TB3MS`
Inflation	$\pi_t = \frac{CPI_t - CPI_{t-12}}{CPI_{t-12}}$	`pi = (CPI - CPI.shift(12)) / CPI.shift(12)`
Rate Volatility	$\sigma_{r,t} = std_{12}(r_t)$	`sigma_r = FEDFUNDS.rolling(12).std()`

—

2. Derived Indicators

Metric	Mathematical Definition	Python Implementation
Liquidity Stress Intensity (LSI)	$LSI_t = \frac{\sigma_{12}(NIV)}{\lvert\mu_{12}(NIV)\rvert + \varepsilon}$	`LSI = NIV.rolling(12).std() / (abs(NIV.rolling(12).mean()) + eps)`
Structural Drag	$Drag_t = 1 - \frac{NIV_t}{\mu_{24}(NIV)}$	`Drag = 1 - (NIV / NIV.rolling(24).mean())`
Impulse (Change in NIV)	$\Delta NIV_t = NIV_t - NIV_{t-1}$	`Impulse = NIV.diff()`

—

3. Calibration Parameters

Parameter	Symbol	Default	Function
Fiscal Sensitivity	α_1	1.0	Responsiveness to investment growth (dG_t).
Monetary Sensitivity	α_2	1.0	Responsiveness to liquidity growth (dA_t).
Rate Resistance	α_3	0.7	Damping effect of interest rate increases.
Spread Friction Weight	β_1	0.4	Sensitivity to yield curve slope.
Real Rate Friction Weight	β_2	0.4	Sensitivity to real interest burdens.
Volatility Friction Weight	β_3	0.2	Sensitivity to rate instability.
Friction Sensitivity	η	1.0	Governs NIV's response to total friction.
Numerical Tolerance	ε	10^{-6}	Stabilizes denominators in ratio computations.

—

4. Rolling Computations and Spectral Windows

Diagnostic	Window Length	Python Operation
LSI Rolling Std	12 months	`NIV.rolling(12).std()`
LSI Rolling Mean	12 months	`NIV.rolling(12).mean()`
Drag Rolling Mean	24 months	`NIV.rolling(24).mean()`
Inflation Lag	12 months	`CPI.shift(12)`
Rate Volatility	12 months	`FEDFUNDS.rolling(12).std()`

—

5. Verification Statement

All computations are fully reproducible under Python 3.11 using the following libraries:

```
pandas >= 2.0
```

```
numpy >= 1.25
matplotlib >= 3.8
fredapi >= 0.5
```

The numerical outputs generated by the engine correspond exactly to the mathematical forms specified here, within rounding error. Every term has been directly tested for sign, range, and dimensional coherence.

—

6. Author's Note

The NIV Engine v6 forms the computational backbone of the Capital Velocity Economics (CVE) framework. It unites activation, regeneration, and frictional components into a live economic throughput measure — a dynamic gauge of how capital truly moves, regenerates, and decays within a system.

Diren Kumaratilleke
Chapel Hill, 2025