A Deterministic Framework for 5-Dimensional Coupled Dynamical Systems:
Theory, Algorithms, and Implementation Specification

Sebastian Klemm

October 22, 2025

Abstract

This document provides a complete, deterministic, and domain-agnostic mathematical framework for simulating coupled dynamical systems in five dimensions. The specification is designed to be unambiguous and directly implementable by any computational agent. We define the state space, coupling mechanisms, evolution operators, numerical integration schemes, stability analysis, and visualization projections with mathematical rigor. The framework is equipped with a template system allowing instantiation for specific domains while maintaining mathematical consistency.

Contents

1 Introduction and Motivation

1.1 Purpose

Real-world systems often involve **five or more strongly coupled variables** that evolve nonlinearly over time. Traditional analysis methods (spreadsheets, 2D plots, intuition) fail when:

- Dimensionality exceeds human visualization capacity ($n \geq 4$)
- Couplings are nonlinear and cross-dimensional
- Temporal evolution exhibits complex attractors or bifurcations
- Multiple scenarios must be compared quantitatively

This framework provides a **computational microscope** for such systems, enabling:

1. Precise numerical integration of 5D dynamics
2. Projection to 3D for visualization
3. Stability and bifurcation analysis
4. Ensemble exploration of parameter spaces
5. Domain-agnostic application via templates

1.2 Design Principles

1. **Determinism:** Given identical inputs, any implementation must produce identical outputs.
2. **Completeness:** All mathematical objects are fully specified; no ambiguities remain.
3. **Domain-agnosticism:** The core framework makes no assumptions about application domain.
4. **Extensibility:** Templates allow domain-specific instantiation without modifying the core.
5. **Implementability:** The specification is directly translatable to code.

2 Mathematical Foundations

2.1 State Space

Definition 2.1 (State Vector)**.** *The system state at time $t \in \mathbb{R}_{\geq 0}$ is represented by a vector*

$$\sigma(t) = \begin{pmatrix} \sigma_1(t) \\ \sigma_2(t) \\ \sigma_3(t) \\ \sigma_4(t) \\ \sigma_5(t) \end{pmatrix} \in \mathbb{R}^5 \tag{1}$$

where each component $\sigma_i(t) \in \mathbb{R}$ represents a system variable.

Definition 2.2 (State Space)**.** *The state space is the Euclidean space*

$$\mathcal{M} = \mathbb{R}^5 \tag{2}$$

equipped with the standard Euclidean metric

$$d(\sigma, \sigma') = \left\|\sigma - \sigma'\right\|_2 = \sqrt{\sum_{i=1}^{5} (\sigma_i - \sigma_i')^2} \tag{3}$$

2.2 Variable Naming and Semantics

Definition 2.3 (Variable Labels)**.** *Each component σ_i is associated with a human-readable label $\ell_i \in \mathcal{L}$ where $\mathcal{L}$ is the set of valid identifier strings. The mapping*

$$L : \{1, 2, 3, 4, 5\} \to \mathcal{L}, \quad i \mapsto \ell_i \tag{4}$$

assigns semantic meaning to components for a specific domain instantiation.

Implementation note: Labels are metadata and do not affect computation.

Definition 2.4 (Dimensionless Normalization)**.** *All variables are assumed to be normalized to dimensionless quantities in $\mathbb{R}$. Dimensional analysis and unit conversion are handled externally to the core framework.*

3 System Dynamics

3.1 General Evolution Equation

Definition 3.1 (First-Order Autonomous System)**.** *The temporal evolution of the state vector is governed by*

$$\frac{d\sigma}{dt} = F(\sigma), \quad \sigma(0) = \sigma_0 \tag{5}$$

where $F : \mathbb{R}^5 \to \mathbb{R}^5$ is a continuously differentiable vector field.

3.2 Coupling Matrix

Definition 3.2 (Linear Coupling Matrix)**.** *The coupling between variables is encoded in a matrix $C \in \mathbb{R}^{5\times5}$ where*

$$C_{ij} = \textit{strength of influence from variable } j \textit{ on variable } i \tag{6}$$

The diagonal elements C_{ii} represent self-dynamics.

Definition 3.3 (Coupling Types)**.** *Each coupling C_{ij} is associated with a coupling type $\tau_{ij} \in \mathcal{T}$ where*

$$\mathcal{T} = \{\mathit{LINEAR}, \mathit{QUADRATIC}, \mathit{EXPONENTIAL}, \mathit{SIGMOID}, \mathit{CUSTOM}\} \tag{7}$$

Definition 3.4 (Coupling Function)**.** *The effective coupling from σ_j to $\frac{d\sigma_i}{dt}$ is computed via*

$$g_{ij}(\sigma_j) = \begin{cases} C_{ij} \cdot \sigma_j & \textit{if } \tau_{ij} = \mathit{LINEAR} \\ C_{ij} \cdot \sigma_j^2 & \textit{if } \tau_{ij} = \mathit{QUADRATIC} \\ C_{ij} \cdot e^{\sigma_j} & \textit{if } \tau_{ij} = \mathit{EXPONENTIAL} \\ C_{ij} \cdot \frac{1}{1+e^{-\sigma_j}} & \textit{if } \tau_{ij} = \mathit{SIGMOID} \\ h_{ij}(\sigma_j) & \textit{if } \tau_{ij} = \mathit{CUSTOM} \end{cases} \tag{8}$$

where $h_{ij} : \mathbb{R} \to \mathbb{R}$ is a user-defined function for custom couplings.

3.3 Vector Field Definition

Definition 3.5 (Coupled Vector Field)**.** *The complete vector field is defined as*

$$F_i(\sigma) = \sum_{j=1}^{5} g_{ij}(\sigma_j) + b_i, \quad i = 1, \ldots, 5 \tag{9}$$

where $b_i \in \mathbb{R}$ is a bias term for component i.

Complete specification:

$$\frac{d\sigma_i}{dt} = \sum_{j=1}^{5} g_{ij}(\sigma_j) + b_i \tag{10}$$

4 Numerical Integration

4.1 Time Discretization

Definition 4.1 (Time Grid). *Time is discretized into a uniform grid*

$$t_n = n \cdot \Delta t, \quad n \in \mathbb{N}_0 \tag{11}$$

where $\Delta t > 0$ is the time step. We denote $\sigma^n = \sigma(t_n)$.

4.2 Integration Scheme

Definition 4.2 (Semi-Implicit Heun Method). *Given σ^n, the next state σ^{n+1} is computed via:*

$$\tilde{\sigma}^{n+1} = \sigma^n + \Delta t \cdot F(\sigma^n) \tag{12}$$

$$\sigma^{n+1} = \sigma^n + \frac{\Delta t}{2}\left(F(\sigma^n) + F(\tilde{\sigma}^{n+1})\right) \tag{13}$$

This is a second-order Runge-Kutta method (Heun's method).

Theorem 4.1 (Local Truncation Error). *The local truncation error of the Heun method is $\mathcal{O}(\Delta t^3)$, and the global error is $\mathcal{O}(\Delta t^2)$.*

Proposition 4.1 (Stability Condition). *For stability, the time step must satisfy*

$$\Delta t \leq \frac{2}{\lambda_{\max}(J)} \tag{14}$$

where $\lambda_{\max}(J)$ is the maximum absolute eigenvalue of the Jacobian $J = \nabla F$.

Default parameter: $\Delta t = 0.01$ (adjustable based on system stiffness).

5 Stability and Bifurcation Analysis

5.1 Fixed Points

Definition 5.1 (Fixed Point). *A state $\sigma^* \in \mathbb{R}^5$ is a fixed point if*

$$F(\sigma^*) = 0 \tag{15}$$

Algorithm 5.1 (Fixed Point Detection). *During simulation, a state σ^n is considered a numerical fixed point if*

$$\|F(\sigma^n)\|_2 < \epsilon_{\textit{fixed}}, \quad \epsilon_{\textit{fixed}} = 10^{-6} \tag{16}$$

5.2 Jacobian and Stability

Definition 5.2 (Jacobian Matrix). *The Jacobian of the vector field at σ is*

$$J(\sigma) = \nabla F(\sigma) = \begin{pmatrix} \frac{\partial F_1}{\partial \sigma_1} & \cdots & \frac{\partial F_1}{\partial \sigma_5} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_5}{\partial \sigma_1} & \cdots & \frac{\partial F_5}{\partial \sigma_5} \end{pmatrix} \in \mathbb{R}^{5\times 5} \tag{17}$$

Definition 5.3 (Analytical Jacobian for Standard Couplings). *For the coupling functions defined in Eq.* (8)*, the partial derivatives are:*

$$\frac{\partial F_i}{\partial \sigma_j} = \begin{cases} C_{ij} & \textit{if } \tau_{ij} = \textit{LINEAR} \\ 2C_{ij} \cdot \sigma_j & \textit{if } \tau_{ij} = \textit{QUADRATIC} \\ C_{ij} \cdot e^{\sigma_j} & \textit{if } \tau_{ij} = \textit{EXPONENTIAL} \\ C_{ij} \cdot \frac{e^{-\sigma_j}}{(1+e^{-\sigma_j})^2} & \textit{if } \tau_{ij} = \textit{SIGMOID} \\ \frac{\partial h_{ij}}{\partial \sigma_j}(\sigma_j) & \textit{if } \tau_{ij} = \textit{CUSTOM} \end{cases} \tag{18}$$

Definition 5.4 (Stability Classification)**.** *Let $\{\lambda_1, \ldots, \lambda_5\}$ be the eigenvalues of $J(\sigma^*)$ at a fixed point. The fixed point is classified as:*

- ***Stable*** *if $\Re(\lambda_i) < 0$ for all i*
- ***Unstable*** *if $\exists i : \Re(\lambda_i) > 0$*
- ***Marginal*** *if $\max_i \Re(\lambda_i) = 0$ and no $\Re(\lambda_i) > 0$*

5.3 Lyapunov Exponents

Definition 5.5 (Maximal Lyapunov Exponent)**.** *The maximal Lyapunov exponent λ_{max} quantifies sensitivity to initial conditions:*

$$\lambda_{max} = \lim_{t \to \infty} \frac{1}{t} \ln \frac{\|\delta\sigma(t)\|}{\|\delta\sigma(0)\|} \tag{19}$$

where $\delta\sigma(t)$ is the deviation from a reference trajectory.

Algorithm 5.2 (Numerical Lyapunov Exponent)**.** *Computed via:*

Initialize $\delta\sigma_0$ small perturbation
for $n = 0, \ldots, N-1$ ***do***
    Evolve σ^{n+1} and $(\sigma + \delta\sigma)^{n+1}$
    $\delta\sigma^{n+1} = (\sigma + \delta\sigma)^{n+1} - \sigma^{n+1}$
    $\lambda_n = \frac{1}{\Delta t} \ln \frac{\|\delta\sigma^{n+1}\|}{\|\delta\sigma^n\|}$
    Renormalize $\delta\sigma^{n+1}$ to prevent overflow
end for
$\lambda_{max} \approx \frac{1}{N\Delta t} \sum_{n=0}^{N-1} \lambda_n$

6 Visualization and Projection

6.1 Dimension Reduction

Definition 6.1 (Projection Operator)**.** *A projection operator* $\Pi : \mathbb{R}^5 \to \mathbb{R}^3$ *maps the 5D state to 3D visualization space:*

$$\mathbf{x} = \Pi(\sigma) = P \cdot \sigma \tag{20}$$

where $P \in \mathbb{R}^{3\times 5}$ *is the projection matrix and* $\mathbf{x} = (x, y, z)^T$*.*

6.2 Standard Projections

Definition 6.2 (Orthogonal Projection)**.** *Projects onto the first three components:*

$$P_{orth} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{21}$$

Definition 6.3 (Isometric Projection)**.** *Preserves relative magnitudes of all dimensions:*

$$P_{iso} = \frac{1}{\sqrt{5}} \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 0 \\ 1 & 0 & -1 & 0 & 1 \end{pmatrix} \tag{22}$$

This matrix has orthonormal rows.

Definition 6.4 (Principal Component Projection)**.** *Let* $\Sigma = \{\sigma^1, \ldots, \sigma^N\}$ *be a trajectory. Compute:*

1. *Mean:* $\bar{\sigma} = \frac{1}{N} \sum_{n=1}^{N} \sigma^n$
2. *Covariance:* $\mathcal{C} = \frac{1}{N-1} \sum_{n=1}^{N} (\sigma^n - \bar{\sigma})(\sigma^n - \bar{\sigma})^T$
3. *Eigendecomposition:* $\mathcal{C} = V \Lambda V^T$ *where* $\Lambda = diag(\lambda_1, \ldots, \lambda_5)$ *with* $\lambda_1 \geq \cdots \geq \lambda_5$
4. *Projection:* $P_{PCA} = V_{1:3}^T$ *(first 3 eigenvectors)*

6.3 Color Mapping

Definition 6.5 (Trajectory Coloring)**.** *Each point* σ^n *on a trajectory is mapped to a color via:*

$$color(\sigma^n) = HSV(h(\sigma^n), s(\sigma^n), v(\sigma^n)) \tag{23}$$

where $h, s, v : \mathbb{R}^5 \to [0, 1]$ *are hue, saturation, and value functions.*

Definition 6.6 (Standard Color Schemes)**.**

- ***Speed-based:*** $h(\sigma) = \frac{\|F(\sigma)\|}{\max_n \|F(\sigma^n)\|}$
- ***Stability-based:*** $h(\sigma) = \frac{\lambda_{\max}(J(\sigma))+1}{2}$
- ***Time-based:*** $h(n) = (n \mod N)/N$

7 Template System for Domain Instantiation

7.1 Template Structure

Definition 7.1 (System Template)**.** *A template $\mathcal{T}$ is a tuple*

$$\mathcal{T} = (L, C, \tau, b, \sigma_0, \Delta t, T_{max}) \tag{24}$$

where:

- $L : \{1, \ldots, 5\} \to \mathcal{L}$ *assigns variable names*
- $C \in \mathbb{R}^{5\times5}$ *is the coupling matrix*
- $\tau \in \mathcal{T}^{5\times5}$ *specifies coupling types*
- $b \in \mathbb{R}^5$ *is the bias vector*
- $\sigma_0 \in \mathbb{R}^5$ *is the default initial condition*
- $\Delta t \in \mathbb{R}_{>0}$ *is the recommended time step*
- $T_{max} \in \mathbb{R}_{>0}$ *is the recommended simulation duration*

7.2 Example Templates

7.2.1 Epidemiological SIR Model

Definition 7.2 (SIR Template)**.** *Consider a population with Susceptible (S), Infected (I), Recovered (R) compartments, plus Vaccination rate (V) and Mobility (M).*

Variable labels:

$$L = \{\mathit{Susceptible}, \mathit{Infected}, \mathit{Recovered}, \mathit{Vaccination}, \mathit{Mobility}\} \tag{25}$$

Coupling matrix and types:

$$C = \begin{pmatrix} -0.3 & -0.5 & 0 & 0.2 & -0.1 \\ 0.3 & -0.2 & 0 & -0.1 & 0.15 \\ 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & -0.05 \end{pmatrix} \tag{26}$$

$$\tau = \begin{pmatrix} \mathit{LIN} & \mathit{QUAD} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} \\ \mathit{QUAD} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} \\ \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} \\ \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} \\ \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} \end{pmatrix} \tag{27}$$

Dynamics interpretation:

$$\frac{dS}{dt} = -0.3S - 0.5SI + 0.2V - 0.1M \tag{28}$$

$$\frac{dI}{dt} = 0.3SI - 0.2I - 0.1V + 0.15M \tag{29}$$

$$\frac{dR}{dt} = 0.2I \tag{30}$$

$$\frac{dV}{dt} = 0 \quad \textit{(external parameter)} \tag{31}$$

$$\frac{dM}{dt} = 0.1I - 0.05M \tag{32}$$

Parameters:

$$b = (0, 0, 0, 0, 0)^T, \quad \sigma_0 = (0.99, 0.01, 0, 0.1, 0.5)^T, \quad \Delta t = 0.1, \quad T_{max} = 100 \tag{33}$$

7.2.2 Financial Market Model

Definition 7.3 (Market Template)**.** *Variables: Interest Rate (r), Inflation (π), Risk Appetite (ρ), Liquidity (L), Market Confidence (C).*

Variable labels:

$$L = \{\mathit{Interest}, \mathit{Inflation}, \mathit{Risk}, \mathit{Liquidity}, \mathit{Confidence}\} \tag{34}$$

Coupling matrix and types:

$$C = \begin{pmatrix} -0.1 & 0.5 & -0.2 & 0.3 & 0.1 \\ -0.3 & -0.1 & 0.2 & -0.1 & -0.2 \\ -0.4 & 0.2 & -0.15 & 0.3 & 0.5 \\ 0.2 & -0.3 & 0.1 & -0.2 & 0.4 \\ -0.3 & -0.4 & 0.3 & 0.2 & -0.1 \end{pmatrix} \tag{35}$$

$$\tau = \begin{pmatrix} \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} \\ \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} \\ \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{SIGM} \\ \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} & \mathit{LIN} \\ \mathit{LIN} & \mathit{LIN} & \mathit{SIGM} & \mathit{LIN} & \mathit{LIN} \end{pmatrix} \tag{36}$$

Parameters:

$$b = (0, 0, 0, 0, 0)^T, \quad \sigma_0 = (0.05, 0.02, 0.3, 0.7, 0.6)^T, \quad \Delta t = 0.05, \quad T_{max} = 50 \tag{37}$$

7.2.3 Predator-Prey Ecosystem

Definition 7.4 (Lotka-Volterra Template)**.** *Variables: Prey (N), Predator (P), Resource (R), Climate (T), Disturbance (D).*

Coupling matrix:

$$C = \begin{pmatrix} 0.2 & -0.1 & 0.3 & 0.1 & -0.2 \\ 0.15 & -0.05 & -0.1 & -0.05 & 0 \\ -0.1 & -0.05 & -0.1 & 0.2 & -0.15 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1 & 0 & -0.3 \end{pmatrix} \tag{38}$$

All couplings are LINEAR except:

- $\tau_{12} = \mathit{QUADRATIC}$ *(prey-predator interaction)*
- $\tau_{21} = \mathit{QUADRATIC}$ *(predator-prey interaction)*

Parameters:

$$b = (0, 0, 0.1, 0, 0)^T, \quad \sigma_0 = (0.5, 0.2, 0.8, 0.5, 0.1)^T, \quad \Delta t = 0.05, \quad T_{max} = 100 \tag{39}$$

8 Implementation Algorithms

8.1 Main Simulation Loop

Algorithm 1 Deterministic 5D System Simulation

Require: Template $\mathcal{T} = (L, C, \tau, b, \sigma_0, \Delta t, T_{\max})$
Require: Projection type $p \in \{\text{ORTH}, \text{ISO}, \text{PCA}\}$
Ensure: Trajectory $\{\sigma^n\}_{n=0}^{N}$, Projected trajectory $\{\mathbf{x}^n\}_{n=0}^{N}$

$N \leftarrow \lfloor T_{\max}/\Delta t \rfloor$
$\sigma^0 \leftarrow \sigma_0$
Initialize arrays: $\Sigma \leftarrow [\sigma^0]$, $\mathbf{X} \leftarrow []$
for $n = 0, \ldots, N-1$ **do**
    Compute vector field:
    for $i = 1, \ldots, 5$ **do**
        $F_i^n \leftarrow b_i + \sum_{j=1}^{5} g_{ij}(\sigma_j^n)$ ▷ Using Eq. (8)
    end for
    Heun predictor:
    $\tilde{\sigma}^{n+1} \leftarrow \sigma^n + \Delta t \cdot F(\sigma^n)$
    Compute vector field at predictor:
    for $i = 1, \ldots, 5$ **do**
        $\tilde{F}_i^{n+1} \leftarrow b_i + \sum_{j=1}^{5} g_{ij}(\tilde{\sigma}_j^{n+1})$
    end for
    Heun corrector:
    $\sigma^{n+1} \leftarrow \sigma^n + \frac{\Delta t}{2}\left(F(\sigma^n) + F(\tilde{\sigma}^{n+1})\right)$
    Append σ^{n+1} to Σ
end for
Compute projection:
if $p = \text{ORTH}$ **then**
    $P \leftarrow P_{\text{orth}}$
else if $p = \text{ISO}$ **then**
    $P \leftarrow P_{\text{iso}}$
else if $p = \text{PCA}$ **then**
    $P \leftarrow \text{ComputePCA}(\Sigma)$ ▷ Algorithm 2
end if
for $n = 0, \ldots, N$ **do**
    $\mathbf{x}^n \leftarrow P \cdot \sigma^n$
    Append $\mathbf{x}^n$ to $\mathbf{X}$
end for
return $(\Sigma, \mathbf{X})$

8.2 PCA Projection Computation

Algorithm 2 Principal Component Analysis Projection

Require: Trajectory $\Sigma = \{\sigma^1, \ldots, \sigma^N\}$
Ensure: Projection matrix $P \in \mathbb{R}^{3\times 5}$
$N \leftarrow |\Sigma|$
$\bar{\sigma} \leftarrow \frac{1}{N}\sum_{n=1}^{N} \sigma^n$ ▷ Mean
Center data:
for $n = 1, \ldots, N$ **do**
    $\hat{\sigma}^n \leftarrow \sigma^n - \bar{\sigma}$
end for
Compute covariance matrix:
$\mathcal{C} \leftarrow \frac{1}{N-1}\sum_{n=1}^{N} \hat{\sigma}^n (\hat{\sigma}^n)^T$
Eigendecomposition:
$(\Lambda, V) \leftarrow \text{eig}(\mathcal{C})$ ▷ $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_5)$
Sort eigenvectors by $\lambda_1 \geq \cdots \geq \lambda_5$
$P \leftarrow V_{1:3}^T$ ▷ First 3 eigenvectors as rows
return P

8.3 Stability Analysis

Algorithm 3 Eigenvalue-Based Stability Classification

Require: State $\sigma \in \mathbb{R}^5$
Require: Coupling matrix C, types τ
Ensure: Stability class $\in \{\text{STABLE}, \text{UNSTABLE}, \text{MARGINAL}\}$
Compute Jacobian:
for $i = 1, \ldots, 5$ **do**
    for $j = 1, \ldots, 5$ **do**
        $J_{ij} \leftarrow \frac{\partial F_i}{\partial \sigma_j}$ ▷ Using coupling-specific derivatives
    end for
end for
Compute eigenvalues:
$\{\lambda_1, \ldots, \lambda_5\} \leftarrow \text{eig}(J)$
$\lambda_{\max} \leftarrow \max_i \Re(\lambda_i)$
if $\lambda_{\max} < -\epsilon_{\text{stable}}$ **then**
    return STABLE
else if $\lambda_{\max} > \epsilon_{\text{stable}}$ **then**
    return UNSTABLE
else
    return MARGINAL
end if
where $\epsilon_{\text{stable}} = 10^{-6}$

9 Ensemble and Parameter Exploration

9.1 Monte Carlo Ensemble

Definition 9.1 (Ensemble Simulation)**.** *To explore sensitivity to initial conditions, generate an ensemble of M trajectories with perturbed initial states:*

$$\sigma_0^{(m)} = \sigma_0 + \mathcal{N}(0, \epsilon^2 I_5), \quad m = 1, \ldots, M \tag{40}$$

where $\mathcal{N}(0, \epsilon^2 I_5)$ is a 5D Gaussian perturbation with standard deviation ϵ.

Default parameters: $M = 100$, $\epsilon = 0.05$.

Algorithm 4 Ensemble Simulation

Require: Template $\mathcal{T}$, Ensemble size M, Perturbation ϵ
Ensure: Set of trajectories $\{\Sigma^{(m)}\}_{m=1}^{M}$
1: **for** $m = 1, \ldots, M$ **do**
2: $\quad \sigma_0^{(m)} \leftarrow \sigma_0 + \mathcal{N}(0, \epsilon^2 I_5)$
3: $\quad \mathcal{T}^{(m)} \leftarrow \mathcal{T}$ with initial condition $\sigma_0^{(m)}$
4: $\quad \Sigma^{(m)} \leftarrow \text{Simulate}(\mathcal{T}^{(m)})$ ▷ Algorithm 1
5: **end for**
6: **return** $\{\Sigma^{(m)}\}_{m=1}^{M}$

9.2 Parameter Sweep

Definition 9.2 (Grid Search)**.** *To explore parameter sensitivity, define a grid over one or more coupling parameters:*

$$\mathcal{G} = \{C_{ij}^{(k)} \mid C_{ij}^{(k)} = C_{ij}^{\min} + k \cdot \Delta C,\ k = 0, \ldots, K-1\} \tag{41}$$

Simulate the system for each parameter value and record outcome metrics.

10 Output Specification and Data Formats

10.1 Trajectory Data

Definition 10.1 (Trajectory Record)**.** *For each time step n, record:*

- *Time:* $t_n = n \cdot \Delta t$
- *State:* $\sigma^n = (\sigma_1^n, \ldots, \sigma_5^n)$
- *Vector field:* $F(\sigma^n)$
- *Projection:* $\mathbf{x}^n = (x^n, y^n, z^n)$
- *Stability:* $s^n \in \{STABLE, UNSTABLE, MARGINAL\}$
- *Eigenvalues:* $\{\lambda_1^n, \ldots, \lambda_5^n\}$

10.2 CSV Export Format

Header:

```
time,s1,s2,s3,s4,s5,F1,F2,F3,F4,F5,x,y,z,stability,eig1_re,eig1_im,...
```

Data row format:

```
<t_n>,<sigma_1^n>,...,<sigma_5^n>,<F_1^n>,...,<F_5^n>,<x^n>,<y^n>,<z^n>,
<stability>,<Re(lambda_1)>,<Im(lambda_1)>,...
```

Numerical precision: All floating-point values must be represented with at least 10 significant digits to ensure reproducibility.

10.3 JSON Export Format

```
{
  "metadata": {
    "template_name": "SIR",
    "dt": 0.1,
    "T_max": 100,
    "projection": "ISO",
    "timestamp": "2025-10-22T00:00:00Z"
  },
  "variable_names": ["S", "I", "R", "V", "M"],
  "trajectory": [
    {
      "time": 0.0,
      "state": [0.99, 0.01, 0.0, 0.1, 0.5],
      "vector_field": [-0.005, 0.003, 0.002, 0.0, 0.001],
      "projection": [0.447, 0.089, 0.671],
      "stability": "STABLE",
      "eigenvalues": [
        {"real": -0.1, "imag": 0.0},
        ...
      ]
    },
    ...
  ]
}
```

11 Validation and Testing

11.1 Reference Solutions

Definition 11.1 (Unit Test Cases)**.** *Implementations must pass the following tests:*

11.1.1 Test 1: Linear Decoupled System

Setup:

$$C = -0.1 \cdot I_5, \quad \tau_{ij} = \text{LINEAR}, \quad b = 0, \quad \sigma_0 = (1, 1, 1, 1, 1)^T \tag{42}$$

Analytical solution:

$$\sigma_i(t) = e^{-0.1t}, \quad \forall i \tag{43}$$

Tolerance: At $t = 10$, require $|\sigma_i^{\text{num}} - e^{-1}| < 10^{-4}$ for all i.

11.1.2 Test 2: Harmonic Oscillator

Setup: (Using only σ_1, σ_2)

$$C = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \sigma_0 = (1, 0, 0, 0, 0)^T \tag{44}$$

Analytical solution:

$$\sigma_1(t) = \cos(t), \quad \sigma_2(t) = -\sin(t) \tag{45}$$

Tolerance: At $t = 2\pi$, require $|\sigma_1 - 1| < 10^{-3}$ and $|\sigma_2| < 10^{-3}$.

11.1.3 Test 3: Fixed Point Convergence

Setup:

$$C = -I_5, \quad b = (1, 1, 1, 1, 1)^T, \quad \sigma_0 = (0, 0, 0, 0, 0)^T \tag{46}$$

Expected behavior: Converge to fixed point $\sigma^* = (1, 1, 1, 1, 1)^T$.

Tolerance: At $t = 10$, require $\|\sigma^{\text{num}} - \sigma^*\| < 10^{-3}$.

11.2 Determinism Verification

Proposition 11.1 (Deterministic Guarantee)**.** *Given identical inputs (template, random seed, floating-point arithmetic), two independent implementations must produce bit-identical outputs.*

Test protocol:

1. Run simulation with fixed random seed
2. Export trajectory to CSV
3. Compute SHA-256 hash of CSV file
4. Compare hash across implementations

12 Implementation Guidelines

12.1 Numerical Considerations

- Use IEEE 754 double-precision (64-bit) floating-point arithmetic throughout
- For eigenvalue computation, use a standard LAPACK-based library (e.g., Eigen, LAPACK, NumPy)
- Matrix operations should follow standard linear algebra conventions (row-major or column-major consistently)
- Random number generation for ensembles must use a seedable generator (e.g., Mersenne Twister MT19937)

12.2 Performance Optimization

- Precompute and cache the coupling function type for each (i, j) pair
- Use SIMD vectorization where available for the main integration loop
- For large ensembles ($M > 1000$), parallelize over trajectories
- Store trajectories in contiguous memory for efficient I/O

12.3 Error Handling

- Check for NaN or Inf values at each time step; halt and report if detected
- Validate that Δt satisfies stability condition (warn if $\Delta t > 2/\lambda_{\max}$)
- Verify template consistency: C must be 5×5, τ must have valid types, etc.

13 Extensions and Future Work

13.1 Adaptive Time Stepping

Implement error-controlled adaptive time stepping using embedded Runge-Kutta methods (e.g., Dormand-Prince 5(4)).

13.2 Stochastic Dynamics

Extend to stochastic differential equations (SDEs):

$$d\sigma = F(\sigma)dt + G(\sigma)dW \tag{47}$$

where W is a 5D Wiener process and G is a diffusion matrix.

13.3 Higher-Order Methods

For stiff systems, consider implicit methods such as backward Euler or BDF.

13.4 Interactive Visualization

Develop a real-time 3D viewer that allows dynamic parameter adjustment and trajectory visualization.

14 Conclusion

This document provides a complete, deterministic, and mathematically rigorous specification for a 5-dimensional dynamical system framework. All components—from the state space definition to numerical integration, stability analysis, visualization, and template instantiation—are fully specified with sufficient detail to enable identical implementations by independent agents.

The framework is designed to be:

- **Deterministic:** Identical inputs yield identical outputs
- **Complete:** No mathematical objects are left undefined
- **Domain-agnostic:** Applicable across disciplines via templates
- **Extensible:** New coupling types and analysis methods can be added
- **Implementable:** Directly translatable to code in any language

By adhering to this specification, implementations will produce reliable, reproducible, and interpretable results for complex coupled systems that would otherwise require speculation or approximation.

A Mathematical Notation Reference

Symbol	Meaning
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathbb{R}_{\geq 0}$	Non-negative reals
$\mathbb{N}$	Natural numbers $\{0, 1, 2, \ldots\}$
$\|\cdot\|_2$	Euclidean norm
$\langle a, b \rangle$	Inner product
$\frac{\partial f}{\partial x}$	Partial derivative of f with respect to x
$\frac{df}{dt}$	Total derivative of f with respect to t
∇f	Gradient of f
I_n	$n \times n$ identity matrix
$\mathcal{O}(\cdot)$	Big-O notation (asymptotic complexity)
$\Re(\lambda)$	Real part of complex number λ
$\text{diag}(a_1, \ldots, a_n)$	Diagonal matrix with entries a_i

B Glossary of Terms

Attractor A set toward which a dynamical system evolves over time.

Bifurcation A qualitative change in system behavior as a parameter varies.

Coupling Interaction between two variables in the system.

Eigenvalue Scalar λ such that $Jv = \lambda v$ for some vector v.

Fixed Point State where $F(\sigma^*) = 0$; no further change occurs.

Jacobian Matrix of first partial derivatives of the vector field.

Lyapunov Exponent Measure of sensitivity to initial conditions.

Projection Mapping from high-dimensional space to lower-dimensional space.

Stability Property that small perturbations decay over time.

State Space Set of all possible states of the system.

Template Pre-configured system specification for a specific domain.

Trajectory Time-ordered sequence of states $\{\sigma(t)\}$.

Vector Field Function F that assigns a velocity vector to each state.

C Implementation Checklist

An implementation is considered complete if it satisfies:

- ✓ Defines 5D state vector $\sigma \in \mathbb{R}^5$
- ✓ Implements coupling matrix C and types τ
- ✓ Computes vector field $F(\sigma)$ via Eq. (9)
- ✓ Implements Heun method (Eqs. (12)–(13))
- ✓ Computes Jacobian J analytically
- ✓ Performs eigenvalue decomposition for stability
- ✓ Implements at least orthogonal and isometric projections
- ✓ Supports at least 3 templates (SIR, Financial, Predator-Prey)
- ✓ Exports CSV and JSON in specified formats
- ✓ Passes all 3 validation tests (Section 8)
- ✓ Produces deterministic outputs (same hash for same inputs)
- ✓ Handles NaN/Inf gracefully
- ✓ Documents all functions with mathematical references