Visualization of Set-valued Dynamical Systems

Nam Do, Jyri Ollanketo
Supervisor: Dr. Kalle Timperi

University of Oulu

Feb, 2026

What we are doing?

Computational and visualization toolbox

- Calculation backend for boundary map evolution of dynamical systems
- Visualization interface with smooth boundary map visualization

Dynamical Systems with Additive Bounded Noise

- Non-linear Dynamical Systems (e.g., Hénon Map, Duffing Map)
- Additive bounded noise with epsilon amplitude

Computational backend

Backend written in Rust to compute the evolution of dynamical systems with additive bounded noise. Features include:

- Calculation of the fixed and periodic orbits of the boundary map with support up to 8-step periodic orbits
- Iterative boundary evolution for dynamical systems
- Numerical approximation of unstable manifold for smooth Jordan curve visualization
- GAIO-based Ulam method calculation for approximating minimal invariant sets (MIS) and their dual repellers
- Fixed and periodic point classification: Unstable, Saddle, Stable

Visualization interface

Front-end provides an interactive interface for visualizing dynamical system evolutions in the web browser

- Smooth boundary evolution visualization from unstable manifold for dynamical systems
- Fixed and periodic points visualization
- Parameter sweeping for system state exploration
- Ulam Grid visualization for approximating the minimal invariant sets and their dual repellers
- Support pre-defined dynamical systems: Hénon Map and Duffing Map, and arbitrary user-defined dynamical systems
- Save visualized systems' states in PNG, and video format when parameter sweeping is enabled

Feature: Boundary Evolution Visualization

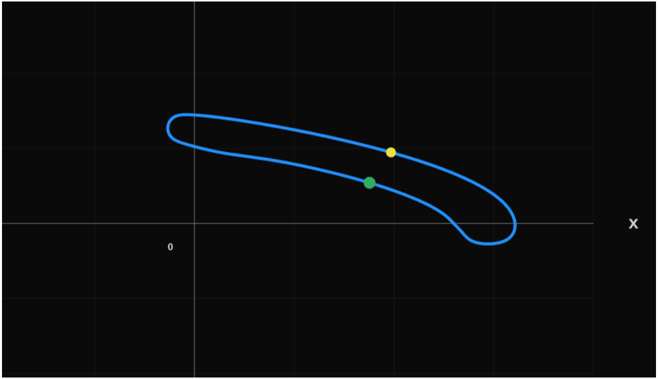
We implemented adaptive approximation of unstable manifold for saddle periodic points

- First, we precompute fixed and periodic points using Newton's method and Davidchack & Lai's Algorithm
- From a saddle periodic point, we construct an initial state from this position with some small perturbation in its unstable eigenvector direction
- Apply the $n-$fold composition E^n of the boundary map where n is the point's period to get the next mapped point
- When distance between two mapped boundary point is too far, we recursively add more points between them to fill the space
- The unstable manifold at the end is shown by a blue Jordan curve for the boundary map

Feature: Boundary Evolution Visualization (continued)

Visualization of the boundary evolution of the Hénon Map with $a = 0.4$, $b = 0.3$ **and** $\epsilon = 0.0625$

- Starts from the saddle (yellow) point and map forward until reach stable points (green)



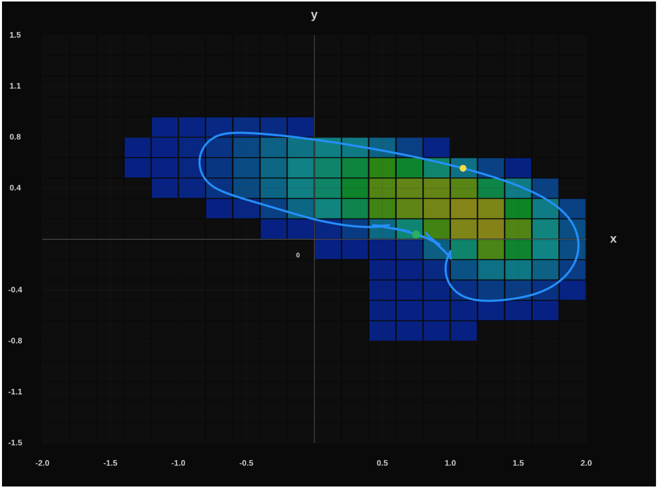
Feature: GAIO-based Ulam Grid Visualization

To approximate the evolution of a minimal invariant set together with its complementary domain of attraction, dual repeller. We proceed the GAIO-based Ulam method as follows:

- With Ulam method, we discretize (partition) the domain of interest into N boxes $B_1, B_2, ..., B_n$
- Build the probability transition matrix where P_{ij} is the probability that B_i maps to B_j
- For each box B_i, sample M points uniformly, map them forward through $p' = f(p)$, find all boxes that intersect with the $\epsilon-$ball: $B_\epsilon(p') = \{x : ||x - p'|| \leq \epsilon\}$, each intersecting box receives a transition count
- Left eigenvector: Solve $P^T v^l = v^l$. Boxes with ${v_i}^l > 0$ approximate the MIS.
- Right eigenvector (absorption probabilities): Computed via power method starting from $\hat{M}$. Boxes with $\hat{a}_i < 1$ approximate the dual repeller M^*.

Feature: GAIO-based Ulam Grid Visualization (continue)

Visualization of MIS using GAIO-based ULAM method with $a = 0.4, b = 0.3$ and $\epsilon = 0.2$, grid $= 20 * 20$ boxes

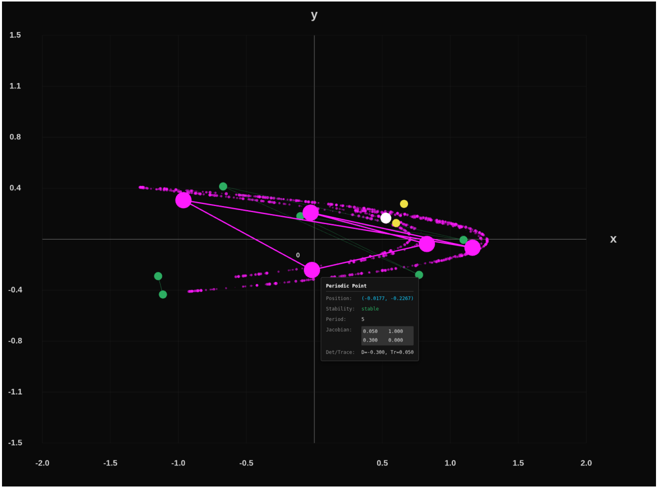


Feature: Fixed and periodic point stability; color classification

- To find fixed and periodic points for boundary map, we solve $E^n(x, y, n_x, n_y) - (x, y, n_x, n_y) = 0$ with $n \geq 1$
- For initial point, perform the grid search over (x, y, z) where $x \in [-2, 2], y \in [-2, 2], z \in [0, 2\pi]$, and construct $p_0 = (x_i, y_j, cos\theta_k, sin\theta_k)$
- We used the Davidchack-Lai algorithm for stabilized Newton iterations with $p_{k+1} = p_k + \Delta p$ where $\Delta p = (\beta||g||I - Dg)^{-1}g$
- Compute 4x4 Jacobian matrix of E^n by composing n individual Jacobians
- Terminate when $||\Delta p|| \leq 10^{-10}$ or max iterations reached

Feature: Fixed and periodic point stability; color classification (continued)

Visualize stability classification with green for stable, yellow for saddle and red for unstable, with orbit loop is in pink



Feature: Parameter Space Exploration

We implemented parameter sweeping to automatically compute the system evolution when one parameter is varying while others stay constant.

- Can choose one of the system parameters to automatically varying (e.g, a, b, ϵ)
- Specify the range r for the increment, decrease ($p \pm r$)
- Choose number of steps to run in the range
- Click "Play" button to run parameter sweeping for visualizing boundary evolution
- "Record" button available to record the boundary evolution when the parameter sweeping is running

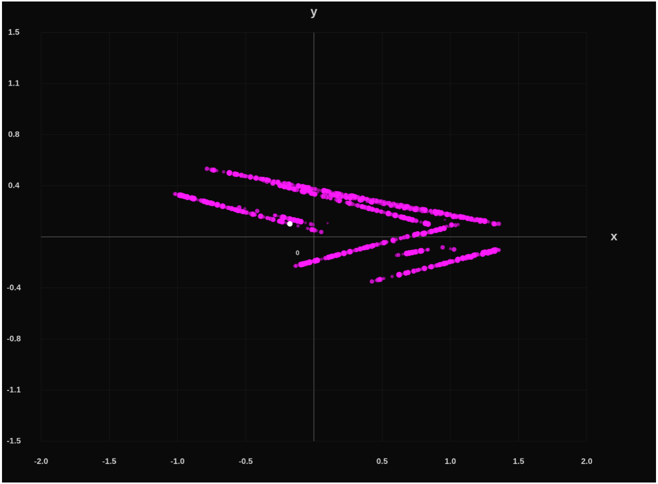
Feature: User-Defined Dynamical Systems

Besides the predefined Hénon Map and Duffing Map, users can arbitrarily enter system equations to visualize the dynamics

- Text input fields for entering system equations with symbolic expression
- Jacobian computed numerically via central differences; $(J^{-1})^T$ derived on-the-fly at each boundary point
- Boundary evolution, periodic orbit search, and unstable manifold are then computed automatically

Feature: User-Defined Dynamical Systems (continued)

Boundary evolution of a user-defined dynamical system (Lozi map) with $a = 1.4, b = 0.3, \epsilon = 0.095$



Future Direction

Our aim is to construct feature-rich toolbox for numerical analysis of set-valued dynamical systems under additive bounded noise. The list of features to be added include the following:

- The current system is limited to 2D visualization. We consider implementing the 3D visualization of the system
- Support for noise geometry other than a noise-ball.
- Cleaner and more intuitive UI for users and researchers.
- More efficient algorithms for backend calculation when slightly changing the system parameters (e.g, a, b, ϵ)
- Better visualizing a bifurcation when singular happens
- Visualize the domain of attraction and the dual repeller.