Protocolo de pruebas para el caso de estudio

Chua fraccionario no suave, candidatos Lur'e/Machado y comparación con Danca

Hidden Attractors Fractional Order

18 de mayo de 2026

Índice

1. Objetivo del documento

Este documento describe cómo ejecutar, documentar e interpretar todas las pruebas disponibles en el repositorio para el caso de estudio del sistema de Chua fraccionario no suave. No sustituye el reporte matemático general; su función es convertir la teoría y los módulos de código en una ruta de trabajo reproducible.

El caso de referencia externa es el atractor oculto reportado por Danca en el sistema de Chua no suave de orden fraccionario. La comparación debe hacerse con cautela: Danca usa integración Caputo mediante Adams–Bashforth–Moulton con historia completa, mientras que la ruta principal del repositorio usa EFORK con memoria truncada L_m. Por tanto, la comparación es geométrica, numérica y metodológica; no es identidad exacta de algoritmos.

2. Estructura operativa del repositorio

La versión de librería está en:

```
version_2/
```

La estructura mínima que se usa en este protocolo es:

Ruta	Papel dentro del protocolo
`hidden_attractors/`	Paquete importable. Incluye modelo, análisis, gráficas, clasificadores, workflows y backends nativos.
`examples/`	Ejemplos cortos para verificar instalación, equilibrios, candidatos y gráficas.
`tools/cli/`	Envolturas CLI mantenidas para workflows públicos.

Ruta	Papel dentro del protocolo
`tools/legacy/`	Scripts de investigación todavía útiles para búsqueda Lur'e, búsqueda sesgada, Machado, Danca y corridas históricas. No todo es API pública.
`outputs/`	Corridas, CSV, JSON, trayectorias, figuras y candidatos ya generados.
`configs/`	Configuraciones YAML para corridas largas o búsquedas extendidas.

Antes de ejecutar pruebas largas conviene instalar la librería en modo editable:

```
cd version_2
python -m pip install -e .
```

Para pruebas de desarrollo:

```
python -m pip install -e ".[dev]"
python -m pytest -q
```

Para habilitar anÃ¡lisis externos opcionales:

```
python -m pip install -e ".[analysis]"
```

3. Sistema de Chua fraccionario no suave

El sistema estudiado es

$$
\begin{aligned}
{}^{C}D_t^q x &= \alpha\,(y - x - f(x))\,, && (1)\\
{}^{C}D_t^q y &= x - y + z, && (2)\\
{}^{C}D_t^q z &= -\beta y - \gamma z, && (3)
\end{aligned}
$$

con $0 < q \leq 1$, y no linealidad no suave

$$
f(x) = m_1 x + \frac{m_0 - m_1}{2}\left(|x+1| - |x-1|\right). \tag{4}
$$

Equivalente en forma saturada,

$$
f(x) = m_1 x + (m_0 - m_1)\,\mathrm{sat}(x), \qquad \mathrm{sat}(x) = \begin{cases} -1, & x < -1,\\ x, & |x| \leq 1,\\ 1, & x > 1. \end{cases} \tag{5}
$$

La librería separa el campo vectorial del contrato fraccionario. El campo $F(X)$ se evalúa en Python mediante funciones como `rhs_piecewise`; el orden q, el paso h, la memoria L_m, el tiempo final T y el burn-in pertenecen al integrador o workflow.

3.1. Parámetros por defecto de la librería

En la parte pública de `version_2`, los parámetros por defecto del modelo no suave son

$$
\alpha = 8.4562, \qquad \beta = 12.0732, \qquad \gamma = 0.0052, \qquad m_0 = -0.1768, \qquad m_1 = -1.1468. \tag{6}
$$

Una forma directa de verlos es:

```
python examples/quickstart_equilibria.py
```

O desde Python:

```
from hidden_attractors import chua_piecewise_parameters

p = chua_piecewise_parameters()
print(p)
```

4. Ver la gráfica del atractor

4.1. Usar una trayectoria ya generada

La librería trabaja con trayectorias en formato CSV con columnas:

$$(t, x, y, z). \tag{7}$$

Ejemplo:

```
python examples/dynamical_analysis_gallery.py ^
  --trajectory-csv outputs/extended_search/
      machado_targeted_verification_lm10_20260515_182252/trajectories/
      branch_0_mu_4p00000_theta_0p00000_reference_attractor.csv ^
  --output-dir outputs/examples/chua_nonsmooth_gallery
```

En Linux/macOS:

```
python examples/dynamical_analysis_gallery.py \
  --trajectory-csv outputs/extended_search/
      machado_targeted_verification_lm10_20260515_182252/trajectories/
      branch_0_mu_4p00000_theta_0p00000_reference_attractor.csv \
  --output-dir outputs/examples/chua_nonsmooth_gallery
```

Este ejemplo genera, típicamente:

- `phase_space_3d.png`;
- `phase_projections.png`;
- `time_series.png`;
- `bifurcation_diagram.png`;
- `summary.json`;
- `bifurcation_points.csv`.

4.2. Código mínimo para graficar una trayectoria

```
from hidden_attractors.io import load_trajectory_csv
from hidden_attractors.plotting import (
    plot_phase_space,
    plot_phase_projections,
    plot_time_series,
)

traj = load_trajectory_csv("ruta/a/trayectoria.csv")

plot_phase_space(traj, "salidas/atractor_3d.png")
plot_phase_projections(traj, "salidas/proyecciones.png")
plot_time_series(traj, "salidas/series_temporales.png")
```

Lectura correcta: una gráfica de atractor muestra que la trayectoria es acotada y no trivial bajo el contrato numérico usado. No demuestra ocultedad.

5. Puntos de equilibrio

5.1. Derivación algebraica

En equilibrio se cumple

$$0 = \alpha(y - x - f(x)), \qquad 0 = x - y + z, \qquad 0 = -\beta y - \gamma z. \tag{8}$$

De la tercera ecuación,

$$z = -\frac{\beta}{\gamma} y, \tag{9}$$

y de la segunda,

$$z = y - x. \tag{10}$$

Entonces,

$$y - x = -\frac{\beta}{\gamma} y \quad \Longrightarrow \quad (\beta + \gamma) y = \gamma x \quad \Longrightarrow \quad y = \frac{\gamma}{\beta + \gamma} x. \tag{11}$$

Además, de la primera ecuación,

$$y = x + f(x). \tag{12}$$

Por tanto,

$$f(x) = y - x = \frac{\gamma}{\beta + \gamma} x - x = -\frac{\beta}{\beta + \gamma} x. \tag{13}$$

Si $x = 0$, se obtiene

$$E_0 = (0, 0, 0). \tag{14}$$

En las ramas exteriores, para $x > 1$,

$$f(x) = m_1 x + (m_0 - m_1), \tag{15}$$

y para $x < -1$,

$$f(x) = m_1 x - (m_0 - m_1). \tag{16}$$

Sea

$$s = -\frac{\beta}{\beta + \gamma}. \tag{17}$$

La condición escalar exterior es

$$m_1 x \pm (m_0 - m_1) = s x. \tag{18}$$

Así,

$$x_+ = -\frac{m_0 - m_1}{m_1 - s}, \qquad x_- = \frac{m_0 - m_1}{m_1 - s}. \tag{19}$$

Finalmente,

$$E_\pm = \left(x_\pm,\, x_\pm + f(x_\pm),\, f(x_\pm)\right). \tag{20}$$

5.2. Líneas para calcularlos con la librería

```
from hidden_attractors import chua_piecewise_parameters
from hidden_attractors.models import equilibria_piecewise, rhs_piecewise
import numpy as np

p = chua_piecewise_parameters()
eqs = equilibria_piecewise(p)

for name, point in eqs.items():
    residual = np.linalg.norm(rhs_piecewise(point, p))
    print(name, point, residual)
```

O directamente:

```
python examples/quickstart_equilibria.py
```

6. Jacobiano y estabilidad fraccionaria local

La pendiente local de la no linealidad por tramos es

$$a(x) = \begin{cases} m_1, & |x| > 1, \\ m_0, & |x| < 1. \end{cases} \tag{21}$$

El Jacobiano por regiÃ³n es

$$J(a) = \begin{bmatrix} -\alpha(1+a) & \alpha & 0 \\ 1 & -1 & 1 \\ 0 & -\beta & -\gamma \end{bmatrix}. \tag{22}$$

Para un sistema fraccionario conmensurado

$${}^{C}D_t^q \xi = J(a)\xi, \tag{23}$$

el criterio de Matignon exige

$$|\arg(\lambda_i)| > \frac{q\pi}{2} \tag{24}$$

para estabilidad local asintótica. Si un equilibrio viola esa condición, es localmente inestable. Sin embargo, que un equilibrio sea inestable no implica que el atractor candidato sea autoexcitado; para eso se necesita verificar si su cuenca intersecta alguna vecindad de equilibrio.

7. Forma Lur'e del sistema

Escribimos

$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \qquad \sigma = r^T X = x, \qquad r = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{25}$$

Separando la parte lineal con pendiente exterior m_1,

$$P = \begin{bmatrix} -\alpha(1+m_1) & \alpha & 0 \\ 1 & -1 & 1 \\ 0 & -\beta & -\gamma \end{bmatrix}, \qquad b = \begin{bmatrix} -\alpha \\ 0 \\ 0 \end{bmatrix}. \tag{26}$$

La no linealidad escalar compatible es

$$\psi(\sigma) = (m_0 - m_1)\,\mathrm{sat}(\sigma). \tag{27}$$

Entonces

$${}^{C}D_t^q X = PX + b\,\psi(r^T X). \tag{28}$$

8. Función de transferencia fraccionaria

Para el bloque lineal fraccionario,

$${}^{C}D_t^q X = PX + bu, \qquad \sigma = r^T X, \tag{29}$$

la transferencia formal es

$$W_q(s) = r^T (s^q I - P)^{-1} b. \tag{30}$$

Usando el parámetro espectral

$$\lambda = s^q, \tag{31}$$

se define

$$\widehat{W}_q(\lambda) = r^T (\lambda I - P)^{-1} b. \tag{32}$$

En frecuencia,

$$\lambda = (\mathrm{i}\omega)^q = \omega^q \left(\cos\frac{q\pi}{2} + \mathrm{i}\sin\frac{q\pi}{2} \right). \tag{33}$$

El código histórico usa tambien la convención

$$W_{\mathrm{code}}(\omega) = r^T (P - \lambda I)^{-1} b, \tag{34}$$

por lo que

$$W_{\mathrm{code}}(\omega) = -\widehat{W}_q(\lambda). \tag{35}$$

Esta diferencia de signo debe cuidarse al comparar ecuaciones del reporte con residuos emitidos por el código.

9. Función descriptiva centrada clásica

Para una entrada centrada

$$\sigma(t) = A\cos(\omega t), \tag{36}$$

la función descriptiva clásica de la saturación escalada es

$$N(A) = \frac{Y_1}{A}, \tag{37}$$

donde Y_1 es el primer armónico de $\psi(A\cos\theta)$.
Para el caso no suave,

$$\psi(\sigma) = M\,\mathrm{sat}(\sigma), \qquad M = m_0 - m_1. \tag{38}$$

Si $0 < A \le 1$,

$$N(A) = M. \tag{39}$$

Si $A > 1$,

$$N(A) = \frac{2M}{\pi}\left[\arcsin\left(\frac{1}{A}\right) + \frac{\sqrt{A^2-1}}{A^2}\right]. \tag{40}$$

La condición armónica, usando la convención de reporte, es

$$1 + \widehat{W}_q((\mathrm{i}\omega)^q)N(A) = 0. \tag{41}$$

Con la convenciÃ³n histÃ³rica del cÃ³digo,

$$1 + W_{\mathrm{code}}(\omega)N(A) = 0, \tag{42}$$

pero recordando que $W_{\mathrm{code}} = -\widehat{W}_q$. En el cÃ³digo, primero se resuelve

$$\Im W_{\mathrm{code}}(\omega) = 0, \tag{43}$$

y luego

$$k = -\frac{1}{\Re W_{\mathrm{code}}(\omega)}. \tag{44}$$

La amplitud se obtiene resolviendo

$$N(A) = k. \tag{45}$$

9.1. LÃneas para obtener semillas centradas

Desde el flujo extendido:

```
cd version_2
python tools/legacy/run_extended_search.py ^
  --config configs/chua_fractional_nonsmooth.yaml
```

En Linux/macOS:

```
cd version_2
python tools/legacy/run_extended_search.py \
  --config configs/chua_fractional_nonsmooth.yaml
```

La parte centrada usa internamente:

- `find_omega_k_candidates`;
- `centered_candidates_from_nyquist`;
- `solve_amplitude_from_k`;
- `build_fractional_seed`.

Archivos esperados:

- `rho_H_diagnostics.csv`;
- `biased_df_candidates.csv`;
- `continuation_summary.csv`;
- `continuation_paths.csv`;
- `extended_search_report.md`.

Aunque el archivo `biased_df_candidates.csv` contiene la familia sesgada, el flujo extendido tambiÃ©n construye candidatos centrados desde Nyquist antes de la continuaciÃ³n.

10. FunciÃ³n descriptiva sesgada

La funciÃ³n descriptiva sesgada usa

$$\sigma(t) = \sigma_0 + A\cos(\omega t). \tag{46}$$

La salida

$$y(t) = \psi(\sigma_0 + A\cos(\omega t)) \tag{47}$$

se expande en Fourier:

$$y(t) = y_{\text{mean}} + \sum_{k=1}^{K} \left(a_k \cos(k\omega t) + b_k \sin(k\omega t)\right). \tag{48}$$

El cÃ³digo define el coeficiente complejo

$$Y_k = a_k - \mathrm{i}b_k. \tag{49}$$

Por tanto,

$$N(A, \sigma_0) = \frac{Y_1}{A}. \tag{50}$$

La ventaja frente a la funciÃ³n descriptiva centrada es que explora oscilaciones alrededor de una media distinta de cero. Esto es importante en sistemas con atractores desplazados respecto al origen.

10.1. CondiciÃ³n armÃ³nica sesgada

La condiciÃ³n usada para filtrar candidatos es

$$R(A, \sigma_0, \omega) = 1 + W_{\text{code}}(\omega) N(A, \sigma_0), \tag{51}$$

y se busca

$$|R(A, \sigma_0, \omega)| \ll 1. \tag{52}$$

AdemÃ¡s se calcula un diagnÃ³stico de armÃ³nicos superiores:

$$\rho_H = \frac{\sum_{k=2}^{K} |W_q(k\omega)|\,|Y_k|}{|W_q(\omega)|\,|Y_1| + \varepsilon}. \tag{53}$$

InterpretaciÃ³n:

- ρ_H pequeÃ±o: la aproximaciÃ³n de primer armÃ³nico es mÃ¡s razonable;
- ρ_H grande: la funciÃ³n descriptiva pierde confiabilidad como generador de semilla;
- ρ_H no prueba existencia de atractor ni ocultedad.

10.2. LÃneas para bÃºsqueda sesgada

El script principal es el mismo:

```
python tools/legacy/run_extended_search.py ^
  --config configs/chua_fractional_nonsmooth.yaml
```

La bÃºsqueda sesgada usa internamente:

- una malla en (A, σ_0, ω);

- opcionalmente Latin Hypercube Sampling;
- familias `classical_biased` y `machado_biased`;
- filtro por `residual_abs`;
- desempate por ρ_H.

Archivos relevantes:

- `biased_df_all_evaluations.csv`: todas las evaluaciones;
- `biased_df_candidates.csv`: mejores candidatos seleccionados;
- `rho_H_diagnostics.csv`: diagnÃ³stico armÃ³nico;
- `plots/rho_H_vs_A.png`;
- `plots/rho_H_vs_omega.png`;
- `plots/harmonic_spectrum_candidate_*.png`.

11. FunciÃ³n descriptiva tipo Machado

La familia tipo Machado se usa como generador auxiliar de semillas. Si $N(A)$ es la funciÃ³n descriptiva base, se define

$$N_\mu(A) = \exp\{\mu \operatorname{Log} N(A)\}, \qquad \mu > 0. \tag{54}$$

En el caso no suave centrado, si $N(A) > 0$, se usa la rama real:

$$N_\mu(A) = N(A)^\mu. \tag{55}$$

En el caso sesgado, como $N(A, \sigma_0)$ puede ser complejo, el cÃ³digo usa

$$N_\mu(A, \sigma_0) = \exp\left(\mu \operatorname{Log}_\ell N(A, \sigma_0)\right), \tag{56}$$

donde ℓ es la rama seleccionada del logaritmo complejo.

11.1. Advertencia teÃ³rica

El parÃ¡metro μ no es el orden fraccionario del sistema. El sistema causal sigue teniendo orden q. El parÃ¡metro μ solo deforma la funciÃ³n descriptiva usada para generar semillas. Por tanto:

$$\mu \neq q. \tag{57}$$

La ruta Machado no prueba ciclos exactos de Caputo. Produce candidatos que deben simularse y verificarse con cuencas.

11.2. LÃneas para activar Machado

En la configuraciÃ³n YAML deben existir valores como:

```
machado:
  mu_values: [0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 2.0, 3.0, 4.0]
  branch: 0
  zero_eps: 1.0e-12

biased_search:
  families:
    - classical_biased
    - machado_biased
```

DespuÃ©s:

```
python tools/legacy/run_extended_search.py ^
  --config configs/chua_fractional_nonsmooth.yaml
```

Para una verificaciÃ³n refinada de corridas Machado:

```
python tools/legacy/run_extended_search.py ^
  --config configs/chua_fractional_nonsmooth.yaml ^
  --mode corrida1_refined_verification
```

12. Otras rutas de generaciÃ³n que no conviene olvidar

AdemÃ¡s de la funciÃ³n descriptiva centrada, sesgada y Machado, el repositorio incluye o conserva estas rutas:

Ruta	Papel
ρ_H-diagnostics	No genera una semilla nueva por sÃ solo, pero filtra la confiabilidad armÃ³nica de una semilla.
Seed cloud search	Simula nubes de condiciones iniciales evitando vecindades de equilibrio para detectar candidatos no triviales.
ContinuaciÃ³n en η	Transporta una semilla desde una no linealidad suavizada o aproximada hasta la no linealidad original.
ContinuaciÃ³n multiparamÃ©trica	Cambia η, q, σ_0 o μ segÃºn la familia del candidato.
Danca ABM replication	Ruta independiente para reproducir el caso reportado por Danca con predictor–corrector ABM.
Refined basin classification	Reanaliza celdas desconocidas comparÃ¡ndolas con trayectorias target.
Robustness overlay	Cambia h, L_m, T para revisar persistencia geomÃ©trica.
Sphere controls	Lanza trayectorias desde esferas alrededor de los equilibrios.
External complexity tools	Usa `nolds` o `antropy` para entropÃas, dimensiones o indicadores de series temporales.

13. Modo sweep: barridos sistemÃ¡ticos antes de elegir candidatos

El modo *sweep* no debe confundirse con una prueba final de ocultedad. Su papel es exploratorio: recorre una familia de parÃ¡metros, genera semillas o trayectorias candidatas, calcula residuos y diagnÃ³sticos, y deja archivos resumidos para decidir quÃ© casos merecen continuaciÃ³n larga, cuencas y verificaciÃ³n alrededor de equilibrios.

La idea general es:

$$\mathcal{P} = \{p_1, p_2, \ldots, p_N\} \quad \Longrightarrow \quad \{\text{semilla}(p_i), \text{residuo}(p_i), \rho_H(p_i), \text{clase}(p_i)\}_{i=1}^{N}. \tag{58}$$

En este repositorio existen varios barridos. Los mÃ¡s importantes para el Chua fraccionario no suave son:

Modo	QuÃ© explora
`q_sweep`	Barre el orden fraccionario q. Para cada q, calcula Nyquist/DF, semilla, continuaciÃ³n y atractor final ligero. Sirve para saber en quÃ© Ã³rdenes fraccionarios el cierre armÃ³nico y la continuaciÃ³n son mÃ¡s prometedores.
`df_compare`	Compara la funciÃ³n descriptiva clÃ¡sica contra la extensiÃ³n tipo Machado para uno o varios valores de μ. Sirve para saber si Machado aporta semillas distintas de las clÃ¡sicas.
`machado_sweep`	Barrido completo de ramas, valores de μ y fases θ. Es mÃ¡s costoso y estÃ¡ pensado para exploraciÃ³n amplia del Chua no suave.
`machado_sweep_fast`	VersiÃ³n reducida del barrido Machado. Usa una malla menor de μ y θ, por lo que sirve como prueba previa antes de una corrida completa.
`positive_x_basin_sweep`	Barrido heredado sobre regiones de cuenca, Ãºtil para explorar quÃ© zonas iniciales caen en clases target. Debe tratarse como herramienta de cartografÃa de cuenca, no como prueba independiente de ocultedad.

13.1. Barrido del orden fraccionario q

El modo `q_sweep` responde a la pregunta:

> Â¿Para quÃ© valores de q la ruta Nyquist/funciÃ³n descriptiva genera semillas continuables hacia un atractor no trivial?

Con la entrada publica sin variables de entorno:

```
hidden-attractors-unified-chua --run-mode q_sweep --q-values 0.97,0.98,0.985,0.99,0.995,1.0
```

Con malla uniforme:

```
hidden-attractors-unified-chua --run-mode q_sweep --q-values linspace:0.96:1.0:17
```

La forma equivalente con modulo de Python es:

```
python -m hidden_attractors.workflows.unified_chua --run-mode q_sweep --q-values
    0.97,0.98,0.985,0.99,0.995,1.0
```

Las salidas esperadas son:

```
q_order_sweep/
  q_sweep_summary.csv
  q_sweep_summary.json
  q_0p98500/
    fig01_nyquist_df_q_0p98500.pdf
    fig02_continuation_progress_q_0p98500.pdf
    fig03_final_attractor_q_0p98500.pdf
    continuation_summary_q_0p98500.json
    final_attractor_q_0p98500.csv
    final_attractor_q_0p98500.npz
    nyquist_df_samples_q_0p98500.csv
```

Lectura recomendada: elegir los valores de q donde el residuo armÃ³nico sea pequeÃ±o, la continuaciÃ³n no diverja y el atractor final sea acotado y no trivial. DespuÃ©s de esa preseleccciÃ³n se ejecutan pruebas de robustez, cuencas y esferas alrededor de equilibrios. No se debe escribir "atractor oculto" Ãºnicamente a partir de `q_sweep`.

13.2. ComparaciÃ³n clÃ¡sica contra Machado: `df_compare`

Este modo compara dos familias de semillas:

$$N_{\mathrm{cl}}(A), \qquad N_{\mu}(A) = N_{\mathrm{cl}}(A)^{\mu}. \tag{59}$$

Se ejecuta, para cada familia, la cadena:

$$\text{Nyquist/DF} \rightarrow \text{semilla} \rightarrow \text{continuaciÃ³n} \rightarrow \text{verificaciÃ³n de ocultedad.} \tag{60}$$

Comando bÃ¡sico:

```
hidden-attractors-unified-chua --run-mode df_compare --q 0.99 --machado-mu 0.5
```

Para varios valores de μ:

```
hidden-attractors-unified-chua --run-mode df_compare --q 0.99 --machado-mu-values
    0.5,0.75,1.0,1.25,1.5,2.0,3.0
```

Salidas principales:

```
df_seed_comparison/
  df_seed_comparison_summary.csv
  df_seed_comparison_summary.json
  classic/
  machado_mu_0p50000/
```

La comparaciÃ³n debe responder:

- si Machado produce amplitudes diferentes;
- si la semilla resultante llega a una trayectoria acotada;
- si la verificaciÃ³n preliminar detecta contacto con vecindades de equilibrio;
- si la geometrÃa final es distinta o equivalente a la semilla clÃ¡sica.

13.3. Barrido Machado completo

El modo `machado_sweep` recorre ramas de Nyquist, Ã³rdenes descriptivos μ y fases θ. Para cada combinaciÃ³n se construye

$$X_{\mathrm{seed}}^{(j,\mu,\theta)} = A_{j,\mu}\,\mathrm{Re}\left(v_j e^{i\theta}\right), \tag{61}$$

donde j identifica la rama de Nyquist. La condiciÃ³n de compatibilidad usada en el barrido es

$$0 < k_j < \Delta^{\mu}, \qquad \Delta = m_0 - m_1. \tag{62}$$

Comando completo:

```
hidden-attractors-unified-chua --run-mode machado_sweep --model piecewise --output-dir
    runs_machado_sweep --q 0.99 --h 0.01 --memory-length 8 --t-transient 80 --t-keep 100
```

Salidas principales:

```
machado_sweep/
  machado_sweep_summary.csv
  machado_sweep_summary.json
  branch_*/mu_*/theta_*/
```

Este barrido debe usarse cuando la bÃºsqueda centrada/sesgada no produce candidatos suficientemente buenos o cuando se quiere justificar una exploraciÃ³n mÃ¡s amplia de fases.

13.4. Barrido Machado rÃ¡pido

El modo `machado_sweep_fast` es la versiÃ³n recomendada para una primera corrida. Usa una malla reducida:

$$\text{rama } 0\text{: } \mu \in \{0.75, 1.00, 1.25, 1.50, 2.00, 3.00\}, \tag{63}$$

$$\text{rama } 1\text{: } \mu \in \{0.50, 0.75, 1.00, 1.10, 1.25, 1.40\}, \tag{64}$$

$$\theta \in \left\{0, \frac{\pi}{3}, \frac{2\pi}{3}, \pi, \frac{4\pi}{3}, \frac{5\pi}{3}\right\}. \tag{65}$$

Comando:

```
hidden-attractors-unified-chua --run-mode machado_sweep_fast --model piecewise --output-dir
    runs_machado_sweep_fast --q 0.99 --h 0.01 --memory-length 8 --t-transient 80 --t-keep
    100
```

Para una prueba mÃnima:

```
hidden-attractors-unified-chua --run-mode machado_sweep_fast --machado-sweep-max-candidates
    1
```

Salidas:

```
machado_sweep_fast/
  machado_sweep_fast_summary.csv
  machado_sweep_fast_summary.json
  branch_*/mu_*/theta_*/
```

13.5. CÃ³mo filtrar un sweep

Un sweep puede generar muchos candidatos. El filtrado recomendado es:

1. eliminar candidatos invÃ¡lidos por incompatibilidad de ganancia;
2. ordenar por `residual_abs`;
3. exigir ρ_H pequeÃ±o cuando estÃ© disponible;
4. conservar Ãºnicamente trayectorias acotadas y no colapsadas;
5. descartar candidatos que converjan a equilibrio;
6. revisar contacto preliminar con vecindades de equilibrio;
7. pasar solo los mejores a continuaciÃ³n larga, robustez y cuencas.

Operativamente, una tabla de sweep debe leerse asÃ:

Columna	InterpretaciÃ³n
`branch`	Rama de Nyquist usada para construir la semilla.
`mu`	Orden descriptivo auxiliar Machado. No es el orden de Caputo.
`theta`	Fase usada para colocar la semilla sobre la aproximaciÃ³n armÃ³nica.
`residual_abs`	TamaÃ±o del error de cierre armÃ³nico $\lvert 1 + W_q N \rvert$. Menor es mejor.

Columna	InterpretaciÃ³n
`rho_H`	Medida relativa de armÃ³nicos superiores. Menor indica mejor consistencia de primer armÃ³nico.
`survived`	Indica si la continuaciÃ³n o simulaciÃ³n preliminar no divergiÃ³.
`hiddenness_status`	Etiqueta operacional. No equivale por sÃ sola a prueba matemÃ¡tica.

13.6. RelaciÃ³n entre sweep y comparaciÃ³n con Danca

Para comparar con Danca, el sweep sirve para localizar candidatos internos bajo el mismo sistema y un orden q cercano al reportado. La comparaciÃ³n debe separar tres cosas:

$$\text{Danca/ABM con historia completa} \neq \text{EFORK con memoria truncada} \neq \text{semilla armÃ³nica tipo Weyl.} \tag{66}$$

Por tanto, el reporte debe presentar:

- el atractor reportado/replicado por Danca;
- el mejor candidato encontrado por `q_sweep`, `df_compare` o `machado_sweep_fast`;
- sus parÃ¡metros $q, h, L_m, T, t_{\text{burn}}$;
- sus residuos armÃ³nicos y ρ_H, si existen;
- sus pruebas de cuenca y esfera alrededor de equilibrios;
- una conclusiÃ³n conservadora: compatible, descartado o pendiente de verificaciÃ³n, no "demostrado" salvo que todas las pruebas de cuenca lo respalden.

14. ContinuaciÃ³n numÃ©rica de un parÃ¡metro

La continuaciÃ³n implementada en `multiparameter_continuation.py` usa una familia

$$ {}^{C}D_t^q X = PX + b\,\psi_\eta(r^T X), \qquad \eta \in [0,1]. \tag{67}$$

Para el modelo no suave, se usa una transiciÃ³n entre saturaciÃ³n suavizada y saturaciÃ³n exacta:

$$\psi_\eta(\sigma) = M\left[(1-\eta)\tanh\left(\frac{\sigma}{w}\right) + \eta\,\mathrm{sat}(\sigma)\right], \qquad M = m_0 - m_1. \tag{68}$$

Casos:

$$\eta = 0: \quad \psi_0(\sigma) = M\tanh(\sigma/w), \tag{69}$$

$$\eta = 1: \quad \psi_1(\sigma) = M\,\mathrm{sat}(\sigma). \tag{70}$$

14.1. Memoria en la continuaciÃ³n

En sistemas de Caputo no basta pasar el Ãºltimo punto de una etapa a la siguiente. El cÃ³digo conserva una ventana discreta de historia:

$$\mathcal{H}_n = \{(t_{n-M}, X_{n-M}), \ldots, (t_n, X_n)\}, \qquad M = \left\lceil \frac{L_m}{h} \right\rceil. \tag{71}$$

Esta ventana se transporta mediante `FractionalHistory`. La continuaciÃ³n sigue siendo una heurÃstica numÃ©rica, no una prueba de existencia exacta.

14.2. Archivos de salida

- `continuation_summary.csv`: una fila por paso de continuaciÃ³n;
- `continuation_paths.csv`: ruta de estados finales;
- columnas relevantes:
  - `eta`;
  - `q`;
  - `chi_name`;
  - `chi_value`;
  - `bounded`;
  - `diverged`;
  - `equilibrium_hit`;
  - `attractor_label`;
  - `memory_carried`;
  - `fft_dominant_frequency`;
  - `psd_entropy`.

15. ContinuaciÃ³n multiparamÃ©trica

El cronograma de continuaciÃ³n se forma con tres posibles direcciones:

$$q_{\text{start}} \to q_{\text{target}}, \tag{72}$$

$$\chi_{\text{start}} \to \chi_{\text{target}}, \tag{73}$$

$$\eta_{\text{start}} \to \eta_{\text{target}}. \tag{74}$$

La variable χ depende de la familia del candidato:

$$\chi = \begin{cases} \mu, & \text{familia Machado,} \\ \sigma_0, & \text{familia sesgada clÃ¡sica.} \end{cases} \tag{75}$$

La configuraciÃ³n tÃpica contiene:

```
continuation:
  eta_start: 0.0
  eta_target: 1.0
  eta_steps: 6
  q_start: 0.9998
  q_target: 0.9998
  q_steps: 1
  chi_steps: 1
  h: 0.01
  memory_length: 10.0
  t_step: 60.0
  smooth_width: 0.2
  max_candidates: 10
```

16. Candidatos finales y filtrado

La librerÃa pÃºblica carga tres candidatos finales mediante

```
hidden-attractors-list-candidates
```

o

```
python examples/list_final_candidates.py
```

Desde Python:

```
from hidden_attractors import load_final_candidate_records

records = load_final_candidate_records()

for r in records:
    print(r.candidate_id, r.route, r.q, r.seed, r.robust_start)
```

16.1. Criterios de filtrado recomendados

Un candidato no debe seleccionarse solo por verse bien. El orden sugerido es:

1. Residuo armÃ³nico pequeÃ±o:
$$|1 + W_q N| \ll 1.$$
2. ρ_H pequeÃ±o:
$$\rho_H < \rho_{\max}.$$
3. Semilla finita:
$$\|X_0\| < \infty.$$
4. ContinuaciÃ³n sobreviviente:
$$\texttt{bounded=True}, \quad \texttt{diverged=False}.$$
5. No convergencia a equilibrio:
$$\texttt{equilibrium\_hit=False}.$$
6. Rango post-transitorio no trivial:
$$\max\{\text{range}_x, \text{range}_y, \text{range}_z\} > \varepsilon.$$
7. Robustez bajo cambios de h, L_m, T.
8. Ausencia de contacto target desde vecindades de equilibrios bajo radios probados.

17. Pruebas de ocultedad

La definiciÃ³n operacional es:

> Un atractor es oculto si su cuenca de atracciÃ³n no intersecta ninguna vecindad abierta de los puntos de equilibrio. Si sÃ intersecta alguna vecindad de un equilibrio, es autoexcitado.

En cÃ³mputo finito no se prueba una vecindad abierta completa. Se prueban radios y muestras:

$$X_0 = E_i + r u_j, \qquad \|u_j\| = 1, \qquad r \in \mathcal{R}. \tag{76}$$

Si alguna trayectoria desde $E_i + r u_j$ cae en la clase target, entonces hay evidencia contra la ocultedad bajo ese contrato.

18. Cuencas de atracciÃ³n

Las cuencas se clasifican con etiquetas operativas:

ID	Etiqueta	InterpretaciÃ³n
0	`equilibrium`	La trayectoria converge o queda cerca de un equilibrio.
1	`target_positive`	La trayectoria cae en la clase target positiva.
2	`target_negative`	La trayectoria cae en la clase target negativa.
3	`infinity`	La trayectoria diverge o excede cota.
4	`unknown`	La clasificaciÃ³n gruesa no decide.
5	`numerical_failure`	Fallo numÃ©rico o datos no finitos.
6	`bounded_other`	Clase refinada: acotada pero no suficientemente cercana al target.

El corte de cuenca tÃpico se hace en un plano, por ejemplo:

$$z = z_*, \qquad (x_0, y_0) \in [x_{\text{mín}}, x_{\text{máx}}] \times [y_{\text{mín}}, y_{\text{máx}}]. \tag{77}$$

Si quedan celdas `unknown`, se usa `refined_basin`.

19. Prueba por esferas alrededor de equilibrios

19.1. EjecuciÃ³n

```
hidden-attractors-sphere-controls --help
```

EjecuciÃ³n con parÃ¡metros explÃcitos:

```
hidden-attractors-sphere-controls ^
  --top-n 3 ^
  --radii 1e-5,3e-5,1e-4,3e-4,1e-3 ^
  --samples-per-radius 500 ^
  --chunks 6 ^
  --q 0.9998 ^
  --h 0.01 ^
  --memory-length 10.0 ^
  --t-final 1500.0 ^
  --t-burn 100.0
```

En Linux/macOS:

```
hidden-attractors-sphere-controls \
  --top-n 3 \
  --radii 1e-5,3e-5,1e-4,3e-4,1e-3 \
  --samples-per-radius 500 \
  --chunks 6 \
  --q 0.9998 \
  --h 0.01 \
  --memory-length 10.0 \
  --t-final 1500.0 \
  --t-burn 100.0
```

19.2. Archivos que produce

- `sphere_plan.csv`: todas las condiciones iniciales sobre esferas;
- `sphere_raw_chunk_*.csv`: resultados parciales por proceso;
- `sphere_raw.csv`: todos los resultados;
- `sphere_cumulative_summary.csv`: conteos acumulados;
- `sphere_decision.csv`: decisiÃ³n operacional por candidato;
- `attractor_robustness_raw.csv`: clasificaciÃ³n de robustez desde el punto robusto;
- `attractor_robustness_summary.csv`;
- `top3_sphere_robustness_summary.json`.

19.3. CÃ³mo leer `sphere_decision.csv`

Columnas importantes:

- `candidate_id`;
- `total_sphere_trajectories`;
- `total_target_hits`;
- `smallest_radius_with_target_hit`;
- `hiddenness_status`.

Lectura:

$$\texttt{total\_target\_hits} > 0 \quad \Longrightarrow \quad \text{no se sostiene ocultedad bajo esas esferas.} \tag{78}$$

$$\texttt{total\_target\_hits} = 0 \quad \Longrightarrow \quad \text{compatible con ocultedad bajo radios y muestras probados.} \tag{79}$$

No debe escribirse "atractor oculto demostrado" solo con esta condiciÃ³n.

20. Pruebas de robustez

20.1. Robustness overlay

Este workflow no prueba ocultedad. Compara la geometrÃa del atractor al modificar:

$$h, \qquad L_m, \qquad T. \tag{80}$$

EjecuciÃ³n:

```
hidden-attractors-robustness-overlay --help
```

Ejemplo:

```
hidden-attractors-robustness-overlay ^
  --q 0.9998 ^
  --divergence-norm 120.0 ^
  --equilibrium-tol 1e-3 ^
  --max-store-points 6000 ^
  --max-metric-points 1000 ^
  --max-section-points 300
```

20.2. Casos de robustez tÃpicos

Caso	h	L_m	T	Lectura
`R0_base`	0.01	10	1500	Contrato base.
`R1_h_finer`	0.005	10	1500	Paso mÃ¡s fino.
`R2_h_coarser`	0.02	10	1500	Paso mÃ¡s grueso.
`R3_Lm_lower`	0.01	5	1500	Menor memoria.
`R4_Lm_higher`	0.01	20	1500	Mayor memoria.
`R5_t_longer`	0.01	10	3000	Tiempo mÃ¡s largo.

20.3. Archivos de salida

- `robustness_overlay_config.json`;
- `metrics_<candidate>.csv`;
- `robustness_overlay_metrics.csv`;
- `robustness_overlay_summary.json`;
- `plots/overlay_<candidate>.png`.

20.4. CÃ³mo decidir si un candidato es robusto

Un candidato es robusto bajo estos contratos si:

$$\texttt{bounded=True}, \quad \texttt{diverged=False}, \quad \texttt{equilibrium\_like=False}, \tag{81}$$

y las distancias relativas son pequeÃ±as:

$$\texttt{range\_relative\_distance} \ll 1, \tag{82}$$

$$\texttt{cloud\_median\_distance\_norm} \ll 1, \tag{83}$$

$$\texttt{section\_median\_distance\_norm} \ll 1. \tag{84}$$

21. Refinamiento de cuencas desconocidas

Si una cuenca tiene muchas celdas `unknown`, se usa:

```
hidden-attractors-refined-basin --help
```

Ejemplo:

```
hidden-attractors-refined-basin ^
  --source-dir outputs/basin_compare_danca_project_h001_grid101_20260517 ^
  --chunks 8 ^
  --tail-fraction-start 0.5 ^
  --max-cloud-points 700 ^
  --max-section-points 250 ^
  --max-score 0.95
```

El mÃ©todo reintegra cada celda desconocida y compara contra dos referencias:

$$\mathcal{A}_+, \qquad \mathcal{A}_-. \tag{85}$$

Calcula:

- distancia entre nubes de cola;
- distancia entre secciones de PoincarÃ©;
- distancia relativa de rangos;
- diferencia relativa de frecuencia FFT dominante;
- score ponderado.

La decisiÃ³n se expresa como:

$$\texttt{matched\_target\_reference} \tag{86}$$

o

$$\texttt{bounded\_noncollapsed\_reference\_mismatch}. \tag{87}$$

22. AnÃ¡lisis adicionales implementados

22.1. FFT y entropÃa espectral

Para una seÃ±al x_n, se resta la media y se usa una ventana de Hann:

$$\tilde{x}_n = (x_n - \bar{x}) w_n. \tag{88}$$

El espectro de potencia discreto es

$$S_k = |\,\mathrm{rFFT}(\tilde{x})_k|^2. \tag{89}$$

La frecuencia dominante es

$$f_{\text{máx}} = f_{\arg\text{máx}_{k\geq 1} S_k}. \tag{90}$$

La entropÃa espectral normalizada se calcula como

$$H_{\mathrm{PSD}} = -\frac{1}{\log N} \sum_k p_k \log(p_k + \varepsilon), \qquad p_k = \frac{S_k}{\sum_j S_j}. \tag{91}$$

22.2. SecciÃ³n de PoincarÃ© operacional

El cÃ³digo usa cruces hacia arriba del plano $x = 0$:

$$x_{n-1} < 0 \leq x_n, \qquad \dot{x}_n > 0. \tag{92}$$

En un sistema fraccionario no se interpreta esta secciÃ³n como mapa de PoincarÃ© exacto con propiedad de semigrupo. Se usa como nube geomÃ©trica para comparar atractores.

22.3. Distancia entre nubes

Dadas dos nubes A y B, se define una distancia simÃ©trica basada en vecinos mÃ¡s cercanos:

$$d(A,B) = \mathrm{median}\left(\{\min_{b\in B} \|a-b\| : a \in A\} \cup \{\min_{a\in A} \|b-a\| : b \in B\} \right). \tag{93}$$

22.4. BifurcaciÃ³n por postprocesamiento

La librerÃa extrae mÃ¡ximos, mÃnimos o muestras de trayectorias ya calculadas:

```
from hidden_attractors.analysis import bifurcation_points_from_trajectories
from hidden_attractors.plotting import plot_bifurcation_diagram
```

Esto no es continuaciÃ³n numÃ©rica formal. Sirve para diagramas exploratorios.

22.5. Medidas externas de complejidad

Si se instala `nolds` o `antropy`:

```
from hidden_attractors.integrations import compute_complexity_measures

metrics = compute_complexity_measures(traj[:, 1], backend="auto")
print(metrics)
```

Estas medidas pueden incluir entropÃa de permutaciÃ³n, entropÃa muestral, dimensiÃ³n de correlaciÃ³n, DFA o estimadores de Lyapunov de serie temporal. No son prueba de ocultedad.

23. CÃ³mo elegir el mejor candidato

Un ranking razonable puede organizarse asÃ:

Prioridad	Criterio	Archivo
1	Residuo armÃ³nico pequeÃ±o	`biased_df_candidates.csv`, `rho_H_diagnostics.csv`
2	ArmÃ³nicos superiores atenuados	`rho_H_diagnostics.csv`
3	Supervivencia por continuaciÃ³n	`continuation_summary.csv`
4	No convergencia a equilibrio	`continuation_summary.csv`
5	Trayectoria acotada no trivial	`metrics_*.csv`, `trajectory_metrics`
6	Robustez ante h, L_m, T	`robustness_overlay_metrics.csv`
7	No contacto desde esferas	`sphere_decision.csv`
8	Cuenca refinada sin contacto local	`refined_basin_summary.json`
9	ComparaciÃ³n geomÃ©trica contra Danca	`danca2017_vs_project_candidates_report.md`

Una regla conservadora:

$$\text{mejor candidato} = \arg\min \left[w_1 |R| + w_2 \rho_H + w_3 d_{\text{cloud}} + w_4 d_{\text{section}} + w_5 I_{\text{contact}} \right], \tag{94}$$

donde $I_{\text{contact}} = +\infty$ si hay contacto target desde vecindades de equilibrio. En tÃ©rminos prÃ¡cticos: cualquier contacto target desde una vecindad de equilibrio pesa mÃ¡s que una buena grÃ¡fica 3D.

24. GrÃ¡ficas recomendadas para cada candidato

Para cada candidato final debe generarse:

1. atractor 3D;
2. proyecciones xy, xz, yz;

3. series temporales;
4. espectro FFT o PSD;
5. secciÃ³n $x = 0$;
6. overlay de robustez;
7. cuenca xy;
8. cuencas xz y yz, si estÃ¡n disponibles;
9. zoom alrededor de E_0, E_+, E_-;
10. esferas alrededor de E_0, E_+, E_-;
11. target hits contra radio;
12. tabla final de contratos numÃ©ricos.

25. ComparaciÃ³n con el atractor reportado por Danca

25.1. Ruta Danca

El script especÃfico es:

```
python tools/legacy/danca2017_chua_abm_replication.py --help
```

EjecuciÃ³n rÃ¡pida no cientÃfica:

```
python tools/legacy/danca2017_chua_abm_replication.py ^
  --job all ^
  --quick ^
  --workers 4 ^
  --output-dir outputs/danca_quick_test
```

EjecuciÃ³n mÃ¡s seria:

```
python tools/legacy/danca2017_chua_abm_replication.py ^
  --job all ^
  --workers 4 ^
  --q 0.9998 ^
  --h 0.05 ^
  --t-final 500.0 ^
  --transient 250.0 ^
  --delta 0.01 ^
  --local-samples-per-unstable-eq 100 ^
  --figure-local-trajectories 80 ^
  --output-dir outputs/danca2017_chua_abm_reproduction
```

25.2. QuÃ© produce

- `run_config.json`;
- `danca_reference_summary.json`;
- `danca_hiddenness_decision.json`;
- `fig03_danca2017_chua_abm_replica.png`;
- `fig03_danca2017_chua_abm_replica.pdf`;
- `project_best_two_figure_summary.csv`;
- `danca2017_vs_project_candidates_report.md`.

25.3. Diferencias metodolÃ³gicas

Aspecto	Danca	Repositorio
Integrador	ABM predictor–corrector	EFORK con memoria truncada; ABM solo en script de rÃ©plica
Memoria	Historia completa de Caputo	Ventana finita L_m
LocalizaciÃ³n	BÃºsqueda por prueba y error reportada	Semillas por Lur'e, DF centrada, DF sesgada, Machado y nubes
Ocultedad	Vecindades de equilibrios con $\delta = 0.01$	Esferas, cuencas, refinamiento y comparaciÃ³n target
Salida	Figura y argumento del artÃículo	CSV/JSON/figuras reproducibles
Lectura	Atractor oculto reportado	Candidato compatible/no compatible bajo contrato

25.4. ComparaciÃ³n mÃnima que debe aparecer en el reporte

Para comparar un candidato del repositorio con Danca, incluir una tabla:

Bloque	Dato	InterpretaciÃ³n
Modelo	mismos $\alpha, \beta, \gamma, m_0, m_1, q$	ComparaciÃ³n vÃ¡lida a nivel de sistema
Integrador	ABM vs EFORK	No son numÃ©ricamente idÃ©nticos
CondiciÃ³n inicial	semilla Danca localizada o no publicada	Si no estÃ¡ publicada, declararlo
Atrayente observado	rangos, FFT, nube, secciÃ³n	ComparaciÃ³n geomÃ©trica
Equilibrios	E_0, E_+, E_-	Base de prueba de ocultedad
Pruebas locales	δ Danca vs radios $\mathcal{R}$	Comparar tamaÃ±os de vecindad
Resultado	target/no target desde vecindades	Decide compatibilidad con ocultedad

26. Flujo completo recomendado

1. Instalar la librerÃa:

```
cd version_2
python -m pip install -e ".[dev]"
```

2. Verificar equilibrios:

```
python examples/quickstart_equilibria.py
```

3. Graficar una trayectoria existente:

```
python examples/dynamical_analysis_gallery.py ^
  --trajectory-csv ruta/a/trayectoria.csv ^
  --output-dir outputs/examples/galeria_prueba
```

4. Ejecutar bÃºsqueda extendida Lur'e/DF/Machado:

```
python tools/legacy/run_extended_search.py ^
  --config configs/chua_fractional_nonsmooth.yaml
```

5. Revisar candidatos:

```
python examples/list_final_candidates.py
hidden-attractors-list-candidates
```

6. Ejecutar robustez:

```
hidden-attractors-robustness-overlay
```

7. Ejecutar esferas de equilibrio:

```
hidden-attractors-sphere-controls ^
  --top-n 3 ^
  --samples-per-radius 500 ^
  --chunks 6
```

8. Refinar cuenca si existen celdas `unknown`:

```
hidden-attractors-refined-basin ^
  --source-dir outputs/basin_compare_danca_project_h001_grid101_20260517 ^
  --chunks 8
```

9. Reproducir Danca:

```
python tools/legacy/danca2017_chua_abm_replication.py ^
  --job all ^
  --workers 4 ^
  --output-dir outputs/danca2017_chua_abm_reproduction
```

10. Comparar Danca contra candidatos del proyecto:

```
python tools/legacy/danca2017_chua_abm_replication.py ^
  --job report ^
  --output-dir outputs/danca2017_chua_abm_reproduction
```

27. Criterio final de redacciÃ³n cientÃfica

La conclusiÃ³n debe redactarse con niveles de evidencia:

- "La funciÃ³n descriptiva predice una semilla" si solo se resolviÃ³ la condiciÃ³n armÃ³nica.
- "La continuaciÃ³n produce una trayectoria acotada no trivial" si sobreviviÃ³ la homotopÃa.
- "El candidato es robusto bajo los contratos probados" si persiste al cambiar h, L_m, T.
- "El candidato es compatible con ocultedad bajo los radios probados" si no hay target hits desde esferas.
- "El candidato no estÃ¡ soportado como oculto bajo este contrato" si hay target hits desde alguna vecindad de equilibrio.
- "Atractor oculto verificado numÃ©ricamente" solo si se documentan todos los equilibrios, radios, cuencas, robustez, integraciÃ³n y ausencia de intersecciÃ³n de cuenca con vecindades probadas. Aun asÃ, debe decirse que es verificaciÃ³n numÃ©rica finita.

28. Checklist para una corrida defendible

1. Â¿Se fijaron $\alpha, \beta, \gamma, m_0, m_1, q$?
2. Â¿Se reportÃ³ el integrador?
3. Â¿Se reportaron $h, L_m, T, t_{\mathrm{burn}}$?
4. Â¿Se calcularon E_0, E_+, E_-?
5. Â¿Se revisÃ³ Matignon localmente?
6. Â¿Se documentÃ³ cÃ³mo se obtuvo la semilla?
7. Â¿Se guardÃ³ $|1 + W_q N|$?
8. Â¿Se guardÃ³ ρ_H?
9. Â¿Se realizÃ³ continuaciÃ³n con memoria?
10. Â¿Se probÃ³ robustez?
11. Â¿Se generaron cuencas?
12. Â¿Se probaron esferas alrededor de cada equilibrio?
13. Â¿Se comparÃ³ contra Danca con el mismo sistema y explicando diferencias de integrador?
14. Â¿Se evitÃ³ declarar ocultedad solo por una grÃ¡fica?

29. Balance armÃ³nico implementado en el repositorio

El balance armÃ³nico es la etapa que convierte la forma de Lur'e en una condiciÃ³n algebraica aproximada para proponer semillas iniciales. No es una prueba de existencia de ciclos lÃmite exactos de Caputo. En el repositorio se usa como aproximaciÃ³n estacionaria tipo Weyl para construir una condiciÃ³n inicial que despuÃ©s se valida con integraciÃ³n causal de Caputo/EFORK o ABM.

Partimos de

$$ {}^{C}D_t^q X = PX + b\psi(r^T X), \qquad \sigma = r^T X. \tag{95} $$

Para un rÃ©gimen dominante supuesto,

$$ \sigma(t) \approx \sigma_0 + A\cos(\omega t), \tag{96} $$

la no linealidad se expande como

$$ \psi(\sigma_0 + A\cos\theta) = Y_0 + \sum_{k=1}^{\infty} \left(a_k \cos(k\theta) + b_k \sin(k\theta)\right), \qquad \theta = \omega t. \tag{97} $$

El cÃ³digo usa coeficientes complejos

$$ Y_k = a_k - \mathrm{i} b_k. \tag{98} $$

La funciÃ³n descriptiva de primer armÃ³nico es

$$ N(A, \sigma_0) = \frac{Y_1}{A}. \tag{99} $$

En el caso centrado, $\sigma_0 = 0$, se escribe simplemente $N(A)$.

La evaluaciÃ³n frecuencial del bloque lineal fraccionario no se hace con $s = \mathrm{i}\omega$ directamente, sino con

$$ \lambda = (\mathrm{i}\omega)^q = \omega^q \exp\left(\mathrm{i}\frac{q\pi}{2}\right). \tag{100} $$

Con la convenciÃ³n de reporte,

$$\widehat{W}_q(\lambda) = r^T(\lambda I - P)^{-1}b. \tag{101}$$

El cierre de primer armÃ³nico queda

$$1 + \widehat{W}_q((\mathrm{i}\omega)^q)N(A,\sigma_0) = 0. \tag{102}$$

La convenciÃ³n histÃ³rica del cÃ³digo define

$$W_{\mathrm{code}}(\omega) = r^T(P - (\mathrm{i}\omega)^q I)^{-1}b = -\widehat{W}_q((\mathrm{i}\omega)^q). \tag{103}$$

Por eso, en los CSV histÃ³ricos el residuo aparece como

$$R(A,\sigma_0,\omega) = 1 + W_{\mathrm{code}}(\omega)N(A,\sigma_0). \tag{104}$$

Al interpretar resultados del cÃ³digo debe respetarse esa convenciÃ³n de signo.

29.1. CondiciÃ³n de Nyquist usada para semillas centradas

Para la funciÃ³n descriptiva centrada se busca primero una frecuencia ω_0 tal que

$$\Im W_{\mathrm{code}}(\omega_0) = 0. \tag{105}$$

DespuÃ©s se calcula

$$k = -\frac{1}{\Re W_{\mathrm{code}}(\omega_0)}. \tag{106}$$

La amplitud se obtiene de

$$N(A) = k. \tag{107}$$

Si k no pertenece al rango admisible de la funciÃ³n descriptiva, la rama se descarta. Para la saturaciÃ³n no suave con $M = m_0 - m_1 > 0$, el cÃ³digo exige

$$0 < k \le M. \tag{108}$$

29.2. DiagnÃ³stico de armÃ³nicos superiores

La condiciÃ³n armÃ³nica sÃ³lo conserva el primer armÃ³nico. Para medir la severidad de esa aproximaciÃ³n se calcula

$$\rho_H = \frac{\displaystyle\sum_{k=2}^{K} |W_q(k\omega)|\,|Y_k|}{|W_q(\omega)|\,|Y_1| + \varepsilon}. \tag{109}$$

La lectura correcta es:

- ρ_H pequeÃ±o: la semilla armÃ³nica es mÃ¡s razonable como punto inicial;
- ρ_H grande: los armÃ³nicos superiores no estÃ¡n suficientemente atenuuados;
- ρ_H no prueba caos, ocultedad ni existencia de ciclo periÃ³dico exacto.

En el flujo extendido, los campos importantes son:

Campo	Significado
`residual_abs`	$\lvert 1 + W_{\mathrm{code}}N\rvert$.
`rho_H`	RazÃ³n de armÃ³nicos superiores.
`harmonic_energy_ratio`	EnergÃa relativa fuera del primer armÃ³nico.
`accepted_by_rhoH`	Filtro lÃ³gico comparado contra el umbral del YAML.

30. CÃ¡lculo explÃcito de semillas iniciales

El repositorio contiene tres mecanismos principales de semilla: centrada clÃ¡sica, sesgada y Machado. AdemÃ¡s conserva bÃºsqueda por nubes de semillas externas a vecindades de equilibrio.

30.1. Semilla centrada clÃ¡sica

Una vez obtenidos (ω_0, k, A), se introduce el bloque lineal modificado

$$P_0 = P + kbr^T. \tag{110}$$

La semilla armÃ³nica se construye con el autovector asociado a

$$\lambda_0 = (\mathrm{i}\omega_0)^q. \tag{111}$$

El problema que se usa es

$$(P_0 - \lambda_0 I)v \approx 0, \qquad r^T v = 1. \tag{112}$$

Luego, para una fase θ,

$$X_{\mathrm{seed}}(\theta) = A\Re\{ve^{\mathrm{i}\theta}\}. \tag{113}$$

En cÃ³digo, esta ruta corresponde a:

- `find_omega_k_candidates`;
- `solve_amplitude_from_k`;
- `build_fractional_seed`.

30.2. Semilla sesgada

Para una semilla sesgada se usa

$$\sigma(t) = \sigma_0 + A\cos(\omega t). \tag{114}$$

La componente media de la no linealidad, $Y_0 = y_{\mathrm{mean}}$, produce un equilibrio medio aproximado $\bar{X}$. El cÃ³digo lo obtiene por mÃnimos cuadrados imponiendo simultÃ¡neamente

$$P\bar{X} + bY_0 = 0, \qquad r^T\bar{X} = \sigma_0. \tag{115}$$

La componente fundamental compleja V se obtiene resolviendo

$$(\lambda I - P)V = bY_1, \qquad r^T V = A. \tag{116}$$

La semilla sesgada queda

$$X_{\mathrm{seed}}(\theta) = \bar{X} + \Re\{Ve^{\mathrm{i}\theta}\}. \tag{117}$$

Esta ruta evita asumir que el atractor oscila alrededor del origen. Es especialmente Ãºtil cuando el candidato se localiza en una regiÃ³n desplazada de fase.

En cÃ³digo, esta construcciÃ³n corresponde a:

- `fourier_coefficients_psi`;
- `N_biased`;
- `reconstruct_biased_lure_seed`;
- `search_biased_candidates`.

30.3. Semilla Machado centrada

Para el caso centrado no suave, la extensiÃ³n tipo Machado usa

$$N_\mu(A) = N(A)^\mu, \qquad \mu > 0. \tag{118}$$

Para cada rama de Nyquist se conserva la misma ω_j y la misma ganancia lineal k_j, pero la amplitud se recalcula con

$$N(A_\mu)^\mu = k_j, \qquad \text{o equivalentemente} \qquad N(A_\mu) = k_j^{1/\mu}. \tag{119}$$

La compatibilidad exigida es

$$0 < k_j < M^\mu. \tag{120}$$

DespuÃ©s se usa el mismo autovector fraccionario:

$$X_{\text{seed},\mu,j}(\theta) = A_{\mu,j} \Re\{v_j e^{\mathrm{i}\theta}\}. \tag{121}$$

30.4. Semilla Machado sesgada

Cuando $N(A, \sigma_0)$ es complejo, se usa

$$N_\mu(A, \sigma_0) = \exp\left(\mu \operatorname{Log}_\ell N(A, \sigma_0)\right), \tag{122}$$

donde ℓ fija la rama del logaritmo complejo. Esta ruta es mÃ¡s delicada porque puede introducir discontinuidades de rama. Por eso el cÃ³digo descarta valores cercanos a cero con `zero_eps`.

30.5. Nubes de semillas

La bÃºsqueda por nubes no usa funciÃ³n descriptiva. Genera condiciones iniciales en una caja

$$[x_{\text{mín}}, x_{\text{máx}}] \times [y_{\text{mín}}, y_{\text{máx}}] \times [z_{\text{mín}}, z_{\text{máx}}], \tag{123}$$

mezclando malla cartesiana y Latin Hypercube Sampling. DespuÃ©s elimina puntos demasiado cercanos a los equilibrios:

$$\min_i \|X_0 - E_i\| > \delta_{\text{eq}}. \tag{124}$$

La clasificaciÃ³n preliminar puede producir etiquetas como `candidate_hidden_like`, pero esa etiqueta sÃ³lo significa que la trayectoria observada fue acotada, no trivial y no pasÃ³ cerca de los equilibrios bajo el contrato corto probado. No equivale a `hidden_verified`.

31. Criterios de robustez: pruebas implementadas y lectura

El repositorio implementa varias familias de robustez. Conviene distinguirlas porque no responden la misma pregunta.

31.1. Robustez por superposiciÃ³n de trayectorias

El workflow pÃºblico `robustness_overlay` compara la geometrÃa del mismo candidato cuando se cambia el contrato numÃ©rico. La lista estÃ¡ndar es:

Caso	h	L_m	T	Lectura
`R0_base`	0.01	10	1500	Referencia.
`R1_h_finer`	0.005	10	1500	Sensibilidad a paso mÃ¡s fino.
`R2_h_coarser`	0.02	10	1500	Sensibilidad a paso mÃ¡s grueso.
`R3_Lm_lower`	0.01	5	1500	Sensibilidad a memoria menor.
`R4_Lm_higher`	0.01	20	1500	Sensibilidad a memoria mayor.
`R5_t_longer`	0.01	10	3000	Persistencia en tiempo largo.

Para cada caso se guardan mÃ©tricas:

- `bounded`, `diverged`, `equilibrium_like`;
- rangos `range_x`, `range_y`, `range_z`;
- varianzas de cola;
- frecuencia FFT dominante;
- entropÃa espectral;
- nÃºmero de puntos de secciÃ³n;
- distancia relativa de rangos frente a `R0_base`;
- diferencia relativa de FFT;
- distancia mediana entre nubes de cola;
- distancia mediana entre secciones de PoincarÃ©.

Un candidato es robusto sÃ³lo en sentido operativo si, al menos:

$$\texttt{bounded=True}, \quad \texttt{diverged=False}, \quad \texttt{equilibrium\_like=False}, \tag{125}$$

para todos los casos probados, y ademÃ¡s las distancias geomÃ©tricas respecto de la referencia no son desproporcionadas. El cÃ³digo no convierte automÃ¡ticamente esta comparaciÃ³n en prueba de ocultedad.

31.2. Robustez por clasificador de cuenca desde el punto candidato

El workflow `sphere_controls` incluye una prueba adicional desde el punto final robusto de cada candidato. Usa cuatro contratos:

Caso	Cambio	Objetivo	Salida
`R0_baseline`	contrato base	clase target inicial	`attractor_robustness_raw.csv`
`R1_longer`	aumenta T	persistencia temporal	`attractor_robustness_summary.csv`
`R2_shorter_memory`	reduce L_m	sensibilidad a memoria	`attractor_robustness_summary.csv`
`R3_refined_h`	reduce h	sensibilidad a paso	`attractor_robustness_summary.csv`

El campo `robust_attractor=True` significa que el candidato cayÃ³ en clase target bajo todos esos contratos. No significa que sea oculto.

31.3. Robustez por esferas alrededor de equilibrios

La prueba decisiva contra ocultedad es local alrededor de cada equilibrio. Para cada equilibrio E_i, cada radio r_j y cada direcciÃ³n unitaria u_k, se integra desde

$$X_0 = E_i + r_j u_k. \tag{126}$$

Si alguna trayectoria se clasifica como `target_positive` o `target_negative`, entonces existe evidencia contra la ocultedad bajo ese contrato:

$$\exists i, j, k: \quad X_0 = E_i + r_j u_k \longrightarrow \texttt{TARGET}. \tag{127}$$

La lectura del archivo `sphere_decision.csv` es:

- `not_supported_by_sphere_equilibrium_test`: hubo contacto target desde alguna vecindad de equilibrio;
- `compatible_with_hiddenness_under_tested_spheres`: no se observÃ³ contacto target bajo los radios y muestras probados.

La segunda etiqueta no es demostraciÃ³n formal; sÃ³lo resume una prueba finita.

31.4. Robustez por refinamiento de cuencas desconocidas

El workflow `refined_basin` reintegra celdas previamente clasificadas como `unknown`. Compara cada trayectoria contra referencias target positiva y negativa mediante un puntaje de geometrÃa:

$$S = \frac{w_c d_c + w_r d_r + w_f d_f + w_s d_s}{w_c + w_r + w_f + w_s}, \tag{128}$$

donde d_c es distancia de nube de cola, d_r distancia relativa de rangos, d_f diferencia relativa de FFT y d_s distancia de secciÃ³n. Con los valores por defecto, se acepta target sÃ³lo si se satisfacen simultÃ¡neamente umbrales como:

$$S \le 0.95, \qquad d_c \le 0.85, \qquad d_r \le 1.50, \qquad d_f \le 1.00, \qquad d_s \le 1.20. \tag{129}$$

Estos umbrales son contratos operativos de clasificaciÃ³n, no teoremas.

32. AnÃ¡lisis espectral, Lyapunov, bifurcaciones y Matignon

32.1. FFT dominante y entropÃa espectral

La API activa calcula FFT sobre la cola de la trayectoria. Para una serie x_n, primero se remueve la media y se aplica ventana de Hann:

$$\tilde{x}_n = (x_n - \bar{x}) w_n. \tag{130}$$

Luego

$$S_k = |\,\mathrm{rFFT}(\tilde{x})_k|^2. \tag{131}$$

La frecuencia dominante es

$$f_{\max} = f_{k^*}, \qquad k^* = \arg\max_{k \ge 1} S_k. \tag{132}$$

La entropÃa espectral normalizada es

$$H = -\frac{1}{\log N_f} \sum_{k \ge 1} p_k \log(p_k + \varepsilon), \qquad p_k = \frac{S_k}{\sum_{j \ge 1} S_j}. \tag{133}$$

El campo correspondiente en los CSV suele aparecer como `fft_peak` o `fft_dominant_frequency`; la entropÃa aparece como `psd_entropy`, aunque en la API activa puede provenir de la FFT con ventana y no necesariamente de Welch.

32.2. PSD de Welch

La PSD de Welch estÃ¡ conservada como anÃ¡lisis opcional en el pipeline legado. Se activa con:

```
hidden-attractors-unified-chua --run-mode balanced --spectral --psd
```

En Linux/macOS:

```
python -m hidden_attractors.workflows.unified_chua --run-mode balanced --spectral --psd
```

La lectura correcta es: FFT y PSD caracterizan contenido frecuencial y ayudan a comparar persistencia espectral, pero no prueban caos ni ocultedad.

32.3. Exponentes de Lyapunov

El repositorio conserva un backend C tipo Benettin para estimaciÃ³n operacional de exponentes de Lyapunov. Se activa desde el pipeline legado con:

```
hidden-attractors-unified-chua --run-mode balanced --lyapunov
```

Para hacer que un fallo detenga la corrida:

```
hidden-attractors-unified-chua --run-mode balanced --lyapunov --lyapunov-strict
```

La interpretaciÃ³n mÃnima es:

$$\lambda_{\text{máx}} > 0 \quad \Rightarrow \quad \text{evidencia operacional de sensibilidad caÃ³tica bajo el contrato probado.} \tag{134}$$

No sustituye la prueba de ocultedad, porque la ocultedad es una propiedad de la cuenca respecto de los equilibrios, no sÃ³lo de la dinÃ¡mica sobre el atractor.

32.4. Diagramas de bifurcaciÃ³n

La API activa contiene post-procesamiento para diagramas de bifurcaciÃ³n. No hace continuaciÃ³n de ramas en sentido estricto; toma trayectorias ya integradas para distintos parÃ¡metros y extrae mÃ¡ximos, mÃnimos o muestras de una observable. MatemÃ¡ticamente, para cada parÃ¡metro ρ_i, se conserva una colecciÃ³n

$$\mathcal{B}_i = \{x(t_k;\rho_i) : t_k \geq t_{\text{burn}},\ x(t_k) \text{ mÃ¡ximo local}\}. \tag{135}$$

La funciÃ³n pÃºblica es:

```
from hidden_attractors.analysis import bifurcation_points_from_trajectories
from hidden_attractors.plotting import plot_bifurcation_diagram
```

Modos disponibles:

- `maxima`: mÃ¡ximos locales;
- `minima`: mÃnimos locales;
- `both`: mÃ¡ximos y mÃnimos;
- `sample`: muestra directa de la cola.

32.5. Equilibrios y grÃ¡fica Matignon

El script de anÃ¡lisis de equilibrios calcula para cada equilibrio:

- regiÃ³n: central, saturaciÃ³n izquierda, saturaciÃ³n derecha o frontera de conmutaciÃ³n;
- autovalores del Jacobiano local;
- margen fraccionario

$$\mu_{\text{Matignon}} = \min_i \left(|\arg(\lambda_i)| - \frac{q\pi}{2} \right); \tag{136}$$

- etiqueta `matignon_stable`.

El archivo producido es `equilibria_summary.csv`. Una gráfica defendible debe mostrar, para cada equilibrio, si

$$\mu_{\text{Matignon}} > 0 \tag{137}$$

o no. Un ejemplo mínimo para generar una grÃ¡fica a partir del CSV es:

```
import csv
import matplotlib.pyplot as plt

rows = list(csv.DictReader(open('equilibria_summary.csv', newline='')))
labels = [r['eq_id'] for r in rows]
margins = [float(r['min_arg_margin']) for r in rows]

plt.figure(figsize=(6,4))
plt.axhline(0.0, color='k', lw=0.8)
plt.bar(labels, margins)
plt.ylabel(r'$\min_i(|\arg\lambda_i|-q\pi/2)$')
plt.xlabel('equilibrio')
plt.tight_layout()
plt.savefig('matignon_equilibria_margin.png', dpi=220)
```

Esta figura no dice si el atractor es oculto; sÃ³lo clasifica estabilidad local de los equilibrios.

33. Inventario de funciones disponibles, activas y opcionales

El repositorio contiene una API activa en `hidden_attractors/` y scripts legacy todavÃa Ãºtiles en `tools/legacy/`. La siguiente tabla resume quÃ© existe y cÃ³mo debe leerse.

Bloque	Estado	FunciÃ³n disponible
`models.chua`	API activa	ParÃ¡metros, campo vectorial y equilibrios del Chua no suave.
`analysis.trajectory`	API activa	Rangos, varianza, FFT, entropÃa espectral, secciones, distancias de nubes y casos de robustez.
`analysis.bifurcation`	API activa	ExtracciÃ³n de puntos para diagramas de bifurcaciÃ³n desde trayectorias ya calculadas.
`plotting.dynamics`	API activa	Retratos de fase, proyecciones, series temporales y diagramas de bifurcaciÃ³n.
`basins.classification`	API activa	Etiquetas `equilibrium`, `target_positive`, `target_negative`, `infinity`, `unknown`, `numerical_failure`.
`workflows.robustness_overlay`	CLI mantenida	SuperposiciÃ³n de trayectorias al variar h, L_m, T.
`workflows.sphere_controls`	CLI mantenida	Pruebas desde esferas alrededor de todos los equilibrios y robustez del candidato.
`workflows.refined_basin`	CLI mantenida	ReclasificaciÃ³n de celdas `unknown` mediante geometrÃa contra referencias target.
`seed_generation`	API activa ligera	DF centrada, semilla sesgada y semillas Machado sin variables de entorno.
`solvers.FractionalHistory`	API activa ligera	Ventana de memoria EFORK; la integracion pesada se ejecuta con backend C.

Bloque	Estado	FunciÃ³n disponible
`workflows.unified_chua`	CLI/API mantenida	Entrada publica sin `\$env`; delega calculos pesados al backend C.
`integrations.external_tools`	API activa opcional	Adaptadores para `nolds`, `antropy`, PyDSTool y referencias externas.
`tools/legacy/run_extended_search.py`	Legacy Ãºtil	BÃºsqueda centrada, sesgada, Machado, ρ_H, continuaciÃ³n multiparamÃ©trica y nubes de semillas.
`tools/legacy/unified_nyquist_hidden_pipeline.py`	Legacy amplio	Pipeline completo con modos `balanced`, `fast`, `basin_only`, `q_sweep`, `df_compare`, `machado_sweep`, `machado_sweep_fast`.
`tools/legacy/danca2017_chua_abm_replication.py`	Legacy de referencia	RÃ©plica ABM de Danca y comparaciÃ³n contra candidatos del proyecto.
`tools/legacy/positive_x_basin_sweep.py`	Pendiente de migraciÃ³n	Barrido de cuenca en regiÃ³n $x > 0$, especialmente alrededor de E_+.
`tools/legacy/chua_frac_lyapunov_efork_benettin.c`	Opcional nativo	EstimaciÃ³n operacional de exponentes de Lyapunov.

33.1. Funciones opcionales desactivadas por defecto

Varias capacidades existen pero no siempre se ejecutan en una corrida normal:

Capacidad	ActivaciÃ³n tÃpica	Salida esperada
PSD de Welch	`hidden-attractors-unified-chua --psd`	`psd_summary.json`, figuras PSD.
ExportaciÃ³n TISEAN	`hidden-attractors-unified-chua --export-tisean`	Series exportadas para anÃ¡lisis externo.
Lyapunov operativo	`hidden-attractors-unified-chua --lyapunov`	Resumen de exponentes y convergencia.
Lyapunov estricto	`hidden-attractors-unified-chua --lyapunov-strict`	Falla si el cálculo estricto falla.
Planos extra de cuenca	`hidden-attractors-unified-chua --basin-planes`	Cortes xy, xz, yz.
Cuenca 3D o vistas extra	`pendienteC`	Figuras adicionales.
IlustraciÃ³n de ocultedad	`hidden-attractors-unified-chua --hidden-illustration`	Figura de ocultedad target/no-target.
Sweep de q	`hidden-attractors-unified-chua --mode q_sweep`	`q_sweep/`
ComparaciÃ³n DF/Machado	`hidden-attractors-unified-chua --mode df_compare`	`df_compare/`
Sweep Machado completo	`hidden-attractors-unified-chua --mode machado_sweep`	`machado_sweep/`
Sweep Machado rÃ¡pido	`hidden-attractors-unified-chua --mode machado_sweep_fast`	`machado_sweep_fast/`

34. Ruta completa recomendada actualizada

Para el caso Chua fraccionario no suave se recomienda ejecutar y documentar en este orden:

1. Fijar sistema, parÃ¡metros, $q, h, L_m, T, t_{\text{burn}}$ y versiÃ³n de cÃ³digo.
2. Calcular equilibrios, Jacobianos, autovalores y margen de Matignon.
3. Generar grÃ¡fica de mÃ¡rgenes de Matignon por equilibrio.
4. Ejecutar `q_sweep` para detectar Ã³rdenes q prometedores.
5. Ejecutar `df_compare` para comparar semilla clÃ¡sica y Machado.
6. Ejecutar `machado_sweep_fast`; si hay candidatos, ejecutar `machado_sweep` completo.
7. Ejecutar `run_extended_search.py` para combinar centradas, sesgadas, Machado, ρ_H, continuaciÃ³n y nubes de semillas.
8. Filtrar candidatos por `residual_abs`, ρ_H, no divergencia, no convergencia a equilibrio y geometrÃa acotada no trivial.
9. Reintegrar candidatos seleccionados con tiempo largo.
10. Calcular FFT, entropÃa espectral y, si se activa, PSD de Welch.
11. Calcular exponentes de Lyapunov si se requiere evidencia adicional de caos.
12. Construir diagramas de bifurcaciÃ³n para parÃ¡metros relevantes $q, \alpha, \beta, \gamma, m_0, m_1$ si hay trayectorias por barrido.
13. Ejecutar `robustness_overlay` para h, L_m, T.
14. Ejecutar `sphere_controls` para todos los equilibrios, radios y muestras.
15. Generar cuencas xy, y si estÃ¡ habilitado, xz, yz o cortes locales.
16. Ejecutar `refined_basin` si hay muchas celdas `unknown`.
17. Comparar todo contra la rÃ©plica ABM de Danca, separando diferencias de mÃ©todo: ABM con historia completa frente a EFORK con memoria truncada.
18. Redactar conclusiÃ³n con nivel de evidencia, sin promover a `hidden_verified` salvo que no haya contacto target desde vecindades de todos los equilibrios bajo los contratos probados.

Referencias

[1] M. F. Danca. *Hidden chaotic attractors in fractional-order systems.* Nonlinear Dynamics, 89(1), 2017, 577. DOI relacionado: `10.1007/s11071-017-3472-7`. Preprint: `arXiv:1804.10769`.

[2] M. Caputo. *Linear Models of Dissipation whose Q is almost Frequency Independent-II.* Geophysical Journal International, 1967.

[3] D. Matignon. *Stability Results for Fractional Differential Equations with Applications to Control Processing.* Computational Engineering in Systems Applications, 1996.

[4] K. Diethelm, N. J. Ford, and A. D. Freed. *A Predictor-Corrector Approach for the Numerical Solution of Fractional Differential Equations.* Nonlinear Dynamics, 2002.

[5] G. A. Leonov and N. V. Kuznetsov. *Hidden Attractors in Dynamical Systems: From Hidden Oscillations in Hilbert-Kolmogorov, Aizerman, and Kalman Problems to Hidden Chaotic Attractor in Chua Circuits*. International Journal of Bifurcation and Chaos, 2013.

[6] M. S. Tavazoei and M. Haeri. *A Proof for Non Existence of Periodic Solutions in Time Invariant Fractional Order Systems*. Automatica, 2009.

[7] J. Tenreiro Machado. *Fractional order describing functions*. Signal Processing, 2015.