Protocolo secuencial de pruebas

Caso de estudio: Chua fraccionario no suave y comparación con Danca

Hidden Attractors Fractional Order

Versión de trabajo: 20 de mayo de 2026

Índice

1. Objetivo y alcance **2**

2. Preparación del entorno **2**

3. Definición del sistema de Chua fraccionario no suave **3**

4. Etapa 1: equilibrios y estabilidad local **3**

5. Etapa 2: atractor de referencia para parámetros dados **4**

6. Etapa 3: transferencia fraccionaria y balance armónico **4**
6.1. Puente Weyl–Caputo . . . 5

7. Etapa 4: cálculo de semillas iniciales **5**
7.1. Semilla centrada clásica . . . 5
7.2. Semilla sesgada . . . 5
7.3. Semilla tipo Machado . . . 6

8. Etapa 5: diagnósticos armónicos y filtro ρ_H **6**

9. Etapa 6: modos sweep **6**
9.1. Barrido del orden fraccionario . . . 7
9.2. Comparación clásica contra Machado . . . 7
9.3. Barrido Machado completo y rápido . . . 7

10. Etapa 7: continuación numérica **7**

11. Etapa 8: filtrado de candidatos finales **8**

12. Etapa 9: construcción y replicabilidad del target **8**
12.1. Referencia target . . . 8
12.2. Estimadores usados para decidir si una trayectoria reproduce el target . . . 9
12.3. Prueba de hit de target por sección . . . 9

13. Etapa 10: robustez del candidato **9**
13.1. Casos estándar de robustez . . . 9
13.2. Criterio de lectura . . . 10

14. Etapa 11: pruebas de ocultedad desde vecindades de equilibrios **10**
14.1. Replicabilidad de hits target desde equilibrios . . . 10

15. Etapa 12: cuencas de atracción **11**

16. Etapa 13: análisis complementarios **11**
16.1. FFT y PSD . . . 11
16.2. Exponentes de Lyapunov . . . 11
16.3. Diagramas de bifurcación . . . 12
16.4. Exportación TISEAN y medidas externas . . . 12

17. Etapa 14: comparación con Danca **12**

18. Etapa 15: criterio final de decisión **12**

19. Resumen de comandos en orden de ejecución **13**

20. Inventario funcional: disponible, activable o legacy **14**

21. Advertencia final **14**

1. Objetivo y alcance

Este documento fija el orden de trabajo para estudiar candidatos a atractores ocultos en el sistema de Chua fraccionario no suave. El flujo se organiza de forma secuencial:

$$\text{modelo} \to \text{equilibrios} \to \text{semillas} \to \text{sweep} \to \text{continuación}$$
$$\to \text{filtrado} \to \text{replicabilidad del target} \to \text{robustez} \to \text{ocultedad}$$
$$\to \text{diagnósticos complementarios} \to \text{comparación Danca}.$$

La función descriptiva, el balance armónico y la extensión tipo Machado se usan como generadores heurísticos de semillas. La verificación final debe hacerse con simulaciones causales de Caputo/EFORK o ABM y con pruebas de cuencas alrededor de todos los equilibrios.

2. Preparación del entorno

Trabajar desde la carpeta activa de la librería:

```
cd version_2
python -m pip install -e .
python -m compileall hidden_attractors examples tests tools/cli
python examples/quickstart_equilibria.py
python examples/list_final_candidates.py
```

El caso de estudio se fija desde objetos de libreria:

```
from hidden_attractors.models import chua_piecewise_parameters
from hidden_attractors.seed_generation import validate_fractional_order

params = chua_piecewise_parameters()
q = validate_fractional_order(0.9998)
```

Comando de compatibilidad para ver la llamada completa del pipeline sin usar variables de entorno:

```
hidden-attractors-unified-chua --model piecewise --q 0.9998 --print-command
```

3. Definición del sistema de Chua fraccionario no suave

El sistema usado como caso de estudio es

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

con $0 < q \leq 1$. La no linealidad no suave es

$$
f(x) = m_1 x + (m_0 - m_1)\,\mathrm{sat}(x), \qquad \mathrm{sat}(x) = \begin{cases} -1, & x < -1,\\ x, & |x| \leq 1,\\ 1, & x > 1. \end{cases} \tag{4}
$$

Los parámetros de referencia de Danca son

$$
\begin{aligned}
\alpha = 8.4562, \quad \beta = 12.0732, \quad \gamma = 0.0052,\\
m_0 = -0.1768, \quad m_1 = -1.1468, \quad q = 0.9998.
\end{aligned} \tag{5}
$$

En forma Lur'e se separa

$$
{}^{C}D_t^q X = PX + b\psi(r^T X), \qquad r^T X = x, \tag{6}
$$

con

$$
P = \begin{bmatrix} -\alpha(1+m_1) & \alpha & 0\\ 1 & -1 & 1\\ 0 & -\beta & -\gamma \end{bmatrix}, \qquad b = \begin{bmatrix} -\alpha\\ 0\\ 0 \end{bmatrix}, \qquad r = \begin{bmatrix} 1\\ 0\\ 0 \end{bmatrix}, \qquad \psi(\sigma) = (m_0 - m_1)\,\mathrm{sat}(\sigma). \tag{7}
$$

4. Etapa 1: equilibrios y estabilidad local

En equilibrio se impone

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

De las dos últimas ecuaciones,

$$
y = \frac{\gamma}{\beta + \gamma} x, \qquad z = -\frac{\beta}{\beta + \gamma} x. \tag{8}
$$

Sustituyendo en la primera,

$$
f(x) = -\frac{\beta}{\beta + \gamma} x. \tag{9}
$$

Por tanto, siempre aparece el equilibrio central

$$
E_0 = (0, 0, 0). \tag{10}
$$

En las regiones saturadas se obtienen

$$
E_+ = \left(x_+, \frac{\gamma}{\beta + \gamma} x_+, -\frac{\beta}{\beta + \gamma} x_+ \right), \qquad E_- = -E_+, \tag{11}
$$

con

$$
x_+ = -\frac{(\beta + \gamma)(m_0 - m_1)}{(\beta + \gamma) m_1 + \beta}. \tag{12}
$$

Estos puntos se aceptan sí sólo si pertenecen a su región: $x_+ > 1$, $x_- < -1$.

El Jacobiano regional es

$$J(a) = \begin{bmatrix} -\alpha(1+a) & \alpha & 0 \\ 1 & -1 & 1 \\ 0 & -\beta & -\gamma \end{bmatrix}, \tag{13}$$

donde $a = m_0$ en la región central y $a = m_1$ en las regiones externas, bajo la convención del modelo anterior. Para un sistema fraccionario conmensurado, el criterio de Matignon exige

$$|\arg(\lambda_i(J))| > \frac{q\pi}{2} \qquad \forall i. \tag{14}$$

Comando recomendado:

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

Salida esperada: puntos E_0, E_+, E_-, autovalores, margen mínimo de Matignon y clasificación local. La gráfica recomendada compara cada autovalor con las rectas frontera

$$\arg(\lambda) = \pm\frac{q\pi}{2}. \tag{15}$$

5. Etapa 2: atractor de referencia para parámetros dados

Antes de buscar ocultedad se debe verificar que el sistema produce una trayectoria acotada no trivial con los parámetros fijados. Para usar una trayectoria ya calculada:

```
python examples/dynamical_analysis_gallery.py ^
  --trajectory-csv outputs/.../trajectory.csv ^
  --output-dir outputs/examples/chua_case_gallery
```

Las figuras mínimas son: atractor 3D, proyecciones xy, xz, yz, series temporales y sección. Esta etapa no prueba ocultedad; sí sólo verifica que existe diámica no trivial para una condición inicial dada.

6. Etapa 3: transferencia fraccionaria y balance armónico

f La transferencia del bloque lineal fraccionario se escribe como

$$W_q(s) = r^T(s^q I - P)^{-1} b. \tag{16}$$

Equivalente, usando el parámetro espectral,

$$\widehat{W}_q(\lambda) = r^T(\lambda I - P)^{-1} b, \qquad \lambda = s^q. \tag{17}$$

En frecuencia no se usa $\lambda = \mathrm{i}\omega$, sino

$$\lambda = (\mathrm{i}\omega)^q = \omega^q \left(\cos\frac{q\pi}{2} + \mathrm{i}\sin\frac{q\pi}{2}\right). \tag{18}$$

Para el balance armónico centrado se propone

$$\sigma(t) = A\cos(\omega t). \tag{19}$$

La salida no lineal se expande como

$$\psi(A\cos\theta) = \sum_{k\geq 1} Y_k \cos(k\theta + \phi_k), \qquad \theta = \omega t. \tag{20}$$

La función descriptiva clásica toma sí sólo el primer armónico:

$$N(A) = \frac{Y_1 e^{\mathrm{i}\phi_1}}{A}. \tag{21}$$

El cierre armónico queda

$$1 + \widehat{W}_q((\mathrm{i}\omega)^q)N(A) = 0. \tag{22}$$

Separando parte imaginaria y real:

$$\Im\left\{\widehat{W}_q((\mathrm{i}\omega)^q)N(A)\right\} = 0, \tag{23}$$

$$\Re\left\{\widehat{W}_q((\mathrm{i}\omega)^q)N(A)\right\} = -1. \tag{24}$$

6.1. Puente Weyl–Caputo

El balance armónico se interpreta como una aproximació estacionaria tipo Weyl. El sistema real integrado es de Caputo, causal, con memoria desde t_0. Por tanto, la solución armÃónica no demuestra un ciclo periódico exacto de Caputo; produce una semilla inicial para simulación y continuación.

7. Etapa 4: cálculo de semillas iniciales

7.1. Semilla centrada clásica

Se buscan frecuencias ω_j tales que

$$\Im\widehat{W}_q((\mathrm{i}\omega_j)^q) = 0. \tag{25}$$

Luego se define

$$k_j = -\frac{1}{\Re\widehat{W}_q((\mathrm{i}\omega_j)^q)}. \tag{26}$$

Si existe A_j tal que

$$N(A_j) = k_j, \tag{27}$$

se calcula el vector armónico v_j resolviendo

$$(P_0 - (\mathrm{i}\omega_j)^q I)v_j = 0, \qquad r^T v_j = 1, \tag{28}$$

con

$$P_0 = P + kbr^T. \tag{29}$$

La semilla queda

$$X_{\text{seed}}(\theta) = A_j \Re\{v_j e^{\mathrm{i}\theta}\}. \tag{30}$$

Llamada de libreria:

```
from hidden_attractors.seed_generation import find_harmonic_seed

seed = find_harmonic_seed(q=0.9998, branch_index=0, method="classic")
print(seed.seed, seed.omega, seed.gain, seed.amplitude)
```

7.2. Semilla sesgada

La función descriptiva sesgada usa

$$\sigma(t) = \sigma_0 + A\cos(\omega t). \tag{31}$$

Se calculan numéricamente los coeficientes de Fourier de

$$y(\theta) = \psi(\sigma_0 + A\cos\theta), \tag{32}$$

con

$$Y_k = a_k - \mathrm{i}b_k, \qquad N(A, \sigma_0) = \frac{Y_1}{A}. \tag{33}$$

Para reconstruir la semilla se resuelve una parte constante y una parte armónica:

$$P\bar{X} + b\bar{y} = 0, \qquad r^T\bar{X} = \sigma_0, \tag{34}$$

$$((\mathrm{i}\omega)^q I - P)V = bY_1, \qquad r^T V = A. \tag{35}$$

Entonces

$$X_{\mathrm{seed}}(\theta) = \bar{X} + \Re\{Ve^{\mathrm{i}\theta}\}. \tag{36}$$

Llamada de libreria:

```
from hidden_attractors.seed_generation import reconstruct_biased_lure_seed

biased = reconstruct_biased_lure_seed(
    q=0.9998,
    amplitude=5.0,
    sigma0=0.0,
    omega=2.04,
)
```

7.3. Semilla tipo Machado

La extensión tipo Machado usa una función descriptiva auxiliar

$$N_\mu(A) = \exp\{\mu \operatorname{Log} N(A)\}, \qquad \mu > 0. \tag{37}$$

En la implementación real para el Chua no suave se usa la rama real cuando $N(A) > 0$:

$$N_\mu(A) = N(A)^\mu. \tag{38}$$

La amplitud se obtiene de

$$N(A_\mu)^\mu = k_j. \tag{39}$$

La semilla final conserva la forma

$$X^{(\mu)}_{\mathrm{seed}}(\theta) = A_\mu \Re\{v_j e^{\mathrm{i}\theta}\}. \tag{40}$$

El parámetro μ no es el orden fraccionario q del sistema de Caputo; sí sólo modifica la familia de semillas.

Llamada de libreria:

```
from hidden_attractors.seed_generation import find_harmonic_seed

machado = find_harmonic_seed(q=0.9998, method="machado", mu=2.0)
```

8. Etapa 5: diagnósticos armónicos y filtro ρ_H

Para evitar semillas dominadas por armónicos superiores se calcula

$$\rho_H = \frac{\sum_{k=2}^{K} |\widehat{W}_q((\mathrm{i}k\omega)^q)|\,|Y_k|}{|\widehat{W}_q((\mathrm{i}\omega)^q)|\,|Y_1| + \varepsilon}. \tag{41}$$

La semilla es más confiable si ρ_H es pequeño. El umbral usado en configuraciones rápidas suele ser

$$\rho_H < 0.1. \tag{42}$$

Este criterio no verifica ocultedad; sí sólo descarta semillas armónicas pobres.

9. Etapa 6: modos sweep

Los barridos se usan antes de invertir tiempo en verificación larga.

9.1. Barrido del orden fraccionario

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

Salida principal:

```
q_order_sweep/q_sweep_summary.csv
q_order_sweep/q_sweep_summary.json
```

Uso: detectar valores de q donde hay cruces Nyquist, semillas con buena continuación y atractores persistentes.

9.2. Comparación clásica contra Machado

```
hidden-attractors-unified-chua ^
  --run-mode df_compare ^
  --model piecewise ^
  --machado-mu 0.5
```

Salida principal:

```
df_seed_comparison/df_seed_comparison_summary.csv
```

Uso: comparar si la semilla clásica o una semilla Machado produce mejor continuación.

9.3. Barrido Machado completo y rápido

```
hidden-attractors-unified-chua ^
  --run-mode machado_sweep_fast ^
  --model piecewise ^
  --output-dir runs_machado_sweep_fast
```

El barrido completo usa más valores de μ y más fases θ. El barrido rápido reduce la malla. En ambos casos se exploran ramas, órdenes descriptivos auxiliares y fases.

10. Etapa 7: continuación numérica

La continuación transporta una semilla desde un sistema auxiliar hacia el sistema objetivo. La familia implementada usa

$$ {}^{C}D_t^q X = PX + b\psi_\eta(r^T X), \qquad \eta \in [0,1], \tag{43} $$

donde $\eta = 0$ representa una aproximación suavizada o auxiliar y $\eta = 1$ representa el sistema no suave original.

La continuación multiparámetro permite variar, según la ruta,

$$ \eta, imesq, imes\chi, \tag{44} $$

donde χ puede ser σ_0 para la ruta sesgada o μ para la ruta Machado.

En sistemas de Caputo no se debe reiniciar cada etapa sí sólo con el último punto. Se transporta una ventana discreta de memoria:

$$ \mathcal{H}_k = X(t_k - M), \ldots, X(t_k). \tag{45} $$

En el código esta ventana aparece como `FractionalHistory` o como una historia EFORK con columnas t, x, y, z.

La extraccion de la ventana es API publica y ligera; la integracion pesada se ejecuta con backend C:

```
from hidden_attractors.solvers import FractionalHistory

history = FractionalHistory.from_trajectory(
    trajectory,
    q=0.9998,
    h=0.01,
    memory_length=8.0,
)
efork_history = history.as_efork_history()
```

Comando de búsqueda extendida:

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

Salidas principales:

```
rho_H_diagnostics.csv
biased_df_candidates.csv
biased_df_all_evaluations.csv
continuation_summary.csv
continuation_paths.csv
seed_cloud_summary.csv
extended_search_report.md
```

11. Etapa 8: filtrado de candidatos finales

Un candidato pasa al bloque de robustez si cumple, como mínimo:

1. residual armónico bajo;
2. ρ_H aceptable;
3. semilla finita;
4. continuación sobreviviente hasta $\eta = 1$;
5. trayectoria acotada;
6. no convergencia inmediata a equilibrio;
7. rango o varianza postransitoria no colapsada.

Una etiqueta como `candidate\_bounded\_nontrivial` significa que existe un candidato dinámico para verificar. No significa `hidden\_verified`.

12. Etapa 9: construcción y replicabilidad del target

Antes de preguntar si el atractor es oculto, se debe comprobar si el atractor candidato se reproduce como target. La idea es fijar una referencia $\mathcal{A}_\star$, obtenida desde la semilla candidata o desde una corrida previa, y después comparar otras trayectorias contra ella.

12.1. Referencia target

Sea $X_\star(t)$ la trayectoria generada desde la semilla candidata. Se descarta el transitorio y se define una nube de referencia

$$\mathcal{C}_\star = \{X_\star(t_k) : t_k \geq t_{\text{burn}}\}. \tag{46}$$

También se construye una sección tipo Poincaré:

$$\Sigma_\star = \{(y, z) : x = 0, \dot{x} > 0, t \geq t_{\text{burn}}\}. \tag{47}$$

12.2. Estimadores usados para decidir si una trayectoria reproduce el target

Dada otra trayectoria $X_i(t)$, se calculan los estimadores:

$$\Delta_{\text{range}} = \frac{\|R_i - R_\star\|}{\|R_\star\| + \varepsilon}, \tag{48}$$

$$\Delta_{\text{FFT}} = \frac{|f_i - f_\star|}{|f_\star| + \varepsilon}, \tag{49}$$

$$d_{\text{cloud}} = \text{median}\{d_{NN}(\mathcal{C}_i, \mathcal{C}_\star), d_{NN}(\mathcal{C}_\star, \mathcal{C}_i)\}, \tag{50}$$

$$d_\Sigma = \text{median}\{d_{NN}(\Sigma_i, \Sigma_\star), d_{NN}(\Sigma_\star, \Sigma_i)\}. \tag{51}$$

Además se revisa:

$$\text{bounded} = 1, \qquad \text{diverged} = 0, \qquad \text{equilibrium\_like} = 0, \qquad \text{noncollapsed\_variance} = 1. \tag{52}$$

La trayectoria reproduce el target si permanece acotada, no colapsa a equilibrio y sus métricas geométricas/espectrales son compatibles con la referencia. Esta comparación es estadística y geométrica; no se exige coincidencia punto a punto porque el sistema es caótico.

12.3. Prueba de hit de target por sección

En las pruebas refinadas se usa una regla adicional basada en secciones. Para una trayectoria de prueba se calcula

$$h_\Sigma = \frac{\#\{p \in \Sigma_i : \text{dist}(p, \Sigma_\star) \le \varepsilon_\Sigma\}}{\#\Sigma_i}. \tag{53}$$

Si

$$\#\Sigma_i \ge N_{\text{mín}}, \qquad h_\Sigma \ge h_{\text{mín}}, \tag{54}$$

la trayectoria se etiqueta como `target\_attractor`. En configuraciones usadas por el proyecto aparecen valores como $N_{\text{mín}} = 20$ y $h_{\text{mín}} = 0.70$, pero deben leerse desde el archivo YAML de cada corrida.

13. Etapa 10: robustez del candidato

La robustez responde: ¿él candidato sigue llegando al mismo target bajo cambios del contrato numérico? No responde todavía si el atractor es oculto.

13.1. Casos estándar de robustez

El conjunto estándar compara:

Caso	Cambio	Propósito
`R0\_base`	contrato base	referencia del candidato
`R1\_h\_finer`	h menor	sensibilidad a paso fino
`R2\_h\_coarser`	h mayor	sensibilidad a paso grueso
`R3\_Lm\_lower`	L_m menor	sensibilidad a memoria corta
`R4\_Lm\_higher`	L_m mayor	sensibilidad a memoria larga
`R5\_t\_longer`	T mayor	persistencia temporal

Comando:

```
python tools/cli/robustness_overlay_c_trajectories.py --job launch
```

Salidas:

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

13.2. Criterio de lectura

Para cada caso R_i se compara contra R_0. El candidato es robusto si:

1. todas las trayectorias relevantes son acotadas;
2. ninguna colapsa a equilibrio;
3. las varianzas postransitorias no colapsan;
4. la distancia relativa de rangos es pequeña;
5. la frecuencia dominante no cambia de forma incompatible;
6. la distancia de nubes y de secciones permanece controlada;
7. la superposición visual conserva la misma geometría global.

Una falla de robustez no prueba que el candidato sea falso, pero bloquea una afirmación fuerte. Una robustez positiva tampoco prueba ocultedad.

14. Etapa 11: pruebas de ocultedad desde vecindades de equilibrios

Después de demostrar que el target es reproducible, se prueba si su cuenca intersecta vecindades de algún equilibrio. Para cada equilibrio E_i se generan condiciones iniciales

$$X_0 = E_i + \rho u, \qquad \|u\| = 1 \tag{55}$$

o dentro de la bola

$$X_0 = E_i + \rho r^{1/3} u, \qquad 0 \le r \le 1. \tag{56}$$

Si alguna trayectoria desde una vecindad de equilibrio reproduce el target, entonces el candidato no debe declararse oculto bajo ese contrato.

Comando API pública:

```
python tools/cli/lure_top3_sphere_robustness.py --help
```

Comando legacy refinado:

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

14.1. Replicabilidad de hits target desde equilibrios

Un solo hit target puede ser un evento aislado. Por eso se marca `robust\_target\_hit` cuando el contacto con el target se reproduce bajo alguno de estos criterios:

1. aparecen al menos tres trayectorias `target\_attractor` en el mismo grupo;
2. el hit aparece tanto con muestreo tipo `ball` como con `sphere\_shell`;
3. el hit se reproduce al cambiar h;

4. el hit se reproduce al cambiar L_m;

5. el hit persiste en una etapa de mayor tiempo final.

Si `robust\_target\_hit=True`, la decisión correcta es bloquear la etiqueta de ocultedad. Si sólo hay hits aislados, la decisión debe ser `inconclusive\_isolated\_hit`. Si no hay hits en los radios probados, la conclusión máxima es `compatible\_with\_hiddenness\_under\_tested\_radii`, no `hidden\_verified`.

15. Etapa 12: cuencas de atracción

Las cuencas se calculan después de la robustez y antes de la discusión final. Deben incluir cortes:

$$xy, \qquad xz, \qquad yz, \tag{57}$$

y, si está habilitado, visualizaciones 3D.

Las clases típicas son:

Clase	Interpretación
0	equilibrio
1	target positivo
2	target negativo
3	infinito/divergente
4	desconocido
5	falla numérica

Las celdas desconocidas se refinan con `refined\_basin`, comparando trayectorias contra referencias target mediante nubes, secciones, rangos y frecuencia dominante.

16. Etapa 13: análisis complementarios

Estos análisis ayudan a caracterizar el candidato, pero no deciden ocultedad por sí solos.

16.1. FFT y PSD

FFT dominante:

$$f_{\text{peak}} = \arg\max_{f>0} |\mathcal{F}[x(t)]|^2. \tag{58}$$

Entropía espectral normalizada:

$$H_s = -\frac{\sum_k p_k \log(p_k + \varepsilon)}{\log N}. \tag{59}$$

Activar PSD de Welch:

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

16.2. Exponentes de Lyapunov

Activar el estimador operacional:

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

La lectura mínima es:

$$\lambda_{\text{máx}} > 0 \quad \Rightarrow \quad \text{evidencia numérica de sensibilidad caótica.} \tag{60}$$

Para sistemas fraccionarios con memoria, este valor debe reportarse junto con $q, h, L_m, T, t_{\text{burn}}$ y el backend usado.

16.3. Diagramas de bifurcación

Los diagramas se hacen por post-procesamiento de trayectorias. Para un parámetro ρ, se extraen máximos, mínimos o muestras postransitorias:

$$(\rho_j, x_{\text{máx},k}), \qquad (\rho_j, x_{\text{mín},k}), \qquad (\rho_j, x(t_k)). \tag{61}$$

No son continuación formal de ramas; sólo resumen cambios de observables.

16.4. Exportación TISEAN y medidas externas

Si se activa exportación externa:

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

Las medidas externas como entropía, dimensión de correlación, DFA o Lyapunov por series temporales deben citar el paquete usado y no mezclarse con el criterio de ocultedad.

17. Etapa 14: comparación con Danca

La comparación con Danca debe mantenerse separada de la ruta EFORK del proyecto.

Danca usa ABM predictor-corrector con historia completa de Caputo. La réplica local se ejecuta con:

```
python tools/legacy/danca2017_chua_abm_replication.py ^
  --job all --workers 4 --q 0.9998 --h 0.05 ^
  --t-final 500 --transient 250 --delta 0.01
```

La comparación debe reportar:

1. mismos parámetros $(\alpha, \beta, \gamma, m_0, m_1, q)$;
2. método numérico: ABM completo contra EFORK con memoria truncada;
3. condición inicial usada para la réplica;
4. geometría del atractor;
5. comportamiento desde vecindades de equilibrios;
6. si la cuenca del target intersecta o no vecindades de equilibrio;
7. diferencias debidas al contrato numérico.

18. Etapa 15: criterio final de decisión

La conclusión debe seguir esta tabla:

Resultado observado	Lectura permitida
La semilla no continúa o diverge	candidato descartado
Trayectoria acotada, pero no reproduce target bajo robustez	candidato no robusto
Target reproducible, pero hay `robust\_target\_hit` desde equilibrio	no soporta ocultedad; candidato autoexcitado bajo el contrato probado
Target reproducible, sin hits target en radios probados	compatible con ocultedad bajo radios probados; no es prueba definitiva
Cuencas, esferas, robustez y comparación Danca son consistentes	candidato fuerte; aún debe reportarse como evidencia numérica, no teorema

19. Resumen de comandos en orden de ejecución

```
# 1. Instalacion y prueba minima
cd version_2
python -m pip install -e .
python examples/quickstart_equilibria.py

# 2. Sweep exploratorio
hidden-attractors-unified-chua --run-mode q_sweep --q-values 0.97,0.98,0.985,0.99,0.995,1.0

# 3. Comparacion de DF clasica/Machado
hidden-attractors-unified-chua --run-mode df_compare --machado-mu 0.5

# 4. Sweep Machado rapido
hidden-attractors-unified-chua --run-mode machado_sweep_fast

# 5. Busqueda extendida
python tools/legacy/run_extended_search.py --config configs/chua_fractional_nonsmooth.yaml

# 6. Robustez del candidato
python tools/cli/robustness_overlay_c_trajectories.py --job launch

# 7. Esferas alrededor de equilibrios
python tools/cli/lure_top3_sphere_robustness.py --help

# 8. Cuenca refinada
python tools/cli/refine_project_basin_classification.py --help

# 9. Replicacion Danca ABM
python tools/legacy/danca2017_chua_abm_replication.py --job all --workers 4
```

20. Inventario funcional: disponible, activable o legacy

Bloque	Estado	Uso
Modelo Chua y equilibrios	API pública	`hidden\_attractors.models.chua`
DF centrada, sesgada y Machado	API pública ligera	`hidden\_attractors.seed\_generation`
Historia fraccionaria EFORK	API pública ligera	`hidden\_attractors.solvers.FractionalHistory`
Trayectorias, FFT, secciones, nubes	API pública	`hidden\_attractors.analysis.trajectory`
Diagramas de bifurcación por post-proceso	API pública	`hidden\_attractors.analysis.bifurcation`
Robustez overlay	Workflow activo	`hidden\_attractors.workflows.robustness\_overlay`
Controles esféricos	Workflow activo	`hidden\_attractors.workflows.sphere\_controls`
Cuenca refinada	Workflow activo	`hidden\_attractors.workflows.refined\_basin`
Backends C EFORK/cuencas	Backend nativo	`hidden\_attractors.native` y `tools/legacy/*.c`
Pipeline unificado Chua	Wrapper API/CLI	`hidden\_attractors.workflows.unified\_chua`; ejecuta integracion/cuencas por C
Continuación multiparámetro	Legacy mantenido	`multiparameter\_continuation.py`; pendiente de migración completa a backend C
Lyapunov Benettin EFORK	Backend C opcional	`chua\_frac\_lyapunov\_efork\_benettin.c`
PSD Welch	Opcion CLI	`hidden-attractors-unified-chua--psd`
TISEAN/exportación externa	Opcion CLI	`hidden-attractors-unified-chua--tisean`
Réplica Danca ABM	Legacy de referencia	`danca2017\_chua\_abm\_replication.py`
Medidas externas nolds/antropy	Adaptador opcional	`hidden\_attractors.integrations.external\_tools`

21. Advertencia final

El orden correcto evita redundancias: primero se construye y reproduce el target; después se mide su robustez; sólo entonces se verifica si la cuenca del target toca vecindades de equilibrios. Un atractor robusto puede ser autoexcitado. Un candidato sin contactos observados con equilibrios es compatible con ocultedad sólo bajo el contrato numérico probado.

Referencias

[1] M. F. Danca. *Hidden Chaotic Attractors in Fractional-Order Systems.* Nonlinear Dynamics, 2017.

[2] J. Tenreiro Machado. *Fractional order describing functions.* Signal Processing, 2014.

[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, 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. Trabajos sobre atractores ocultos y autoexcitados en sistemas dinámicos.

[6] G. Benettin, L. Galgani, A. Giorgilli, J.-M. Strelcyn. *Lyapunov Characteristic Exponents for Smooth Dynamical Systems and for Hamiltonian Systems*. Meccanica, 1980.