



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

CENTRO DE FÍSICA APLICADA Y TECNOLOGÍA AVANZADA

CÁLCULO DEL MÁXIMO EXPONENTE DE
LYAPUNOV EN MATHEMATICA

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

Licenciada en Tecnología

PRESENTA:

Natalia Isabel Ramírez Estrada

TUTOR:

Dr. José Luis Aragón Vera



C F A T A

Centro de Física Aplicada y Tecnología Avanzada, Querétaro,
2022



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

*A mi mamá y mi hermana
A mis amigos por estar conmigo en todo este proceso
A mis compañeritos de la escuela, gracias por ser mis amigos,
es gracias a ustedes que la experiencia universitaria fue maravillosa.
A mis profesores, sobre todo a mi tutor el Dr. José Luis Aragón
y a la UNAM por todo el conocimiento
y buenas experiencias que me dejaron.
Es gracias a ustedes que es posible el presente trabajo.
Gracias.
Natalia.*

Agradecimientos

Quiero agradecer al Dr. Mario Alan Quiroz Juárez, CFATA-UNAM Juriquilla por su asesoría en cuanto al algoritmo del máximo exponente, también quiero agradecer a mi tutor el Dr. José Luis Aragón Vera por su guía, por brindarme sus conocimientos y su apoyo.

Declaración de autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea el resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

Natalia Isabel Ramírez Estrada. Centro de Física Aplicada y Tecnología Avanzada,
Querétaro, 2022

Resumen

En el presente trabajo se desarrolló un paquete en el software *Wolfram Mathematica* que contiene dos funciones: `MLE[]` y `LE[]`. La primera calcula el máximo exponente de Lyapunov y la segunda calcula el espectro completo de los exponentes de Lyapunov, para sistemas dinámicos continuos y autónomos. Los exponentes de Lyapunov sirven como una característica cuantitativa de la dinámica de un sistema al medir divergencia o convergencia de trayectorias vecinas de la solución, lo que los ha hecho un indicador de caos.

Con estas funciones se llena un espacio existente en las librerías de *Wolfram Mathematica*, que puede ser de utilidad en diversas áreas, brindando una herramienta para determinar el tipo de dinámica del sistema particular.

Índice general

1. Objetivo	1
2. Introducción	5
2.1. Sistemas dinámicos	5
2.1.1. Puntos fijos	10
2.2. Análisis de estabilidad lineal	10
2.2.1. Sistemas lineales homogéneos con coeficientes constantes	10
2.3. Teoría de Floquet	13
2.4. Teoría de Lyapunov	14
2.5. Antecedentes	16
2.6. Objetivo	17
2.7. Motivación	17
2.8. Planteamiento del problema	18
3. Marco teórico e implementación	19
3.1. Cálculo del máximo exponente de Lyapunov	19
3.1.1. Ecuación variacional	19
3.2. Cálculo del espectro de los exponentes de Lyapunov	21
3.3. Implementación en Mathematica	22
3.3.1. Programación en <i>Mathematica</i>	22
3.3.2. Máximo exponente de Lyapunov	24
3.3.3. Espectro completo de los exponentes de Lyapunov	26
4. Resultados	29
4.1. Sistema de Lorenz	29
4.2. Sistema de Rössler	31
4.3. Sistema con un atractor extraño y un toro invariante	33
4.4. Sistema electrocardiograma	35
4.4.1. Ciclo límite	36
4.4.2. Ciclo límite con doblamiento de período	37
4.4.3. Toro	37
4.4.4. Atractor extraño	38
4.4.5. Punto fijo	39

5. Discusión y conclusiones	41
A. Códigos	43
A.1. Máximo exponente de Lyapunov en Mathematica	43
A.2. Espectro completo de los exponentes de Lyapunov	44
Bibliografía	47

Objetivo

El objetivo de esta tesis es desarrollar un paquete en el software *Wolfram Mathematica*, para calcular numéricamente el máximo exponente de Lyapunov y el espectro completo de exponentes de Lyapunov, de un sistema dinámico continuo y autónomo, propuesto por el usuario, y así tener al alcance una herramienta que ayude a identificar la dinámica y estabilidad de sistemas de ese tipo.

Lista de símbolos

Tabla 1.1: Lista de símbolos

MEL	Máximo exponente de Lyapunov
EL	Exponentes de Lyapunov
λ_i	Exponente de Lyapunov i -ésimo
λ_1	Máximo exponente de Lyapunov
J	Matriz Jacobiana
ERK(i)	ExplicitRungeKutta de orden i
τ	Intervalo corto de tiempo de integración para el cálculo de los EL
k	Número de iteraciones para el cálculo de los EL
$\dot{\Phi}$	Ecuación variacional
MLE[]	Función en <i>Mathematica</i> que calcula el máximo exponente de Lyapunov
LE[]	Función en <i>Mathematica</i> que calcula los exponentes de Lyapunov

Introducción

La teoría de sistemas dinámicos se originó a finales del siglo XIX con preguntas fundamentales acerca de la estabilidad y evolución del sistema solar. El responder a esas preguntas llevó al desarrollo de un poderoso campo con aplicaciones en física, biología, astronomía, entre otras áreas [1]. En las siguientes líneas se presentan los aspectos generales de esta teoría.

2.1. Sistemas dinámicos

Un sistema dinámico es aquel cuyo estado cambia en el tiempo [2]. Los sistemas que evolucionan en el tiempo son estudiados por una rama de la física conocida como dinámica.

La forma en la que se describen este tipo de sistemas es mediante un sistema de ecuaciones diferenciales de primer orden, que se representan de forma vectorial de la siguiente manera:

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}) \tag{2.1}$$

Los estados a un tiempo t están completamente especificados por los valores de n variables reales representadas por \mathbf{x} , que es un vector que contiene a todas las variables del sistema, es decir, $\mathbf{x} = (x_1, x_2, \dots, x_n)$, igualmente $\mathbf{F} = (f_1, f_2, \dots, f_n)$ donde a $\mathbf{F}(\mathbf{x})$ se le asocia un sistema de ecuaciones diferenciales ordinarias [3]. Si las funciones contenidas en \mathbf{F} son independientes del tiempo, como en el caso (2.1), se dice que el sistema es autónomo. En caso contrario, en el que una o más funciones contenidas en \mathbf{F} dependa explícitamente del tiempo, se dice que el sistema es no autónomo y el lado derecho de la ecuación (2.1) se escribe como $\mathbf{F}(\mathbf{x}, t)$.

Consideremos el ejemplo de la ecuación de un oscilador armónico forzado: $\ddot{x} + \alpha\dot{x} + \beta x = \cos(\omega t)$. Haciendo $y = \dot{x}$, se obtiene $\dot{y} + \alpha y + \beta x = \cos(\omega t)$, por lo que puede escribirse como un sistema de ecuaciones de primer orden:

$$\begin{aligned} \dot{x} &= y, \\ \dot{y} &= -\alpha y - \beta x + \cos(\omega t), \end{aligned}$$

2. INTRODUCCIÓN

que es no autónomo. En el caso que $\cos(\omega t) = 0$ (o simplemente $\omega = 0$), se obtiene un sistema autónomo.

El cambio de variable aplicado en el ejemplo anterior se puede generalizar, de tal suerte que puede probarse que cualquier ecuación diferencial ordinaria de orden n puede transformarse en un sistema de n ecuaciones diferenciales ordinarias de primer orden [4].

Sin abundar en el tema, se menciona que una noción general de los sistemas dinámicos requiere hablar de los siguientes tópicos:

1. Espacio fase.
2. Tiempo (discreto o continuo).
3. Ley de evolución en el tiempo.

El espacio fase, es un espacio en el que cada punto representa un posible estado del sistema.

Existen dos tipos de sistemas dinámicos, continuos y discretos; los sistemas continuos son modelados por sistemas de ecuaciones diferenciales de primer orden y los discretos por mapeos donde cada valor del tiempo discreto tiene asociado un estado siguiente del sistema dinámico.

La ley de evolución del tiempo se refiere a una regla que nos permite determinar el estado del sistema a cada momento de tiempo t , a partir de sus estados en los tiempos anteriores. El estado final del sistema dependerá solamente del estado inicial y del tiempo de evolución del sistema [5].

Usualmente, para visualizar el comportamiento general de un sistema dinámico, se emplea el retrato fase, que es una representación gráfica de las curvas en espacio fase que corresponden a trayectorias del sistema con diferentes condiciones iniciales. En otras palabras, es la representación gráfica de las soluciones del sistema, siendo una solución las curvas diferenciables que satisface las condiciones iniciales y la ecuación diferencial.

Como ejemplos se presentan los siguientes sistemas [6]:

Oscilador armónico simple

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= -\omega^2 x.\end{aligned}\tag{2.2}$$

Péndulo simple

$$\begin{aligned}\dot{x} &= y, \\ \dot{y} &= -\omega^2 \sin x,\end{aligned}\tag{2.3}$$

cuyos retratos fase se presentan en la Fig. 2.1.

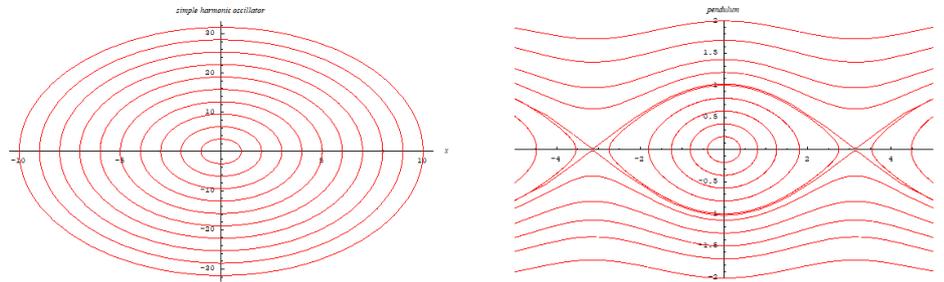


Figura 2.1: Retratos fase de (2.2), izquierda, y (2.3), derecha.

Otra característica topológica importante en los retratos fase son los atractores. Éstos son colecciones invariantes de estados, o bien puntos invariantes en el espacio fase hacia los cuales los estados vecinos en una base de atracción dada se aproximan asintóticamente en el curso de la evolución del sistema. En otras palabras, es una base de atracción que como su nombre lo dice atraen todas las trayectorias vecinas [7, 8].

El concepto de atractor es importante para describir el comportamiento de un sistema dinámico, además de que es una buena fuente de información; por ejemplo, la dimensión de los atractores indica el tipo de movimiento que describe. Se pueden dividir los atractores en dos grupos: los regulares y los extraños. Los regulares tienen dimensiones enteras y los extraños pueden tener dimensiones no enteras. En cuanto a los regulares, hay tres tipos de movimiento asociado con ellos:

1. Punto de equilibrio
2. Movimiento periódico
3. Movimiento cuasiperiódico

Éstos se ilustran en la Fig. 2.2. En *a)* se muestra un punto de equilibrio o punto fijo. Como su nombre lo indica son puntos que atraen las trayectorias vecinas.

El *b)* muestra un ciclo límite. Los ciclos límite son trayectorias cerradas aisladas. Con aislado nos referimos a que no existen trayectorias vecinas que sean también cerradas. Estos atractores describen movimientos periódicos. Una característica importante de los ciclos límite es que solamente se encuentran en fenómenos no lineales; en sistemas lineales pueden existir órbitas cerradas, pero no son aisladas.

Los toros corresponden a movimientos cuasiperiódicos que lucen como una dona en el espacio tridimensional como se presenta en *c)*. Esta dinámica corresponde a un sistema que evoluciona en el tiempo casi como un sistema periódico pero la periodicidad no se cumple de manera exacta. Esto tiene como consecuencia que las trayectorias llenen densamente una región.

Con respecto a los atractores extraños, como el que se muestra en *d)*, estos consisten de trayectorias irregulares que permanecen dentro de una región acotada del espacio fase. A diferencia de los movimientos que acabamos de discutir, las trayectorias no forman un punto, una curva o una superficie, sino a figuras que se repiten recursivamente

2. INTRODUCCIÓN

y que contienen una imagen de sí mismas en cada una de sus partes, conocidos como fractales.

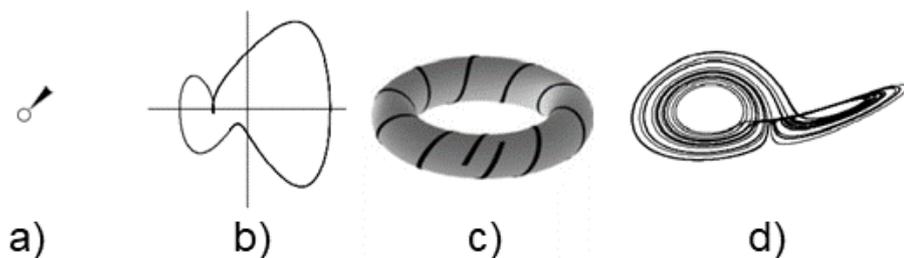


Figura 2.2: Atractores que se pueden encontrar en el espacio fase tridimensional

La estabilidad de estos atractores depende de, entre otras cosas, si las trayectorias vecinas a ellos se acercan o se alejan del atractor; si se acercan se dice que el atractor es estable y si se alejan que es inestable, aunque puede haber casos mixtos [7, 9, 10, 8, 11].

La estabilidad de un sistema dinámico ha sido por mucho tiempo un tema de interés para los matemáticos. Henry Poincaré, a finales de 1800, introdujo una nueva forma de ver este problema, que enfatizaba cuestiones cualitativas, más que cuantitativas. Por ejemplo, en lugar de preguntarse por la posición exacta de los planetas a todo tiempo, se preguntó: “¿El sistema solar es estable por siempre o algunos planetas saldrán eventualmente disparados al infinito”.

Además Poincaré fue el primero en vislumbrar la posibilidad del caos. Aunque no hay definición universalmente aceptada de lo que es el caos, este se puede definir como un sistema determinista que exhibe comportamiento aperiódico el cual tiene una sensible dependencia a las condiciones iniciales.

El caos se popularizó gracias al trabajo de Edward Lorenz, un matemático y meteorólogo estadounidense. Una de sus conferencias se tituló *Predecibilidad: ¿el aleteo de una mariposa en Brasil puede provocar un tornado en Texas?*, por lo que ahora se le conoce como *efecto mariposa* a la alta sensibilidad a las condiciones iniciales. Está además la coincidencia de que el atractor del sistema de Lorenz luce como las alas de una mariposa, como se puede ver en la Fig. 2.3.

Durante sus estudios del clima, Lorenz descubrió su carácter impredecible, debido a que las soluciones del sistema no eran atraídas ni a un punto fijo, ni a un ciclo límite ni a un toro, sino a un atractor completamente diferente: un atractor extraño. El atractor del sistema de Lorenz ahora es conocido como atractor de Lorenz [12, 13].

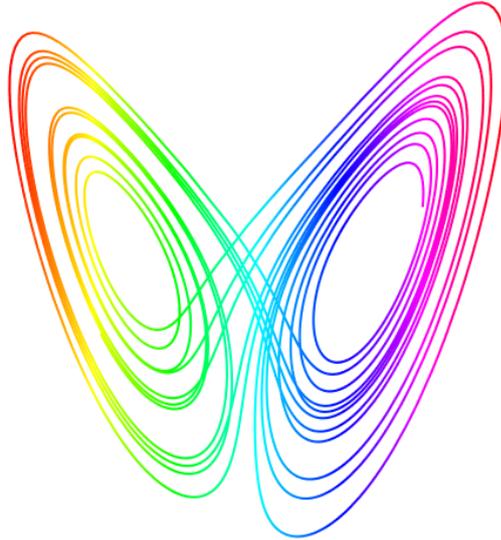


Figura 2.3: Atractor de Lorenz

El caos ha provocado fascinación desde hace mucho tiempo, tal vez gran parte de esto se debe al hecho de que, los atractores no tienen un aspecto irregular, como se esperaría si se atiende a lo que la palabra caos evoca, sino que tienen apariencia muy estética [8]. Aunque el caos fue el último tipo de dinámica en ser descubierto, este tipo de movimiento se puede presentar en sistemas muy simples [14], lo que lleva a preguntarse: ¿Cuáles son los requerimientos para que pueda haber caos?

Para sistemas continuos los requerimientos para que pueda haber caos son: no linealidad y que el sistema tenga dimensión ≥ 3 .

La posibilidad de caos forma parte de la naturaleza, ya que la mayoría de los fenómenos son no lineales [15], por lo que se puede manifestar también en varios campos de la ciencia.

Como el caos puede encontrarse en sistemas muy simples la detección y caracterización del caos se ha vuelto un tema importante. Una herramienta importante para detectar la presencia de caos, son los exponentes de Lyapunov, que se presentan más adelante. Antes es necesario detenernos en el concepto de estabilidad. Esto debido a que usualmente los sistemas físicos están sometidos a perturbaciones, por lo que es importante identificar el tipo de estabilidad que un sistema posee [16].

La estabilidad de los sistemas dinámicos se ha estudiado mediante 3 enfoques: estabilidad lineal, estabilidad de soluciones periódicas (teoría de Floquet) y estabilidad según Lyapunov. A continuación se establecerán conceptos necesarios para comprender los tres enfoques.

2.1.1. Puntos fijos

Los puntos fijos también conocidos como puntos críticos, puntos de equilibrio o puntos estacionarios, son importantes para analizar el comportamiento local del sistema. El punto fijo es una solución constante o de equilibrio de un sistema.

Si se tiene un sistema:

$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}), \mathbf{x} \in \mathbb{R}$$

Los puntos fijos \mathbf{x}^* están definidos por $\mathbf{F}(\mathbf{x}^*) = 0$ [8].

Un sistema puede no tener puntos fijos, uno o un número finito de puntos fijos o infinitos puntos fijos. Por ejemplo:

$\dot{x} = 5$ no tiene puntos fijos.

$\dot{x} = x$ tiene un punto fijo ($x_0 = 0$).

$\dot{x} = x^2 - 1$ tiene dos puntos fijos ($x_0 = \pm 1$).

$\dot{x} = \sin x$ tiene infinitos puntos fijos ($x_0 = \pm n\pi$, donde n es un número natural).

2.2. Análisis de estabilidad lineal

Un punto fijo x_0 se dice que es estable si para un $\epsilon > 0$ dado, existe un $\delta > 0$ dependiente de ϵ tal que $\forall t \geq t_0, \|x(t) - x_0(t)\| < \epsilon$, siempre que $\|x(t_0) - x_0(t_0)\| < \delta$. Entonces, cuando un punto fijo es estable las trayectorias del sistema no se alejan, al contrario de lo que sucede cuando el sistema es inestable.

La estabilidad lineal se basa en el estudio del sistema linealizado alrededor de un punto de equilibrio. Consideremos el sistema (2.1), desarrollando $\mathbf{F}(\mathbf{x})$ en serie de Taylor alrededor de un punto de equilibrio x_0 y despreciando los términos de orden mayor a 1, entonces el sistema (2.4) linealizado alrededor de x_0 es:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \tag{2.4}$$

Donde \mathbf{A} es una matriz llamada matriz Jacobiana evaluada en el punto de equilibrio x_0 . La estabilidad o inestabilidad del punto de equilibrios se infiere de un análisis de los eigenvalores de \mathbf{A} [17].

2.2.1. Sistemas lineales homogéneos con coeficientes constantes

Consideremos un sistema de la forma (2.4), donde \mathbf{A} es una matriz constante de $n \times n$, asumimos que todos los elementos de \mathbf{A} son reales.

Si $n = 1$ el sistema se reduce a una ecuación de primer orden:

$$\dot{x} = ax$$

Su solución es $x = ce^{at}$ y la constante c puede ser encontrada mediante las condiciones iniciales $x(0) = x_0$. Por lo tanto la solución general sería $x(t) = x_0e^{at}$.

Este sistema tiene un sólo punto fijo el cual es $x = 0$ mientras $a \neq 0$.

Si $a < 0$ entonces las soluciones tienden a $x = 0$ cuando t incrementa, en estos casos se dice que $x = 0$ es una solución de equilibrio asintóticamente estable.

Si $a > 0$ entonces las soluciones tienden a alejarse de $x = 0$ cuando t incrementa, en estos casos se dice que $x = 0$ es una solución inestable.

Para sistemas de n ecuaciones la situación es similar, pero un poco más complicada [4].

Si el sistema (2.4) tiene solución $\mathbf{x}(t)$ lo que quiere decir que satisface la ecuación (2.4), cualquier combinación lineal de las soluciones del sistema es también una solución por el principio de superposición.

De ahí que mientras el sistema sea lineal se considera la solución trivial:

$$\mathbf{x}(t) = \mathbf{v}x^{\lambda t} \quad (2.5)$$

Donde $\mathbf{x}(t)$ es el vector de soluciones, \mathbf{v} es el vector columna con componentes $\mathbf{v} = (v_1, v_2, \dots, v_n)^t$, λ es un número. Si la solución (2.5) existe, éstas corresponden a un movimiento exponencial a lo largo de la línea generada por el vector \mathbf{v} donde la tasa de crecimiento está determinada por λ .

Si se sustituye (2.5) en (2.4) se obtiene:

$$\begin{aligned} \lambda \mathbf{v}e^{\lambda t} &= \mathbf{A}\mathbf{v}e^{\lambda t} \\ (\mathbf{A} - \lambda \mathbf{I})\mathbf{v} &= 0 \end{aligned} \quad (2.6)$$

Para que se obtenga una solución no trivial se debe cumplir que:

$$\det(\mathbf{A} - \lambda \mathbf{I}) \neq 0$$

Si se desarrolla (2.6) se obtiene una ecuación polinomial de grado n en λ que se conoce como ecuación característica de la matriz \mathbf{A} . Las raíces de la ecuación característica son los eigenvalores, los eigenvectores \mathbf{v} se obtienen resolviendo (2.6) para cada eigenvalor.

La estabilidad del sistema (2.4) está definida por el valor de los eigenvalores. Por ejemplo, para un sistema 2x2 con coeficientes reales hay tres casos principales que pueden ocurrir.

- Los eigenvalores son reales y tienen signos opuestos. Por lo tanto $x = 0$ es un punto silla
- Los eigenvalores son diferentes, reales y con el mismo signo. Por lo tanto $x = 0$ es un nodo
- Los eigenvalores tienen valores complejos con parte real diferente de cero. Por lo tanto $x = 0$ es un punto espiral.

Para ejemplificar, consideremos como ejemplo el sistema:

$$\dot{\mathbf{x}} = \begin{pmatrix} -3 & \sqrt{2} \\ \sqrt{2} & -2 \end{pmatrix} \mathbf{x}. \quad (2.7)$$

2. INTRODUCCIÓN

La ecuación característica se obtiene de

$$\det \begin{pmatrix} -3 - \lambda & \sqrt{2} \\ \sqrt{2} & -2 - \lambda \end{pmatrix} = (-3 - \lambda)(-2 - \lambda) - 2 = (\lambda + 1)(\lambda + 4) = 0,$$

por lo que los eigenvalores son $\lambda_1 = -1$ y $\lambda_2 = -4$.

Resolviendo (2.6) para $\lambda_1 = -1$ se obtiene el eigenespacio correspondiente a ese eigenvalor es generado por

$$\mathbf{v}^{(1)} = \begin{pmatrix} 1 \\ \sqrt{2} \end{pmatrix}.$$

Haciendo lo mismo para $\lambda_2 = -4$ se obtiene:

$$\mathbf{v}^{(2)} = \begin{pmatrix} -\sqrt{2} \\ 1 \end{pmatrix}.$$

El conjunto de soluciones fundamentales del sistema (2.7) es, entonces:

$$\mathbf{x}^{(1)} = \begin{pmatrix} 1 \\ \sqrt{2} \end{pmatrix} e^{-t}, \mathbf{x}^{(2)} = \begin{pmatrix} -\sqrt{2} \\ 1 \end{pmatrix} e^{-4t}.$$

Y la solución general del sistema (2.7) es:

$$\mathbf{x}(t) = c_1 \mathbf{x}^{(1)} + c_2 \mathbf{x}^{(2)} = c_1 \begin{pmatrix} 1 \\ \sqrt{2} \end{pmatrix} e^{-t} + c_2 \begin{pmatrix} -\sqrt{2} \\ 1 \end{pmatrix} e^{-4t}.$$

Como los eigenvalores son distintos, reales y con el mismo signo, el punto fijo es un nodo, y dado que los los eigenvalores son negativos, se trata de un nodo estable. La representación en el espacio fase del sistema (2.7) se muestra en la Fig. 2.4.

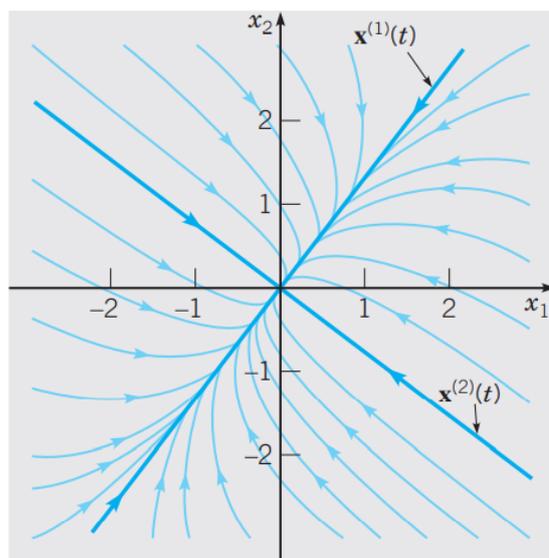


Figura 2.4: Retrato fase del sistema (2.7)

2.3. Teoría de Floquet

La teoría de Floquet se utiliza para identificar la estabilidad de una trayectoria cerrada, la cual corresponde a sistemas con soluciones periódicas. Cuando una solución periódica es ligeramente perturbada, si la perturbación se extingue a medida que el tiempo incrementa, la solución periódica es estable, de lo contrario es inestable [18].

Los sistemas de Floquet son sistemas donde un sistema como en (2.1) tiene solución periódica

$$\mathbf{x}_p(t) = \mathbf{x}_p(t + T).$$

Para identificar la estabilidad de esta solución periódica, se perturba la solución para quedar de la manera:

$$\mathbf{x}(t) = \mathbf{x}_p(t) + \tilde{\mathbf{x}}(t). \quad (2.8)$$

Donde $\tilde{\mathbf{x}}$ es una pequeña perturbación. Si se sustituye (2.8) en (2.1), despejando para $\dot{\tilde{\mathbf{x}}}$ y desarrollando $\mathbf{F}(\tilde{\mathbf{x}})$ en series de Taylor (linealizando sobre la solución periódica). El sistema lineal asociado para la perturbación es el siguiente:

$$\dot{\tilde{\mathbf{x}}}(t) = \mathbf{D}(t)\tilde{\mathbf{x}}(t). \quad (2.9)$$

Donde \mathbf{D} está definida como sigue:

$$\mathbf{D}(t) = \left. \frac{\partial \mathbf{F}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}_p}. \quad (2.10)$$

Podemos observar que se trata de la matriz jacobiana evaluada en la solución periódica \mathbf{x}_p y, debido a la periodicidad de \mathbf{x}_p , \mathbf{D} es una matriz con coeficientes periódicos, es decir, $\mathbf{D}(t) = \mathbf{D}(t + T)$

La idea básica detrás de la teoría de estabilidad de Floquet es el supuesto de que la periodicidad de la matriz \mathbf{D} permite observar el comportamiento de $\tilde{\mathbf{x}}$, sólo en tiempos discretos $t = 0, T, 2T, \dots$

El sistema lineal n dimensional asociado para la perturbación (2.9) tiene n soluciones linealmente independientes $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_n$ por lo que, por el principio de superposición, la solución puede escribirse como $\tilde{\mathbf{x}} = c_1\tilde{\mathbf{x}}_1 + c_2\tilde{\mathbf{x}}_2 + \dots + c_n\tilde{\mathbf{x}}_n$. O en forma matricial:

$$\tilde{\mathbf{x}}(t) = \tilde{\mathbf{X}}(t)\mathbf{c}. \quad (2.11)$$

Donde $\tilde{\mathbf{X}}$ es la matriz de soluciones $\tilde{\mathbf{X}}(t) = [\tilde{\mathbf{x}}_1(t) \ \tilde{\mathbf{x}}_2(t) \ \dots \ \tilde{\mathbf{x}}_n(t)]$ y \mathbf{c} es el vector de constantes $\mathbf{c} = \{c_1, c_2, \dots, c_n\}$.

Si $\tilde{\mathbf{x}}(t)$ es solución del sistema (2.9), $\tilde{\mathbf{x}}(t + T)$ también es solución, siempre que $\mathbf{D}(t) = \mathbf{D}(t + T)$.

Para la matriz asociada a las correspondientes soluciones fundamentales, se introduce la siguiente notación

$$\Phi(t) = [\varphi_1(t) \ \varphi_2(t) \ \dots \ \varphi_n(t)],$$

con

$$\Phi(0) = \mathbf{I} \quad (2.12)$$

Si se aplica el mismo razonamiento que con $\tilde{\mathbf{x}}(t)$ y $\tilde{\mathbf{x}}(t+T)$ entonces se puede establecer una matriz constante de $n \times n$, \mathbf{C} , tal que

$$\Phi(t+T) = \Phi(t)\mathbf{C}. \quad (2.13)$$

Donde \mathbf{C} es una matriz que mapea $\Phi(t)$ en $\Phi(t+T)$. Debido a las condiciones iniciales (2.12), se puede deducir de (2.13) que

$$\Phi(T) = \Phi(0)\mathbf{C} = \mathbf{C}.$$

A la matriz \mathbf{C} se le conoce como matriz de monodromía.

Si el periodo T es conocido, se integra (2.9) sobre el periodo T , y aplicando (2.12) se obtiene \mathbf{C} directamente.

La estabilidad en este caso viene dada por los eigenvalores σ de la matriz \mathbf{C} , los cuales son conocidos como multiplicadores de Floquet. Si $|\sigma_j| < 1$ el sistema es asintóticamente estable, pero si $|\sigma_j| > 1 \forall j = 1, 2, \dots, n$ el sistema es inestable [9].

2.4. Teoría de Lyapunov

Los exponentes de Lyapunov (también conocidos como coeficientes de Lyapunov), son medidas que caracterizan la tasa promedio de crecimiento o decaimiento de pequeñas perturbaciones de las órbitas de un sistema dinámico. Desde que fueron introducidos por Lyapunov, han sido ampliamente utilizados para el estudio de sistemas dinámicos porque son una herramienta útil para investigar el tipo de estabilidad de conjuntos límite, y clasificarlos de acuerdo con su dinámica [19, 20]. Debido a que los exponentes de Lyapunov miden la separación de dos trayectorias con condiciones iniciales cercanas (la del sistema dinámico y la de la pequeña perturbación), calcularlos nos provee una medida directa de la sensibilidad a las condiciones iniciales, la cual, entre otras cosas, es una característica del caos. Por esta razón, los exponentes de Lyapunov son también la herramienta más útil para el diagnóstico de sistemas caóticos [21, 22].

Un sistema dinámico n dimensional definido por n ecuaciones diferenciales de primer orden, tiene n exponentes de Lyapunov ($\lambda_i, i = 1, \dots, n$). Donde cada exponente (λ_i) representa la tasa de crecimiento o decaimiento de pequeñas perturbaciones a lo largo de cada uno de los ejes principales en el espacio de estados del sistema. Usualmente se ordenan del más grande al más pequeño ($\lambda_1 \geq \lambda_2 \geq \dots \lambda_n$) [23].

$\lambda_i > 0$ implica que existe una tasa exponencial promedio de divergencia entre las trayectorias vecinas en el espacio fase cerca de un atractor, entonces, si al menos un exponente de Lyapunov es positivo, es indicador de caos. Por el contrario, si $\lambda_i < 0$ las trayectorias convergen; en el caso que $\lambda_i = 0$ las trayectorias ni convergen ni divergen [24].

En caso de que el sistema dinámico a analizar tenga múltiples atractores, los valores de los exponentes de Lyapunov se aproximarán a alguno de los atractores, lo que determina a cuál de los múltiples atractores se aproximen está definido por la condición inicial [20].

Para cualquier atractor (de un sistema dinámico autónomo) que no sea un punto fijo, un exponente de Lyapunov es siempre cero, que corresponde a la dirección tangente del flujo definido por el sistema dinámico; lo que indica que la perturbación hecha a lo largo de la trayectoria ni converge ni diverge [24, 25, 23], además, para cualquier atractor la suma de los exponentes de Lyapunov es negativa ya que la contracción del atractor debe ser mayor que la expansión; en otras palabras, el sistema no crecerá sin límites ni divergirá hasta infinito [23, 24].

Debido a las características anteriores, un atractor extraño (asociado a un sistema caótico) debe tener al menos 3 exponentes de Lyapunov (Al menos un valor positivo, al menos un valor cero y al menos un valor negativo para que la suma sea negativa), que es una de las características para que exista caos, siendo $(\lambda_1, \lambda_2, \lambda_3) = (+, 0, -)$, con $\sum_i^n \lambda_i < 0$ [24, 25]. Para el caso en que al menos dos exponentes de Lyapunov sean positivos, esa condición se define como hipercaos. Donde un atractor es hipercaótico si tiene al menos dos exponentes de Lyapunov positivos, de esta manera la dimensión mínima del sistema para hipercaos es 4, o sea $(\lambda_1, \lambda_2, \lambda_3, \lambda_4) = (+, +, 0, -)$ [24, 26].

De manera que las magnitudes de los exponentes de Lyapunov cuantifican la dinámica de un atractor brindándonos información de si el sistema es regular (estado estacionario, periódico o cuasiperiódico) o caótico. Generalmente, el número de exponentes de Lyapunov con valor cero, de sistemas no caóticos, indican la dimensión del atractor, por ejemplo: un punto de equilibrio o punto fijo tiene dimensión 0, ya que no tiene ningún exponente cero, un ciclo límite tiene dimensión 1, porque tiene un exponente cero, un 2-toro tiene dimensión 2 y un k-toro tiene dimensión k porque tienen 2 y k exponentes cero, respectivamente. Esto se puede apreciar de mejor manera en la Tabla 2.1, [23, 17, 22, 19, 27, 24].

De esta manera, el espectro de los coeficientes de Lyapunov puede diferenciar entre distintos tipos de atractores y se pueden utilizar para revelar inestabilidades y bifurcaciones de algún sistema que depende de controlar algún parámetro. Por ejemplo, si un toro bidimensional estable se convierte en un atractor extraño, existe un punto crítico en el que el espectro de los coeficientes de Lyapunov cambia de $\{0, 0, -\}$ a $\{+, 0, -\}$ [9].

De todo el espectro de los coeficientes de Lyapunov, el más importante es el máximo (λ_1) . La magnitud de (λ_1) especifica la tasa exponencial promedio máxima de divergencia de trayectorias en un atractor y, por lo tanto, detecta qué tanto se alejan las trayectorias. Por esta razón el máximo exponente de Lyapunov (MEL) se utiliza usualmente como medida local de la inestabilidad de un sistema dinámico, lo que lo ha convertido en un indicador de caos. De ahí que la mayoría de los algoritmos estén desarrollados para calcular sólo el MEL. Sin embargo, el cálculo del espectro completo brinda un retrato completo de la dinámica del sistema ya que caracteriza el comportamiento asintótico de éste [25, 19, 23].

2. INTRODUCCIÓN

Si la órbita estudiada es regular (puntos fijos, sistemas periódicos y cuasiperiódicos) $\lambda_1 = 0$ aun cuando sea un punto fijo ($\lambda_i < 0$ todos los exponentes negativos) debido a que λ_1 tiende a cero siguiendo una ley de potencias $\lambda_1 \propto t^{-1}$. Por el contrario, si es caótica $\lambda_1 > 0$, lo que implica divergencia exponencial de órbitas cercanas, donde para valores relativamente pequeños de tiempo t el MEL usualmente variará irregularmente, pero para tiempos t suficientemente grandes el MEL tenderá a estabilizarse a un valor positivo constante.

Además de utilizar el MEL como criterio de caos o regularidad de una órbita, al valor del MEL se le puede dar un significado físico, definiendo el siguiente tiempo:

$$t_L = \frac{1}{\lambda_1}. \quad (2.14)$$

La ecuación anterior brinda un estimado del tiempo que necesita un sistema dinámico para volverse caótico (conocido como tiempo Lyapunov) y en la práctica mide el tiempo que necesitan las órbitas cercanas del sistema para divergir exponencialmente [19].

Tabla 2.1: Exponentes de Lyapunov de los distintos tipos de conjuntos límite

Tipo de atractor	Dinámica	Exponentes de Lyapunov	Dimensión
Punto fijo	Estado estacionario	$0 > \lambda_1 \geq \dots \geq \lambda_n$	0
Ciclo límite	Periódico	$\lambda_1 = 0$ $0 > \lambda_2 \geq \dots \geq \lambda_n$	1
Toro	Cuasiperiódico	$\lambda_1 = \lambda_2 = 0$ $0 > \lambda_3 \geq \dots \geq \lambda_n$	2
k-Toro	Cuasiperiódico	$\lambda_1 = \dots = \lambda_k = 0$ $0 > \lambda_{k+1} \geq \dots \geq \lambda_n$	k
Atractor extraño	Caótico	$\lambda_1 > 0$ $\sum_i^n \lambda_i < 0$	No entera

2.5. Antecedentes

Para el cálculo de los exponentes de Lyapunov se han desarrollado distintos algoritmos [19, 28, 29]. Un resumen de los métodos propuestos puede hallarse en la sección 7 de [23].

En cuanto a el cálculo numérico del máximo exponente de Lyapunov, la mayoría de los códigos existentes requieren un conocimiento más que básico de programación,

debido a que la mayoría se encuentran escritos en lenguajes como C, C++, Fortran y, en el mejor de los casos, Matlab [30].

Con respecto al software *Mathematica*, el primero en usarlo para calcular los exponentes de Lyapunov fue M. Sandri, en 1996. El autor calculó los exponentes de Lyapunov para 2 casos particulares, los sistemas de Lorenz y el de Rössler, que se sabe que es caótico el primero e hipercaótico el segundo.

Posteriormente, en el año 2008, una versión modificada de los dos métodos propuestos en [20] en *Mathematica* fue propuesto en [31] y aplicado al cálculo del máximo exponente y todo el espectro de los exponentes de Lyapunov de 3 sistemas: Lorenz, Rössler y de un sistema químico no isotérmico. En ese trabajo proponen usar la función *NDSolve*, para resolver las ecuaciones diferenciales, en lugar del método de tamaño de paso fijo de Roman Maeder conocido como RKStep. La modificación les permitió tener resultados más precisos y a la vez redujeron substancialmente los tiempos de cómputo.

Finalmente, en el año 2013, se publica otro trabajo en donde se calcula el máximo exponente de Lyapunov en *Wolfram Mathematica*, también para los sistemas de Lorenz y de Rössler. Se utilizaron librerías actualizadas y se obtuvo el valor del máximo exponente como un promedio sobre un conjunto de trayectorias.

En estos trabajos anteriores en *Mathematica* se realizaron cálculos de sistemas dinámicos en específico. Nuestro paquete es una generalización en el que el usuario podrá definir el sistema y la función regresará ya sea el máximo exponente de Lyapunov o todo el espectro de los exponentes de Lyapunov, lo que lo hace más práctico.

2.6. Objetivo

El objetivo de esta tesis es implementar un paquete en el software *Wolfram Mathematica* que calcule numéricamente el máximo exponente de Lyapunov, y de todo el espectro de exponentes, de un sistema dinámico, con el fin de contar con una herramienta general para identificar el tipo de dinámica del sistema.

2.7. Motivación

En la naturaleza abundan los sistemas caóticos debido a que muchos modelos de fenómenos naturales son intrínsecamente no lineales. Los sistemas caóticos tienen una sensible dependencia a las condiciones iniciales, es decir, aunque dos trayectorias inicien muy cercanamente, divergirán exponencialmente. Esa impredecibilidad tiene su origen en la no linealidad de las ecuaciones, de ahí que detectar caos y estimar el tiempo límite de predicción ha sido una tarea importante y desafiante.

Identificar la estabilidad de un sistema suele ser muy importante en diversas ramas de la ciencia. Por ejemplo, en ingeniería de control se requiere estabilizar a un sistema no

estable, por lo que es necesario contar con un parámetro que identifique la estabilidad o inestabilidad del sistema.

Debido a que los exponentes de Lyapunov son un indicador de la presencia o ausencia de caos, calcularlos es la herramienta más importante en el diseño y análisis de sistemas no lineales [21, 23, 32, 33, 34].

Además de ser indicadores de caos, los coeficientes de Lyapunov son útiles para catalogar sistemas con comportamientos de estado estacionario, como lo son: puntos fijos asintóticamente estables, ciclos límite asintóticamente estables y toros n dimensionales asintóticamente estables [24].

2.8. Planteamiento del problema

El análisis de estabilidad lineal y el de floquet son limitados, la estabilidad según Lyapunov tiene más alcance debido a que se puede determinar la estabilidad no sólo de puntos fijos o soluciones periódicas, sino también de dinámicas cuasiperiódicas y caóticas [24]. Existen paquetes o funciones que calculan el máximo exponente de Lyapunov, o todo el espectro, en distintos lenguajes de programación, sin embargo, debido a su amplio uso y a las bondades de su lenguaje de programación, se propone el uso de *Wolfram Mathematica*. Como se mencionó anteriormente, ya existen propuestas del uso de este software para el cálculo del MEL y de todo el espectro, pero son limitadas a casos particulares. En la presente tesis se propone una función para el cálculo del MEL y de todo el espectro de exponentes de Lyapunov, para cualquier sistema dinámico, continuo y autónomo.

Marco teórico e implementación

3.1. Cálculo del máximo exponente de Lyapunov

El algoritmo utilizado para el cálculo del máximo exponente de Lyapunov se basa en el método variacional [24], que describimos a continuación.

3.1.1. Ecuación variacional

La ecuación variacional es una ecuación diferencial lineal en forma matricial con entradas que dependen del tiempo. Es la linealización del campo vectorial a lo largo de la trayectoria $\phi_t(x_0, t_0)$, por lo que si la trayectoria cambia, también lo hace la ecuación variacional.

Consideremos un sistema de n ecuaciones diferenciales ordinarias

$$\dot{x} = f(x, t), \quad (3.1)$$

$$x(t_0) = x_0, \quad (3.2)$$

con solución $\phi_t(x_0, t_0)$, es decir

$$\dot{\phi}_t(x_0, t_0) = f(\phi_t(x_0, t_0), t), \quad (3.3)$$

$$\phi_t(x_0, t_0) = x_0. \quad (3.4)$$

Derivando (3.3) con respecto a x_0 , se obtiene:

$$D_{x_0} \dot{\phi}_t(x_0, t_0) = D_{x_0} f(\phi_t(x_0, t_0), t) D_{x_0} \phi_t(x_0, t_0), \quad (3.5)$$

$$D_{x_0} \phi_{t_0}(x_0, t_0) = \mathbf{I}, \quad (3.6)$$

definiendo $\Phi_t(x_0, t_0) = D_{x_0} \phi_t(x_0, t_0)$. Entonces (3.5) se convierte en:

$$\dot{\Phi}_t(x_0, t_0) = D_x f(\phi_t(x_0, t_0), t) \Phi_t(x_0, t_0), \quad (3.7)$$

$$\Phi_{t_0}(x_0, t_0) = \mathbf{I}, \quad (3.8)$$

3. MARCO TEÓRICO E IMPLEMENTACIÓN

Que se conoce como ecuación variacional, y puede escribirse de la siguiente manera:

$$\dot{\Phi}_t(x_0, t_0) = \mathbf{J}\Phi_t(x_0, t_0), \quad (3.9)$$

$$\Phi_{t_0}(x_0, t_0) = \mathbf{I}, \quad (3.10)$$

donde \mathbf{J} es la matriz Jacobiana del sistema (3.1).

Dado que la condición inicial es la matriz identidad (\mathbf{I}), la solución $\Phi_t(x_0, t_0)$ de la ecuación variacional es la matriz de transición de estados del sistema lineal (3.7). De ello se deduce que una perturbación δx_0 , de x_0 , evoluciona de la siguiente manera

$$\delta x(t) = \Phi_t(x_0, t_0)\delta x_0. \quad (3.11)$$

La perturbación debe ser pequeña, por ejemplo del orden de 10^{-15} [8] o de una distancia en el espacio fase de $10^{-7} - 10^{-6}$ [19]. En los trabajos anteriores realizados en *Mathematica* se escogieron valores entre 0 y 1 [20, 31, 30]. En este trabajo se escogió una perturbación con un tamaño de un valor del orden 10^{-16} .

La perturbación δx_0 se puede interpretar como una perturbación infinitesimal del sistema original (3.1) o como un vector solución del sistema linealizado (3.7).

Para obtener la solución de la ecuación variacional usualmente se calculan simultáneamente las soluciones de la ecuación variacional y del sistema original obteniendo el siguiente sistema:

$$\begin{cases} \dot{\mathbf{x}} \\ \dot{\Phi} \end{cases} = \begin{cases} f(x, t) \\ D_x f(x, t)\Phi \end{cases}, \quad (3.12)$$

integrando a ecuación (3.12) desde la condición inicial:

$$\begin{cases} x(t_0) \\ \Phi(t_0) \end{cases} = \begin{cases} x_0 \\ \mathbf{I} \end{cases}. \quad (3.13)$$

Para calcular el MLE de una órbita con condición inicial $x(t_0)$, se sigue simultáneamente la evolución de la órbita y de un vector desviación δx_0 , desde su órbita con condición inicial δx_0 , como se infiere de la ecuación (3.11). Para ello, se resuelve simultáneamente la ecuación variacional, para la evolución en el tiempo del vector desviación, y el sistema de ecuaciones del sistema dinámico, para la evolución en el tiempo de la órbita. Haciendo esto, puede probarse que [19]:

$$\lambda_1(t) = \lim_{t \rightarrow \infty} \frac{1}{t} \ln \frac{\|\delta x(t)\|}{\|\delta x_0\|} = \lim_{t \rightarrow \infty} \ln \frac{\|\Phi_t(x_0, t_0)\delta x_0\|}{\|\delta x_0\|}. \quad (3.14)$$

Hay que recalcar que el vector desviación $\delta x(t)$ a tiempos $t > 0$ es determinado por la acción de la ecuación variacional sobre el vector desviación δx_0 , de acuerdo a la ecuación (3.11) [19, 24].

Para ejemplificar el procedimiento, consideremos el sistema de Lorenz, que está bien caracterizado; se sabe que es caótico y que el valor de su MEL es $\lambda_1 = 1.5$ [30].

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 16(y - x) \\ x\left(\frac{4592}{100} - z\right) - y \\ xy - 4z \end{bmatrix}, \quad (3.15)$$

con la condición inicial

$$\dot{\mathbf{X}}(t_0) = x_0 = \begin{bmatrix} \dot{x}(t_0) \\ \dot{y}(t_0) \\ \dot{z}(t_0) \end{bmatrix} = \begin{bmatrix} \frac{2}{10} \\ \frac{3}{10} \\ \frac{5}{10} \end{bmatrix}. \quad (3.16)$$

La matriz Jacobiana de (3.15) es

$$\mathbf{J} = \begin{bmatrix} -16 & 16 & 0 \\ 45.92 - z & -1 & -x \\ y & x & -4 \end{bmatrix}, \quad (3.17)$$

con lo que la ecuación variacional queda como

$$\dot{\Phi} = \mathbf{J}\Phi = \begin{bmatrix} \dot{\Phi}_{11} & \dot{\Phi}_{12} & \dot{\Phi}_{13} \\ \dot{\Phi}_{21} & \dot{\Phi}_{22} & \dot{\Phi}_{23} \\ \dot{\Phi}_{31} & \dot{\Phi}_{32} & \dot{\Phi}_{33} \end{bmatrix} = \begin{bmatrix} -16\Phi_{11} + 16\Phi_{21} & -16\Phi_{12} + 16\Phi_{22} & -16\Phi_{13} + 16\Phi_{23} \\ (45.92 - z)\Phi_{11} - \Phi_{21} - x\Phi_{31} & (45.92 - z)\Phi_{12} - \Phi_{22} - x\Phi_{32} & (45.92 - z)\Phi_{13} - \Phi_{23} - x\Phi_{33} \\ y\Phi_{11} + x\Phi_{21} - 4\Phi_{31} & y\Phi_{12} + x\Phi_{22} - 4\Phi_{32} & y\Phi_{13} + x\Phi_{23} - 4\Phi_{33} \end{bmatrix}. \quad (3.18)$$

Ahora, siguiendo (3.12) y (3.13), se resuelven simultáneamente (3.18) y (3.15). Una vez obtenido $\dot{\Phi}$, se calcula (3.11) para, finalmente calcular el MEL mediante (3.14).

3.2. Cálculo del espectro de los exponentes de Lyapunov

Para el cálculo del espectro de coeficientes de Lyapunov se usa un método estándar, que se basa en la evolución temporal de más de un vector desviación, los cuales se mantienen linealmente independientes, lo que se consigue si esos vectores son ortogonales; por lo que se orthogonaliza mediante el método de Gram-Schmidt [19].

Calcular el espectro completo de los exponentes de Lyapunov presenta complicaciones por el hecho de que, para valores grandes de tiempo, los valores de los exponentes tienden al máximo, lo que, de alguna manera, contamina el cálculo y produce resultados no acertados. Para resolver este problema, en lugar de calcular los exponentes de Lyapunov de orden 1 (vectores), como en el caso del algoritmo para calcular el máximo exponente de Lyapunov, se calculan los exponentes de Lyapunov de orden p , donde $1 \geq p \geq n$, que describen la tasa promedio de crecimiento de un volumen p dimensional en el espacio tangente [20].

Teniendo un sistema n dimensional como (3.1) con condición inicial (3.2).

Para obtener el espectro de los coeficientes de Lyapunov se sigue el siguiente procedimiento:

3. MARCO TEÓRICO E IMPLEMENTACIÓN

1. Se construye la ecuación variacional (3.9) y se establece como condición inicial la matriz identidad (\mathbf{I}). Si este no fuera el caso, como condición inicial se escoge un conjunto de n vectores ortonormales $\{v_1^0, \dots, v_n^0\}$ y se construye la matriz de $n \times n$ $\mathbf{V}_0 = [v_1^0, \dots, v_n^0]$.
2. Se resuelve (3.12) con las condiciones iniciales (3.13) integrando en un intervalo de tiempo corto τ . A las soluciones respectivas las denotamos como x_1 y Φ_1 .
3. Usando el proceso de Gram-Schmidt, se ortonormalizan los vectores columna de Φ_1 para obtener \mathbf{V}_1 .
4. Se almacenan las respectivas normas, $N_1(1), \dots, N_n(1)$, obtenidas del proceso de ortonormalización del paso anterior.
5. Se integra nuevamente (3.12), usando como condiciones iniciales las soluciones x_1 y \mathbf{V}_1 , en el intervalo de tiempo τ , para obtener x_2 y Φ_2 . Se repiten los pasos 3 y 4, para obtener $N_1(2), \dots, N_n(2)$.
6. Este proceso se repite hasta k veces para obtener $N_1(k), \dots, N_n(k)$, con lo que podemos obtener los exponentes de Lyapunov como:

$$\lambda_i = \lim_{k \rightarrow \infty} \frac{1}{k\tau} \sum_{m=1}^k \ln N_i(m). \quad (3.19)$$

3.3. Implementación en Mathematica

3.3.1. Programación en *Mathematica*

Mathematica es una plataforma para computación técnica muy poderosa y versátil para el cálculo numérico y simbólico. Los cálculos se basan en funciones ya predefinidas, con la posibilidad de que el usuario defina sus propias funciones. El sistema incluye cerca de 6,000 funciones predeterminadas, que cubren prácticamente todas las áreas de la computación técnica. El sistema está basado en el llamado *Wolfram Language*, que es un lenguaje de programación que unifica varios paradigmas de la programación, como la programación procedural, simbólica, basada en reglas y funcional (veremos un ejemplo de la primera y la última más adelante, al implementar el proceso de Gram-Schmidt). Una introducción muy completa a *Mathematica* y su lenguaje de programación, puede hallarse en [35].

Para este trabajo se desarrollaron dos funciones, `MLE[in1, in2, ...]`, con la que se calcula el máximo exponente de Lyapunov, y `LE[in1, in2, ...]`, con la que se calcula todo el espectro de exponentes. En donde `in1`, `in2`, etc., son las entradas de la función, como el sistema dinámico, las condiciones iniciales y otras que se describen en las siguientes secciones. Debido a la naturaleza de los sistemas dinámicos involucrados, que son principalmente no lineales, el cálculo simbólico no puede ser aplicado y

los cálculos son numéricos y las funciones se programaron en una combinación de programación procedural y basada en reglas. Hay que señalar que debido también a que muchas aplicaciones consideran sistema caóticos, los métodos numéricos usados para la solución del sistema de ecuaciones diferenciales ordinarias asociado a esos problemas suelen ser especiales. Los desarrolladores de *Mathematica* argumentan que usar la opción `Automatic` permite al sistema escoger el método más adecuado para el problema particular, pero siempre es recomendable que el usuario corrobore resultados usando métodos específicos incluidos en *Mathematica*, que puede consultar en [36].

Algunos comentarios respecto a la implementación del proceso Gram-Schmidt, que se requiere para el cálculo del espectro de exponentes de Lyapunov, son pertinentes aquí. Si se tiene un conjunto de vectores arbitrarios $\{u_1, \dots, u_n\}$, a partir de estos, el proceso de Gram-Schmidt produce un conjunto de vectores ortogonales $\{v_1, \dots, v_n\}$, de la siguiente manera:

$$v_1 = u_1, \quad v_n = u_n - \sum_{k=1}^{n-1} \frac{\langle v_k, u_n \rangle}{\|v_k\|^2} v_k, \quad n \geq 2. \quad (3.20)$$

La función de `Orthogonalize[]` de *Mathematica* realiza este proceso pero produce un conjunto ortonormal de vectores, con lo que perdemos las normas que, de acuerdo con (3.19), son necesarias para calcular los exponentes de Lyapunov. La única solución posible es, entonces, implementar el proceso (3.20), cuya inestabilidad numérica es bien conocida [37]. Se han desarrollado métodos alternativos, como el proceso de Gram-Schmidt modificado, para mejorar la estabilidad, pero es de notar que en el cálculo de los exponentes de Lyapunov, de acuerdo con el paso 3 del procedimiento descrito en la Sección 3.2, se ortogonaliza una matriz Φ de $n \times n$, donde n es el número de ecuaciones del sistema dinámico. En este caso, dado que difícilmente se tendrán casos de sistemas dinámicos con más de 10 ecuaciones (el ejemplo con más ecuaciones considerado en este trabajo contiene 4 ecuaciones), la acumulación de errores por redondeo es despreciable.

Podemos, entonces, implementar el método clásico (3.20), usando programación procedural, como sigue:

```

GS[vec_List] := Module[{n = Length[vec], s, i, j},
  s = Array[0, {n, n}];
  s[[1]] = vec[[1]];
  For[i = 2, i <= n, i++, s[[i]] = vec[[i]] - Sum[vec[[j]].s[[j]]/s[[j]].s[[j]], {j, 1, i-1}];
  s
]

```

O usando programación funcional, como sigue:

```

oneStepOrthogonalize[vct_List, {}] := vct
oneStepOrthogonalize[vct_List, vecmat_] :=
  Fold[(#1 - (vct.#2)/(#2.#2)*#2)&, vct, vecmat]
GS[vec_List] :=

```

3. MARCO TEÓRICO E IMPLEMENTACIÓN

```
Fold[Append[#1, oneStepOrthogonalize[#2, #1]]&, {}, vec]
```

En ambos casos, la función `GS[]` recibe una matriz cuyos renglones son los vectores que se van a ortogonalizar y devuelve la matriz cuyos renglones son ortogonales. Los tiempos de ejecución de ambas implementaciones son muy semejantes, como se muestra en el siguiente ejemplo, en donde llamaremos `GS1[]` a la implementación procedural y `GS2[]` a la implementación basada en reglas.

Consideremos el siguiente conjunto de vectores:

$$\{u_1, u_2, u_3\} = \{(1, 1, 0), (0, 1, 1), (1, 0, 1)\}. \quad (3.21)$$

El proceso de ortogonalización con `GS1[]` es:

```
In [] := U = {{1, 1, 0}, {0, 1, 1}, {1, 0, 1}};
In [] := GS1[U]
Out [] := {{1, 1, 0}, {-1/2, 1/2, 1}, {2/3, -2/3, 2/3}}
In [] := Timing[GS1[U]]
Out [] := {0.000089, {{1, 1, 0}, {-1/2, 1/2, 1}, {2/3, -2/3, 2/3}}}
```

Y usando `GS2[]`:

```
In [] := U = {{1, 1, 0}, {0, 1, 1}, {1, 0, 1}};
In [] := GS2[U]
Out [] := {{1, 1, 0}, {-1/2, 1/2, 1}, {2/3, -2/3, 2/3}}
In [] := Timing[GS2[U]]
Out [] := {0.000092, {{1, 1, 0}, {-1/2, 1/2, 1}, {2/3, -2/3, 2/3}}}
```

En las siguientes dos secciones se describen las funciones `MLE[]` y `LE[]` desarrolladas en este trabajo, y se ejemplifica el uso de ambas con el sistema de Lorentz. El código completo de cada función se presenta en los Apéndices [A.1](#) y [A.2](#), respectivamente.

3.3.2. Máximo exponente de Lyapunov

La función `MLE[]` calcula el máximo exponente de Lyapunov y tiene los siguientes parámetros de entrada:

```
MLE[Sys, CI, Ti, Tf, vars, OptionsPattern[{ "Metodo" -> "Automatic",
"PhaseSpace_Plot" -> "False" }]]
```

Donde:

Sys: Lista con las ecuaciones del sistema dinámico.

CI: Lista con las condiciones iniciales.

Ti: Tiempo inicial de integración (usualmente se inicia en $Ti=0$).

Tf: Tiempo total de integración.

vars: Lista de variables dependientes del sistema dinámico.

"Metodo" → "Automatic": El método predeterminado para resolver el sistema dinámico es el "Automatic", y es el recomendado por *Mathematica*. Si se desea usar otro método, se introduce el nombre [36].

"PhaseSpace_Plot" → "False": Valor predeterminado para desplegar gráficas, si se requiere visualizar las gráficas del espacio fase, se introduce "True".

Consideremos como ejemplo del uso de la función al sistema de Lorenz (3.15), con condiciones iniciales (3.16). El arreglo con el sistema y las condiciones iniciales se define así:

```
In []:= F = {16 (y[t] - x[t]), x[t] (45.92 - z[t]) - y[t],  
x[t] y[t] - 4 z[t]};  
CI = {2/10, 3/10, 5/10};  
var={x[t], y[t], z[t]};
```

Iniciando en $t = 0$ hasta un tiempo máximo de $t = 1000$, el cálculo del MEL, con los valores predeterminados para el método numérico (**Automatic**) y para el despliegue de gráficos (**False**), el cálculo se realiza de la siguiente manera.

```
In []:= MLE[F, CI, 0, 200, var ]  
Out []:= 1.49181
```

Consideremos ahora el mismo ejemplo pero usando el método **ExplicitRungeKutta** y mostrando gráficas del espacio fase:

```
In []:= MLE[F, CI, 0, 200, var, "Metodo"→"ExplicitRungeKutta",  
"PhaseSpace_Plot"→"True"]  
Out []:= 1.47376
```

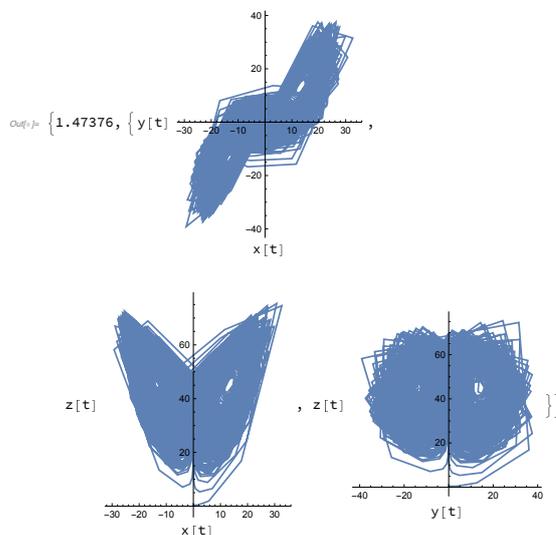


Figura 3.1: Gráficas en espacio fase que despliega la función `MLE[]`, para el ejemplo del sistema de Lorenz

3.3.3. Espectro completo de los exponentes de Lyapunov

La función `LE[]` calcula el espectro completo de los exponentes de Lyapunov y tiene los siguientes parámetros de entrada:

```
LE[Sys_ , CIS_ ,  $\tau$ _ , k_ , vars_List , OptionsPattern [ { "Metodo" ->
"Automatic" , "Lyapunov_Plot" -> "False" } ]]
```

Donde, tomando en cuenta el algoritmo descrito en la Sección 3.2, se tiene:

Sys: Lista con las ecuaciones del sistema dinámico.

CIS: Lista con las condiciones iniciales.

τ : Intervalo corto de tiempo en el que se resuelve la ecuación variacional (Pasos 2 y 5).

k: Número de iteraciones (paso 6).

vars: Lista de variables dependientes del sistema dinámico.

"Metodo" → "Automatic": El método predeterminado para resolver el sistema dinámico es el "Automatic", y es el recomendado por *Mathematica*. Si se desea usar otro método, se introduce el nombre [36].

"Lyapunov_Plot" → "False": Valor predeterminado para desplegar gráficas, si se requiere visualizar las gráficas del espacio fase, se introduce "True".

Consideremos el mismo ejemplo del sistema de Lorenz (3.15), con condiciones iniciales (3.16), que se usó para calcular el MEL en la sección anterior. Considerando $\tau = 0.1$ y $k = 1000$ iteraciones, se tiene:

```
In []:= LE[F, CI, 0.1, 1000, var]
Out []:= {1.4316, 0.001556, -22.393}
```

Si usamos las mismas entradas, pero usamos el método `ExplicitRungeKutta` de orden 4, se obtiene

```
In []:= LE[F, CI, 0.1, 1000, var, "Metodo" ->
      {"ExplicitRungeKutta", "DifferenceOrder" -> 5}]
Out []:= {1.44003, -0.008244, -22.4318}
```

En resumen, las funciones `MLE[]` y `LE[]` son la principal aportación de este trabajo. A diferencia de otras implementaciones en *Mathematica* que se han propuesto [20, 31], que consideran ejemplos particulares de sistemas dinámicos, las funciones desarrolladas aquí pueden aplicarse a cualquier sistema dinámico.

Debido a las características de algunos sistemas dinámicos, principalmente en regímenes caóticos, se pueden producir avisos, *warnings*, durante los cálculos, que son resultado de las limitaciones de los métodos numéricos para esos sistemas. Si bien *Mathematica* recomienda usar el método `Automatic`, en esos casos es conveniente que el usuario experimente usando otros métodos y mayores o menores tiempos.

En algunos casos de sistemas caóticos, se recomienda convertir los valores decimales de los parámetros del sistema dinámico a racionales, usando la función `Rationalize[]`. Esto debido a la extrema sensibilidad del sistema a las condiciones iniciales.

Resultados

En esta sección se presentan aplicaciones de las funciones desarrolladas en este trabajo a problemas conocidos y se comparan los resultados con los publicados, cuando es el caso. Para cada problema, los métodos usados para resolver el sistema fueron: `Automatic`, `ExplicitRungeKutta` y `Dormand-Prince`.

Sobre estos métodos, podemos mencionar lo siguiente. Con `Automatic`, *Mathematica* intenta encontrar el método más adecuado para el cálculo, pero puede presentar problemas de convergencia [30, 38]. El método `ExplicitRungeKutta` es uno de los métodos más utilizados para resolver sistemas caóticos, sobre todo el de orden 4 (llamado *RK4*), porque es más preciso [39]. Finalmente, el método `Dormand-Prince` es el método más utilizado en el programa *Matlab*, en donde es conocido como *Ode45* [40].

El método y el número de iteraciones o tiempo final de los resultados presentados enseguida, se fijaron usando como guía los valores reportados en la literatura, es decir, se ajustaron para obtener una mejor aproximación a estos. En el caso en que no se cuente con valores reportados, se recomienda usar el método por defecto (`Automatic`) y un tiempo máximo $t = 1000$, en el caso del MEL, o $k = 1000$, para el espectro completo, y simplemente verificar el signo de los resultados, dependiendo del régimen en el que se encuentra el sistema, de acuerdo con la Tabla 2.1. Este es el caso, por ejemplo, del sistema de [41]; para el régimen caótico sabemos que el MEL debe ser positivo y para el estado estacionario el MEL tiende a cero y el espectro completo debe producir valores negativos. Finalmente, si el uso de `Automatic` produce *warnings*, se recomienda usar `ExplicitRungeKutta` o `Dormand-Prince` o el método que sea más recomendable para el sistema a resolver, algunos otros métodos son: `Adams`, `BDF`, `StiffnessSwitching` por mencionar algunos.

4.1. Sistema de Lorenz

Como una primera aplicación, consideremos el sistema de Lorenz (3.15), que usamos en los ejemplos de las Secciones 3.3.2 y 3.3.3. Este fue desarrollado por Edward Lorenz en 1963, como un modelo matemático simplificado de la convección atmosférica.

4. RESULTADOS

Usaremos las mismas condiciones iniciales (3.16) pero, a diferencia de los ejemplos anteriores, consideraremos tanto tiempos como iteraciones mayores, con lo que los tiempos de cálculo aumentan considerablemente.

Definimos el sistema, las condiciones iniciales y las variables independientes:

Listing 4.1: Definición del sistema de Lorenz

```
In [] := F = {16 (y[t] - x[t]), x[t] (4592/100 - z[t]) - y[t],  
             x[t] y[t] - 4 z[t]};  
CI = {2/10, 3/10, 5/10};  
var = {x[t], y[t], z[t]};
```

Para el cálculo del MEL, usamos el método `Automatic`, comenzando en $t = 0$, hasta $t = 10\,000$:

```
In [] := MLE[F, CI, 0, 10000, var]  
Out [] := 1.5002942090717714899
```

Para el cálculo de los EL, con el método `Automatic` para un $\tau = 0.1$ y $k = 10\,000$ en *Mathematica* quedaría como sigue:

```
In [] := LE[F, CI, 0.1, 10000, var]  
Out [] := {1.50078, -0.00121574, -22.4996}
```

En la Tabla 4.1 se comparan los valores obtenidos usando las funciones `MEL[]` y `EL[]`, desarrolladas en este trabajo, con los reportados en la literatura.

Tabla 4.1: Comparación de los resultados obtenidos en este trabajo para el sistema de Lorenz, con los reportados en la literatura. Se utilizó el método `Automatic`, con un tiempo de integración de 0 – 10000 para el primer caso y $\tau = 0.1$ y $k = 10\,000$, para el segundo.

Referencia	Máximo exponente	Espectro completo
Sandri [20]	1.46574	$\lambda_1 = 1.48245$ $\lambda_2 = -0.0001195$ $\lambda_3 = -22.4654$
H. Binous & N. Zakia [31]	1.48665	$\lambda_1 = 1.48381$ $\lambda_2 = -0.000694557$ $\lambda_3 = -22.4916$
Dubeibe [30]	1.501	
Este trabajo	1.50029421	$\lambda_1 = 1.50078$ $\lambda_2 = -0.00121574$ $\lambda_3 = -22.4996$

4.2. Sistema de Rössler

El sistema de Rössler fue desarrollado por el matemático Suizo Otto Rössler, en 1976, como el sistema mínimo requerido para exhibir caos [26]. El sistema es el siguiente:

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} y - x \\ x + 0.15y \\ 0.2 + z(x - 10) \end{bmatrix}. \quad (4.1)$$

Para este sistema, consideraremos las mismas condiciones iniciales que para el sistema de Lorenz.

El sistema, con las condiciones iniciales y las variables independientes se definen como sigue:

$$\begin{aligned} \mathbf{In} &:= \mathbf{F} = \{-y[t] - z[t], x[t] + \frac{15}{100} y[t], \\ &\frac{2}{10} + z[t] (x[t] - 10)\}; \\ \mathbf{CI} &= \{\frac{2}{10}, \frac{3}{10}, \frac{5}{10}\}; \\ \mathbf{var} &= \{x[t], y[t], z[t]\}; \end{aligned}$$

El valor de referencia para el MEL en la literatura es $\lambda = 0.09$ [30], por lo que usándolo como guía, se obtiene la mejor aproximación usando el método `Dorman-Prince`,

4. RESULTADOS

comenzando en $t = 0$, hasta $t = 700$. Como se mencionó anteriormente, éste método es el equivalente al *ODE45*, que es el método por defecto de *Matlab* y su aplicación no es directa, ya que requiere definir primero la llamada Tabla de Butcher [42]:

```
In []:= DOPRIamat = {{1/5}, {3/40, 9/40}, {44/45, -56/15, 32/9},
{19372/6561, -25360/2187, 64448/6561, -212/729},
{9017/3168, -355/33, 46732/5247, 49/176, -5103/18656},
{35/384, 0, 500/1113, 125/192, -2187/6784, 11/84}};
DOPRIbvec = {35/384, 0, 500/1113, 125/192, -2187/6784, 11/84, 0};
DOPRIcvec = {1/5, 3/10, 4/5, 8/9, 1, 1};
DOPRIevec = {71/57600, 0, -71/16695, 71/1920, -17253/339200,
22/525, -1/40};
DOPRICoefficients[5, p_] := N[{DOPRIamat, DOPRIbvec, DOPRIcvec,
DOPRIevec}, p];
```

Con esto, se realiza el cálculo del MEL:

```
In []:= MLE[F,CI,0,700,var,"Metodo" -> {"ExplicitRungeKutta",
"DifferenceOrder" -> 5, "Coefficients" -> DOPRICoefficients,
"StiffnessTest" -> False}]
Out []:= 0.090443
```

Para el espectro completo, usamos el método *ExplicitRungeKutta* de orden 5 para un $\tau = 0.1$ y $k = 100\,000$, este método con estas características de integración fueron escogidos para presentarse debido a que era la mejor aproximación a $\lambda_1 = 0.09$. En *Mathematica* quedó como sigue:

```
In []:= LE[F,CI,0.1,100000,var,"Metodo"->{"ExplicitRungeKutta",
"DifferenceOrder" -> 5}]
Out []:= {0.0899867,0.000323853,-9.80163}
```

En la Tabla 4.2 se comparan los valores obtenidos con los reportados en la literatura.

Tabla 4.2: Comparación de los resultados obtenidos en este trabajo para el sistema de Rössler. Para el MEL se utilizó el método *Dorman-Prince*, con un tiempo de integración de 0 – 700, ya para el espectro completo, utilizó el método *RK5*, con $\tau = 0.1$ y $k = 100\ 000$.

Referencia	Máximo exponente	Espectro completo
H. Binous & N. Zakia [31]	0.0883907	$\lambda_1 = 0.092337$ $\lambda_2 = 0.0194685$ $\lambda_3 = -9.83146$
Dubeibe [30]	0.088	
Este trabajo	0.090443	$\lambda_1 = 0.0899867$ $\lambda_2 = 0.000323853$ $\lambda_3 = -9.80163$

4.3. Sistema con un atractor extraño y un toro invariante

Es un sistema definido por 3 ecuaciones diferenciales ordinarias no lineales, en el que un atractor extraño coexiste con un toro, que fue propuesto por J.C. Sprott [43].

El sistema es el siguiente:

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} y + 2xy + xz \\ 1 - 2x^2 + yz \\ x - x^2 - y^2 \end{bmatrix}. \quad (4.2)$$

Como se mencionó, este sistema tiene dos atractores: un toro y un atractor extraño; los valores de los exponentes de Lyapunov son determinados por la cercanía de las condiciones iniciales a uno de los dos atractores.

Para el caso del atractor extraño, la condición inicial es:

$$\dot{\mathbf{X}}(0) = x_0 = \begin{bmatrix} \dot{x}(0) \\ \dot{y}(0) \\ \dot{z}(0) \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}. \quad (4.3)$$

Entonces el sistema, las condiciones iniciales y las variables independientes se definieron como sigue:

```
In []:= F = {y[t] + 2 x[t] y[t] + x[t] z[t],
1 - 2 x[t]^2 + y[t] z[t], x[t] - x[t]^2 - y[t]^2};
CI = {2, 0, 0};
var = {x[t], y[t], z[t]};
```

4. RESULTADOS

Para el cálculo del MEL, se usó el método `Automatic`, desde $t = 0$, hasta $t = 10\,000$:

```
In []:= MLE[F, CI, 0, 10000, var ]]
```

```
Out []:= 0.0539878
```

Para el cálculo de los EL, se usó el método `ExplicitRungeKutta` de orden 5 para un $\tau = 0.1$ y $k = 10\,000$:

```
In []:= LE[F, CI, 0.1, 10000, var, "Metodo" -> {"ExplicitRungeKutta",
"DifferenceOrder" -> 5}]
```

```
Out []:= {0.0540864, -0.00647083, -0.158119}
```

En la Tabla 4.3 se comparan los valores obtenidos, con los valores reportados en la literatura:

Tabla 4.3: Comparación de los exponentes de Lyapunov del sistema de Sprott con atractor extraño. Para el MEL se utilizó el método `Automatic`, con un tiempo de integración de $0 - 10\,000$. Para los EL se utilizó el método ERK(5) con $\tau = 0.1$ y $k = 10\,000$.

Referencia	Máximo exponente	Espectro completo
Sprott [43]		$\lambda_1 = 0.0540$ $\lambda_2 = 0$ $\lambda_3 = -0.1575$
Este trabajo	0.0539878	$\lambda_1 = 0.0540864$ $\lambda_2 = -0.00647083$ $\lambda_3 = -0.158119$

Para el caso del toro, las condiciones iniciales son:

$$\dot{\mathbf{X}}(0) = x_0 = \begin{bmatrix} \dot{x}(0) \\ \dot{y}(0) \\ \dot{z}(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \quad (4.4)$$

Entonces el sistema, las condiciones iniciales y las variables independientes se definieron como sigue:

```
In []:= F = {y[t] + 2 x[t] y[t] + x[t] z[t],
1 - 2 x[t]^2 + y[t] z[t], x[t] - x[t]^2 - y[t]^2};
CI = {1, 0, 0};
var = {x[t], y[t], z[t]};
```

Para el cálculo del MEL, se usó el método `Automatic`, desde $t = 0$, hasta $t = 10\,000$:

In [] := MLE[F, CI, 0, 10000, var }

Out [] := 0.000821872

El cálculo de los EL, con el método `ExplicitRungeKutta` de orden 3 para $\tau = 0.1$ y $k = 100\,000$ queda como sigue:

In [] := LE[F, CI, 0.1, 100000, var, "Metodo" -> {"ExplicitRungeKutta", "DifferenceOrder" -> 3}]

Out [] := {0.000822625, -0.00006387, -0.000732392}

En la Tabla 4.4, se comparan los valores obtenidos con los valores reportados en la literatura:

Tabla 4.4: Comparación de los exponentes de Lyapunov del sistema de Sprott con un toro.

Para el MEL se utilizó el método `Automatic` con un tiempo de integración de 0 – 10 000.

Para los EL se utilizó el método `ERK(3)` con $\tau = 0.1$ y $k = 100\,000$.

Referencia	Máximo exponente	Espectro completo
Sprott [43]		$\lambda_1 = 0$ $\lambda_2 = 0$ $\lambda_3 = 0$
Este trabajo	0.000821872	$\lambda_1 = 0.000822625$ $\lambda_2 = -0.00006387$ $\lambda_3 = -0.000732392$

4.4. Sistema electrocardiograma

Este sistema constituye un modelo para generar señales de electrocardiogramas basado en un sistema de reacción-difusión discretizado [41].

Las ecuaciones del modelo son:

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_1 - x_2 - Cx_1x_2 - x_1x_2^2 \\ Hx_1 - 3x_2 + Cx_1x_2 + x_1x_2^2 + \beta(x_4 - x_2) \\ x_3 - x_4 - Cx_3x_4 - x_3x_4^2 \\ Hx_3 - 3x_4 + Cx_3x_4 + x_3x_4^2 + 2\beta(x_2 - x_4) \end{bmatrix}, \quad (4.5)$$

donde $C = 1.35$, $\beta = 4$, y el parámetro H es un parámetro de control, cuya variación permite diferentes dinámicas, como se muestra en la Tabla 4.5.

Tabla 4.5: Tipo de dinámica, de acuerdo a el parámetro de control H.

H	Tipo de atractor
7	Ciclo límite
2.74	Ciclo límite doble periodo
2.72972	Toro
2.7126	Atractor extraño
8	Estado estacionario

4.4.1. Ciclo límite

Para el ciclo límite, el sistema, las condiciones iniciales y las variables independientes se definen como sigue:

In [] := c=135/100;

B=4;

H=7;

F={x₁[t]-x₂[t]-c x₁[t] x₂[t]-x₁[t] x₂[t]²,

H x₁[t]-3 x₂[t]+c x₁[t] x₂[t]+x₁ x₂[t]²+B (x₄[t]-x₂[t]),

x₃[t]-x₄[t]-c x₃[t] x₄[t]-x₃[t] x₄[t]²,

H x₃[t]-3 x₄[t]+c x₃[t] x₄[t]+x₃[t] x₄[t]²+2 B (x₂[t]-x₄[t])};

CI={0,0,1/10,0};

var = {x₁[t], x₂[t], x₃[t], x₄[t]};

Para el cálculo del MEL, se utilizó el método **Dorman-Prince**, desde $t = 0$, hasta $t = 10\ 000$, donde, como se hizo en el ejemplo del sistema de Rössler, hay que definir la Tabla de Butcher:

In [] := DOPRIamat = {{1/5}, {3/40, 9/40}, {44/45, -56/15, 32/9},

{19372/6561, -25360/2187, 64448/6561, -212/729},

{9017/3168, -355/33, 46732/5247, 49/176, -5103/18656},

{35/384, 0, 500/1113, 125/192, -2187/6784, 11/84}}};

DOPRIbvec = {35/384, 0, 500/1113, 125/192, -2187/6784, 11/84, 0};

DOPRIcvec = {1/5, 3/10, 4/5, 8/9, 1, 1};

DOPRIevec = {71/57600, 0, -71/16695, 71/1920, -17253/339200,

22/525, -1/40};

DOPRICoefficients [5, p_] := N[{DOPRIamat, DOPRIbvec, DOPRIcvec,

DOPRIevec}, p];

Con esto, se realiza el cálculo del MEL:

In [] := MLE[F, CI, 0, 10000, var, "Metodo" -> "ExplicitRungeKutta",
"DifferenceOrder" -> 5, "Coefficients" -> DOPRICoefficients,

```
"StiffnessTest" -> False}]
Out[]:= 0.000417035
```

Para el cálculo de los EL, se utilizó el método `ExplicitRungeKutta` de orden 6, para $\tau = 0.1$ y $k = 100\ 000$:

```
In[]:= LE[F,CI,0.1,100000,var,"Metodo"->{"ExplicitRungeKutta",
"DifferenceOrder" -> 6}]
Out[]:= {0.000446032,-1.07774,-1.08214,-8.8622}
```

4.4.2. Ciclo límite con doblamiento de período

Para el caso del ciclo límite con doblamiento de período, el sistema, las condiciones iniciales y las variables independientes se definen como:

```
In[]:= c=135/100;
B=4;
H=Rationalize[2.74];
F={x1[t]-x2[t]-c x1[t] x2[t]-x1[t] x2[t]^2,
H x1[t]-3 x2[t]+c x1[t] x2[t]+x1 x2[t]^2+B (x4[t]-x2[t]),
x3[t]-x4[t]-c x3[t] x4[t]-x3[t] x4[t]^2,
H x3[t]-3 x4[t]+c x3[t] x4[t]+x3[t] x4[t]^2+2 B (x2[t]-x4[t])};
CI={0,0,1/10,0};
var={x1[t],x2[t],x3[t],x4[t]};
```

Para el cálculo del MEL, se utilizó el método `Automatic`, comenzando en $t = 9000$, hasta $t = 100\ 000$. Esto resultó ser lo más conveniente, ya que en este régimen se presenta un transiente que afecta al resultado:

```
In[]:= MLE[F,CI,0,100000,var]
Out[]:= 0.000435418
```

Para el cálculo de los EL se utilizó el método `ExplicitRungeKutta` de orden 4 para un $\tau = 0.1$ y $k = 500\ 000$:

```
In[]:= LE[F,CI,0.1,500000,var,"Metodo"->{"ExplicitRungeKutta",
"DifferenceOrder" -> 4}]
Out[]:= {0.000513375,-0.0241631,-0.025934,-13.5982}
```

4.4.3. Toro

Para este caso, el sistema, las condiciones iniciales y las variables independientes se definen como sigue:

```
In[]:= c=135/100;
B=4;
H=Rationalize[2.72972];
```

4. RESULTADOS

```
F={x1[t]-x2[t]-c x1[t] x2[t]-x1[t] x2[t]^2,
H x1[t]-3 x2[t]+c x1[t] x2[t]+x1 x2[t]^2+B (x4[t]-x2[t]),
x3[t]-x4[t]-c x3[t] x4[t]-x3[t] x4[t]^2,
H x3[t]-3 x4[t]+c x3[t] x4[t]+x3[t] x4[t]^2+2 B (x2[t]-x4[t])};
CI={0,0,1/10,0};
var = {x1[t],x2[t],x3[t],x4[t]};
```

Para el cálculo del MEL, se utiliza el método **Dorman-Prince**, comenzando en $t = 9000$, hasta $t = 100\,000$. Como es usual en este caso, primero se define la Tabla de Butcher:

```
In[]:= DOPRIamat = {{1/5}, {3/40, 9/40}, {44/45, -56/15, 32/9},
{19372/6561, -25360/2187, 64448/6561, -212/729},
{9017/3168, -355/33, 46732/5247, 49/176, -5103/18656},
{35/384, 0, 500/1113, 125/192, -2187/6784, 11/84}};
DOPRIbvec = {35/384, 0, 500/1113, 125/192, -2187/6784, 11/84, 0};
DOPRIcvec = {1/5, 3/10, 4/5, 8/9, 1, 1};
DOPRIevec = {71/57600, 0, -71/16695, 71/1920, -17253/339200,
22/525, -1/40};
DOPRICoefficients[5, p_] := N[{DOPRIamat, DOPRIbvec, DOPRIcvec,
DOPRIevec}, p];
```

Con esto, se calcula el MEL como sigue:

```
In[]:= MLE[F,CI,9000,100000,var,"Metodo"->{"ExplicitRungeKutta",
"DifferenceOrder" -> 5, "Coefficients" -> DOPRICoefficients,
"StiffnessTest" -> False}]
Out[]:= 0.17570818654944081729
```

Para el cálculo de los EL se utiliza el método **Automatic**, para $\tau = 0.1$ y $k = 100\,000$:

```
In[]:= LE[F,CI,0.1,100000,var]
Out[]:= {0.176459, 0.00017024, -0.637751, -12.0002}
```

Los resultados que se obtuvieron para este caso son los de una dinámica caótica, debido a que $\lambda_1 > 0$, lo cuál no coincide con los valores esperados $\lambda_1 = \lambda_2 = 0$, de acuerdo con la Tabla 2.1. Las dinámicas cuasiperiódicas están al borde del caos, por lo que creemos que esa es la razón de que el cálculo numérico produzca un valor correspondiente a un sistema caótico y no a una dinámica cuasiperiódica.

4.4.4. Atractor extraño

Para la dinámica caótica, el sistema, las condiciones iniciales y las variables independientes se definen como sigue:

```
In[]:= c=135/100;
B=4;
H=Rationalize[2.7126];
```

```

F={x1[t]-x2[t]-c x1[t] x2[t]-x1[t] x2[t]^2,
H x1[t]-3 x2[t]+c x1[t] x2[t]+x1 x2[t]^2+B (x4[t]-x2[t]),
x3[t]-x4[t]-c x3[t] x4[t]-x3[t] x4[t]^2,
H x3[t]-3 x4[t]+c x3[t] x4[t]+x3[t] x4[t]^2+2 B (x2[t]-x4[t])};
CI={0,0,1/10,0};
var = {x1[t],x2[t],x3[t],x4[t]};

```

Para el cálculo del MEL, se utiliza el método **Dorman-Prince**, desde $t = 0$, hasta $t = 100\,000$. La Tabla de Butcher es:

```

In[]:= DOPRIamat = {{1/5}, {3/40, 9/40}, {44/45, -56/15, 32/9},
{19372/6561, -25360/2187, 64448/6561, -212/729},
{9017/3168, -355/33, 46732/5247, 49/176, -5103/18656},
{35/384, 0, 500/1113, 125/192, -2187/6784, 11/84}};
DOPRIbvec = {35/384, 0, 500/1113, 125/192, -2187/6784, 11/84, 0};
DOPRIcvec = {1/5, 3/10, 4/5, 8/9, 1, 1};
DOPRIevec = {71/57600, 0, -71/16695, 71/1920, -17253/339200,
22/525, -1/40};
DOPRICoefficients[5, p_] := N[{DOPRIamat, DOPRIbvec, DOPRIcvec,
DOPRIevec}, p];

```

Con esto, se calcula el MEL:

```

In[]:= MLE[F,CI,0,100000,var,"Metodo"->{"ExplicitRungeKutta",
"DifferenceOrder" -> 5, "Coefficients" -> DOPRICoefficients,
"StiffnessTest" -> False}]
Out[]:= 0.17961834864063063337

```

Para el cálculo de los EL se utiliza el método **Automatic**, para $\tau = 0.1$ y $k = 100\,000$:

```

In[]:= LE[F,CI,0.1,100000,var]
Out[]:= {0.173493, 0.0000467256, -0.650737, -12.0288}

```

4.4.5. Punto fijo

Para este caso, el sistema, las condiciones iniciales y las variables independientes se definen como sigue:

```

In[]:= c=135/100;
B=4;
H = 8;
F={x1[t]-x2[t]-c x1[t] x2[t]-x1[t] x2[t]^2,
H x1[t]-3 x2[t]+c x1[t] x2[t]+x1 x2[t]^2+B (x4[t]-x2[t]),
x3[t]-x4[t]-c x3[t] x4[t]-x3[t] x4[t]^2,
H x3[t]-3 x4[t]+c x3[t] x4[t]+x3[t] x4[t]^2+2 B (x2[t]-x4[t])};
CI={0,0,1/10,0};
var = {x1[t],x2[t],x3[t],x4[t]};

```

4. RESULTADOS

Para el cálculo del MEL, se utiliza el método **Automatic**, desde $t = 0$, hasta $t = 1\ 000\ 000$:

In [] := MLE[F, CI, 0, 1000000, var]

Out [] := -0.0000112229

Se escogió este resultado en el que se utilizó un tiempo muy largo para este caso debido a que, aún cuando $\lambda_i < 0$ después de ciertos valores de tiempo λ_1 tiende a cero siguiendo una ley de potencias $\lambda_1 \propto t^{-1}$ con la función MEL[] lo que implica que el sistema tiene una dinámica regular.

Para tiempos más cortos $\lambda_1 < 0$ lo que indica que es un punto fijo, pero conforme vas aumentando los tiempos de integración λ_1 se aproxima a cero.

Para el cálculo de los EL se utiliza el método **Automatic** para $\tau = 0.1$ y $k = 100\ 000$:

In [] := LE[F, CI, 0.1, 100000, var]

Out [] := { -0.00831073, -0.00921649, -3.32343, -7.55039 }

La Tabla 4.6 presenta los valores obtenidos para el MEL y los EL para todas las dinámicas presentes en el sistema del electrocardiograma:

Tabla 4.6: Resultados del sistema 4.5.

H	Dinámica	MEL	EL	Características de integración
7	Periódica (Ciclo límite)	$\lambda_1 = 0.000417035$	$\lambda_1 = 0.000446032$ $\lambda_2 = -1.07774$ $\lambda_3 = -1.08214$ $\lambda_4 = -8.8622$	MEL: Dorman-Prince, $t_i = 0$, $t_f = 10\ 000$ EL: ERK(6), $\tau = 0.1$, $k = 100\ 000$
2.74	Periódica (Ciclo límite doble periodo)	$\lambda_1 = 0.000435418$	$\lambda_1 = 0.000513375$ $\lambda_2 = -0.0241631$ $\lambda_3 = -0.025934$ $\lambda_4 = -13.5982$	MEL: Automatic, $t_i = 9000$, $t_f = 100\ 000$ EL: ERK(4), $\tau = 0.1$ y $k = 500\ 000$
2.72972	Cuasiperiódica (Toro)	$\lambda_1 = 0.17570818654944081729$	$\lambda_1 = 0.176459$ $\lambda_2 = 0.00017024$ $\lambda_3 = -0.637751$ $\lambda_4 = -12.0002$	MEL: Dorman-Prince, $t_i = 9000$, $t_f = 100\ 000$ EL: Automatic, $\tau = 0.1$ y $k = 100\ 000$
2.7126	Caos (Atractor extraño)	$\lambda_1 = 0.17961834864063063337$	$\lambda_1 = 0.173493$ $\lambda_2 = 0.0000467256$ $\lambda_3 = -0.650737$ $\lambda_4 = -12.0288$	MEL: Dorman-Prince, $t_i = 0$, $t_f = 100\ 000$ EL: Automatic, $\tau = 0.1$, $k = 100\ 000$
8	Estado estacionario (Punto fijo)	$\lambda_1 = -0.0000112229$	$\lambda_1 = -0.00831073$ $\lambda_2 = -0.00921649$ $\lambda_3 = -3.32343$ $\lambda_4 = -7.55039$	MEL: Automatic, $t_i = 0$, $t_f = 1\ 000\ 000$ EL: Automatic, $\tau = 0.1$, $k = 100\ 000$

Discusión y conclusiones

En este trabajo de desarrollaron dos funciones en *Mathematica*, `MLE[]` y `LE[]`, para calcular, respectivamente, el máximo exponente de Lyapunov y el espectro completo de exponentes. Si bien, hay antecedentes en la literatura del cálculo de éstos exponentes con *Mathematica*, se trata de propuestas desarrolladas para resolver casos particulares. Las funciones que proponemos se aplican a cualquier sistema dinámico; tres de las entradas que requieren esas funciones son: el sistema dinámico, las variables dependientes y las condiciones iniciales. Con estas funciones se llena un hueco que existía en las librerías públicas de *Mathematica* [44], para el cálculo de exponentes de Lyapunov.

La ventaja principal de las funciones propuestas, como ya se mencionó, es que son generales en el sentido de que pueden usarse con cualquier sistema dinámico, planteado como un sistema de ecuaciones diferenciales ordinarias de primer orden. La principal desventaja de nuestra propuesta es que muchos de los sistemas dinámicos de interés presentan complicaciones en su solución numérica y no existe un método numérico universal, es decir, que produzca buenos resultados con cualquier sistema. Debido a esto, la precisión de los resultados que `MLE[]` y `LE[]` arrojan, depende en gran medida del usuario; del método numérico que se elija para resolver el sistema de ecuaciones diferenciales y de los tiempos de integración. Dada esta situación, en los siguientes párrafos se proponen algunas sugerencias que pueden ser de utilidad.

Respecto a los métodos para resolver el sistema de ecuaciones diferenciales, si el usuario desconoce cuál es el método más adecuado para su problema, *Mathematica* recomienda el uso de `Automatic`, en una primera instancia, ya que esa opción tratará de buscar el método que produzca los mejores resultados para el problema particular. Si `Automatic` arroja *warnings* que puedan afectar el resultado, se recomienda ya sea `ExplicitRungeKutta` o `Dorman-Prince`. Este último, en nuestra experiencia, es el más estable y es el equivalente a `ODE45`, que es el método que usa *MATLAB* por defecto. Hay que mencionar también que, en la gran mayoría de los casos los *warnings* no afectan el resultado.

Respecto a los tiempos de cálculo, para el cálculo del MEL se recomienda utilizar tiempos finales T_f grandes, como $T_f \geq 1000$, tomando en cuenta que T_f mayores aumentan los tiempos de cálculo, sobre todo para caos en donde la complejidad computacional

5. DISCUSIÓN Y CONCLUSIONES

aumenta con el tiempo.

Por otro lado, si se tiene información acerca del transiente, se puede considerar un tiempo inicial T_i diferente de 0. Esto fue el caso en los cálculos para el sistema 4.4, donde se conocía la duración del transiente.

Para el caso de los EL, de las pruebas realizadas se concluye que entre más pequeño sea τ más iteraciones (k) se necesitará para que el cálculo se estabilice.

En resumen, las funciones propuestas pueden ser de gran utilidad en diversas áreas en donde se requiera identificar el tipo de dinámica y la presencia del caos; en esos casos no es importante la precisión sino el signo de los exponentes. En el caso en donde la precisión del resultado es importante, dada la generalidad de las funciones, la interacción del usuario y las pruebas son inevitables.

A.1. Máximo exponente de Lyapunov en Mathematica

Listing A.1: Código para calcular el máximo exponente de Lyapunov en Mathematica

```

MLE[Sys_, CI_, Ti_, Tf_, vars_List, OptionsPattern[{"Metodo" ->
"Automatic", "PhaseSpace_Plot" -> "False"}]] := Module[{S=Sys, J,
X0=CI, TI=Ti, T=Tf, d, v, phit, phitp, ec1, ec2,  $\phi_0$ , ini1, ini2, syst, var,
solt, dev, DX0, DXT, MEL, i, j, G, l},

d=Length[Sys];
J=Outer[D, Sys, vars];
v=Table[ $\phi_{j+(i-1)d}[t]$ , {i, 1, d}];
phit=Transpose[Table[v, {j, 1, d}]];
phitp=Flatten[J. phit];
ec1=Table[D[vars[[i]], t]==S[[i]], {i, 1, d}];
ec2=Table[D[ $\phi_i[t]$ , t]==phitp[[i]], {i, 1,  $d^2$ }];
 $\phi_0$ =Flatten[IdentityMatrix[d]];
ini1=Table[vars[[i]]==X0[[i]].t->0, {i, 1, d}];
ini2=Table[ $\phi_i[0]==\phi_0[[i]]$ , {i, 1,  $d^2$ }];
syst=Join[ec1, ec2, ini1, ini2];
var=Join[vars, Flatten[phit]];
solt=NDSolve[syst, var, {t, TI, T}, Method->OptionValue["Metodo"]];
dev=Table[Random[]*10-16, d];
DX0 = CI+dev;
DXT=Transpose[First[Table[v, {j, 1, d}]].solt/.t->T];
MEL=Log[Norm[DXT.DX0]/Norm[DX0]]/T;
If[OptionValue["PhaseSpace_Plot"]=="False", Return[MEL],
l=Subsets[vars, {2}];
G={};

```

```

For [ i=1, i<=Length [ l ] , i++, G=Append [ G, Labeled [ ParametricPlot [
Evaluate [ l [[ i ] ] /. solt ] , { t, TI, T } ] , l [[ i ] ] , { Bottom, Left } ] ] ];
Return [ { MEL, G } ] ]

```

A.2. Espectro completo de los exponentes de Lyapunov

Listing A.2: Código para calcular el espectro completo de los exponentes de Lyapunov en Mathematica

```

LE [ Sys_ , CIS_ ,  $\tau$ _ , k_ , vars_List , OptionsPattern [ { "Metodo" ->
"Automatic" , "Lyapunov_Plot" -> "False" } ] ] := Module [ { F=Sys, J ,
CIF=CIS, svar=vars, T= $\tau$ , r=k, d, v, phit , phitp , ec1 , ec2 ,  $\phi$ 0, ini1 , ini2 ,
syst , var , vars0 , VN, solt , Phit , U, norms , ListLCE , i , j } ,
d=Length [ Sys ] ;
J=Outer [ D, Sys , vars ] ;
v=Table [  $\phi_{j+(i-1)}$  d ] [ t ] , { i , 1 , d } ] ;
phit=Transpose [ Table [ v , { j , 1 , d } ] ] ;
phitp=Flatten [ J . phit ] ;
ec1=Table [ D [ vars [[ i ] ] , t ] == F [[ i ] ] , { i , 1 , d } ] ;
ec2=Table [ D [  $\phi_i$  [ t ] , t ] == phitp [[ i ] ] , { i , 1 , d2 } ] ;
 $\phi$ 0=Flatten [ IdentityMatrix [ d ] ] ;
ini1=Table [ vars [[ i ] ] == CIF [[ i ] ] /. t -> 0 , { i , 1 , d } ] ;
ini2=Table [  $\phi_i$  [ 0 ] ==  $\phi$ 0 [[ i ] ] , { i , 1 , d2 } ] ;
syst=Join [ ec1 , ec2 , ini1 , ini2 ] ;
var=Join [ vars , Flatten [ Transpose [ Table [ v , { j , 1 , d } ] ] ] ] ;
vars0=vars /. t -> 0 ; VN={ } ;
Do [
solt=NDSolve [ syst , var , { t , 0 , T } , Method -> OptionValue [ "Metodo" ] ] ;
Phit=Map [ First , Table [ v /. solt /. t -> T , { j , 1 , d } ] ] ;
U = GS [ Phit ] ;
norms = Map [ Norm , U ] ;
VN = Append [ VN, norms ] ;
Phit = U / norms ;
ini1=Map [ First , Table [ vars0 [[ i ] ] == vars [[ i ] ] /. solt /. t -> T ,
{ i , 1 , d } ] ] ;
ini2=Table [  $\phi_i$  [ 0 ] == Flatten [ Transpose [ Phit ] ] [[ i ] ] , { i , 1 , d2 } ] ;
syst=Join [ ec1 , ec2 , ini1 , ini2 ] ;
, { r } ] ;
ListLE=Rest [ FoldList [ Plus , 0 , Log [ VN ] ] ] / ( T Range [ r ] ) ;
If [ OptionValue [ "Lyapunov_Plot" ] == "False" , Return [ Last [ ListLE ] ] ,

```

```
{Last[ListLE], ListLinePlot[Transpose[ListLE]]}
```


Bibliografía

- [1] Michael Brin and Garrett Stuck. *Introduction to dynamical systems*. Cambridge university press, 2002. [5](#)
- [2] David K Arrowsmith, Colin M Place, CH Place, et al. *An introduction to dynamical systems*. Cambridge university press, 1990. [5](#)
- [3] Ernesto A Lacomba. Los sistemas dinámicos¿ qué son y para qué sirven. *Miscelánea matemática*, 32:39–50, 2000. [5](#)
- [4] W. E. Boyce and R. C. DiPrima. *ELEMENTARY DIFFERENTIAL EQUATIONS AND BOUNDARY VALUE PROBLEMS*. John Wiley and Sons, 10 edition, 2017. [6](#), [11](#)
- [5] Anatole Katok and Boris Hasselblatt. *Introduction to the modern theory of dynamical systems*. Number 54 in Encyclopedia of Mathematics and its Applications. Cambridge university press, 1997. [6](#)
- [6] Eric W. Weisstein. "phase portrait." From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/PhasePortrait.html>. [6](#)
- [7] Eric W. Weisstein. "attractor." From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/Attractor.html>. [7](#), [8](#)
- [8] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Perseus Books, 1994. [7](#), [8](#), [9](#), [10](#), [20](#)
- [9] John H Argyris, Gunter Faust, Maria Haase, and Rudolf Friedrich. *An exploration of dynamical systems and chaos: completely revised and enlarged second edition*. Springer, 2015. [8](#), [14](#), [15](#)
- [10] Eric W. Weisstein. "torus." From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/Torus.html>. [8](#)
- [11] Vadim S Anishchenko and GI Strelkova. Attractors of dynamical systems. In *1997 1st International Conference, Control of Oscillations and Chaos Proceedings (Cat. No. 97TH8329)*, volume 3, pages 498–503. IEEE, 1997. [8](#)

BIBLIOGRAFÍA

- [12] María Lourdes Bruzco et al. El efecto mariposa y sus implicaciones estratégicas en el contexto organizacional. *Revista Ciencias Estratégicas*, 20(27), 2012. [8](#)
- [13] Kevin J Dooley. The butterfly effect of the "butterfly effect". *Nonlinear dynamics, psychology, and life sciences*, 13(3):279, 2009. [8](#)
- [14] Clark Robinson. *Dynamical systems: stability, symbolic dynamics, and chaos*. CRC press, 1998. [9](#)
- [15] Pedro Henrique Oliveira Silva, Lucas Giovanni Nardo, Samir Angelo Milani Martins, Erivelton Geraldo Nepomuceno, and Matjaž Perc. Graphical interface as a teaching aid for nonlinear dynamical systems. *European Journal of Physics*, 39(6):065105, 2018. [9](#)
- [16] Kathleen T Alligood, Tim D Sauer, and James A Yorke. *Chaos*. Springer, 1996. [9](#)
- [17] GC Layek. *An introduction to dynamical systems and chaos*. Springer, 2015. [10](#), [15](#)
- [18] Jiang Hao Wu and Mao Sun. Floquet stability analysis of the longitudinal dynamics of two hovering model insects. *Journal of the Royal Society Interface*, 9(74):2033–2046, 2012. [13](#)
- [19] Ch Skokos. The lyapunov characteristic exponents and their computation. In *Dynamics of Small Solar System Bodies and Exoplanets*, pages 63–135. Springer, 2010. [14](#), [15](#), [16](#), [20](#), [21](#)
- [20] Marco Sandri. Numerical calculation of lyapunov exponents. *Mathematica Journal*, 6(3):78–84, 1996. [14](#), [15](#), [17](#), [20](#), [21](#), [27](#), [31](#)
- [21] Alan Hastings, Carole L Hom, Stephen Ellner, Peter Turchin, and H Charles J Godfray. Chaos in ecology: is mother nature a strange attractor? *Annual review of ecology and systematics*, pages 1–33, 1993. [14](#), [18](#)
- [22] Alan Wolf, Jack B Swift, Harry L Swinney, and John A Vastano. Determining lyapunov exponents from a time series. *Physica D: nonlinear phenomena*, 16(3):285–317, 1985. [14](#), [15](#)
- [23] Jonathan B Dingwell. Lyapunov exponents. *Wiley encyclopedia of biomedical engineering*, 2006. [14](#), [15](#), [16](#), [18](#)
- [24] Thomas S Parker and Leon Chua. *Practical numerical algorithms for chaotic systems*. Springer-Verlag, 1989. [14](#), [15](#), [18](#), [19](#), [20](#)
- [25] Jingjun Lou and Shijian Zhu. Three conditions lyapunov exponents should satisfy. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 37033, pages 1611–1614, 2003. [15](#)

-
- [26] Christophe Letellier and Otto E Rossler. Hyperchaos. *Scholarpedia*, 2(8):1936, 2007. 15, 31
- [27] Antonio Politi. Lyapunov exponent. *Scholarpedia*, 8(3):2722, 2013. 15
- [28] Giancarlo Benettin, Luigi Galgani, Antonio Giorgilli, and Jean-Marie Strelcyn. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. part 1: Theory. *Meccanica*, 15(1):9–20, 1980. 16
- [29] Giancarlo Benettin, L. Galgani, Antonio Giorgilli, and Marie Strelcyn. Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. part 2: Numerical application. *Meccanica*, 15:21–30, 03 1980. 16
- [30] FL Dubeibe. Cálculo del máximo exponente de lyapunov con mathematica. *Revista Colombiana de Física*, 45(1), 2013. 17, 20, 29, 31, 33
- [31] H Binous and N Zakia. Improved method for lyapunov exponents computation. *Wolfram Library Archive* [<http://library.wolfram.com/infocenter/MathSource/7109/>], 2008. 17, 20, 27, 31, 33
- [32] Wen-wen Tung, Jianbo Gao, Jing Hu, and Lei Yang. Detecting chaos in heavy-noise environments. *Physical Review E*, 83(4):046210, 2011. 18
- [33] Stefanos Boccaletti, Celso Grebogi, Y-C Lai, Hector Mancini, and Diego Maza. The control of chaos: theory and applications. *Physics reports*, 329(3):103–197, 2000. 18
- [34] Chutipon Pukdeboon. A review of fundamentals of lyapunov theory. *The Journal of Applied Science*, 10(2):55–61, 2011. 18
- [35] Mathematica and Wolfram Language: Fast introduction for Math students. <https://www.wolfram.com/language/fast-introduction-for-math-students/en/>. 22
- [36] Introduction to Advanced Numerical Differential Equation Solving in the Wolfram Language—Wolfram Language Documentation. <https://reference.wolfram.com/language/tutorial/NDSolveIntroductoryTutorial.html>. 23, 25, 26
- [37] Wolfram Research. Orthogonalize. <https://reference.wolfram.com/language/ref/Orthogonalize.html>, 2007. 23
- [38] Wolfram Research. Method. <https://reference.wolfram.com/language/ref/Method.html>, 2007. 29
- [39] Sadiq A Mehdi and Rabiha Saleem Kareem. Using fourth-order runge-kutta method to solve lü chaotic system. *American Journal of Engineering Research (AJER)*, 6(1):72–77, 2017. 29
-

BIBLIOGRAFÍA

- [40] Wolfram Research. ExplicitRungeKutta Method for NDSolve—Wolfram Language Documentation. <https://reference.wolfram.com/language/tutorial/NDSolveExplicitRungeKutta.html>. 29
- [41] MA Quiroz-Juárez, O Jiménez-Ramírez, R Vázquez-Medina, V Breña-Medina, JL Aragón, and RA Barrio. Generation of ecg signals from a reaction-diffusion model spatially discretized. *Scientific reports*, 9(1):1–10, 2019. 29, 35
- [42] "ExplicitRungeKutta" Method for NDSolve. <https://reference.wolfram.com/language/tutorial/NDSolveExplicitRungeKutta.html>. 32
- [43] J.C. Sprott. A dynamical system with a strange attractor and invariant tori. *Physics Letters A*, 378(20):1361–1363, 2014. 33, 34, 35
- [44] Wolfram Research. Wolfram Library Archive. <https://library.wolfram.com/infocenter/MathSource>. 41