

# Proyecto Conacyt

Iván Uriel Olvera Pérez

December 15, 2023

## Contents

<b>1 Preliminares</b>	<b>3</b>
1.1 Planteamiento del Problema . . . . .	3
1.2 Pregunta a Responder . . . . .	3
1.3 Hipótesis . . . . .	3
1.4 Objetivo . . . . .	4
1.5 Objetivos Específicos . . . . .	4
1.6 Metodología . . . . .	4
<b>2 Estado del Arte</b>	<b>5</b>
2.1 Teoría de Sistemas Dinámicos . . . . .	5
2.2 Aprendizaje Automático y Reconstrucción de Sistemas . . . . .	5
2.3 Reconstrucción de ECG y Otros Sistemas Biomédicos . . . . .	5
2.4 Desafíos y Perspectivas . . . . .	6
2.5 Conclusión . . . . .	6
<b>3 Marco teórico</b>	<b>7</b>
3.1 Cálculo de los coeficientes de mínimos cuadrados . . . . .	7
3.2 Calculo de los coeficientes de mínimos cuadrados en su forma matricial . . . . .	9
3.3 Equivalencia entre Mínimos Cuadrados y una Red Neuronal de una sola capa con activación lineal . . . . .	12
3.4 Mínimos Cuadrados . . . . .	12
3.5 Red Neuronal de una sola capa con activación lineal . . . . .	12
3.5.1 Interpretación . . . . .	12
3.6 Simulación Numérica de la Ecuación de Difusión Unidimensional	13
3.7 Parámetros de la simulación . . . . .	13
3.7.1 Método de Diferencias Finitas . . . . .	14
3.7.2 Simulación del modelo de difusión unidimensional . . . . .	14
3.8 Modelo de difusión Bidimensional . . . . .	15
3.8.1 Simulacion del modelo . . . . .	15
3.9 El Modelo Diferencial de Laplace . . . . .	17
3.9.1 Simulación del Modelo . . . . .	17

<b>4 Metodología</b>	<b>18</b>
4.1 Teorema de Takens . . . . .	18
4.2 Elección del retraso de tiempo . . . . .	19
4.3 Elección de la dimensión de inmersión . . . . .	19
4.4 Función de Autocorrelación . . . . .	19
4.5 SINDy (Sparse Identification of Nonlinear Dynamics) . . . . .	20
4.6 Teoría de los Polinomios de Taylor . . . . .	21
4.7 Teoría de las Series de Fourier . . . . .	22
4.8 Teoría de los Polinomios de Padé . . . . .	22
4.9 Simulación numérica de ECG . . . . .	23
4.9.1 El Modelo . . . . .	23
4.9.2 Método Numérico . . . . .	23
4.9.3 Resultados de la Simulación . . . . .	24
4.10 Generación del Vector del Electrocardiograma . . . . .	24
4.10.1 Formulación del Vector del ECG . . . . .	24
4.10.2 Visualización del ECG . . . . .	25
<b>5 Resultados</b>	<b>26</b>
5.1 Reconstrucción del sistema de lorenz a partir de datos de simulación . . . . .	26
5.1.1 Barrido de pasos de integración dejando fijos los demás parámetros . . . . .	26
5.1.2 Barrido de condiciones iniciales . . . . .	32
5.1.3 Simulación Usando h buena . . . . .	32
5.1.4 Usando h mala . . . . .	33
5.2 Reconstrucción del espacio de estado . . . . .	35
5.3 Comparación de la reconstrucción con la trayectoria original . . . . .	36
5.3.1 Reconstrucción de ECG con base de taylor . . . . .	37
5.3.2 Análisis de la Reconstrucción con Delay de 500 . . . . .	38
5.4 Mejora en la Reconstrucción de Sistemas con Métodos de Takens y SINDy Personalizado . . . . .	38
5.4.1 Potencial y Novedad en la Aplicación de Fourier y SINDy para Datos de Simulación . . . . .	39
5.4.2 Discusión . . . . .	40
5.5 Avances en la Reconstrucción con Métodos de Takens y SINDy Usando Polinomios de Padé . . . . .	40
5.5.1 Análisis de la Reconstrucción con Delay de 400 . . . . .	40
5.6 Teoría y Aplicación de Polinomios de Hermite en SINDy . . . . .	41
5.6.1 Fundamentos de los Polinomios de Hermite . . . . .	41
5.6.2 Implementación en SINDy . . . . .	41
5.6.3 Análisis de la Reconstrucción con Delay de 300 . . . . .	42
5.7 Incorporación de Polinomios de Legendre de Tercer Orden en SINDy . . . . .	42
5.7.1 Principios de los Polinomios de Legendre . . . . .	42
5.7.2 Aplicación en SINDy . . . . .	42
5.7.3 Análisis de la Reconstrucción con Delay de 400 . . . . .	43

# 1 Preliminares

## 1.1 Planteamiento del Problema

El corazón es uno de los órganos más vitales y complejos del cuerpo humano. Una interpretación precisa de su actividad eléctrica puede ser esencial para el diagnóstico temprano y la prevención de enfermedades cardíacas. A pesar de los avances en medicina y tecnología, las enfermedades cardíacas siguen siendo una de las principales causas de muerte en el mundo. Por ello, el desarrollo de dispositivos o sistemas que contribuyan a nuevos tratamientos que mejoren la calidad de vida de las personas es fundamental, y para diseñarlos es esencial comprender los mecanismos de funcionamiento del corazón. En este sentido, el modelado matemático y la simulación por computadora, como procesos que ayudan a describir el funcionamiento del corazón, han jugado un papel muy importante en el impulso de los avances en la investigación de la fisiología cardíaca.

El corazón humano puede considerarse un sistema complejo, pero robusto, que tiene muchas maneras de ser modelado [1,2]. En varios de estos modelos debe incluirse el efecto de millones de células, lo que demanda una alta capacidad de cómputo. Dada esta importante limitación, se han explorado otras alternativas de modelado que demandan menos recursos y describen la dinámica cardiaca macroscópicamente. Una de las señales más importantes que relaciona la actividad eléctrica del corazón a nivel macroscópico es la señal de electrocardiograma (ECG). Un solo ciclo de esta señal refleja la contracción y relajación del corazón, que conduce a la acción de bombeo.

Existen varios esfuerzos que integran osciladores acoplados para modelar la señal ECG. Sin embargo, ninguno de los modelos publicados ha derivado descripciones matemáticas directamente de registros clínicos. Esta posibilidad puede ofrecer una herramienta poderosa para capturar variables esenciales de la dinámica eléctrica cardiaca que habiliten el entendimiento de mecanismos claves para su funcionamiento. A este respecto, la implementación de algoritmos de inteligencia artificial para construir modelos matemáticos a partir de señales discretas ha emergido como un método no convencional en teoría de sistemas no lineales y ha demostrado resultados sobresalientes.

## 1.2 Pregunta a Responder

¿Es posible construir un modelo matemático que describa la actividad eléctrica del corazón a nivel macroscópico utilizando algoritmos de aprendizaje automático entrenados con registros electrocardiográficos clínicos?

## 1.3 Hipótesis

Los algoritmos de aprendizaje automático entrenados con registros clínicos de electrocardiogramas permitirán la reconstrucción de un modelo matemático de la actividad eléctrica del corazón a nivel macroscópico en condiciones sanas.

## **1.4 Objetivo**

Desarrollar un modelo dinámico que describa la actividad eléctrica de un corazón macroscópicamente, utilizando métodos de inteligencia artificial, en particular, algoritmos de aprendizaje automático entrenados a partir de electrocardiogramas clínicos.

## **1.5 Objetivos Específicos**

1. Investigar y establecer la relación entre dos algoritmos de aprendizaje automático: método de mínimos cuadrados y método de matrices dispersas para la reconstrucción de ecuaciones diferenciales.
2. Analizar la reconstrucción numérica de ecuaciones diferenciales en modelos de juguete generados por el método de matrices dispersas.
3. Combinar el método de coordenadas retardadas y el método de matrices dispersas para la reconstrucción de modelos matemáticos a partir de estados incompletos.
4. Aplicar el método de matrices dispersas a registros de electrocardiogramas saludables, con el objetivo de construir un sistema de ecuaciones que describa la dinámica observada.

## **1.6 Metodología**

La investigación se desarrollará en varias etapas correspondientes a cada objetivo específico.

1. Se explorará la relación teórica y práctica entre el método de mínimos cuadrados y el método de matrices dispersas para sistemas no lineales (SINDy, por sus siglas en inglés), utilizando simulaciones y análisis numérico.
2. Se empleará un modelo matemático reportado en la literatura para generar registros de electrocardiograma sintéticos y entrenar el algoritmo de aprendizaje automático basado en matrices dispersas. Se analizará la calidad de la reconstrucción de las ecuaciones diferenciales bajo varias condiciones, tales como: a) estados incompletos, b) diferentes tiempos de muestreo, y c) diferentes regímenes de comportamiento.
3. Se combinará el método de coordenadas retardadas y el método de matrices dispersas para la reconstrucción del modelo matemático a partir de estados incompletos, como es el caso de la señal ECG. Adicionalmente, se realizará un análisis de sensibilidad variando el parámetro de retardo en el método de coordenadas retardadas para determinar su influencia en la reconstrucción del modelo matemático vía SINDy.
4. Se llevará a cabo la reconstrucción del modelo matemático de la actividad eléctrica del corazón en condición sana usando el método de matrices

dispersas. El algoritmo de aprendizaje automático será entrenado con registros de electrocardiogramas clínicos y las variables de estado faltantes se construirá con el método de coordenadas retardadas. Los modelos computacionales desarrollados se programarán en el lenguaje de programación Python.

## 2 Estado del Arte

La reconstrucción de sistemas dinámicos a partir de datos ha evolucionado considerablemente en la última década, impulsada por los avances en la teoría de sistemas dinámicos, algoritmos de aprendizaje automático y técnicas de procesamiento de señales. Esta sección proporciona una visión general de los desarrollos recientes en este campo, con una atención particular a la reconstrucción de sistemas dinámicos complejos como los electrocardiogramas.

### 2.1 Teoría de Sistemas Dinámicos

Los avances en la comprensión matemática de los sistemas dinámicos han mejorado nuestra capacidad para modelar y reconstruir fenómenos complejos a partir de datos. Las técnicas modernas de análisis de series temporales, como el embebido de Takens, han permitido a los investigadores reconstruir atractores extraños y comprender la dinámica subyacente de sistemas caóticos[6]. Además, los métodos de reducción de dimensión como el análisis de componentes principales (PCA) y el análisis de componentes independientes (ICA) han demostrado ser valiosos para identificar las dinámicas esenciales en sistemas de alta dimensión[7, 8].

### 2.2 Aprendizaje Automático y Reconstrucción de Sistemas

En el contexto del aprendizaje automático, los métodos de aprendizaje profundo han emergido como una herramienta poderosa para la reconstrucción de sistemas dinámicos. Las redes neuronales profundas, especialmente las redes recurrentes como LSTM y GRU, han sido aplicadas con éxito para modelar la secuencialidad y la memoria de largo plazo en datos temporales[9, 10]. Estos métodos han sido cruciales en la reconstrucción de ECG, pero también han encontrado aplicaciones en la predicción del tiempo, la modelización de la dinámica del tráfico y la predicción de trayectorias de objetos en movimiento[11, 12, 13].

### 2.3 Reconstrucción de ECG y Otros Sistemas Biomédicos

Particularmente en el campo biomédico, la reconstrucción de ECG ha visto un avance notable con la aplicación de redes neuronales convolucionales y redes generativas adversas (GANs) para mejorar la calidad y la utilidad clínica de las señales reconstruidas[14]. Además, la fusión de señales multimodales, donde las señales PPG y ECG se combinan para mejorar la reconstrucción de la señal

cardíaca, ha abierto nuevas vías para diagnósticos más precisos y monitoreo en tiempo real[15].

La reconstrucción de sistemas que representan la dinámica de electrocardiogramas (ECG) se ha convertido en un campo de interés creciente en la comunidad científica debido a su importancia en el análisis de señales biomédicas. Se han explorado varios métodos utilizando técnicas avanzadas de aprendizaje automático y aprendizaje profundo. A continuación, se presenta un resumen de algunos estudios preliminares relevantes:

Un enfoque significativo ha sido el uso de modelos de aprendizaje profundo basados en sujetos para la reconstrucción de ECG, empleando arquitecturas como la memoria a largo plazo bidireccional (Bi-LSTM). Un estudio de este tipo probó su modelo utilizando registros de la base de datos MIMIC III, destacando la variabilidad individual de las formas de onda del ECG[1].

La reconstrucción de la señal ECG real con alta precisión se ha logrado también mediante algoritmos basados en Redes Neuronales Artificiales (ANN). Los resultados han demostrado una alta exactitud, con un Error Cuadrático Medio (RMS) menor a 13 y un Coeficiente de Correlación (CC) mayor al 99.7%[2].

Además, se ha investigado la reconstrucción de ECG utilizando LSTM, mostrando un rendimiento similar a estudios previos que utilizaban métodos de regresión lineal, pero centrándose en la viabilidad de reconstruir señales obtenidas a través de dispositivos de parche[3].

Otras investigaciones han propuesto el uso de radar de Onda Continua Modulada en Frecuencia (FMCW) junto con Redes Neuronales Convolucionales (CNN) para reconstruir señales ECG, con experimentos exitosos realizados en individuos sanos[4].

Finalmente, varios estudios han abordado la reconstrucción de señales ECG a partir de señales PPG, utilizando métodos como la Transformada de Coseno Discreta (DCT) y aprendizaje de diccionario conjunto entre dominios, para alinear las señales PPG con picos R del ECG[5].

## 2.4 Desafíos y Perspectivas

A pesar de estos avances, existen desafíos significativos, como la necesidad de manejar la variabilidad interindividual en datos biomédicos y la dependencia de grandes conjuntos de datos anotados para el entrenamiento de modelos de aprendizaje profundo. Además, la interpretación de los modelos complejos y la garantía de su robustez y generalización siguen siendo áreas activas de investigación[16, 17].

## 2.5 Conclusión

Los avances en la reconstrucción de sistemas dinámicos a partir de datos han llevado a descubrimientos significativos y aplicaciones prácticas en diversos campos. La convergencia de la teoría matemática, las técnicas computacionales y

el conocimiento específico del dominio seguirá siendo un enfoque clave para la investigación futura y el desarrollo tecnológico.

### 3 Marco teórico

#### 3.1 Cálculo de los coeficientes de mínimos cuadrados

Primero, se define la ecuación del modelo lineal:

$$y = mx + b \quad (1)$$

Donde  $y$  es la variable dependiente,  $x$  es la variable independiente,  $m$  es la pendiente de la línea y  $b$  es la intersección en el eje  $y$ .

Luego, se busca minimizar la suma de los cuadrados de las diferencias entre los valores observados y los valores predichos por el modelo lineal. Esto se puede expresar matemáticamente como:

$$\min \sum_{i=1}^n (y_i - mx_i - b)^2 \quad (2)$$

Donde  $n$  es el número de observaciones,  $y_i$  es el valor observado de la variable dependiente para la  $i$ -ésima observación y  $x_i$  es el valor observado de la variable independiente para la  $i$ -ésima observación.

Para encontrar los valores de  $m$  y  $b$  que minimizan esta expresión, se derivan parcialmente con respecto a  $m$  y  $b$  e igualan a cero:

$$\frac{\partial}{\partial m} \sum_{i=1}^n (y_i - mx_i - b)^2 = 0 \quad (3)$$

$$\frac{\partial}{\partial b} \sum_{i=1}^n (y_i - mx_i - b)^2 = 0 \quad (4)$$

Derivando la primera ecuación parcialmente con respecto a  $m$ , se obtiene:

$$\frac{\partial}{\partial m} \sum_{i=1}^n (y_i - mx_i - b)^2 = \sum_{i=1}^n 2(y_i - mx_i - b)(-x_i) \quad (5)$$

$$= -2 \sum_{i=1}^n x_i(y_i - mx_i - b) \quad (6)$$

Igualando esta expresión a cero, se obtiene:

$$\sum_{i=1}^n x_i(y_i - mx_i - b) = 0 \quad (7)$$

Derivando la segunda ecuación parcialmente con respecto a  $b$ , se obtiene:

$$\frac{\partial}{\partial b} \sum_{i=1}^n (y_i - mx_i - b)^2 = \sum_{i=1}^n 2(y_i - mx_i - b)(-1) \quad (8)$$

$$= -2 \sum_{i=1}^n (y_i - mx_i - b) \quad (9)$$

Igualando esta expresión a cero, se obtiene:

$$\sum_{i=1}^n (y_i - mx_i - b) = 0 \quad (10)$$

Resolviendo estas ecuaciones simultáneamente, se obtienen las siguientes soluciones:

$$m = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad (11)$$

$$b = \frac{\sum_{i=1}^n y_i - m \sum_{i=1}^n x_i}{n} \quad (12)$$

Para resolver estas ecuaciones, se pueden seguir los siguientes pasos:

1. Calcular las sumas de los valores de  $x$ ,  $y$ ,  $xy$  y  $x^2$ :

$$\sum_{i=1}^n x_i = x_1 + x_2 + \dots + x_n \quad (13)$$

$$\sum_{i=1}^n y_i = y_1 + y_2 + \dots + y_n \quad (14)$$

$$\sum_{i=1}^n x_i y_i = x_1 y_1 + x_2 y_2 + \dots + x_n y_n \quad (15)$$

$$\sum_{i=1}^n x_i^2 = x_1^2 + x_2^2 + \dots + x_n^2 \quad (16)$$

2. Sustituir las sumas en las ecuaciones para  $m$  y  $b$ :

$$m = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad (17)$$

$$= \frac{n(x_1 y_1 + x_2 y_2 + \dots + x_n y_n) - (x_1 + x_2 + \dots + x_n)(y_1 + y_2 + \dots + y_n)}{n(x_1^2 + x_2^2 + \dots + x_n^2) - (x_1 + x_2 + \dots + x_n)^2} \quad (18)$$

$$b = \frac{\sum_{i=1}^n y_i - m \sum_{i=1}^n x_i}{n} \quad (19)$$

$$= \frac{y_1 + y_2 + \dots + y_n - m(x_1 + x_2 + \dots + x_n)}{n} \quad (20)$$

3. Simplificar las expresiones:

$$m = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad (21)$$

$$= \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad (22)$$

$$b = \frac{\sum_{i=1}^n y_i - m \sum_{i=1}^n x_i}{n} \quad (23)$$

$$= \frac{\sum_{i=1}^n y_i - \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \sum_{i=1}^n x_i}{n} \quad (24)$$

4. Resolver las expresiones:

$$m = \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \quad (25)$$

$$b = \frac{\sum_{i=1}^n y_i - \frac{n \sum_{i=1}^n x_i y_i - (\sum_{i=1}^n x_i)(\sum_{i=1}^n y_i)}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} \sum_{i=1}^n x_i}{n} \quad (26)$$

### 3.2 Calculo de los coeficientes de mínimos cuadrados en su forma matricial

Primero, se define la ecuación del modelo lineal:

$$y = X\beta \quad (27)$$

Donde  $y$  es el vector de variables dependientes,  $X$  es la matriz de variables independientes, y  $\beta$  es el vector de coeficientes que se desea estimar.

Luego, se busca minimizar la suma de los cuadrados de las diferencias entre los valores observados y los valores predichos por el modelo lineal. Esto se puede expresar matemáticamente como:

$$\min ||y - X\beta||^2 \quad (28)$$

Donde  $||\cdot||^2$  representa la norma al cuadrado.

Para encontrar los valores de  $\beta$  que minimizan esta expresión, se derivan parcialmente con respecto a  $\beta$  e igualan a cero:

$$\frac{\partial}{\partial \beta} ||y - X\beta||^2 = 0 \quad (29)$$

Expandiendo la expresión de la norma al cuadrado, se obtiene:

$$||y - X\beta||^2 = (y - X\beta)^T (y - X\beta) \quad (30)$$

$$= y^T y - 2\beta^T X^T y + \beta^T X^T X \beta \quad (31)$$

Derivando esta expresión parcialmente con respecto a  $\beta$ , se obtiene:

$$-2X^T y + 2X^T X\beta = 0 \quad (32)$$

Igualando esta expresión a cero, se obtiene:

$$X^T X\beta = X^T y \quad (33)$$

Posteriormente se resuelve esta ecuación para Beta siguiendo los siguientes pasos:

1. Multiplicar ambos lados de la ecuación por la matriz inversa de  $X^T X$ :

$$(X^T X)^{-1} X^T X\beta = (X^T X)^{-1} X^T y \quad (34)$$

2. Simplificar la expresión de la izquierda utilizando la propiedad de la matriz inversa:

$$I\beta = (X^T X)^{-1} X^T y \quad (35)$$

Donde  $I$  es la matriz identidad.

3. Se obtiene la solución para  $\beta$ :

$$\beta = (X^T X)^{-1} X^T y \quad (36)$$

Para resolver esta ecuación, se pueden seguir los siguientes pasos:

1. Definir la matriz  $X$  y el vector  $y$ :

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \quad (37)$$

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (38)$$

Donde  $n$  es el número de observaciones y  $p$  es el número de variables independientes.

2. Calcular la matriz  $X^T X$  y el vector  $X^T y$ :

$$X^T X = \begin{bmatrix} n & \sum_{i=1}^n x_{i1} & \sum_{i=1}^n x_{i2} & \dots & \sum_{i=1}^n x_{ip} \\ \sum_{i=1}^n x_{i1} & \sum_{i=1}^n x_{i1}^2 & \sum_{i=1}^n x_{i1}x_{i2} & \dots & \sum_{i=1}^n x_{i1}x_{ip} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^n x_{ip} & \sum_{i=1}^n x_{i1}x_{ip} & \sum_{i=1}^n x_{i2}x_{ip} & \dots & \sum_{i=1}^n x_{ip}^2 \end{bmatrix} \quad (39)$$

$$X^T y = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_{i1}y_i \\ \sum_{i=1}^n x_{i2}y_i \\ \vdots \\ \sum_{i=1}^n x_{ip}y_i \end{bmatrix} \quad (40)$$

3. Resolver la ecuación matricial para  $\beta$ :

3.1. Calcular la matriz inversa de  $X^T X$ :

$$(X^T X)^{-1} \quad (41)$$

3.2. Multiplicar la matriz inversa por  $X^T y$ :

$$(X^T X)^{-1} X^T y \quad (42)$$

Para calcular la matriz inversa de  $X^T X$ , se pueden seguir los siguientes pasos:

3.2.1. Calcular el determinante de  $X^T X$ :

$$|X^T X| \quad (43)$$

3.2.2. Calcular la matriz de cofactores de  $X^T X$ :

$$C = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1p} \\ C_{21} & C_{22} & \dots & C_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ C_{p1} & C_{p2} & \dots & C_{pp} \end{bmatrix} \quad (44)$$

Donde  $C_{ij}$  es el cofactor correspondiente al elemento  $x_{ij}$  de  $X^T X$ .

3.3.3. Calcular la matriz adjunta de  $X^T X$ :

$$\text{adj}(X^T X) = C^T \quad (45)$$

3.3.4. Calcular la matriz inversa de  $X^T X$ :

$$(X^T X)^{-1} = \frac{1}{|X^T X|} \text{adj}(X^T X) \quad (46)$$

### 3.3 Equivalencia entre Mínimos Cuadrados y una Red Neuronal de una sola capa con activación lineal

Para demostrar la equivalencia entre un modelo de mínimos cuadrados y una red neuronal con una sola neurona y función de activación lineal, primero necesitamos entender qué hacen estos modelos.

#### 3.4 Mínimos Cuadrados

El método de los mínimos cuadrados busca minimizar la suma de las diferencias cuadradas entre los valores observados y los valores predichos por el modelo. Para un conjunto de datos  $\{(x_i, y_i)\}$ , el objetivo de los mínimos cuadrados es encontrar los parámetros del modelo que minimizan la función de coste:

$$C(\boldsymbol{\theta}) = \sum_i (y_i - f(x_i; \boldsymbol{\theta}))^2, \quad (47)$$

donde  $f(x_i; \boldsymbol{\theta})$  es el modelo predictivo que depende de los parámetros  $\boldsymbol{\theta}$ .

#### 3.5 Red Neuronal de una sola capa con activación lineal

Por otro lado, una red neuronal con una sola capa y función de activación lineal se define como:

$$\hat{y}_i = f(x_i; \boldsymbol{\theta}) = \boldsymbol{\theta}^T \mathbf{x}_i, \quad (48)$$

donde  $\mathbf{x}_i$  es el vector de entrada,  $\boldsymbol{\theta}$  es el vector de pesos, y  $\hat{y}_i$  es la salida de la red.

La función de coste para el entrenamiento de la red neuronal es, a menudo, la suma de las diferencias cuadradas entre los valores observados y los valores predichos, al igual que en los mínimos cuadrados:

$$C(\boldsymbol{\theta}) = \sum_i (y_i - \hat{y}_i)^2 = \sum_i (y_i - \boldsymbol{\theta}^T \mathbf{x}_i)^2. \quad (49)$$

Por lo tanto, el problema de minimizar la función de coste en la red neuronal es equivalente al problema de los mínimos cuadrados.

##### 3.5.1 Interpretación

Podemos interpretar esta equivalencia de la siguiente manera: el método de mínimos cuadrados y la red neuronal de una sola capa con activación lineal son ambos métodos lineales que intentan encontrar la mejor línea (o hiperplano en dimensiones superiores) que se ajuste a los datos. La diferencia principal radica en la forma en que se optimizan los parámetros: en los mínimos cuadrados se resuelve una ecuación normal, mientras que en la red neuronal se utiliza la optimización por descenso de gradiente. Sin embargo, en términos de la función de coste y la forma del modelo, son equivalentes.

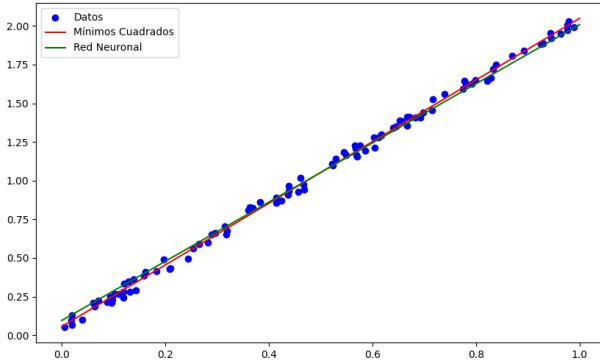


Figure 1: Representación gráfica de la equivalencia entre el método de mínimos cuadrados y una red neuronal de una sola capa con activación lineal. Los datos se representan en azul, la línea de ajuste de mínimos cuadrados en rojo y la línea de ajuste de la red neuronal en verde. Como puede observarse, ambas técnicas producen resultados similares en este caso.

Como se muestra en la Figura 1, ambas técnicas buscan una función lineal que minimice la distancia cuadrada total entre las predicciones del modelo y los datos observados. Ambos métodos son capaces de aprender un modelo predictivo a partir de los datos y generar predicciones precisas, lo que refuerza la idea de su equivalencia.

### 3.6 Simulación Numérica de la Ecuación de Difusión Unidimensional

La ecuación de difusión unidimensional, dada por  $\frac{\partial u}{\partial t} = D \frac{\partial^2 u}{\partial x^2}$ , fue resuelta numéricamente usando el método de diferencias finitas.

### 3.7 Parámetros de la simulación

Los parámetros utilizados en la simulación fueron los siguientes:

- $D$ : El coeficiente de difusión. Este determina la rapidez con la que se difunde el material a través del medio.
- $L$ : La longitud del dominio espacial. La ecuación se resuelve en el intervalo  $[0, L]$ .
- $T$ : El tiempo final. Se resuelve la ecuación para el intervalo de tiempo  $[0, T]$ .

- $N_x$  y  $N_t$ : El número de puntos de malla en el espacio y el tiempo, respectivamente. El dominio espacial y temporal se discretiza en  $N_x$  y  $N_t$  puntos, respectivamente.
- $dx$  y  $dt$ : Los tamaños de los pasos espaciales y temporales. Estos se calculan a partir de  $L$ ,  $T$ ,  $N_x$  y  $N_t$ .
- $r$ : El número de Courant. Se utiliza para garantizar la estabilidad del esquema numérico. Para la estabilidad del método de diferencias finitas aplicado a la ecuación de difusión, se necesita que  $r = D \frac{dt}{dx^2} \leq 0.5$ .

### 3.7.1 Método de Diferencias Finitas

El método de diferencias finitas es un enfoque numérico para resolver ecuaciones diferenciales. Consiste en reemplazar las derivadas en la ecuación diferencial con diferencias finitas que aproximan estas derivadas.

Para la ecuación de difusión, la derivada temporal se aproxima por una diferencia hacia adelante en el tiempo:

$$\frac{\partial u}{\partial t} \approx \frac{u_{i,n+1} - u_{i,n}}{dt}, \quad (50)$$

y la segunda derivada espacial se aproxima por una diferencia central en el espacio:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1,n} - 2u_{i,n} + u_{i-1,n}}{dx^2}. \quad (51)$$

Al reemplazar estas aproximaciones en la ecuación de difusión, obtenemos una ecuación para el valor futuro  $u_{i,n+1}$  en términos de los valores actuales y vecinos  $u_{i-1,n}$ ,  $u_{i,n}$  y  $u_{i+1,n}$ :

$$u_{i,n+1} = r \cdot u_{i-1,n} + (1 - 2r) \cdot u_{i,n} + r \cdot u_{i+1,n}. \quad (52)$$

La solución numérica se obtiene resolviendo estas ecuaciones para cada punto de la malla en cada paso de tiempo.

### 3.7.2 Simulación del modelo de difusión unidimensional

La simulación numérica del modelo de difusión unidimensional se realizó utilizando el método de diferencias finitas. La ecuación de difusión  $\partial u / \partial t = D \partial^2 u / \partial x^2$  se resolvió en el intervalo  $[0, 10]$ , con un coeficiente de difusión  $D = 1$ , condiciones de frontera  $u(0, t) = u(1, t) = 0$  para todo  $t$ , y la condición inicial  $u(x, 0)$  como una función gaussiana centrada en la mitad del intervalo.

El parámetro de estabilidad utilizado para la simulación es  $r = D \Delta t / \Delta x^2$ , donde  $\Delta t$  es el tamaño de paso temporal y  $\Delta x$  es el tamaño de paso espacial.

La figura muestra cómo la condición inicial se difunde en el espacio a medida que pasa el tiempo. En el inicio, la solución es una función gaussiana, concentrada en la mitad del intervalo. Con el tiempo, la solución se difunde

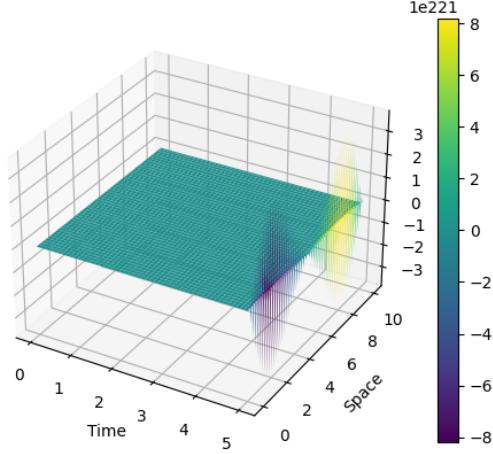


Figure 2: Evolución de la solución del modelo de difusión en el tiempo.

a lo largo de todo el intervalo, reduciendo la concentración en el centro y aumentándola en las regiones anteriormente vacías. Esto es característico de los procesos de difusión, donde las partículas tienden a moverse desde regiones de alta concentración a regiones de baja concentración.

### 3.8 Modelo de difusión Bidimensional

La ecuación de difusión en dos dimensiones se puede escribir de la siguiente manera:

$$\frac{\partial u}{\partial t} = D \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \quad (53)$$

donde  $D$  es el coeficiente de difusión,  $u$  es la concentración de la sustancia que se está difundiendo,  $x$  y  $y$  son las coordenadas espaciales, y  $t$  es el tiempo.

#### 3.8.1 Simulacion del modelo

En esta sección, presentamos una simulación numérica de un modelo de difusión bidimensional. La ecuación de difusión es una ecuación parabólica que describe cómo una cantidad (como el calor, la concentración de una sustancia, etc.) se propaga en el espacio a lo largo del tiempo. En dos dimensiones, la ecuación de difusión se puede escribir como:

$$\frac{\partial u}{\partial t} = \alpha \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

donde  $u(x, y, t)$  es la cantidad que se difunde,  $t$  es el tiempo,  $x$  y  $y$  son las coordenadas espaciales, y  $\alpha$  es el coeficiente de difusión.

Para resolver esta ecuación numéricamente, utilizamos el método de las diferencias finitas. Este método aproxima las derivadas en la ecuación de difusión con diferencias finitas en una malla espaciotemporal. La malla espacial se crea con  $N = 100$  puntos en cada dirección del espacio, y el tamaño de paso temporal se elige de manera que se cumpla la condición de estabilidad para el método de las diferencias finitas.

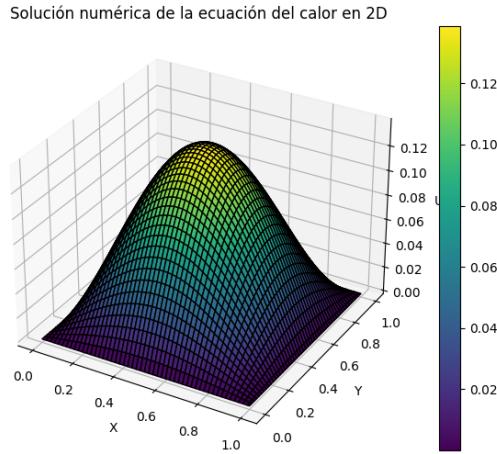


Figure 3: Solución numérica de la ecuación de difusión en 2D.

La solución se inicializa con una función seno 2D, y luego se actualiza en cada paso de tiempo de acuerdo con la ecuación de difusión. El resultado es una simulación de cómo la cantidad  $u$  se propaga en el espacio a lo largo del tiempo.

La Figura 3 muestra la solución numérica de la ecuación de difusión en 2D al final de la simulación.

## 3.9 El Modelo Diferencial de Laplace

La ecuación de Laplace es una ecuación diferencial parcial de segundo orden que se presenta en muchas áreas de la física y la ingeniería, como en problemas de electrostática, mecánica de fluidos y transferencia de calor.

En dos dimensiones, la ecuación de Laplace se puede escribir como:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \quad (54)$$

donde  $u(x, y)$  es la función que queremos encontrar, y  $x$  e  $y$  son las variables espaciales. En este caso, las dos derivadas parciales de segundo orden representan la curvatura de la función  $u$  en las direcciones  $x$  e  $y$ , respectivamente. La ecuación establece que la suma de estas curvaturas es cero, es decir, la función  $u$  es armónica.

Las soluciones de la ecuación de Laplace, también conocidas como funciones armónicas, tienen la propiedad de que en cada punto, su valor es igual al promedio de sus valores en los puntos vecinos. Esto es lo que hace que la ecuación de Laplace sea útil para describir fenómenos de equilibrio, donde las cantidades tienden a distribuirse de manera uniforme.

La ecuación de Laplace se resuelve a menudo con condiciones de frontera dadas, que especifican el valor de la función  $u$  en los límites del dominio. Estas pueden ser condiciones de Dirichlet, donde se especifica el valor de  $u$  en la frontera, o condiciones de Neumann, donde se especifica la derivada normal de  $u$  en la frontera. La solución de la ecuación de Laplace con las condiciones de frontera dadas proporciona una descripción del comportamiento del sistema en todo el dominio.

### 3.9.1 Simulación del Modelo

Para simular el modelo, resolvemos numéricamente la ecuación de Laplace en el dominio bidimensional  $[0, 1] \times [0, 1]$ . Aplicamos las condiciones de frontera  $u(x, y) = \sin(\pi x) \sinh(\pi y)$ .

Utilizamos el método de diferencias finitas para aproximar las derivadas en la ecuación de Laplace. El dominio se discretiza en una malla de puntos y, en cada iteración del método, se actualiza el valor de la función  $u(x, y)$  en cada punto interno de la malla como el promedio de los valores de  $u$  en los puntos vecinos. Este procedimiento se repite hasta que la solución converge, es decir, hasta que la diferencia máxima entre la solución vieja y la nueva es menor que una tolerancia predefinida de  $10^{-5}$ .

La Figura 4 muestra la solución numérica de la ecuación de Laplace. En el eje  $X$  y el eje  $Y$  representamos el espacio bidimensional en el que se resuelve la ecuación, mientras que en el eje  $Z$  mostramos el valor de la solución  $u(x, y)$  en cada punto  $(x, y)$  de la malla. La gráfica visualiza cómo la función  $u(x, y)$  se propaga desde las condiciones de frontera hacia el interior del dominio. La convergencia de la solución numérica confirma que la ecuación de Laplace describe

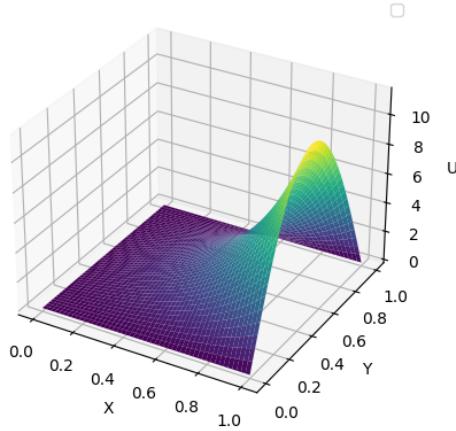


Figure 4: Gráfico de la solución numérica de la ecuación de Laplace.

correctamente el comportamiento de la función bajo las condiciones de frontera dadas.

## 4 Metodología

### 4.1 Teorema de Takens

El teorema de Takens, propuesto por Floris Takens en 1981, es un resultado fundamental en la teoría de sistemas dinámicos que proporciona una justificación teórica para la técnica de reconstrucción del espacio de fase a partir de series temporales.

El teorema establece que bajo ciertas condiciones generales, es posible reconstruir la dinámica de un sistema a partir de una serie de observaciones unidimensionales de dicho sistema. Específicamente, si tenemos un sistema dinámico de dimensión  $m$  dado por las ecuaciones:

$$\frac{dx}{dt} = f(x) \quad (55)$$

donde  $x \in R^m$  y  $f : R^m \rightarrow R^m$  es una función suficientemente suave, y solo podemos observar una única función escalar  $h : R^m \rightarrow R$  del estado del sistema, es decir, medimos  $y(t) = h(x(t))$  para todo  $t$ .

Entonces, según el teorema de Takens, el sistema dinámico original y el sistema dinámico "reconstruido" dado por las ecuaciones

$$\begin{aligned}
\frac{dy_1}{dt} &= y_2, \\
\frac{dy_2}{dt} &= y_3, \\
&\vdots \\
\frac{dy_{2m-1}}{dt} &= y_{2m}, \\
\frac{dy_{2m}}{dt} &= g(y_1, y_2, \dots, y_{2m}),
\end{aligned} \tag{56}$$

donde  $y_i(t) = y(t - (i - 1)\tau)$  para  $i = 1, 2, \dots, 2m$  y  $\tau > 0$  es un retraso de tiempo, son difeomorfos. Esto significa que hay un mapeo biyectivo y diferenciable entre los estados del sistema original y los estados del sistema reconstruido.

La importancia del teorema de Takens reside en que proporciona una base teórica sólida para la reconstrucción del espacio de fase a partir de series de tiempo unidimensionales, una técnica ampliamente utilizada en el análisis de sistemas dinámicos y series de tiempo.

## 4.2 Elección del retraso de tiempo

La elección del retraso de tiempo  $\tau$  es crucial para la calidad de la reconstrucción. Un método común para elegir  $\tau$  es el método del primer mínimo de la función de información mutua promedio (AMI), que se define como:

$$I(\tau) = \sum_{i,j} p(i,j) \log \left( \frac{p(i,j)}{p(i)p(j)} \right) \tag{57}$$

donde  $p(i)$  y  $p(j)$  son las probabilidades de que la serie temporal tome los valores  $i$  y  $j$  respectivamente, y  $p(i,j)$  es la probabilidad conjunta. El valor de  $\tau$  que minimiza  $I(\tau)$  se elige como el retraso de tiempo.

## 4.3 Elección de la dimensión de inmersión

La elección de la dimensión de inmersión  $m$  también es crucial para la calidad de la reconstrucción. Un método común para elegir  $m$  es el método de la falsa vecindad más cercana (FNN), que se basa en la idea de que si  $m$  es demasiado pequeño, los puntos que son vecinos cercanos en el espacio reconstruido no serán vecinos cercanos en el espacio de estado original.

## 4.4 Función de Autocorrelación

La función de autocorrelación (ACF, por sus siglas en inglés) es una herramienta fundamental en el análisis de series de tiempo, utilizada para medir la relación lineal entre valores de la misma serie en diferentes puntos en el tiempo. Matemáticamente, la ACF para una serie de tiempo  $\{X_t\}$  se define como:

$$\rho(\tau) = \frac{C(\tau)}{C(0)} \quad (58)$$

donde  $\rho(\tau)$  es el coeficiente de autocorrelación para un desfase (lag)  $\tau$ , y  $C(\tau)$  es la función de autocovarianza dada por:

$$C(\tau) = \frac{1}{N} \sum_{t=1}^{N-\tau} (X_t - \bar{X})(X_{t+\tau} - \bar{X}) \quad (59)$$

Aquí,  $N$  es el número total de observaciones en la serie,  $X_t$  es el valor de la serie en el tiempo  $t$ ,  $\bar{X}$  es el promedio de la serie, y  $\tau$  es el desfase.

La función de autocorrelación varía en el rango  $[-1, 1]$ , donde un valor cercano a 1 indica una fuerte correlación positiva, un valor cercano a -1 indica una fuerte correlación negativa, y un valor cercano a 0 indica que no hay correlación lineal a ese desfase particular.

La ACF es particularmente útil para identificar patrones recurrentes en los datos, como estacionalidad o ciclos, y también es un componente clave en la identificación de modelos adecuados para el análisis de series de tiempo, como los modelos ARIMA (Modelos Autoregresivos Integrados de Media Móvil).

## 4.5 SYNDy (Sparse Identification of Nonlinear Dynamics)

El algoritmo SINDy (Sparse Identification of Nonlinear Dynamics) es un enfoque para descubrir las leyes dinámicas que gobiernan un sistema a partir de los datos de medición. El proceso general de SINDy se puede dividir en los siguientes pasos:

1. **Recopilación de datos:** Los datos pueden ser mediciones temporales de las variables de estado del sistema. Por ejemplo, para un sistema de  $n$  variables, los datos estarían en forma de una matriz  $X \in R^{m \times n}$ , donde  $m$  es el número de puntos de tiempo y  $n$  es el número de variables de estado.

2. **Cálculo de las derivadas:** SINDy requiere tanto los datos  $X$  como sus derivadas  $\dot{X}$ . Las derivadas pueden ser calculadas usando métodos numéricos como la diferenciación finita.

3. **Construcción de la biblioteca de funciones:** SINDy intenta representar la dinámica del sistema como una combinación lineal de funciones no lineales de las variables de estado. Por lo tanto, necesitamos construir una biblioteca de estas funciones. Por ejemplo, para un sistema de una variable, la biblioteca de funciones podría incluir funciones como  $[1, x, x^2, x^3, \sin(x), \cos(x), e^x]$ .

4. **Ajuste de los datos al modelo:** Aquí es donde SINDy realmente brilla. El objetivo es encontrar la combinación más escasa de las funciones en la biblioteca que pueda describir la dinámica de los datos. Este problema se puede formular como un problema de optimización de mínimos cuadrados con una penalización de la norma  $L_1$  para fomentar la escasez. En otras palabras, queremos resolver el problema:

$$\underset{\xi}{\text{minimize}} \|\dot{X} - \Theta(X)\xi\|_2^2 + \lambda \|\xi\|_1 \quad (60)$$

donde  $\Theta(X)$  es la biblioteca de funciones evaluada en los datos,  $\xi$  es el vector de coeficientes que estamos tratando de encontrar, y  $\lambda$  es un parámetro que controla el equilibrio entre la precisión de la aproximación y la escasez de la solución.

**5. Extracción del modelo:** Una vez resuelto el problema de optimización, los coeficientes  $\xi$  que obtenemos nos dan el modelo del sistema. La dinámica del sistema se puede escribir como:

$$\dot{X} = \Theta(X)\xi \quad (61)$$

Este modelo es una representación escasa de la dinámica del sistema en términos de la biblioteca de funciones.

## 4.6 Teoría de los Polinomios de Taylor

Los polinomios de Taylor son una herramienta poderosa en análisis matemático para aproximar funciones a través de polinomios. La idea central detrás de los polinomios de Taylor es que cualquier función suficientemente suave puede ser aproximada por un polinomio en las cercanías de un punto dado.

Sea  $f(x)$  una función y  $a$  un punto en su dominio. El polinomio de Taylor de grado  $n$  para  $f$  alrededor del punto  $a$  se define como:

$$T_n(x) = \sum_{k=0}^n \frac{f^{(k)}(a)}{k!} (x-a)^k \quad (62)$$

donde  $f^{(k)}(a)$  denota la  $k$ -ésima derivada de  $f$  evaluada en  $a$ , y  $k!$  es el factorial de  $k$ .

El polinomio de Taylor proporciona una aproximación de  $f(x)$  cerca de  $a$ , y su precisión aumenta con el grado  $n$ . Para  $n$  infinito y bajo ciertas condiciones, la serie de Taylor, que es la suma infinita de los términos del polinomio de Taylor, converge a  $f(x)$  en algún intervalo alrededor de  $a$ .

La fórmula del resto de Taylor, que expresa el error de la aproximación, es también una parte importante de la teoría:

$$R_n(x) = f(x) - T_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!} (x-a)^{(n+1)} \quad (63)$$

donde  $c$  es un punto entre  $a$  y  $x$ .

Esta aproximación se utiliza en varios campos, como en cálculos numéricos, para resolver ecuaciones diferenciales, en optimización y en el análisis de algoritmos.

## 4.7 Teoría de las Series de Fourier

Las series de Fourier son una herramienta fundamental en el análisis matemático y la teoría de señales, utilizadas para representar una función periódica como la suma infinita de senos y cosenos. Estas series llevan el nombre de Jean-Baptiste Joseph Fourier, quien introdujo la idea de que cualquier función periódica puede ser representada como una serie de funciones trigonométricas.

Sea  $f(x)$  una función periódica con periodo  $T$  y definida en el intervalo  $[-\frac{T}{2}, \frac{T}{2}]$ . La serie de Fourier de  $f$  se define como:

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left( a_n \cos \left( \frac{2\pi n x}{T} \right) + b_n \sin \left( \frac{2\pi n x}{T} \right) \right) \quad (64)$$

donde los coeficientes  $a_n$  y  $b_n$  se calculan mediante las fórmulas:

$$a_0 = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) dx \quad (65)$$

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \cos \left( \frac{2\pi n x}{T} \right) dx, \quad n \geq 1 \quad (66)$$

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \sin \left( \frac{2\pi n x}{T} \right) dx, \quad n \geq 1 \quad (67)$$

La convergencia de la serie de Fourier depende de las propiedades de la función  $f$ . En muchos casos prácticos, la serie converge a  $f(x)$  en cada punto donde  $f$  es continua, y al promedio de los límites superior e inferior en los puntos donde  $f$  es discontinua.

Las series de Fourier tienen aplicaciones extensas en ingeniería, física, y matemáticas, especialmente en el análisis de señales temporales y espaciales, en el procesamiento de señales, y en la solución de ecuaciones diferenciales parciales.

## 4.8 Teoría de los Polinomios de Padé

Los polinomios de Padé, nombrados en honor al matemático francés Henri Padé, son aproximaciones racionales de funciones que se utilizan en diversos campos de la matemática y la física. A diferencia de las series de Taylor, que son aproximaciones polinómicas, los polinomios de Padé utilizan cocientes de polinomios para proporcionar una mejor aproximación de una función en un rango más amplio.

Dada una función  $f(x)$  y su serie de Taylor:

$$f(x) = \sum_{n=0}^{\infty} c_n (x - a)^n, \quad (68)$$

donde  $c_n$  son los coeficientes de la serie de Taylor, un polinomio de Padé de orden  $[m/n]$  es un cociente de dos polinomios:

$$P_{[m/n]}(x) = \frac{\sum_{i=0}^m a_i(x-a)^i}{1 + \sum_{j=1}^n b_j(x-a)^j}, \quad (69)$$

donde  $m$  y  $n$  son enteros no negativos que determinan los grados de los polinomios del numerador y el denominador, respectivamente, y  $a_i$ ,  $b_j$  son los coeficientes a determinar.

Estos coeficientes se encuentran de tal manera que la serie de Taylor de  $P_{[m/n]}(x)$  coincida con la serie de Taylor de  $f(x)$  hasta el término de mayor grado posible. Es decir, se busca que el error entre la función  $f(x)$  y la aproximación  $P_{[m/n]}(x)$  sea de orden superior al mayor grado presente en el numerador y el denominador.

La aproximación de Padé puede ser especialmente útil para funciones que tienen singularidades o son difíciles de aproximar bien con series de Taylor en ciertos intervalos.

## 4.9 Simulación numérica de ECG

Este estudio se enfoca en la simulación numérica de señales de electrocardiograma (ECG) utilizando un modelo de reacción-difusión discretizado. El objetivo es simular los principales marcapasos del corazón y reproducir señales de ECG tanto de corazones sanos como de pacientes con trastornos del ritmo conocidos.

### 4.9.1 El Modelo

El modelo utilizado [18] es una versión discretizada de un sistema de reacción-difusión, descrito en el documento, que se traduce en un sistema de ecuaciones diferenciales ordinarias (EDO). Las variables  $x_1$ ,  $x_2$ ,  $x_3$  y  $x_4$  representan actividades eléctricas en diferentes partes del corazón. Las ecuaciones del sistema son:

$$\begin{aligned}\dot{x}_1 &= x_1 - x_2 - Cx_1x_2 - x_1x_2^2, \\ \dot{x}_2 &= Hx_1 - 3x_2 + Cx_1x_2 + x_1x_2^2 + B(x_4 - x_2), \\ \dot{x}_3 &= x_3 - x_4 - Cx_3x_4 - x_3x_4^2, \\ \dot{x}_4 &= Hx_3 - 3x_4 + Cx_3x_4 + x_3x_4^2 + 2B(x_2 - x_4),\end{aligned}$$

donde  $C$ ,  $B$  y  $H$  son parámetros que controlan la dinámica del sistema.

### 4.9.2 Método Numérico

Para la simulación numérica, se emplea el método de Runge-Kutta de cuarto orden (RK4). Este es un método iterativo para la aproximación de soluciones de EDOs, ampliamente utilizado debido a su balance entre precisión y eficiencia computacional.

#### 4.9.3 Resultados de la Simulación

Se inicializa el sistema con condiciones iniciales y se simula su evolución temporal. Los resultados se visualizan en una serie de gráficos que muestran la variación de las variables  $x_1$ ,  $x_2$ ,  $x_3$  y  $x_4$  con respecto al tiempo. Estos gráficos proporcionan una representación visual del comportamiento dinámico del sistema y son cruciales para comprender la actividad eléctrica simulada del corazón.

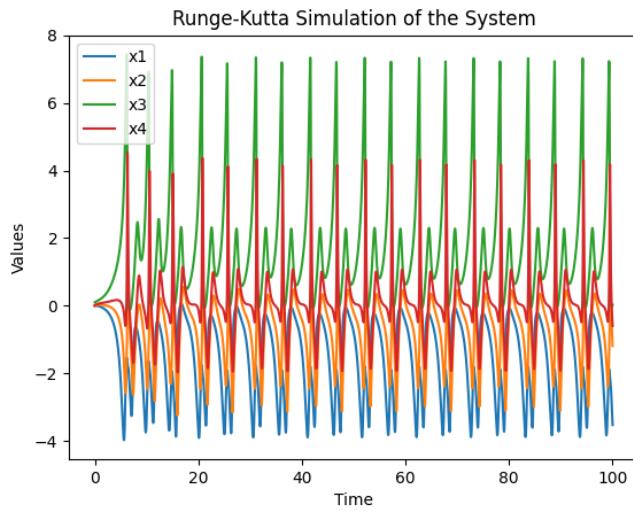


Figure 5: Resultados de la simulación numérica del ECG utilizando el método RK4.

### 4.10 Generación del Vector del Electrocardiograma

En esta sección, nos enfocamos en la formulación matemática utilizada para generar el vector del electrocardiograma (ECG) a partir de las variables dinámicas del sistema y su representación gráfica.

#### 4.10.1 Formulación del Vector del ECG

El vector del ECG se calcula como una combinación lineal de las variables del sistema, representando diferentes aspectos de la actividad eléctrica del corazón. La fórmula utilizada es:

$$ECG = a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4, \quad (70)$$

donde  $x_1$ ,  $x_2$ ,  $x_3$  y  $x_4$  son las variables del sistema obtenidas de la simulación numérica, y  $a_1$ ,  $a_2$ ,  $a_3$ ,  $a_4$  son coeficientes que ponderan la contribución de cada variable al ECG. En este estudio, los valores de los coeficientes son:

- $a_1 = -0.024$
- $a_2 \geq 0.0216$
- $a_3 = -0.0012$
- $a_4 = 0.12$

Estos coeficientes han sido determinados para reflejar de manera efectiva la señal de ECG.

#### 4.10.2 Visualización del ECG

La señal de ECG resultante se visualiza en un gráfico, mostrando cómo varía con el tiempo. Esta visualización es crucial para interpretar la dinámica de la actividad eléctrica del corazón simulada. El gráfico se genera de la siguiente manera:

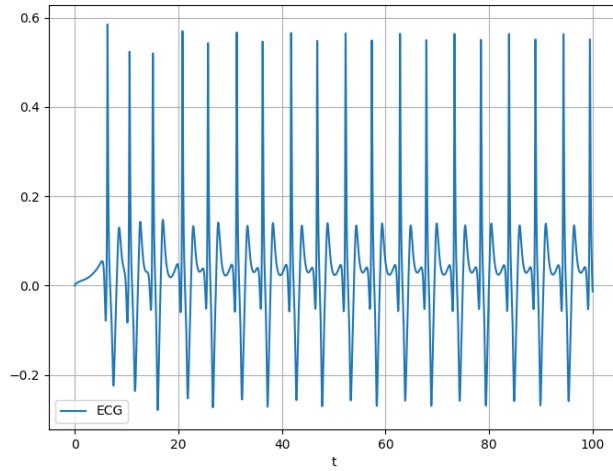


Figure 6: Representación gráfica del ECG calculado, mostrando la variación de la señal en función del tiempo.

El gráfico proporciona una representación visual intuitiva de la señal de ECG, facilitando la comparación con patrones de ECG reales y el análisis de posibles anomalías o características distintivas.

## 5 Resultados

### 5.1 Reconstrucción del sistema de lorenz a partir de datos de simulación

Las ecuaciones de Lorenz son un sistema de ecuaciones diferenciales ordinarias que describen la evolución temporal de tres variables. Este sistema es famoso por su comportamiento caótico y se utiliza a menudo como un ejemplo canónico de sistemas dinámicos complejos.

Supón que tenemos un conjunto de datos de simulación que representa la evolución temporal de las tres variables en el sistema de Lorenz. Estos datos podrían haberse generado mediante la integración numérica de las ecuaciones de Lorenz con un método de integración como el método de Runge-Kutta.

Nuestro objetivo es reconstruir las ecuaciones de Lorenz a partir de estos datos de simulación. Para hacer esto, podemos utilizar una técnica llamada Sparse Identification of Nonlinear Dynamics (SINDy). SINDy es un método de regresión que puede identificar ecuaciones diferenciales subyacentes a partir de datos de tiempo.

En Python, la biblioteca PySINDy proporciona una implementación de SINDy que podemos utilizar para este propósito. Primero, cargariamos los datos de simulación en Python. Luego, podríamos utilizar la función `SINDy()` de PySINDy para ajustar un modelo SINDy a estos datos. Este modelo identificará las ecuaciones diferenciales que mejor describen la evolución de las variables en el tiempo.

Un aspecto importante a tener en cuenta es que el éxito de esta reconstrucción dependerá de la calidad de los datos de simulación. Si los datos son ruidosos o si la resolución temporal no es suficiente, SINDy puede tener dificultades para identificar las ecuaciones correctas.

Es importante destacar que este enfoque no requiere ningún conocimiento previo sobre la forma de las ecuaciones de Lorenz. SINDy puede identificar estas ecuaciones puramente a partir de los datos de simulación.

#### 5.1.1 Barrido de pasos de integración dejando fijos los demás parámetros

Aquí está un ejemplo de cómo se vería este proceso haciendo un barrido del paso de integración para ver su afectación de los coeficientes reconstruidos con PySINDy en código de Python:

```
import numpy as np
import pandas as pd
from scipy.integrate import odeint
import pysindy as ps

# Definir el sistema de Lorenz
def lorenz_system(z, t, sigma, rho, beta):
    x, y, z = z
```

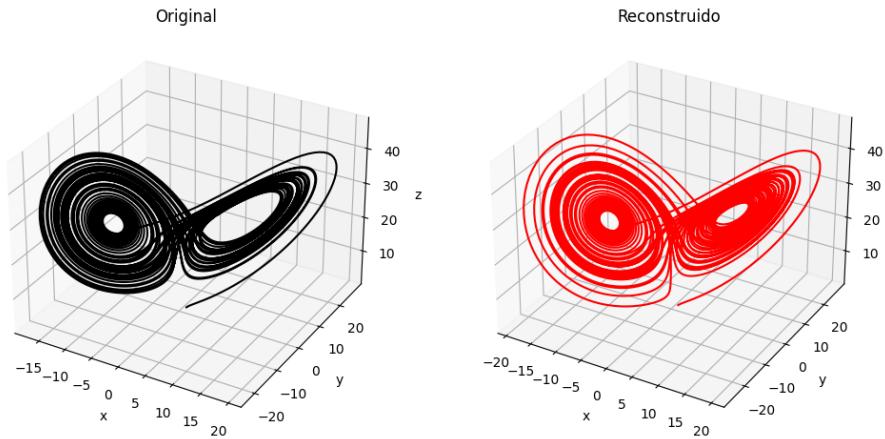


Figure 7: Comparacion de retrato fase reconstruido contra el original.

```

dx_dt = sigma * (y - x)
dy_dt = x * (rho - z) - y
dz_dt = x * y - beta * z
return [dx_dt, dy_dt, dz_dt]

# Parámetros del sistema de Lorenz
sigma = 10
rho = 28
beta = 8/3

# Establecer el parámetro de sparsity
sparsity = 0.01

# Crear un optimizador con el parámetro de sparsity deseado
optimizer = ps.STLSQ(threshold=sparsity)

# Crear un objeto SINDy
model = ps.SINDy(optimizer=optimizer)

# Inicializar un DataFrame para almacenar los coeficientes
coefficients_df = None

# Tiempo de simulación fijo
simulation_time = 1000

# Realizar un barrido de pasos de integración
for h in np.arange(0.001, 0.5, 0.001):

```

```

# Calcular el número de puntos en el vector de tiempo
num_points = int(simulation_time / h)

# Generar datos de simulación con el paso de integración actual
t = np.linspace(0, simulation_time, num_points)
z0 = [1, 1, 1]
simulated_data = odeint(lorenz_system, z0, t, args=(sigma, rho, beta))

# Entrenar el modelo SINDy con los datos de simulación
model.fit(simulated_data, t=t)

# Obtener los coeficientes del modelo SINDy entrenado
coefficients = model.coefficients().flatten()

# Crear un DataFrame temporal para almacenar los coeficientes actuales
temp_df = pd.DataFrame(coefficients.reshape(1, -1))

# Agregar los coeficientes al DataFrame
if coefficients_df is None:
    coefficients_df = temp_df
else:
    coefficients_df = pd.concat([coefficients_df, temp_df], ignore_index=True)

column_names = []
for eq_name in ['dx', 'dy', 'dz']:
    for feature_name in ['1', 'x', 'y', 'z', 'xx', 'xy', 'xz', 'yy', 'yz', 'zz']:
        column_names.append(f'{eq_name}_{feature_name}')

# Asignar los nombres de las columnas al DataFrame
coefficients_df.columns = column_names

# Crear nombres de filas para el DataFrame
row_names = []
for h_index, h in enumerate(np.arange(0.001, 0.5, 0.001)):
    row_names.append(f'h_{h_index + 1}')

# Asignar los nombres de las filas al DataFrame
coefficients_df.index = row_names

# Mostrar el DataFrame
print(coefficients_df)
#Guardar en un csv
coefficients_df.to_csv('coefficients.csv', index=True)

```

Este código Python se utiliza para reconstruir las ecuaciones de Lorenz utilizando el método de identificación dispersa de la dinámica no lineal (SINDy)

implementado en la biblioteca PySINDy. En primer lugar, se importan las bibliotecas necesarias y se define el sistema de Lorenz. A continuación, se ajusta el modelo SINDy a los datos de la simulación y se obtienen los coeficientes del modelo. Finalmente, se guarda un archivo CSV con los coeficientes obtenidos.

El código se puede descomponer en varios pasos:

1. **Importar las bibliotecas necesarias:** El código importa `numpy` y `pandas` para la manipulación de datos, `scipy.integrate.odeint` para resolver numéricamente el sistema de Lorenz y `pysindy` para la reconstrucción del modelo.
2. **Definición del sistema de Lorenz:** Se define una función en Python para el sistema de Lorenz, que toma como entrada el estado actual del sistema y devuelve las derivadas del estado.
3. **Parámetros del sistema de Lorenz:** Se definen los parámetros  $\sigma$ ,  $\rho$  y  $\beta$  para el sistema de Lorenz.
4. **Configuración del modelo SINDy:** Se establece un parámetro de "sparsity" que controla la cantidad de términos en el modelo resultante. Luego, se crea un optimizador SINDy con este parámetro y se instancia un objeto SINDy.
5. **Inicialización del DataFrame de Coeficientes:** Se inicializa un DataFrame de pandas vacío para almacenar los coeficientes de SINDy.
6. **Simulación de los Datos y Ajuste del Modelo:** Para varios pasos de integración, se generan datos de simulación del sistema de Lorenz y se ajusta el modelo SINDy a estos datos. Los coeficientes del modelo se almacenan en el DataFrame.
7. **Guardar los coeficientes:** Se asignan nombres a las columnas y filas del DataFrame, que se imprime y se guarda en un archivo CSV.

Este código puede ser útil para explorar cómo la calidad de la reconstrucción del modelo SINDy depende del paso de integración de los datos de simulación.

La gráfica generada en la figura 8 muestra cómo varían los coeficientes de las derivadas en función del paso de integración utilizado en la simulación del sistema de Lorenz. Cada subgráfico corresponde a una de las derivadas:  $dx/dt$ ,  $dy/dt$  y  $dz/dt$ .

En el subgráfico correspondiente a  $dx/dt$ , se observa que los coeficientes asociados a los términos lineales ( $x$ ) y de interacción ( $xy$ ) se mantienen cercanos a cero para pasos de integración pequeños. Sin embargo, a medida que el paso de integración aumenta, estos coeficientes adquieren valores significativos y estables, lo que sugiere que la presencia de estos términos es importante en la descripción de la dinámica de  $x$  en el sistema de Lorenz. Por otro lado, los coeficientes asociados a los términos cuadráticos ( $xx$ ) y de interacción cruzada ( $xz$ ) se mantienen cercanos a cero en todo el rango de pasos de integración, lo que indica que estos términos tienen un impacto menos significativo.

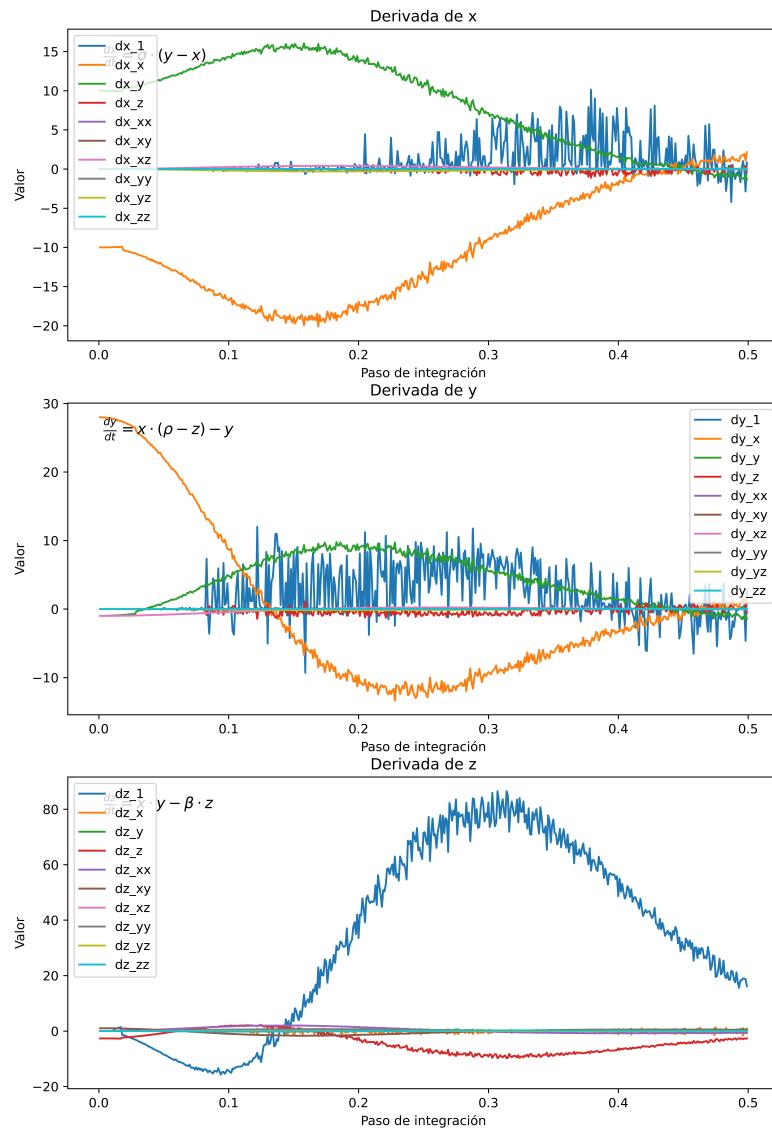


Figure 8: Gráfica de los coeficientes de las derivadas en función del paso de integración.

En el subgráfico correspondiente a  $dy/dt$ , se observa un patrón similar. Los coeficientes asociados a los términos lineales ( $\rho$  y  $z$ ) y de interacción ( $xy$ ) aumentan y se estabilizan a medida que el paso de integración aumenta. Sin embargo, los coeficientes asociados a los términos cuadráticos ( $yy$  y  $yz$ ) y de interacción cruzada ( $yz$ ) permanecen cercanos a cero.

En el subgráfico correspondiente a  $dz/dt$ , los coeficientes siguen un patrón similar a los subgráficos anteriores. Los coeficientes asociados a los términos lineales ( $xy$ ) y de interacción ( $xz$ ) aumentan y se estabilizan a medida que el paso de integración aumenta, mientras que los coeficientes asociados a los términos cuadráticos ( $zz$ ) y de interacción cruzada ( $yz$ ) permanecen cercanos a cero.

Estos resultados sugieren que el tamaño del paso de integración utilizado en la simulación puede influir en la precisión y en la importancia relativa de los diferentes términos en las ecuaciones de Lorenz. Es importante seleccionar un paso de integración apropiado que capture adecuadamente la dinámica del sistema y que permita una reconstrucción precisa de las ecuaciones.

En resumen, la gráfica proporciona información sobre cómo los coeficientes de las derivadas en las ecuaciones de Lorenz varían con el paso de integración utilizado en la simulación. Esto puede ayudar a comprender la sensibilidad de las ecuaciones a diferentes tamaños de paso y a identificar los términos más relevantes en la descripción de la dinámica del sistema.

### 5.1.2 Barrido de condiciones iniciales

1. Se importan las librerías necesarias. Numpy para las operaciones matemáticas, pandas para el manejo de datos, scipy.integrate para resolver ecuaciones diferenciales ordinarias y pysindy (un módulo para la identificación escasa de sistemas no lineales).
2. Se define el sistema de Lorenz, una serie de ecuaciones diferenciales que describen el movimiento de partículas en un fluido.
3. Se establecen los parámetros del sistema de Lorenz:  $\sigma = 10$ ,  $\rho = 28$ , y  $\beta = \frac{8}{3}$ .
4. Se establece un parámetro de escasez (sparsity) de 0.01 para el modelo SINDy. SINDy es una técnica de modelado que busca ecuaciones diferenciales que describen un conjunto de datos dado. El parámetro de escasez controla qué tan escaso (es decir, cuántos coeficientes son cero) es el modelo.
5. Se crea una función para calcular y almacenar los coeficientes del modelo SINDy. Primero, se simulan los datos utilizando el sistema de Lorenz y luego se ajusta el modelo SINDy a estos datos. Los coeficientes se almacenan en un dataframe de pandas.
6. Se fija el tiempo de simulación en 100 y se genera un vector de tiempo con un número de puntos calculado a partir de este tiempo de simulación.
7. Se definen las condiciones iniciales fijas para  $x_0$ ,  $y_0$ ,  $z_0$ .
8. Se crean dataframes vacíos para almacenar los coeficientes de cada barrido.
9. Se definen los nombres de las columnas del dataframe.
10. Se realiza un barrido en las condiciones iniciales para  $x$ ,  $y$ ,  $z$  respectivamente, y en cada barrido se calculan y almacenan los coeficientes utilizando la función definida en el paso 5.
11. Los coeficientes obtenidos para cada barrido se guardan en un archivo .csv.

### 5.1.3 Simulación Usando h buena

Las gráficas muestran la variación de los coeficientes del modelo SINDy, ajustados al sistema de Lorenz, frente a diferentes condiciones iniciales. Cada subgráfica corresponde a una combinación específica de una condición inicial ( $x$ ,  $y$ , o  $z$ ) y una ecuación del sistema de Lorenz (' $dx$ ', ' $dy$ ', ' $dz$ ').

La evolución de las líneas en las gráficas proporciona información sobre cómo los aspectos del sistema de Lorenz son sensibles a las condiciones iniciales. Algunos coeficientes pueden mostrar cambios significativos, indicando una alta sensibilidad, mientras que otros pueden permanecer constantes, lo que sugiere robustez.

El paso de integración en el cálculo del modelo es crucial para la precisión y significado de los resultados.

A continuación, se muestra la gráfica generada:

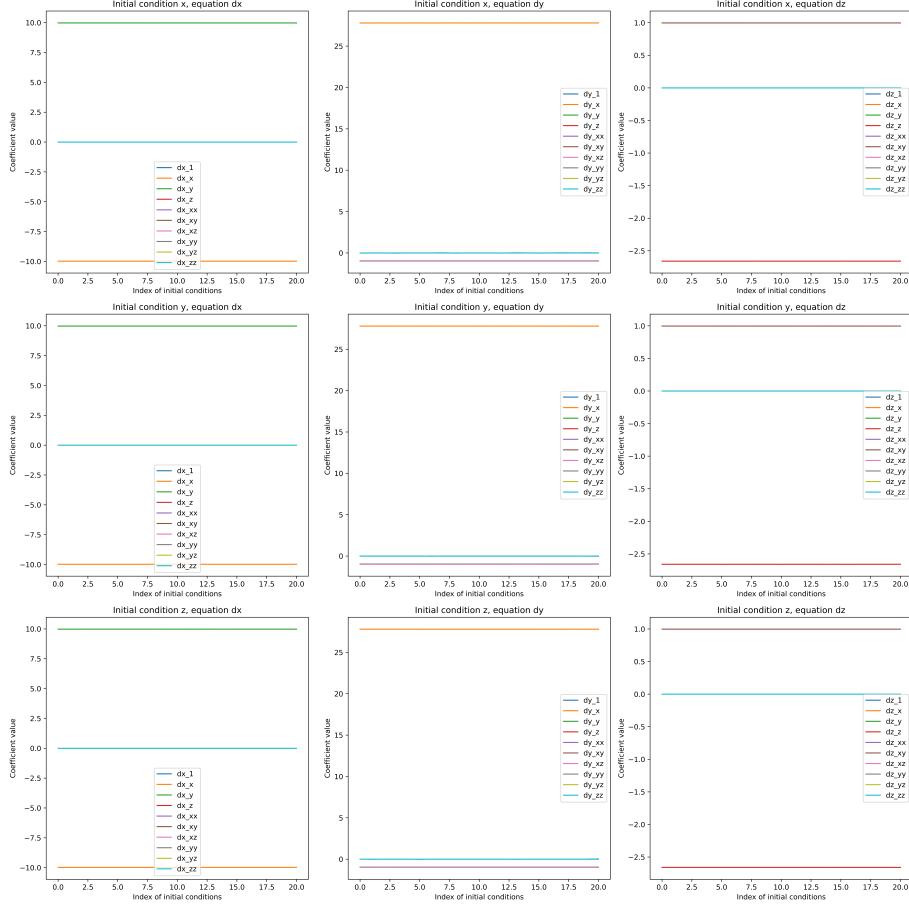


Figure 9: Visualización de los coeficientes del sistema de Lorenz para diferentes condiciones iniciales.

#### 5.1.4 Usando $h$ mala

Las gráficas generadas representan cómo los coeficientes del modelo SINDy, ajustados al sistema de Lorenz, cambian al variar las condiciones iniciales. Cada subgráfica corresponde a una combinación única de una condición inicial (x, y, o z) y una ecuación del sistema de Lorenz ('dx', 'dy', 'dz').

Cuando se utilizó un paso de integración ( $h$ ) adecuado, se observó que los coeficientes del modelo SINDy se mantenían bastante constantes, independien-

temente de las condiciones iniciales. Esto sugiere que el sistema de Lorenz es robusto frente a variaciones en las condiciones iniciales, lo que es una característica deseable en un sistema físico.

Sin embargo, cuando se utilizó un paso de integración inadecuado, los coeficientes del modelo SINDy mostraron grandes variaciones en función de las condiciones iniciales. Esta sensibilidad a las condiciones iniciales podría ser indicativa de errores numéricos introducidos por el paso de integración inadecuado.

La siguiente gráfica muestra estos resultados:



Figure 10: Visualización de los coeficientes del sistema de Lorenz para diferentes condiciones iniciales, utilizando un paso de integración inadecuado.

## 5.2 Reconstrucción del espacio de estado

El método de Takens se utiliza para reconstruir el espacio de estado a partir de la serie temporal  $x(t)$ . Se prueba una gama de valores de retraso de tiempo, y para cada valor de retraso, se reconstruye el espacio de estado y se compara con el espacio de estado original. El retraso de tiempo que produce la reconstrucción más cercana al espacio de estado original se elige como el mejor retraso de tiempo.

Las siguientes figuras muestran las trayectorias reconstruidas para diferentes valores de retraso de tiempo. Cada figura contiene 6 subplots, y hay un total de 5 figuras.

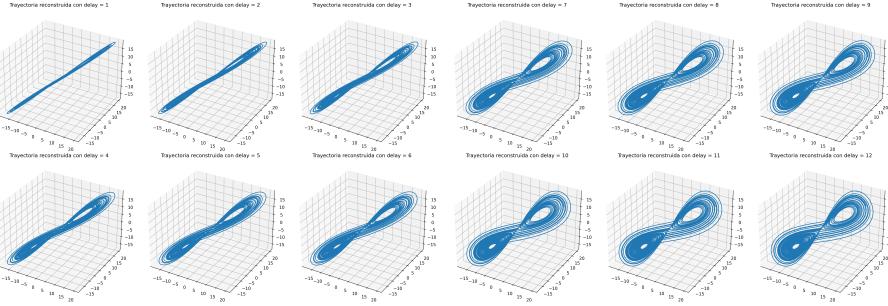


Figure 11: Trayectorias reconstruidas para los primeros 6 valores de retraso de tiempo.

Figure 12: Trayectorias reconstruidas para los siguientes 6 valores de retraso de tiempo.

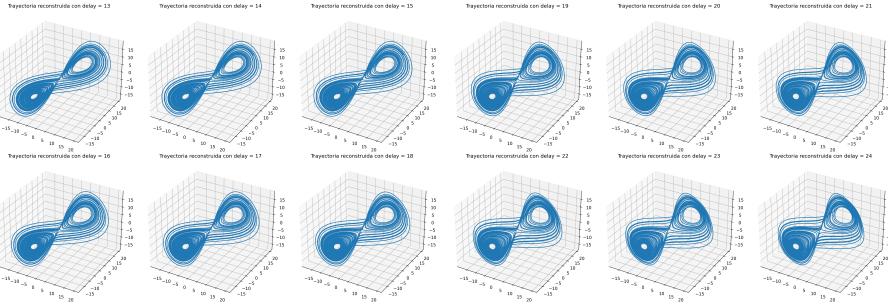


Figure 13: Trayectorias reconstruidas para los siguientes 6 valores de retraso de tiempo.

Figure 14: Trayectorias reconstruidas para los siguientes 6 valores de retraso de tiempo.

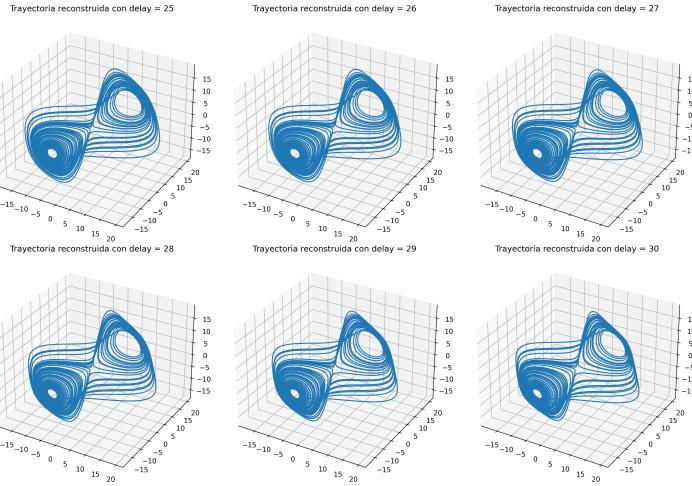


Figure 15: Trayectorias reconstruidas para los últimos valores de retraso de tiempo.

Estas figuras muestran que el retraso de tiempo tiene un impacto significativo en la calidad de la reconstrucción del espacio de estado. Algunas reconstrucciones se parecen más a la trayectoria original que otras, lo que indica que el método de Takens ha sido exitoso en reconstruir el espacio de estado a partir de una única serie temporal.

### 5.3 Comparación de la reconstrucción con la trayectoria original

Finalmente, se compara la trayectoria reconstruida con el mejor retraso de tiempo con la trayectoria original.

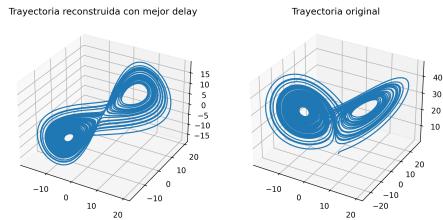


Figure 16: Comparación de la trayectoria reconstruida con el mejor retraso de tiempo (izquierda) y la trayectoria original (derecha).

La figura muestra la trayectoria reconstruida con el mejor retraso de tiempo y la trayectoria original. Se puede observar que la trayectoria reconstruida se parece bastante a la trayectoria original, lo que indica que el método de Takens ha sido exitoso en reconstruir el espacio de estado a partir de una única serie temporal.

### 5.3.1 Reconstrucción de ECG con base de taylor

Realizamos un barrido de delays de Takens desde 100 hasta 1000, con incrementos de 100 utilizando la base por default de la librería de SYNDy que se refiere a los polinomios de Taylor. Sin embargo, se encontró que todos los delays, excepto el de 500, resultaban en colapsos durante la simulación del modelo, probablemente debido a desbordamientos numéricos. El delay de 500 permitió una simulación exitosa, aunque la calidad de la reconstrucción fue pobre.

### 5.3.2 Análisis de la Reconstrucción con Delay de 500

La única reconstrucción que se logró simular fue con un delay de Takens de 500. A continuación, se presenta la gráfica de la evolución temporal y el retrato de fase de esta reconstrucción:

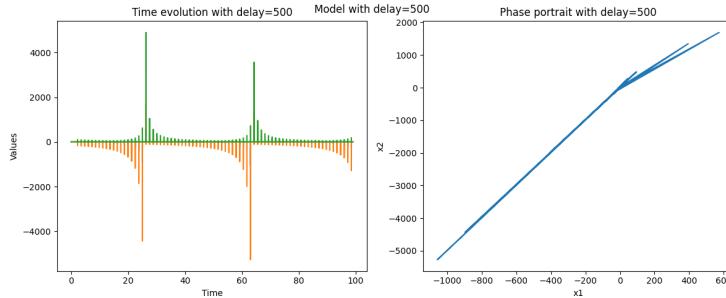


Figure 17: Evolución temporal y retrato de fase de la reconstrucción con delay de Takens de 500.

La gráfica muestra una representación visual de la reconstrucción, evidenciando la baja calidad de la misma. Esta inadecuación se manifiesta en una falta de coherencia y precisión en la representación de la dinámica del ECG original.

La experimentación con diferentes delays de Takens revela limitaciones en la capacidad de reconstruir con precisión el sistema de ecuaciones del ECG. La reconstrucción con un delay de 500, aunque exitosa en términos de evitar colapsos numéricos, proporciona resultados insatisfactorios. Esto subraya la necesidad de una selección cuidadosa del delay y una metodología robusta para la identificación de sistemas dinámicos a partir de series temporales.

## 5.4 Mejora en la Reconstrucción de Sistemas con Métodos de Takens y SINDy Personalizado

Este apartado se dedica a la mejora lograda en la reconstrucción de sistemas de ecuaciones mediante el uso combinado del método de Takens y una versión personalizada del método SINDy, que incorpora funciones trigonométricas y la función identidad, siguiendo una base similar a la de Fourier.

Para la reconstrucción, se emplea el método de Takens para transformar la serie temporal del ECG en un espacio de fase multidimensional. Posteriormente, aplicamos el método SINDy con una librería personalizada que incluye funciones seno, coseno y la función identidad. Esto permite una mejor captura de las dinámicas oscilatorias y no lineales del sistema.

Se realiza un barrido de delays de Takens desde 300 hasta 1000, con incrementos de 100. Se descubre que el delay óptimo para la reconstrucción es 300. Con este delay, la calidad de la reconstrucción mejora significativamente en comparación con los intentos anteriores.

La reconstrucción conseguida con un delay de Takens de 300 se muestra a continuación:

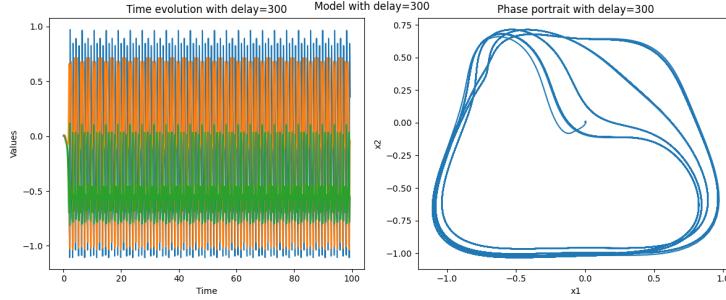


Figure 18: Evolución temporal y retrato de fase de la reconstrucción con delay de Takens de 300 utilizando la librería personalizada en SINDy.

Esta gráfica evidencia una mejora notable en la reconstrucción. Se observan oscilaciones periódicas que, aunque no replican exactamente la dinámica del ECG original, representan un avance significativo respecto a los intentos previos. La presencia de estas oscilaciones sugiere que la inclusión de funciones trigonométricas y la función identidad permite captar mejor la naturaleza oscilatoria del ECG.

#### 5.4.1 Potencial y Novedad en la Aplicación de Fourier y SINDy para Datos de Simulación

El enfoque innovador de combinar series de Fourier y SINDy personalizado presenta un potencial significativo para el análisis de datos de simulación, especialmente en el contexto de electrocardiogramas (ECG). Esta metodología no solo mejora la reconstrucción de sistemas dinámicos, sino que también ofrece una nueva perspectiva para interpretar datos de ECG.

La inclusión de funciones trigonométricas en el método SINDy permite una captura más precisa de las dinámicas oscilatorias y no lineales. Al aplicar series de Fourier en conjunto con SINDy, podemos descomponer la actividad eléctrica del corazón en sus componentes frecuenciales fundamentales, ofreciendo así una visión más detallada y matizada.

La transformación de los datos de simulación a través de esta metodología avanzada facilita la identificación de patrones subyacentes y anomalías en los ritmos cardíacos, lo que podría ser crucial para aplicaciones médicas como el diagnóstico precoz de arritmias cardíacas.

Este enfoque también abre nuevas posibilidades para el refinamiento de modelos cardíacos en la investigación y el desarrollo de tratamientos personalizados, basándose en la capacidad única de esta metodología para descomponer y analizar la compleja actividad eléctrica del corazón.

La representación en series de Fourier, obtenida a través del uso de SINDy

con funciones trigonométricas, proporciona una herramienta poderosa para analistas y médicos, permitiendo una comprensión más profunda y una intervención más efectiva en la salud cardíaca.

#### 5.4.2 Discusión

El uso de una librería personalizada en SINDy, con funciones trigonométricas y la función identidad, demuestra ser una estrategia eficaz para mejorar la reconstrucción de sistemas dinámicos a partir de series temporales. Aunque la reconstrucción con un delay de 300 no captura por completo la dinámica exacta del ECG, las oscilaciones periódicas observadas son un indicativo prometedor de su potencial para estudios más detallados y aplicaciones prácticas en el análisis de ECG.

### 5.5 Avances en la Reconstrucción con Métodos de Takens y SINDy Usando Polinomios de Padé

Esta sección se centra en la reconstrucción mejorada de sistemas de ecuaciones mediante la combinación del método de Takens y una versión personalizada del método SINDy, que incorpora polinomios de Padé de primer y segundo orden.

La metodología consiste en transformar la serie temporal del ECG en un espacio de fase multidimensional utilizando el método de Takens, esta vez con una dimensión de reconstrucción aumentada a 4. Posteriormente, aplicamos el método SINDy con una librería personalizada que incluye polinomios de Padé de primer y segundo orden, que son funciones racionales conocidas por su capacidad para aproximar de manera eficiente funciones complejas.

Se realiza un barrido de delays de Takens desde 100 hasta 1000, con incrementos de 100. Se descubre que el delay óptimo para la reconstrucción es 400. Con este delay y la dimensión de reconstrucción incrementada, la calidad de la reconstrucción mejora sustancialmente en comparación con los intentos anteriores.

#### 5.5.1 Análisis de la Reconstrucción con Delay de 400

La reconstrucción conseguida con un delay de Takens de 400 se muestra a continuación:

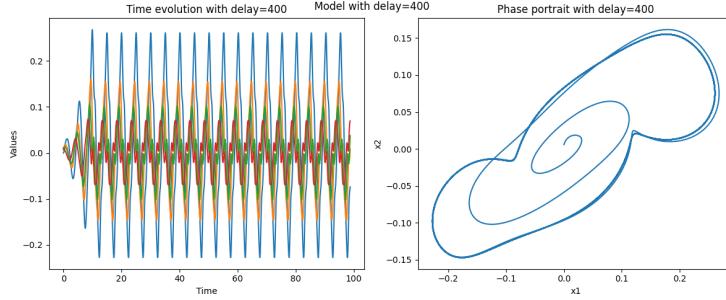


Figure 19: Evolución temporal y retrato de fase de la reconstrucción con delay de Takens de 400 utilizando polinomios de Padé en la librería SINDy.

Esta gráfica evidencia una mejora significativa en la reconstrucción. Se observan oscilaciones periódicas que incluyen picos intermedios más parecidos al ECG original, un avance notable respecto a los intentos previos. Aunque la reconstrucción aún no es exacta, muestra un progreso importante hacia la captura precisa de la dinámica del ECG.

La inclusión de polinomios de Padé en la librería personalizada de SINDy, junto con el aumento de la dimensión de reconstrucción de Takens, ha demostrado ser una estrategia eficaz para mejorar la reconstrucción de sistemas dinámicos a partir de series temporales. El éxito observado con un delay de 400 indica un paso significativo hacia una representación más precisa de la señal de ECG, lo que es prometedor para futuras investigaciones y aplicaciones prácticas en el análisis de ECG.

## 5.6 Teoría y Aplicación de Polinomios de Hermite en SINDy

La integración de polinomios de Hermite en el método SINDy representa una poderosa extensión para el análisis de sistemas dinámicos. Los polinomios de Hermite, particularmente hasta el tercer orden, ofrecen una base sólida para capturar las características no lineales de los datos.

### 5.6.1 Fundamentos de los Polinomios de Hermite

Los polinomios de Hermite son una secuencia de polinomios ortogonales que surgen en la teoría de probabilidades y la física cuántica. Son especialmente útiles para representar distribuciones de probabilidad y procesos estocásticos. Hasta el tercer orden, estos polinomios pueden modelar interacciones no lineales complejas y fenómenos oscilatorios en sistemas dinámicos.

### 5.6.2 Implementación en SINDy

En SINDy, los polinomios de Hermite se utilizan como parte de la biblioteca de funciones para identificar las ecuaciones diferenciales subyacentes de un sistema

dinámico. Al incluir estos polinomios hasta el tercer orden, SINDy puede capturar de manera efectiva dinámicas complejas y no lineales, lo que resulta en modelos más precisos y robustos.

La aplicación de los polinomios de Hermite en SINDy no solo mejora la precisión en la modelización de sistemas dinámicos, sino que también permite una mejor comprensión de las interacciones no lineales y los procesos estocásticos involucrados.

### 5.6.3 Análisis de la Reconstrucción con Delay de 300

La reconstrucción conseguida con un delay de Takens de 300 se muestra a continuación:

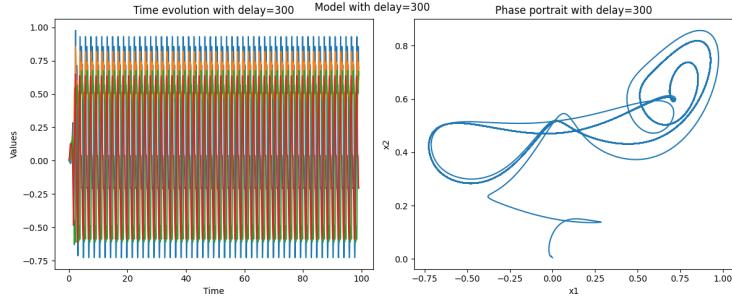


Figure 20: Evolución temporal y retrato de fase de la reconstrucción con delay de Takens de 300 utilizando polinomios de hermite en la librería SINDy.

## 5.7 Incorporación de Polinomios de Legendre de Tercer Orden en SINDy

La aplicación de polinomios de Legendre en la metodología SINDy representa una técnica avanzada para analizar y modelar sistemas dinámicos. Los polinomios de Legendre, especialmente hasta el tercer orden, proporcionan una base robusta para describir dinámicas complejas y fenómenos oscilatorios.

### 5.7.1 Principios de los Polinomios de Legendre

Los polinomios de Legendre son soluciones a la ecuación diferencial de Legendre y forman un conjunto de polinomios ortogonales sobre el intervalo  $[-1, 1]$ . Su relevancia en la física y matemáticas es notable, especialmente en problemas que involucran esferas o campos gravitatorios.

### 5.7.2 Aplicación en SINDy

En SINDy, los polinomios de Legendre hasta el tercer orden pueden ser utilizados para enriquecer la biblioteca de funciones, permitiendo la captura de patrones más complejos en los datos. Estos polinomios ofrecen una forma efectiva de

modelar interacciones no lineales y son particularmente útiles en sistemas con simetrías o propiedades de conservación.

La integración de estos polinomios mejora la habilidad de SINDy para desentrañar la estructura subyacente de los sistemas dinámicos, proporcionando una herramienta más poderosa para el análisis y la modelización.

### 5.7.3 Análisis de la Reconstrucción con Delay de 400

La reconstrucción conseguida con un delay de Takens de 400 se muestra a continuación:

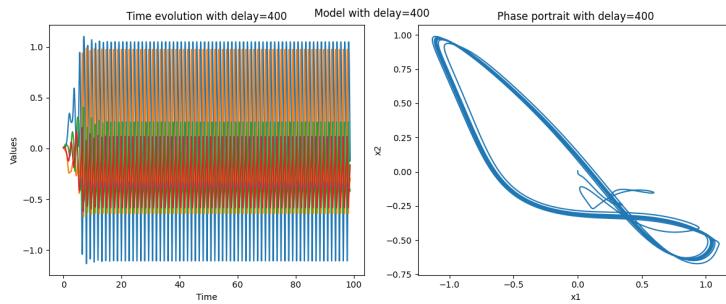


Figure 21: Evolución temporal y retrato de fase de la reconstrucción con delay de Takens de 400 utilizando polinomios de legeandre en la librería SINDy.

## References

- [1] Robust Reconstruction of Electrocardiogram Using Subject-Based Deep Learning Model. *Frontiers*, 2019, <https://www.frontiersin.org/articles/10.3389/fphys.2019.01020/full>.
- [2] Lead Reconstruction Using Artificial Neural Networks for High Precision ECG Signal. *Sensors*, 2020, <https://www.mdpi.com/1424-8220/20/12/3515>.
- [3] Reconstruction of 12-Lead Electrocardiogram from a Three-Lead Patch Device Using LSTM. *MDPI*, 2020, <https://www.mdpi.com/2079-6374/10/4/40>.
- [4] ECG Signal Reconstruction Using FMCW Radar and Convolutional Neural Networks. *IEEE Xplore*, 2020, <https://ieeexplore.ieee.org/document/9127815>.
- [5] PPG2ECGps: An End-to-End System for PPG-to-ECG Conversion. *MDPI*, 2020, <https://www.mdpi.com/2306-5729/5/2/39>.
- [6] F. Takens, "Detecting strange attractors in turbulence," *Dynamical Systems and Turbulence*, 1981, Springer, Berlin, Heidelberg.

- [7] K. Pearson, "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine*, 1901.
- [8] P. Comon, "Independent Component Analysis, a New Concept?" *Signal Processing*, 1994.
- [9] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, 1997.
- [10] K. Cho et al., "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [11] L. A. Smith, "What might we learn from climate forecasts?" *Proceedings of the National Academy of Sciences*, 2002.
- [12] E. I. Vlahogianni et al., "Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, 2014.
- [13] A. Alahi et al., "Social LSTM: Human Trajectory Prediction in Crowded Spaces," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [14] I. Goodfellow et al., "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, 2014.
- [15] Y. Zhang et al., "Multimodal Learning for Clinical Decision Support," *Journal of Healthcare Engineering*, 2016.
- [16] R. Caruana et al., "Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission," *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.
- [17] I. Goodfellow, Y. Bengio, A. Courville, "Deep Learning," MIT Press, 2016.
- [18] M.A. Quiroz-Juárez, O. Jiménez-Ramírez, R. Vázquez-Medina, V. Breña-Medina, J.L. Aragón, y R.A. Barrio. "Generation of ECG signals from a reaction-diffusion model spatially discretized," *Nature*, 2019. DOI: <https://doi.org/10.1038/s41598-019-55448-5>