UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Soluciones de las Ecuaciones de Saint-Venant para Canales Abiertos Basadas en Aprendizaje Profundo y Un Criterio Para el Análisis de Estabilidad

Juan Sebastián Villarreal Godoy

Matemáticas

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Matemático

Quito, 19 de agosto de 2025

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Soluciones de las Ecuaciones de Saint-Venant para Canales Abiertos

Basadas en Aprendizaje Profundo y Un Criterio Para el Análisis de Estabilidad

Juan Sebastián Villarreal Godoy

Julio César Ibarra Fiallo, Ph.D. (c) _____

Luis Servando Espín Torres, Ph.D. (c) _____

John Skukalek, Ph.D. _____

Quito, 19 de agosto de 2025

DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos:	Juan Sebastián Villarreal Godoy
Código:	00326102
C.I.:	1722213558
Fecha:	Quito, 19 de agosto de 2025

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en `http://bit.ly/COPETheses`

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on `http://bit.ly/COPETheses`

RESUMEN

El estudio de las Ecuaciones de Saint Venant siempre han sido de gran curiosidad debido a sus bastas aplicaciones en mecánica de fluidos, como en el diseño de canales y tuberías, y el hecho de que es un sistema que no tiene solución analítica. En este trabajo se buscó usar una metodología con redes neuronales para encontrar las soluciones a las Ecuaciones de Saint Venant basado en un criterio de estabilidad. Este criterio de estabilidad tiene su origen en un método numérico tradicional conocido como método de Lax-Friedrichs , el cual también sirvió como referencia para medir el rendimiento del método planteado. Se hizo un estudio de varias propiedades del método propuesto junto con un estudio de estabilidad de ambos métodos. El mejor modelo obtuvo un error cuadrático medio (MSE) respecto al método tradicional de 0.0008 y 0.0038 para cada una de las soluciones de las ecuaciones, y un diferencia de valor máximo de 0.1128 y 0.2178 respectivamente con una convergencia relativamente rápida. Se concluyó que el modelo tiene una buena aproximación promedio al método de Lax-Friedrichs, sin embargo, falta mejorar el modelo porque en ciertas zonas el método propuesto pierde el comportamiento de la solución del método tradicional, lo que también se ve reflejado en la diferencia de valores máximos.

***Palabras clave*:** Redes Neuronales, Ecuaciones Diferenciales Parciales, Ecuaciones de Saint-Venant, Estabilidad, Aprendizaje Profundo, Canales Abiertos.

ABSTRACT

Study of the Saint Venant Equations has always been of great curiosity due to its many applications in fluid mechanics, such as channels and pipes design, and the fact that they are a system that has no analytical solution. In this work, we sought to use a methodology with neural networks to find the solutions to the Saint Venant Equations based on a stability criterion. This stability criterion has its origin in a traditional numerical method known as the Lax-Friedrichs method, which also served as a reference to measure the performance of the proposed method. A study of several properties of the proposed method was carried out together with a stability study of both methods. The best model obtained a mean square error (MSE) with respect to the traditional method of 0.0008 and 0.0038 for each of the solutions of the equations, and a maximum value difference of 0.1128 and 0.2178 respectively with a relatively fast convergence. It was concluded that the model has a good average approximation to the Lax-Friedrichs method, however, the model needs to be improved because in certain areas the proposed method loses the behavior of the solution of the traditional method, which is also reflected in the difference of maximum values.

***Keywords*:** Neural Networks, Partial Differential Equations, Saint-Venant Equations, Stability, Deep Learning, Open Channels.

Índice general

Índice de figuras

Índice de tablas

DEDICATORIA

A mi abuelita Flor Alba Revelo, desde donde me estés viendo, esto es fruto de tu entrega y dedicación a tu familia.

AGRADECIMIENTO

Aprecio enormemente y deseo manifestar mi agradecimiento a mis padres Héctor y Gabriela. Han sido mi fortaleza constantemente y la base de lo que he logrado ser y conseguir. Todo lo que soy y lo que espero lograr es en gran parte porque soy su hijo. De igual forma, doy gracias a mi familia por su apoyo absoluto en cada una de mis decisiones.

Además, quiero agradecer en gran medida a mis tutores para este proyecto, Julio Ibarra y Servando Espín. Su guía y paciencia han sido clave en el desarrollo tanto de este trabajo como de mi persona. El carisma y el conocimiento que me han brindado han sido, sin duda, una fuente de inspiración y motivación durante este proceso.

Quiero agradecer de igual manera a John Skukalek, pues ha sido mi mayor mentor durante el estudio de mi carrera, y deseo manifestar mi gratitud a José Mora, José Álvarez, David Egas y Danny Navarrete, su confianza en mí me ha permitido estudiar esta carrera.

Finalmente, agradezco a José Palacios y Emilio Paredes porque gracias a su gran contribución, este trabajo se pudo realizar.

Capítulo 1

Introducción

1.1. Ecuaciones Diferenciales Parciales

Debido a que las Ecuaciones de Saint-Venant se aplican en tres dimensiones espaciales y una temporal, todas las definiciones y desarrollo van a estar efectuados para espacios Euclideanos $\mathbb{R}^n$. Además, nuestro interés estará en funciones cuya imagen sea un subconjunto de los números reales por lo que si no se hace referencia a otra cosa, se va a asumir que la norma aplicada y el producto punto van a ser los euclideanos.

Definición 1.1.1. Sea $x \in \mathbf{R}^n$ tal que $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. La norma euclideana $(||\cdot||)$ relacionada con el producto escalar $(\langle \cdot\, , \cdot \rangle)$ aplicada a $\mathbf{x}$ se define como:

$$||\mathbf{x}|| = \langle x, x \rangle^{1/2} = \left(\sum_{i=1}^{n} (x_i)^2 \right)^{1/2} \tag{1.1}$$

Como es de esperarse, las ecuaciones diferenciales tienen como base la derivada. Para definirla, es necesario primero el concepto de límite [36].

Definición 1.1.2. Una función $f : E \subseteq \mathbb{R}^n \to \mathbb{R}$ tiende hacia el límite $L \in \mathbb{R}$ en $\mathbf{a} \in \mathbb{R}^n$ si

para todo $\varepsilon > 0$ existe algún $\delta > 0$ tal que, para todo $\mathbf{x} \in E$, si $0 < ||\mathbf{x} - \mathbf{a}|| < \delta$, entonces $|f(\mathbf{x}) - L| < \varepsilon$. Esto se puede escribir como $f(\mathbf{x}) \to L$ cuando $x \to a$ o

$$\lim_{x \to a} f(\mathbf{x}) = L$$

Cuando $n = 1$, se tiene la noción básica de la derivada

Definición 1.1.3. Sea una función $f : U \subseteq \mathbb{R} \to \mathbb{R}$. La **derivada** de f en $a \in U$ se define como

$$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h} \tag{1.2}$$

si es que el límite existe. Se dice que f es diferenciable en U si existe la derivada para todo $a \in U$.

A partir de aquí, U siempre va a representar un conjunto abierto en $\mathbb{R}^n$ [30].

Definición 1.1.4. Un conjunto $U \subseteq \mathbb{R}^n$ se dice que es **abierto** si para todo $x \in U$ existe un número $r \in \mathbb{R}$ y un conjunto $N_r = \{y \in \mathbb{R}^n \mid ||x - y|| < r\}$ tal que $N_r \subseteq U$

Cuando el dominio de la función real es subconjunto de los espacios multidimensionales $\mathbb{R}^n$ se puede usar el concepto de derivada parcial

Definición 1.1.5. Sea una función $u : U \subseteq \mathbb{R}^n \to \mathbb{R}$ y sea $\mathbf{x} \in U$ tal que $\mathbf{x} = (x_1, x_2, \ldots, x_n)$. Se dice que $\mathbf{x}$ se encuentra representada en la base canónica $\{\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}\}$ de $\mathbb{R}^n$, donde e_i es el vector en $\mathbb{R}^n$ cuyas entradas son cero excepto en la posición i cuya entrada es uno. Esto se puede interpretar equivalentemente como

$$\mathbf{x} = x_1\mathbf{e_1} + x_2\mathbf{e_2} + \cdots + x_n\mathbf{e_n}$$

Se define la **derivada parcial** de $u(\mathbf{x})$ respecto a x_i para $i = 1, 2 \ldots, n$ y $h \in \mathbb{R}$ como

$$\frac{\partial u}{\partial x_i}(\mathbf{x}) = \lim_{h \to 0} \frac{u(\mathbf{x} + h\mathbf{e_i}) - u(\mathbf{x})}{h} \tag{1.3}$$

Dado que el límite exista.

También se usará una notación equivalente para las derivadas parciales como subíndices. Por ejemplo:

$$\frac{\partial u}{\partial x_i} = u_{x_i}$$

$$\frac{\partial^3 u}{\partial x_i \partial x_j \partial x_k} = u_{x_i x_j x_k}$$

Y para algunos conceptos teóricos, una forma útil de representar derivadas parciales es con la notación multi-índice [38].

Definición 1.1.6. Un vector de la forma $\alpha = (\alpha_1, \ldots, \alpha_n)$ donde cada elemento α_i , para $i = 1, \ldots, n$, es un entero no-negativo se dice que es un **multi-índice** al que se asocia un orden $|\alpha| = \alpha_1 + \cdots + \alpha_n$

Sea α un multíndice, se tiene que

$$D^\alpha u(\mathbf{x}) = \frac{\partial^{|\alpha|} u(\mathbf{x})}{\partial x_1^{\alpha_1} \ldots \partial x_n^{\alpha_n}} = \partial_{x_1}^{\alpha_1} \ldots \partial_{x_n}^{\alpha_n} u \tag{1.4}$$

Una forma útil de representación porque se puede asociar al conjunto de todas las derivadas parciales dado un orden k

$$D^k u(\mathbf{x}) = \{D^\alpha u(\mathbf{x}) \mid |\alpha| = k\}$$

El número total de derivadas parciales para $x \in \mathbb{R}^n$ y orden máximo k es n^k. Con esta idea, se puede presentar a $D^k u$ como un punto de R^{n^k} y dependiendo del contexto, se puede organizar a $D^k u$ como una matriz, así $D^k u \in \mathbb{R}^{\underbrace{n \times \cdots \times n}_{k \ veces}}$.

Una de las ventajas de representar este conjunto en forma de tupla es que cuando $k = 1$

$$D^1 u = \nabla u = (u_{x_1}, \ldots, u_{x_n}) \tag{1.5}$$

se tiene al vector gradiente que se definirá posteriormente.

Aclarada la notación, se prosigue a explorar algunos conceptos más. En el campo de las ecuaciones diferenciales parciales es útil definir lo que es un espacio vectorial y un operador [17].

Definición 1.1.7. Un **espacio vectorial** sobre un campo K ($\mathbb{R}$ o $\mathbb{C}$) es un conjunto no vacío X de elementos x, y (llamados vectores) junto dos operaciones algebraicas: adición de vectores y multiplicación de vectores por escalares.

La adición de vectores asocia cada par ordenado de vectores (x, y) un vector $x + y$ y satisface las siguientes propiedades

1. $x + y = y + x$
2. $x + (y + z) = (x + y) + z$
3. Existe un vector 0 llamado el vector cero que satisface $x + 0 = x$
4. Para todo vector x existe un vector $-x$ que satisface $x + (-x) = 0$

La multiplicación por escalares asocia cada vector x y un escalar $\alpha \in K$ un vector αx llamado el producto de α y x que satisface

1. $\alpha(\beta x) = (\alpha\beta)x$
2. $1x = x$

3. $\alpha(x+y) = \alpha x + \alpha y$

4. $(\alpha+\beta)x = \alpha x + \beta x$

Definición 1.1.8. Un **operador** es una función $T : V \to W$ donde V y W son espacios vectoriales.

Finalmente, se puede definir una ecuación diferencial parcial (PDE)[13].

Definición 1.1.9. Una **ecuación diferencial parcial** o PDE de orden k es una ecuación que contiene una función de dos o más variables y algunas de sus derivadas parciales y puede ser expresada como

$$F(D^k u(\mathbf{x}), D^{k-1}u(\mathbf{x}), \ldots, Du(\mathbf{x}), u(\mathbf{x}), \mathbf{x}) = 0 \tag{1.6}$$

Donde $u : U \subseteq \mathbb{R}^n \to \mathbb{R}$ se conoce como la función objetivo o incógnita y $F : \mathbb{R}^{n^k} \times \mathbb{R}^{n^{k-1}} \times \cdots \times \mathbb{R} \times U \to \mathbb{R}$ es una función que define la expresión y es conocida. F también es conocida como el operador diferencial de la ecuación diferencial.

El objetivo, cuando se trata con una ecuación diferencial parcial es el encontrar todas las funciones u que satisfagan la Ecuación1.6. También, se pueden añadir condiciones que nos permitan aislar ciertas soluciones de interés y que incluso podrían tomar en cuenta parte o la totalidad de la frontera o borde de U (∂U). Esto es muy útil, especialmente cuando se pueden encontrar condiciones que aseguren la existencia y la unicidad de la solución.

1.1.1. Problemas de Valor Inicial

Este tipo de problemas está basado en que generalmente se tiene como una de las variables independientes al tiempo $x_1 = t$. Una condición inicial hace referencia a que se tiene información de la función incógnita o de sus derivadas parciales en un tiempo inicial, que podemos llamar t_0. Un problema de valor inicial hace referencia a una ecuación diferencial parcial que tiene una o

más condiciones iniciales. Así, para una PDE de orden k se puede escribir

$$\begin{cases} F(D^k u(\mathbf{x}), \ldots, \mathbf{x}) = 0 \\ \frac{\partial^m}{\partial t^m} u(t, x_2, \ldots, x_n) = g_m(x_2, \ldots, x_n) \quad \text{para } t = t_0,\ m < k \end{cases} \tag{1.7}$$

Donde, el número de condiciones para $t = t_0$ dependen del problema.

1.1.2. Problemas de Valor en la Frontera

Para este tipo de problemas, primero se requiere la noción de frontera o borde. Para esto, es necesario definir la clausura de un conjunto [30].

Definición 1.1.10. Sea $E \subseteq \mathbb{R}^n$. Un punto $p \in \mathbb{R}^n$ es llamado **punto límite** de E si para todo $r \in \mathbb{R}$ y $N_r = \{x \in \mathbb{R}^n \mid ||p - x|| < r\}$, existe $q \in N_r$ tal que $q \neq p$ y $q \in E$

Definición 1.1.11. Sea $U \subseteq \mathbb{R}^n$ un conjunto. La **clausura** de U es la unión del conjunto U con el conjunto de todos los puntos límite de U, U'. Es decir, la clausura $\bar{U} = U \cup U'$.

Por lo tanto, como se puede interpretar, el borde de un conjunto abierto es precisamente el conjunto de los puntos límite. O definido de una manera más formal

Definición 1.1.12. Sea $U \subseteq \mathbb{R}^n$ un conjunto. Se conoce como la frontera o borde de U al conjunto ∂U definido como

$$\partial U = \overline{U} \cap \overline{U^c} \tag{1.8}$$

Donde U^c representa el complemento de U.

Estos conceptos son importantes porque se puede extender el problema de la ecuación diferencial a la clausura de U. De forma similar, se puede adaptar una ecuación diferencial para que esté definida en un conjunto que esté dentro del conjunto abierto U, y que no sea necesariamente abierto. En cualquier caso, se puede llamar a dicho conjunto no necesariamente abierto, Ω. Un problema de valor en la frontera es una ecuación diferencial cuya función incógnita está definida en Ω y está restringida a condiciones en el borde [38].

Uno de los problemas de valor en la frontera más conocidos es el de la condición de borde de Dirichlet

$$\begin{cases} F(D^k u(\mathbf{x}), \ldots, u(\mathbf{x}), \mathbf{x}) = 0 & \text{en } \Omega \\ u(\mathbf{x}) = g(\mathbf{x}) & \text{sobre } \partial\Omega \end{cases} \tag{1.9}$$

Donde la condición nos indica los valores de la función incógnita en la frontera. Otro de estos problemas es cuando la condición de borde es de Neumann.

$$\begin{cases} F(D^k u(\mathbf{x}), \ldots, u(\mathbf{x}), \mathbf{x}) = 0 & \text{en } \Omega \\ \frac{\partial u}{\partial \mathbf{n}} = h(\mathbf{x}) & \text{sobre } \partial\Omega \end{cases} \tag{1.10}$$

Que da información respecto a la derivada respecto a un vector normal que apunta hacia afuera de la frontera $\partial\Omega$.

1.2. Obtención de Ecuaciones de Saint-Venant

1.2.1. Canales abiertos y su clasificación

En lo que concierne a este estudio, el enfoque estará en el análisis de flujo de agua en un canal abierto, que se refiere al movimiento de volúmenes o cantidades de agua en una construcción cuya sección superior está abierta y permite al flujo tener una superficie libre que está sometida a presión atmosférica. Se supondrá que la dependencia espacial de propiedades o variables a analizar en el canal solo ocurre en la dimensión en la dirección que transcurre el flujo. Dicho esto, se procede a definir conceptos de flujos en canales abiertos [8].

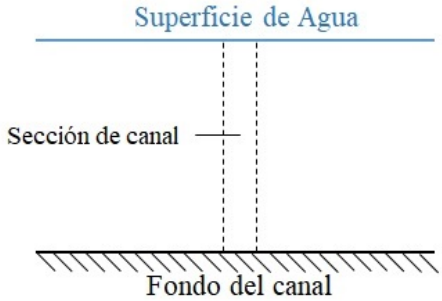


Figura 1.1: Canal Abierto

Primero, para estudiar el flujo de agua se requiere analizar cómo funciona todo el sistema (Figura 1.1). Dentro del canal, se tiene la superficie libre o superficie del agua que es la zona más alta a la que se eleva el agua. Por otro lado, la zona en la que el agua llega a su punto más bajo (la pared inferior del canal) se le conoce como el lecho de canal o fondo de canal. La porción que se estudia se conoce como sección de canal, que involucra al área perpendicular al flujo de agua. Por último, la profundidad de flujo se define como la distancia vertical en $\mathbb{R}$, desde el punto más bajo del fondo del canal hasta la superficie libre de una sección de canal.

De esta manera, el flujo puede clasificarse dependiendo del comportamiento de la profundidad de flujo a través del canal en distintas secciones de canal. Cuando la profundidad de flujo no varía a lo largo de la dimensión espacial se cataloga al flujo como uniforme. Si no hay variación respecto al tiempo, entonces el flujo se conoce como permanente. Por otro lado, si hay variación temporal el flujo se denomina no permanente y si hay variación espacial el flujo se denomina variado. Finalmente, un flujo en el que la profundidad de flujo cambia abruptamente en el espacio se conoce como rápidamente variado, pero si la variación es ligera el flujo es gradualmente variado.

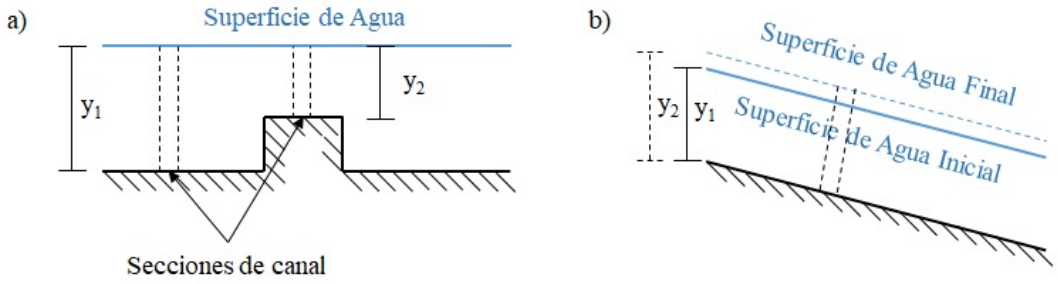


Figura 1.2: Tipos de flujo. a) Flujo variado (rápidamente variado); b) Flujo no permanente

En la Figura 1.2 a) se puede ver un ejemplo de un flujo rápidamente variado donde hay un cambio de profundidad de flujo siguiendo la dimensión espacial a lo largo del flujo desde $y_1 \in \mathbb{R}$ hasta $y_2 \in \mathbb{R}$, y nuevamente a y_1. Por otro lado, en la Figura 1.2 b) se tiene un flujo no permanente en el que la profundidad de flujo cambia en dos estados temporales distintos desde y_1 hasta y_2. En este trabajo, se analizará específicamente al flujo gradualmente variado no permanente al cuál se referirá solo como gradualmente variado (Figura 1.3).

1.2.2. Ecuación de Continuidad

La ecuación de continuidad se deriva de la ley de conservación de masa. Si se considera el sistema de la Figura 1.3, se tiene un flujo gradualmente variado que también cambia a medida que transcurre el tiempo. Si la dimensión espacial en la dirección de flujo es x, se puede analizar el comportamiento del flujo de agua en un canal dentro de un intervalo $I = [a, b]$.

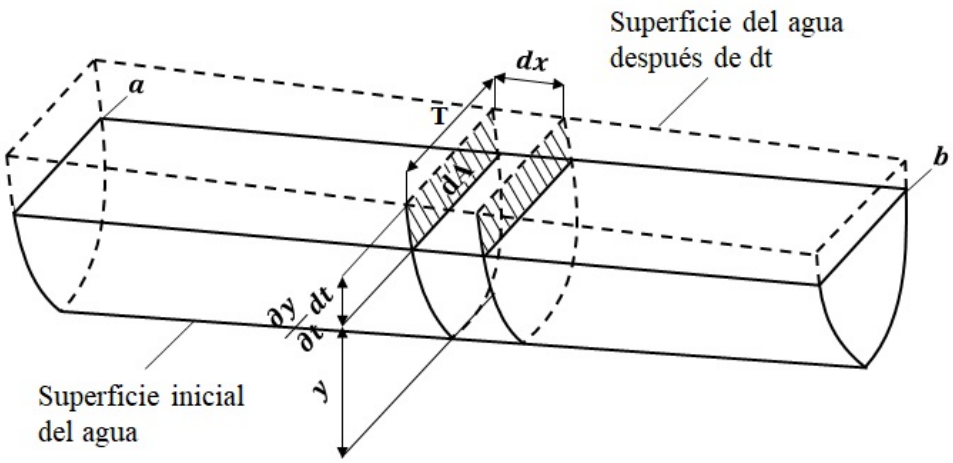


Figura 1.3: Flujo gradualmente variado no permanente y acercamiento infinitesimal a sección de canal

$$\begin{array}{ccccccc} \text{Tasa de cambio} & & \text{Tasa a la que} & & \text{Tasa a la que} & & \text{Masa que} \\ \text{de masa} & = & \text{la masa} & - & \text{la masa} & + & \text{viene de los} \\ \text{respecto al tiempo} & & \text{entra en } a & & \text{sale en } b & & \text{exteriores} \end{array} \tag{1.11}$$

Sin embargo, considerando una densidad constante del agua, se puede analizar el flujo en términos de volumen, ya que $m = \rho V$, donde m representa masa, ρ la densidad y V el volumen, todo en referencia al agua. Como la densidad es constante y no nula, se simplifica en la conservación de masa y se puede expresar de la siguiente manera.

$$\begin{array}{ccccccc} \text{Tasa de cambio} & & \text{Tasa a la que} & & \text{Tasa a la que} & & \text{Volumen que} \\ \text{de volumen} & = & \text{el volumen} & - & \text{el volumen} & + & \text{viene de los} \\ \text{respecto al tiempo} & & \text{entra en } a & & \text{sale en } b & & \text{exteriores} \end{array} \tag{1.12}$$

En este caso, no se considerará que hay volúmenes o masas de agua externas, es decir, no hay entrada o salida desde o hacia otros canales. La variabilidad del volumen en consideración del tiempo se le conoce como caudal y será representado como Q. La ley de la conservación de masa se puede expresar como

$$\frac{\partial V}{\partial t} = Q_a - Q_b \tag{1.13}$$

En este punto es necesario definir la integral. Para este se requiere dos conceptos.

Definición 1.2.1. Sean $a, b \in \mathbb{R}$, tal que $a < b$. Una **partición** en $[a, b]$ es una colección finita de puntos de $[a, b]$ donde uno es a y otro es b.

Definición 1.2.2. Suponiendo que f es acotada sobre $[a, b]$, tal que $a, b \in \mathbb{R}$, es decir si $\forall x \in [a, b]$, $|f(x)| \leq C$. Y $P = \{t_1, t_2, \ldots, t_n\}$ es una partición de $[a, b]$ tal que $n \in \mathbb{N}$. Sea

$$m_i = \text{ínf}\{f(x) : t_{i-1} \leq x \leq t_i\} \tag{1.14}$$

$$M_i = \sup\{f(x) : t_{i-1} \leq x \leq t_i\} \tag{1.15}$$

La **suma inferior** L de f para P, se define poniendo

$$L(f, P) = \sum_{i=1}^{n} m_i(t_i - t_{i-1}) \tag{1.16}$$

La **suma superior** U de f para P, se define poniendo

$$U(f, P) = \sum_{i=1}^{n} M_i(t_i - t_{i-1}) \tag{1.17}$$

Definición 1.2.3. Una función f acotada sobre $[a, b]$ es **integrable** sobre $[a, b]$ si

$$\sup\{L(f, P) | P \text{ es una partición de } [a, b]\}$$

Es igual a

$$\text{ínf}\{U(f, P) | P \text{ es una partición de } [a, b]\}$$

Esta igualdad se conoce como **integral** y se denota por

$$\int_a^b f \tag{1.18}$$

Aunque esta es la definición de integral, en términos de aplicaciones es muy útil verla desde la perspectiva de Suma de Riemann.

Definición 1.2.4. Una función $f : \mathbb{R}^n \to \mathbb{R}$ es continua en $y \in \mathbb{R}^n$ si

$$\lim_{x \to y} f(x) = f(y) \tag{1.19}$$

Teorema 1. Supóngase que f es continua en $[a, b]$. Entonces para todo $\varepsilon > 0$ existe algún $\delta > 0$ tal que, si $P = \{t_0, \ldots, t_n\}$ es una partición cualquiera de $[a, b]$ con todas las longitudes $t_i - t_{i-1} < \delta$, entonces,

$$\left| \sum_{i=1}^{n} f(x_i)(t_i - t_{i-1}) - \int_a^b f(x)dx \right| < \varepsilon \tag{1.20}$$

Para $x_i \in [t_{i-1}, t_i]$

Regresando al canal. La parte izquierda de 1.13 se puede expresar en función del ancho superficial T, que es el ancho del canal en la superficie libre, y y que en este caso es la profundidad de flujo, tal como se representa en la Figura 1.3. De esta manera se puede expresar el cambio del volumen respecto al tiempo como el aporte de el cambio de la profundidad de flujo en todo el canal con la idea de que todos los aportes se ven reflejados con la integral.

$$\frac{\partial V}{\partial t} = \int_a^b T \frac{\partial y}{\partial t} dx \tag{1.21}$$

Por otro lado, la parte derecha de la ecuación 1.13 al asumir la continuidad y diferenciabilidad

del caudal y buenas propiedades de la derivada, se puede expresar como

$$Q_a - Q_b = -\int_a^b \frac{\partial Q}{\partial x} dx \tag{1.22}$$

Esto se da debido al segundo teorema fundamental del cálculo

Teorema 2. Si f es integrable sobre $[a, b]$ y f=g' para alguna función g entonces

$$\int_a^b f = g(b) - g(a) \tag{1.23}$$

Al combinar las ecuaciones 1.21 y 1.25 se obtiene

$$\int_a^b T\frac{\partial y}{\partial t} dx + \int_a^b \frac{\partial Q}{\partial x} dx = 0$$

$$\int_a^b \left(T\frac{\partial y}{\partial t} + \frac{\partial Q}{\partial x} \right) dx = 0$$

$$\frac{\partial Q}{\partial x} + T\frac{\partial y}{\partial t} = 0 \tag{1.24}$$

Debido a que la expresión de la integral es independiente de los límites $x = a$ y $x = b$ que se escojan, se puede concluir que la expresión dentro de la integral es igual a cero. Por otra parte, el caudal se puede definir como

$$Q = Au \tag{1.25}$$

Donde A representa el área transversal en una sección de canal determinada, también conocida como área mojada y u es la velocidad a la que fluye el agua. Ahora se tiene que

Teorema 3. Sean f y g derivables en $x \in \mathbb{R}^{\times}$, entonces fg también es son derivables en x y

$$(fg)'(x) = f'(x)g(x) + f(x)g'(x) \tag{1.26}$$

Al usar la expresión en 1.25 y la regla de derivación del producto en 1.24 se obtiene

$$u\frac{\partial A}{\partial x} + A\frac{\partial u}{\partial x} + T\frac{\partial y}{\partial t} = 0$$

En este punto, se puede definir una nueva variable conocida como la profundidad hidráulica $D \in \mathbb{R}$, cuya funcionalidad se basa en que tiene es una expresión conocida que depende de la geometría del canal.

$$D = \frac{A}{T} \tag{1.27}$$

Finalmente, se puede apreciar que el término $\frac{\partial A}{\partial x}$ depende únicamente de la variación de la profundidad de flujo ya que el ancho de canal es constante en un pedazo infinitesimal. La ecuación resultante se conoce como Ecuación de Continuidad para Flujo Gradualmente Variado no Permanente para Canal Abierto.

$$uT\frac{\partial y}{\partial x} + DT\frac{\partial u}{\partial x} + T\frac{\partial y}{\partial t} = 0$$

$$u\frac{\partial y}{\partial x} + D\frac{\partial u}{\partial x} + \frac{\partial y}{\partial t} = 0 \tag{1.28}$$

De esta manera se puede obtener una Ecuación de Continuidad dependiente de la geometría del canal. En el caso de un canal rectangular, se tiene que el diámetro hidraúlico es igual a la profundidad de flujo

$$u\frac{\partial y}{\partial x} + y\frac{\partial u}{\partial x} + \frac{\partial y}{\partial t} = 0 \tag{1.29}$$

En un canal triangular, el diámetro hidraúlico es igual a la mitad de la profundidad de flujo.

$$u\frac{\partial y}{\partial x} + \frac{y}{2}\frac{\partial u}{\partial x} + \frac{\partial y}{\partial t} = 0 \tag{1.30}$$

Por último, en un canal trapezoidal con base menor/inferior b tiene un diámetro hidraúlico $D = \frac{(b+T)y}{2T}$ por lo que la ecuación de continuidad resultante sería

$$u\frac{\partial y}{\partial x} + y\left(\frac{b+T}{2T}\right)\frac{\partial u}{\partial x} + \frac{\partial y}{\partial t} = 0 \tag{1.31}$$

1.2.3. Ecuación Dinámica para Flujo

La segunda Ecuación de Saint-Venant es conocida como la Ecuación Dinámica para Flujo. Esta es derivada de la conservación de energía, y una gran ventaja es que el comportamiento de un canal es similar al de una tubería en este aspecto tal como se resalta en la Figura 1.4.

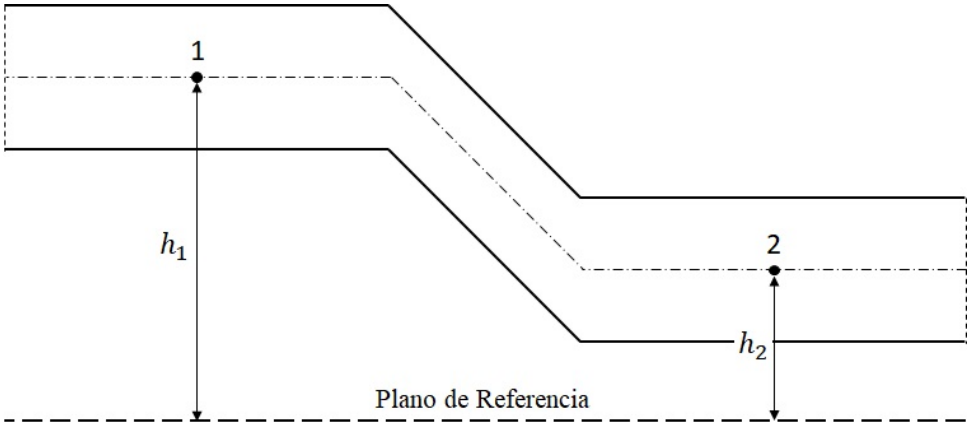


Figura 1.4: Conservación de Energía en Tubería

Si es que no se toman en cuenta fuerzas que disipen energía, entonces el Principio o Ecuación de Bernoulli indica que entre dos puntos (Figura 1.4) de una tubería se conserva la energía mecánica. Es decir, se conserva la energía de flujo, energía cinética y energía potencial gravitacional.

$$P_1 V + \frac{1}{2} m u_1^2 + m g h_1 = P_2 V + \frac{1}{2} m u_2^2 + m g h_2 \tag{1.32}$$

Cada subíndice especifica el punto de la tubería en el que la variable se mide. P representa la presión, V el volumen de agua, m la masa de agua, u la velocidad de flujo y h la altura del eje de la tubería tomado desde un plano de referencia. Aunque esta expresión expresa concretamente la conservación de energía, no suele ser muy útil ya que especificar el volumen o masa de agua en un fluido puede ser engorroso. Es por esto que esta ecuación suele estar expresado en términos de presiones o en términos de altura, para usar la densidad ρ que es mucho más práctica.

$$P_1 + \frac{1}{2} \rho u_1^2 + \rho g h_1 = P_2 + \frac{1}{2} \rho u_2^2 + \rho g h_2 \tag{1.33}$$

$$\frac{P_1}{\rho g} + \frac{u_1^2}{2g} + h_1 = \frac{P_2}{\rho g} + \frac{u_2^2}{2g} + h_2 \tag{1.34}$$

Dentro de un canal abierto, la presión se mantiene constante ya que el sistema no es cerrado, es decir, el flujo en el canal se encuentra a presión atmosférica y no hay presión manométrica. Por esta razón, los términos de energía de flujo en dos puntos distintos en un canal será la misma.

En este estudio, se considerará también un término de pérdida de energía asociado a las fuerzas de fricción que ocurren dentro del canal. Además, debido a que el flujo a analizar es gradualmente variado y no permanente, se incorpora un término relacionado al trabajo producido por la fuerza que efectúa el cambio de caudal dentro del canal. Estos términos se expresan de forma diferencial y es por esto que convenientemente se analiza la conservación de energía dentro de un espacio infinitesimal como generalización para cualquier par de puntos dentro del canal tal como se observa en la Figura 1.5.

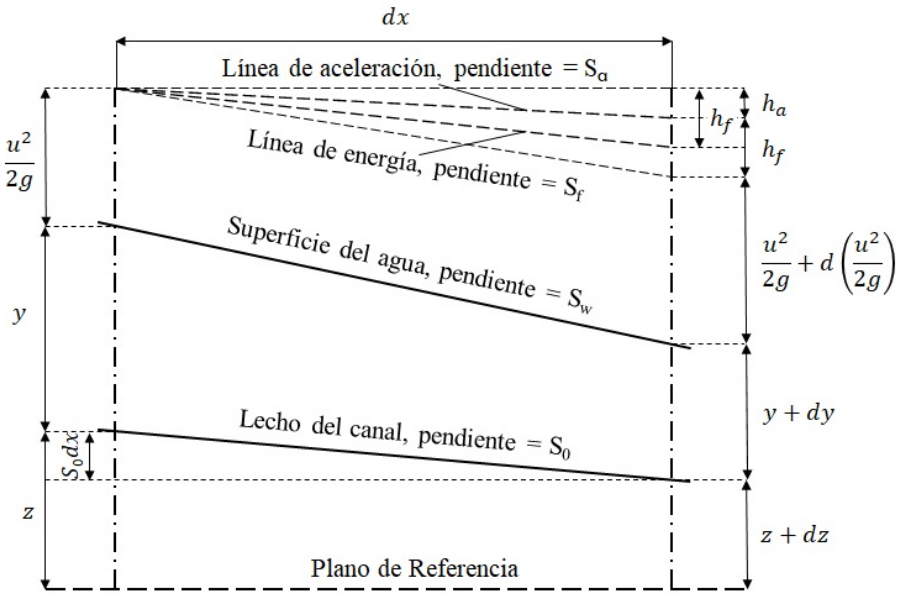


Figura 1.5: Conservación de Energía Expresado en Términos de Alturas en Flujo Gradualmente Variado

Como se puede observar una de las ventajas de expresar los términos de energía en base a sus alturas, es que se puede relacionar el incremento infinitesimal con pendientes de líneas dibujadas entre alturas referentes a términos de energía equivalentes. De forma ventajosa, las líneas asociadas a los términos de energía potencial gravitacional son precisamente la superficie del agua (con pendiente S_w) y el lecho de canal (con pendiente S_0). Por otro lado, la altura

asociada a la pérdida de energía por fricción se puede escribir como

$$h_f = S_f dx \tag{1.35}$$

El término asociado al trabajo se puede obtener como el producto punto de la fuerza ($F = ma$) asociada con la aceleración del flujo ($a = \frac{\partial u}{\partial t}$) con el diferencial espacial (Fdx). Como se mencionó en la Ecuación de Bernoulli, cuando se habla de flujos es útil trabajar con la densidad más que con la masa, así que el término de trabajo por volumen puede ser expresado como $\rho\frac{\partial u}{\partial t}dx$. En términos de altura o cabeza, se tiene

$$h_a = \frac{1}{g}\frac{\partial u}{\partial t}dx \tag{1.36}$$

De esta manera la conservación de energía resulta en

$$z + y + \frac{u^2}{2g} = z + dz + y + dy + \frac{u^2}{2g} + d\left(\frac{u^2}{2g}\right) + h_a + h_f \tag{1.37}$$

Al desarrollar los términos, y observar que $-\frac{\partial z}{\partial x} = S_0$

$$d\left(z + y + \frac{u^2}{2g}\right) = -\frac{1}{g}\frac{\partial u}{\partial t}dx - S_f dx$$

$$\approx \frac{\partial z}{\partial x} + \frac{\partial y}{\partial x} + \frac{1}{2g}\frac{\partial u^2}{\partial x} + \frac{1}{g}\frac{\partial u}{\partial t} + S_f = 0$$

Finalmente la Ecuación Dinámica de Flujo se expresa como

$$g\frac{\partial y}{\partial x} + u\frac{\partial u}{\partial x} + \frac{\partial u}{\partial t} + g(S_f - S_0) = 0 \tag{1.38}$$

1.2.4. Planteamiento del Problema

Lo que hace tan atractivo a la investigación respecto a las Ecuaciones de Saint-Venant es el hecho de que no tienen una solución determinística (analítica). Es decir, no se conoce expresiones de la profundidad de flujo $y(x,t)$ y de la velocidad $u(x,t)$ que resuelvan la ecuación.

El problema que se va a resolver durante este trabajo va a ser encontrar una solución aproximada de las Ecuaciones de Saint-Venant como un problema de valor inicial aplicadas a un caso específico de flujo de agua en un canal rectangular sujetas a condiciones iniciales, tanto en la profundidad de flujo como en la velocidad. La solución se dará mediante una aproximación con redes neuronales. El caso específico y las condiciones iniciales se escogieron para simular el comportamiento de cómo una acumulación de agua en un canal se deja fluir nuevamente y de forma rápida. Es decir, si por ejemplo se quitara una obstrucción dentro del canal.

$$\begin{cases} u\frac{\partial y}{\partial x} + y\frac{\partial u}{\partial x} + \frac{\partial y}{\partial t} = 0 \\ g\frac{\partial y}{\partial x} + u\frac{\partial u}{\partial x} + \frac{\partial u}{\partial t} + g(S_f - S_0) = 0 \\ y(x,0) = \begin{cases} -\frac{1}{25}x^2 + 2 & \text{si } 0 \leq x < 5 \\ 1 & x \geq 5 \end{cases} \\ u(x,0) = 0 \end{cases} \tag{1.39}$$

Aunque ya existen métodos numéricos que son capaces de aproximar una solución al sistema de ecuaciones de forma eficiente, las redes neuronales permiten métodos de aproximación que dan como solución a una función diferenciable, métodos que pueden ser fácilmente adaptables para problemas en los que incluso se tenga dificultades con los métodos numéricos tradicionales, y además abre posibilidades para futuras investigaciones de cómo las redes podrían extraer características de las aproximaciones. Las ecuaciones de Saint Venant son de las tantas ecuaciones diferenciales aplicadas en mecánica de fluidos que no tiene solución analítica por lo que realizar este trabajo podría abrir las puertas a estudios hacia las demás ecuaciones sin solución. Además, hay pocos estudios respecto a encontrar soluciones aproximadas de estas ecuaciones con la ayuda de redes neuronales [4], [14], [26] por lo que hay mucho que se puede seguir profundizando.

1.3. Redes Neuronales

Una red neuronal, correctamente descrita como red neuronal artificial o red neuronal artificialmente generada (ANN) [28], es un modelo de procesamiento de datos basado en el funciona-

miento de las neuronas biológicas que se encuentran en el córtex cerebral de los mamíferos. La neurona consta del cuerpo o soma, las dendritas o entradas y el axón o salida (Figura 1.6). El axón llega a terminales presinápticas, es decir, a la zona previa a la sinapsis o área de contacto entre dos neuronas. La neurona que manda la señal se llama célula presináptica y la neurona que recibe se llama célula postsináptica [11], [38].

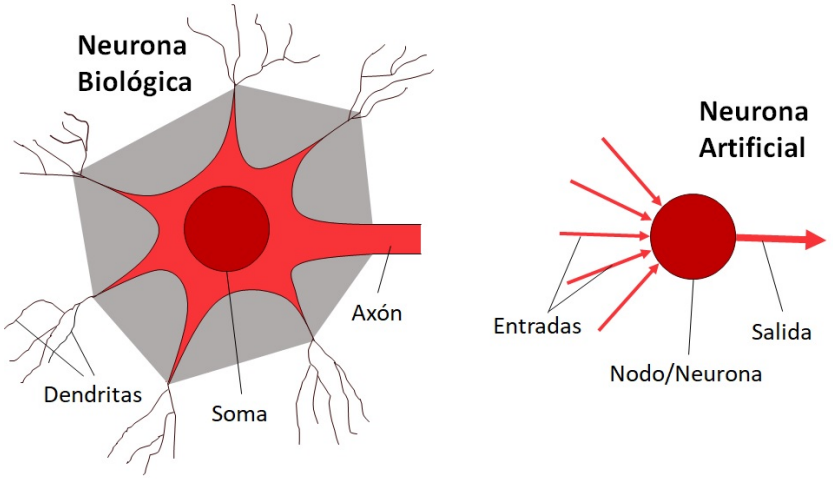


Figura 1.6: Comparación entre neurona biológica y neurona artificial

Las redes neuronales artificiales son una estructura de procesamiento que trata de simular la interacción de neuronas que se encuentran interconectadas mediante sinapsis y que transmiten información mediante señales eléctricas, para poder reproducir sistemas que puedan aprender (Figura 1.7) [11]. Fueron implementadas por primera vez en la década de los cuarenta, y fueron evolucionando junto con nuevos descubrimientos del funcionamiento de las células nerviosas del cerebro. Aunque los estudios y sus aplicaciones fueron prometedores desde su creación, el desarrollo de las redes siempre sufrió de interrupciones [28]. Sin embargo, el uso de esta estructura volvió a tener un auge debido a que se han convertido en una parte fundamental del desarrollo de Inteligencia Artificial.

La arquitectura de estas redes consta de las neuronas como elementos o unidades de procesamiento y el sistema completo está constituido por capas de neuronas conectadas entre ellas como se puede ver en la Figura 1.7. La red recibe información o señales en una capa de entrada que es transmitida y procesada hacia la última capa donde se ejecutan los valores de salida. Entre estas

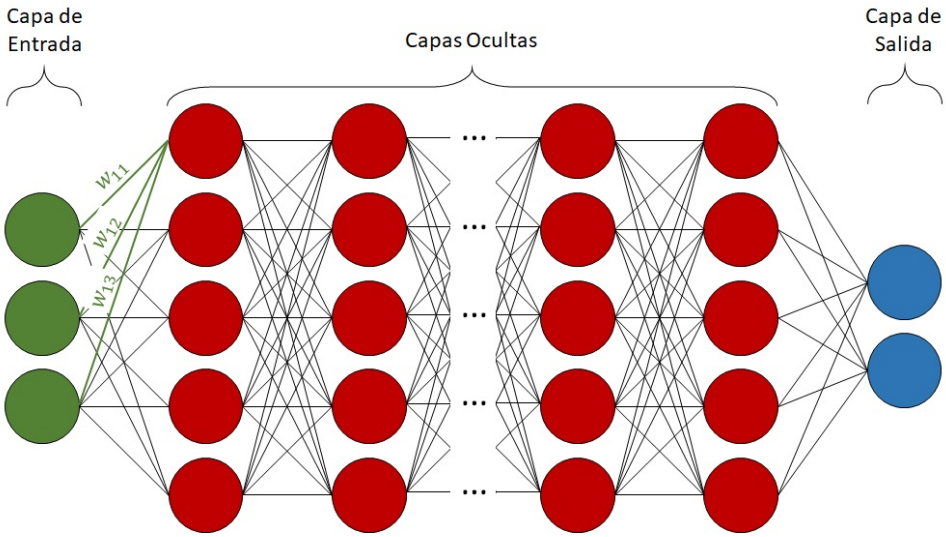


Figura 1.7: Arquitectura red neuronal

capas puede o no haber capas intermedias conocidas como capas ocultas, aunque lo que da tanta utilidad a este paradigma es el hecho de que haya estas capas. Las conexiones se generan entre capas y cada una lleva consigo un valor conocido como peso que multiplica las señales o valores de salida de las neuronas de las capas anteriores. También, se puede añadir valores de sesgos dentro de cada neurona que se suman a dichos productos. De forma similar a las neuronas biológicas, las neuronas artificiales pueden asociarse a un valor de activación para poder transmitir información. Si es que la señal supera este valor de activación, entonces la neurona se activa y transmite la información [11]. Una forma práctica de replicar esto artificialmente es con una función de activación que precisamente transforma el valor de las señales dependiendo de qué tan relevante es en la activación de la neurona. El funcionamiento de una neurona artificial se puede ver en la Figura 1.8.

Dos tipos de arquitectura son comúnmente usadas, la FeedForward y la FeedBack. En una red FeedForward toda la información fluye únicamente en la dirección que viene desde la capa de entrada hacia la capa de salida. Un ejemplo de este tipo de redes es la Red Neuronal en Cascada que transmite toda la información hacia delante, y lo que le caracteriza es que la capa de entrada tiene uno o varios nodos que, de forma conjunta, tienen conexiones con absolutamente todos los nodos de las capas siguientes [28] como se puede ver en la Figura 1.9. Por otro lado, en una red

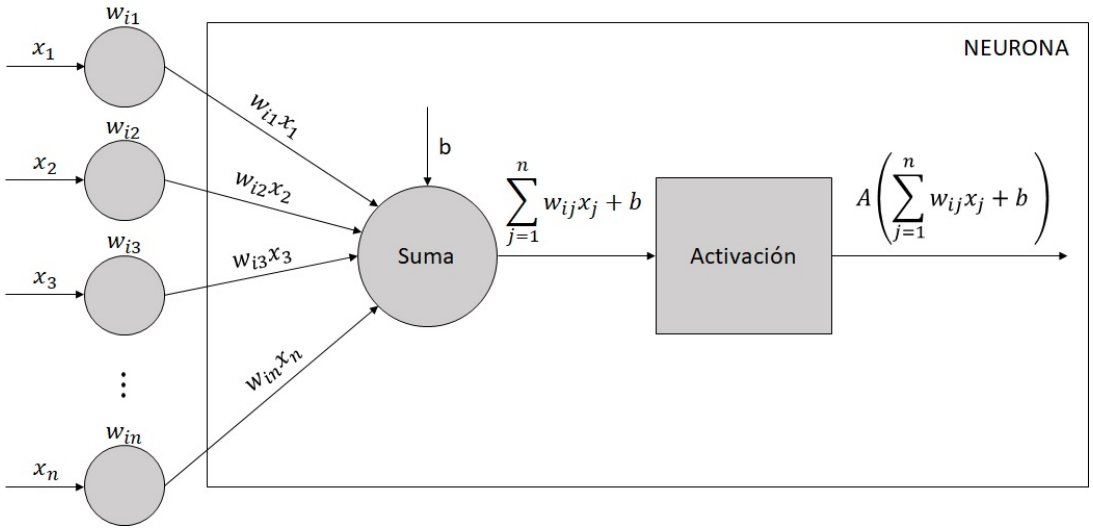


Figura 1.8: Funcionamiento de una neurona

FeedBack se pueden tener conexiones tipo bucle o de realimentación hacia neuronas de capas anteriores. Un ejemplo se puede observaar en la Figura 1.10. Este tipo de conexiones provocan que el sistema pueda tener memoria y claramente influyen en el proceso de entrenamiento de la red [10].

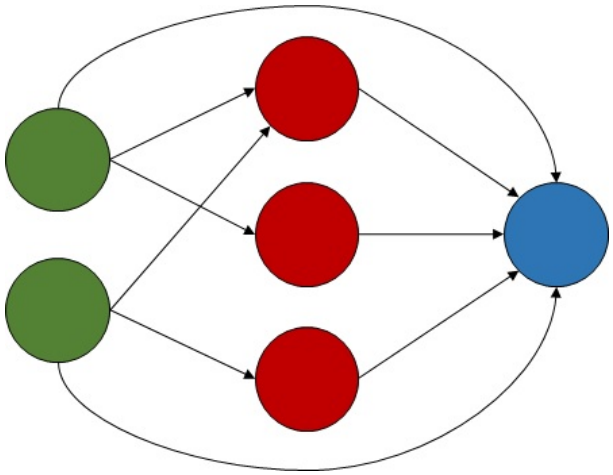


Figura 1.9: Red neuronal tipo cascada

Como ya se mencionó, lo que hace especiales a las redes neuronales artificiales es que tienen la capacidad de aprender y auto-organizarse. Generalmente, para encontrar la solución de un problema (computacional) se busca un conjunto de reglas o instrucciones que se aplique a valores de entrada que llevan al hallazgo de dicha solución. Por otro lado, las redes tratan de encontrar el conjunto de instrucciones necesarias y acoplarlas en un modelo que trata de ajustarse dentro

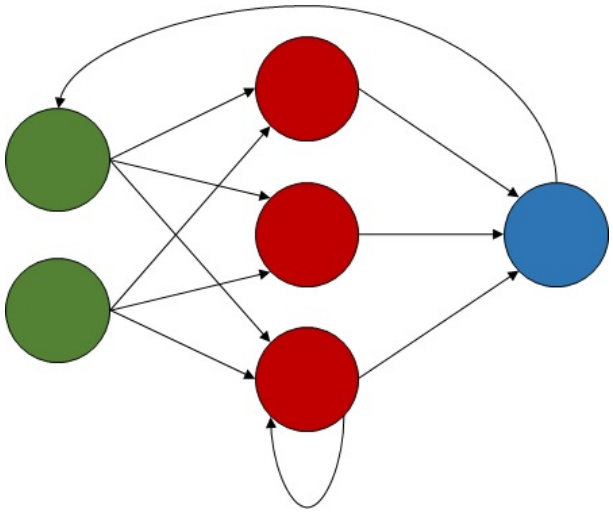


Figura 1.10: Red neuronal tipo Feed Back

de la estructura, a partir de un conjunto de entradas y salidas, que pueden volver a ser usadas para encontrar soluciones a otros problemas [38].

Las formas en las que aprende una red se pueden resumir en cuatro tipos: supervisado, no supervisado, por refuerzo y por competición. Algunos autores suelen considerar a los dos últimos como tipos de aprendizajes supervisados, pero aquí se ha decidido darles un énfasis particular.

El aprendizaje supervisado asume que hay un tipo de conexión entre la entrada y la salida de la red, y que la información de entrenamiento que entra a la red puede ajustar los parámetros del modelo mediante una señal de enseñanza dada esta conexión. El aprendizaje no-supervisado busca alguna particularidad o propiedad de la red, elemento o característica asociada, que permita ajustar los parámetros una vez se condicione a la salida a cumplir con dicha propiedad. Es decir, en este caso no se da un ejemplo de entrenamiento específico a la salida del sistema para enseñar al modelo [35].

Por otra parte, el aprendizaje por refuerzo es similar al supervisado, con la diferencia en que al indicar la tolerancia con la que se puede aceptar una salida comparada con la salida

deseada, la señal de error es solo binaria (si se decide clasificar al aprendizaje supervisado en las características de la señal de salida, este aprendizaje solo sería un tipo de aprendizaje supervisado). Si la señal de salida no se encuentra dentro de la tolerancia, la señal de error se indica solo como falsa lo que avisa al sistema que tiene que reajustar sus parámetros.

Por último, el aprendizaje por competición consiste en que la red tenga varias neuronas en la capa de salida. Los valores de salida de cada una de estas neuronas son comparados bajo algún criterio y se escoge a la neurona dominante como la que tiene el valor más cercano al esperado [38]. De esta forma, el aprendizaje se da por selección y se entrena con varias salidas a la vez. En cualquier caso, el ajuste de los pesos generalmente se da por un algoritmo de retropropagación que usa una función de coste o pérdida que cuantifica el error de la predicción y un paso de optimización que actualiza los pesos [35].

Todas estas características permiten definir alternativamente a una ANN como una estructura de procesamiento de información paralelamente distribuida en la forma de un grafo dirigido donde [38]:

1. Los nodos se conocen como elementos de procesamiento
2. Los enlaces entre nodos se conocen como conexiones y son unidireccionales (Enfocado más en FeedForward).
3. Cada elemento de procesamiento puede recibir cualquier número de conexiones
4. Cada elemento de procesamiento puede tener cualquier número de conexiones salientes, siempre y cuando la salida sea la misma.
5. Los elementos de procesamiento pueden tener memoria local
6. Cada elemento de procesamiento puede tener una función de transferencia que usa la memoria local para producir una única señal para las conexiones salientes.

Esta estructura fue la base para múltiples estudios de las redes neuronales multicapas, que son

redes con más de una capa oculta, y llevó al desarrollo de lo que se conoce ahora como Deep Learning. El Deep Learning es un tipo de Machine Learning que es parte fundamental para lo que se espera de una máquina para que sea inteligente. La idea es que cuando se usa más capas en una red neuronal (más profundidad), más características asociadas a cada neurona puede aprender la red [10]. De esta manera, las redes neuronales han pasado a representar una categoría de algoritmos dentro del Machine Learning donde al entrenarlas se consideran expertas en la categoría de información que analizan [21].

Su peculiar forma de aprendizaje hace que las ANN sean herramientas eficaces al momento de encontrar patrones o descubrir tendencias que son difíciles de entender para el humano [28] y sus cálculos pueden ser ejecutados en paralelo [10]. Además, una de las características más interesantes y una de las ventajas de tener un sistema de tipo red es que si una unidad de procesamiento de las redes neuronales falla, las otras neuronas pueden suplir su ausencia e incluso se ha visto casos en los que pueden reactivarla [21]. Esto hace que el modelo de la red sea bastante robusto a posibles disturbios.

Estas ventajas han llevado a las ANN a ser aplicadas, entre los temas más profundizados, en procesamiento natural del lenguaje y el habla [5], [25], [34] además de reconocimiento, clasificación y transformación de imágenes [18], [37]. Otro tema concurrido es la seguridad digital y ciberseguridad que se encuentra en auge de investigación [1], [22], [32], [33] y tomará aun más relevancia con el desarrollo de la computación cuántica. La habilidad para detectar patrones ha sido clave para fomentar su uso en algoritmos y software para detección y diagnóstico de enfermedades mediante imágenes [2], [23], [24], [31]. También ha podido ser implementado en áreas como ingenierías, tecnología [12], [19], [20] y biología [9].

Este trabajo se centrará en usar redes neuronales para poder obtener soluciones en ecuaciones diferenciales. Aunque las redes neuronales tienen un fundamento matemático, es interesante ver cómo se aplican dentro de la misma área. El Machine Learning, como herramienta dentro de la Matemática, no se ha explotado lo suficiente comparado con otros campos aunque sí se han

hecho investigaciones importantes[3], [6], [7]. Este estudio tratará de profundizar un poco más en su utilidad.

1.3.1. Optimización

Como se mencionó en la anterior sección, lo que hace especiales a las redes neuronales es que encuentran las instrucciones para que la entrada se convierta en salida. Es decir, que aunque ya se ha visto la estructura de la red, realmente lo que se requiere es encontrar los parámetros de los pesos y los sesgos que provoquen que la entrada se haga realmente la salida.

Para esto, lo que se tiene que hacer es entrenar a la red. Por lo que es necesario definir una expresión que precisamente le diga a la ANN si es que su modelo realmente está prediciendo los valores. Esta expresión es la función de pérdida. Aunque no hay una sola manera de representarla, se puede decir que esta función es una representación del error que tiene la salida. Por ejemplo, si el aprendizaje es supervisado se puede tener una función de pérdida que compare el valor de salida de la red ,que depende de los parámetros de la red, con el valor esperado en forma de un error. Entonces la clave está en que mientra más pequeño sea el error, mejor sería el modelo. Así que se tendría que encontrar parámetros que minimicen la función de pérdida.

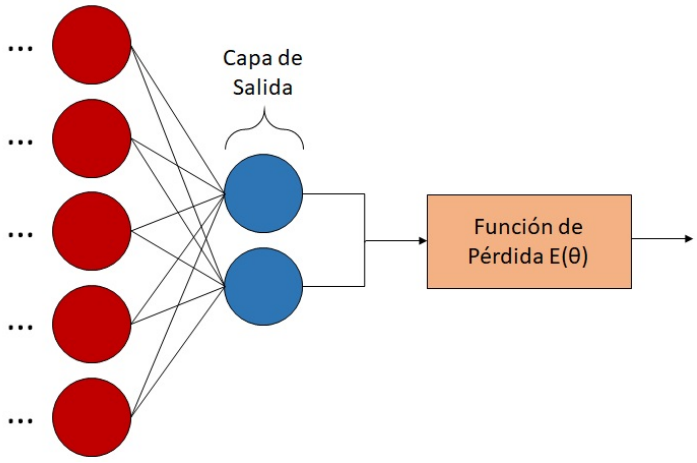


Figura 1.11: Módulo con función de pérdida

Sea $\theta \in \Theta \subseteq \mathbb{R}^d$ un vector que contiene a todos los parámetros de una red y sea $E(\theta) : \Theta \subseteq \mathbb{R}^d \to \mathbb{R}$ la función de pérdida, entonces el tipo de problema descrito anteriormente se puede

ver como

$$\text{Minimizar } E(\theta)$$

$$\text{Dado que } \theta \in \Theta$$

O simplemente

$$\min_{\theta \in \Theta} L(\theta) \tag{1.40}$$

Y para resolver este tipo de problemas existen los algoritmos de optimización que van variando los valores de los parámetros en iteraciones hasta que la imagen de la función llegue a un valor pequeño adecuado. Uno de estos algoritmos es el de Gradiente Descendente.

Definición 1.3.1. Sea $f : \mathbb{R}^n \to \mathbb{R}$. El **gradiente** de f en un punto $x \in \mathbb{R}^n$ definido como $x = (x_1, \ldots, x_n)$ se describe como

$$\nabla f(x) = (f_{x_1}(x), \ldots, f_{x_n}(x)) \tag{1.41}$$

Definición 1.3.2. La **derivada direccional** de una función derivable $f : \mathbb{R}^n \to \mathbb{R}$ en un punto $x \in \mathbb{R}^n$ en la dirección de un vector unitario $u \in \mathbb{R}^n$ es

$$D_u f(x) = \lim_{h \to 0} \frac{f(x + hu) - f(x)}{h} \tag{1.42}$$

El Algoritmo de Gradiente Descendente se basa en los siguientes dos teoremas

Teorema 4. Si $f : \mathbb{R}^n \to \mathbb{R}$ es una función derivable para $x \in \mathbb{R}^n$ y sea $u \in \mathbb{R}^n$ un vector unitario, entonces

$$D_u f(x) = \nabla f(x) \cdot u \tag{1.43}$$

Teorema 5. El valor máximo de la derivada direccional $D_u f(x)$ de una función $f : \mathbb{R}^n \to \mathbb{R}$ dado un vector unitario $u \in \mathbb{R}^n$ es $||\nabla f||$ en dirección del gradiente.

Demostración. Dada la Ecuación 1.43 se puede escribir

$$D_u f(x) = \nabla f(x) \cdot u = ||\nabla f(x)|| cos\theta$$

Donde θ es el ángulo entre $\nabla f(x)$ y u. Claramente, el valor de $D_u f(x)$ se maximiza cuando $\cos\theta = 1$, es decir cuando el vector unitario está en dirección del gradiente. □

El algoritmo del Gradiente Descendente se basa en el hecho de que si se comienza en un punto $\theta_t \in \Theta$ y se quiere buscar el mínimo de la función, en este caso la función de pérdida $E(\theta)$, entonces uno se puede aproximar al mínimo acercándose en la dirección donde f desciende más rápido. Por el Teorema 5 se sabe que es en la dirección contraria al gradiente. Así para todos los parámetros se puede buscar el mínimo iterando

$$\theta_{t+1} = \theta_t - \gamma g_t \tag{1.44}$$

Donde γ se conoce como la tasa de aprendizaje y $g_t = \nabla_\theta E(\theta_t)$.

Aunque este método se usa bastante, presenta ciertos problemas especialmente con las funciones de pérdida de las redes neuronales. Desde el punto de vista de la tasa de aprendizaje, si es que se toma un valor muy grande podría ocurrir que los saltos no lograran estabilizar el método, provocando que cada vez solo se llegue a puntos no lo suficientemente cercanos del mínimo en distintas direcciones. Por otro lado, si se toma un valor muy pequeño podría ocurrir que el algoritmo avance tan lento que el tiempo de convergencia sea alto como se puede ver en la Figura 1.12.

Como era de esperarse, también hay problemas con los valores del gradiente. Si es que el valor del gradiente es muy grande, puede provocar el fenómeno de gradiente explosivo donde la actualización de los parámetros se haría de forma tan agresiva que podrían diverger. Si es que el valor de gradiente disminuye mucho, se provoca el fenómeno de desvanecimiento de gradiente y

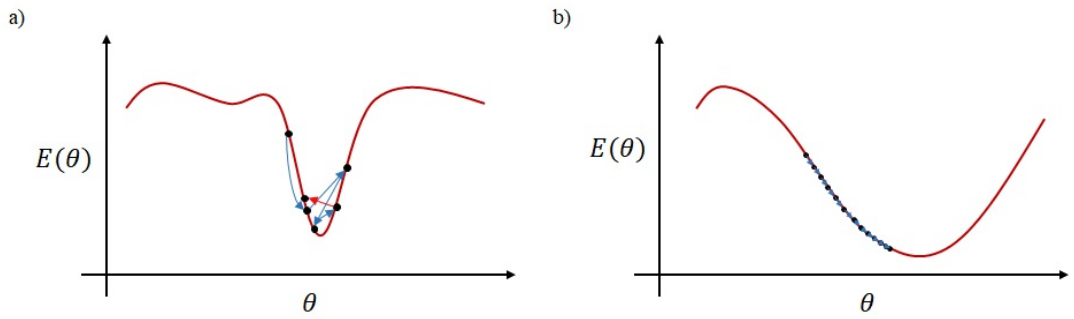


Figura 1.12: a) Divergencia por tasa de aprendizaje alta; b) Tardanza por tasa de aprendizaje baja

el método podría demorarse mucho en llegar al mínimo. Además, algo problemático que puede estar relacionado con el desvanecimiento del gradiente es el hecho de que tal vez el método pueda estancarse en un punto crítico no muy óptimo (donde el gradiente es igual a cero) como un punto silla.

Por estas razones, se han efectuado métodos más eficientes basados en el método de gradiente descendente que puedan tratar mejor estos problemas [29]. El primero es el Algoritmo de Gradiente Descendente con Momentum. Este establece que la actualización de los parámetros se da de la forma

$$\theta_{t+1} = \theta_t - \gamma m_t \tag{1.45}$$

Donde

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{1.46}$$

Y a β_1 se le conoce como la constante de decaimiento. La idea detrás de este método es generar un parámetro m_t que pondera los gradientes de todos los puntos anteriores. De esta forma, al toparse con zonas donde el gradiente disminuye o aumenta mucho su valor, el registro de los gradientes anteriores no permiten que el paso crezca o decrezca demasiado.

Otro algoritmo usado es el AdaGrad, donde la actualización de los parámetros se da por

$$\theta_{t+1} = \theta_t - \gamma_t g_t \tag{1.47}$$

$$\gamma_t = \frac{\eta}{\epsilon + \sqrt{v_t}} \tag{1.48}$$

$$v_t = v_{t-1} + {g_t}^2 \tag{1.49}$$

Donde ϵ es solo una constante que controla la división por cero. En este caso, se tiene una expresión en la que la tasa de aprendizaje ahora se ajusta continuamente. Al igual que en el algoritmo anterior, se tiene un término que acumula gradientes pero ahora en un promedio aritmético. g_t^2 en este caso representa $g_t \cdot g_t$. La idea es que cuando se está en una zona con mucha pendiente, entonces la tasa se hace más pequeña, asegurando que el paso no se haga muy largo. Cuando se está en una zona menos empinada, entonces se tiene una tasa que no decrece demasiado.

De este algoritmo evolucionó otro en el que se consideró hacer una acumulación de gradientes similar al Algoritmo de Gradiente Descendente con Momentum. Este es el Algoritmo RMSprop y se define como

$$\theta_{t+1} = \theta_t - \frac{\eta}{\epsilon + \sqrt{v_t}} g_t \tag{1.50}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2){g_t}^2 \tag{1.51}$$

Finalmente, tanto el Algoritmo de Gradiente Descendente con Momentum y el Algoritmo RMSprop se combinaron para dar paso al Algoritmo ADAM junto con una pequeña modificación. Este se define como

$$\theta_{t+1} = \theta_t - \frac{\gamma \hat{m}_t}{\epsilon + \sqrt{\hat{v}_t}} \tag{1.52}$$

Sin embargo, lo que los autores del algoritmo comentan es que al inicializar el algoritmo con $v_0 = m_0 = 0$, se tiene un sesgo con los parámetros cercanos a cero, lo que genera problemas [16]. Por lo que se hace una modificación

$$\hat{m_t} = \frac{m_t}{1 - {\beta_1}^t} \tag{1.53}$$

$$\hat{v_t} = \frac{v_t}{1 - {\beta_2}^t} \tag{1.54}$$

Que controlan el valor esperado de los valores iniciales de las iteraciones. ADAM presenta las

ventajas para evitar de manera más efectiva problemas respecto a la tasa de aprendizaje y el gradiente. Aunque no es un método infalible, sin duda su aporte ha sido significativo en redes neuronales.

Capítulo 2

Metodología

De forma resumida, primero se va a encontrar expresiones para las pendientes S_f y S_0. Luego, se va a resolver el sistema con un método numérico conocido para tener como referencia la solución aproximada y además, con el análisis de estabilidad del método, poder discretizar el dominio para usarlo en el método de redes neuronales. Finalmente, se analizará a profundidad el método que usa redes neuronales para obtener la solución del sistema.

2.1. Pendientes de Lecho y Energía

Primero, como observación dada en Vente Chow [8] y la razón por la que se pudo obtener la Ecuación Dinámica de Flujo en función de S_0, es que las pendientes de lecho en los canales de forma general, no son muy pronunciadas. De esta manera se puede aproximar la pendiente

$$S_0 \approx \operatorname{sen}\theta \tag{2.1}$$

Donde θ representa el ángulo entre el lecho de canal con un plano horizontal de referencia.

Por otro lado, para obtener la pendiente de energía S_f se tienen que hacer algunas asunciones y

aproximaciones. La primera, es que la pendiente de energía se puede aproximar a la pendiente de energía del flujo si fuera uniforme y permanente. Aunque realmente no se ha encontrado que esto tenga algún fundamento teórico, el pequeño error que se genera de esta asunción compensa el arduo procedimiento que hay que hacer para obtener la pendiente real. Cabe mencionar que este error es mínimo siempre y cuando el flujo ocurra en la dirección de caída de la pendiente.

Como ya se mencionó, un flujo uniforme permanente está caracterizado porque no hay variación de la profundidad de flujo de ningún tipo aunque en la realidad, la profundidad de flujo es sensible a muchas cosas. Para que sea constante se suele asumir que el área mojada, la distribución de velocidad y el caudal de cada sección se mantienen constantes. Esto se puede traducir a que la velocidad se mantiene constante y que las pendientes tanto de superficie, como de lecho y de energía son iguales. Ya que la energía cinética debería crecer a medida que se pierde energía potencial, y esto no ocurre en estas condiciones ideales, entonces el cambio faltante debe reflejarse en la pendiente de energía. Esto provoca que todas las pendientes del sistema sean iguales. En este caso esa pendiente se llamará S_f.

La ventaja de asumir que la pendiente S_f se puede aproximar a la pendiente de energía de flujo uniforme permanente es que empíricamente se encontró una relación directa entra la velocidad de flujo y la pendiente, conocida como ecuación de flujo uniforme.

$$u \propto R^x S_f^y \tag{2.2}$$

Donde $\propto$ es el símbolo de proporcionalidad, x y y son números reales y R se conoce como el radio hidráulico y está definido como

$$R = \frac{A}{P} \tag{2.3}$$

Donde P es el perímetro mojado que se define como el perímetro de contacto del agua con las paredes del canal.

Esta relación menciona dos cosas. La primera que es intuitiva, es que la velocidad depende de

que tan pronunciada es la pendiente del canal. Por otro lado, nos indica que el área y el perímetro mojado son fundamentales. Si se mantiene el área mojada y se aumenta el perímetro mojado, se aumenta la interacción entre el flujo y las paredes, lo que se traduce en más pérdida de energía. De igual forma, si se mantiene el perímetro mojado y aumenta el área mojada, se tiene que se favorece el flujo de agua con el mismo contacto del flujo con el canal. Esta relación nos permite encontrar una ecuación

$$u = KR^x S_f^y \tag{2.4}$$

Donde K representa la constante de proporcionalidad. Tanto K, como x y y tienen valores que dependerán del sistema de flujo. Sin embargo, se han encontrado ecuaciones que suelen dar buenos resultados de manera general. En este trabajo usaremos la Ecuación de Manning

$$u = \frac{1}{c} R^{2/3} S_f^{1/2} \tag{2.5}$$

Donde c es el coeficiente de Manning y está relacionado con la fricción. Es decir, que es un valor que depende de los obstáculos y del material del canal.

De esta forma, la pendiente se representa como

$$S_f = c^2 u^2 R^{-4/3} \tag{2.6}$$

Finalmente, el problema a tratar es en un canal rectangular. Usando la notación de la sección 1.3.2, se tiene que

$$R = \frac{Ty}{T + 2y} \tag{2.7}$$

y la expresión de la pendiente y del sistema serían

$$S_f = c^2 u^2 \left(\frac{Ty}{T + 2y} \right)^{-4/3} \tag{2.8}$$

$$\begin{cases} u\frac{\partial y}{\partial x} + y\frac{\partial u}{\partial x} + \frac{\partial y}{\partial t} = 0 \\ g\frac{\partial y}{\partial x} + u\frac{\partial u}{\partial x} + \frac{\partial u}{\partial t} + g\left(man^2u^2\left(\frac{Ty}{T+2y}\right)^{-4/3} - \operatorname{sen}\theta\right) = 0 \\ y(x,0) = \begin{cases} -\frac{1}{25}x^2 + 2 & \text{si } 0 \leq x < 5 \\ 1 & \text{si } x \geq 5 \end{cases} \\ u(x,0) = 0 \end{cases} \tag{2.9}$$

2.2. Método de Lax-Friedrichs

Definición 2.2.1. Sea $U : \mathbb{R}^n \times (0,\infty) \to \mathbb{R}^m$ y sea $A_i(x,t)$ una matriz $m \times m$ para $i = 1, \ldots, n$ y sea $F : \mathbb{R}^n \times (0,\infty) \to \mathbb{R}^m$. Se dice que el sistema

$$U_t + \sum_{i=1}^{n} A_i(x,t)U_{x_i} = F(x,t)$$

es un **sistema hiperbólico de ecuaciones de primer orden** si es que la matriz

$$A(x,t;\xi) \equiv \sum_{i=1}^{n} A_i(x,t)\xi_i$$

tiene valores propios reales para todo $\xi_i \in \mathbb{R}$

Dada esta definición, se puede ver que el sistema de Ecuaciones de Saint-Venant es un sistema hiperbólico de ecuaciones de primer orden [13]. El sistema se puede reescribir como

$$\begin{pmatrix} u_t \\ y_t \end{pmatrix} + \begin{pmatrix} y(x,t) & u(x,t) \\ u(x,t) & g \end{pmatrix} \begin{pmatrix} u_x \\ y_x \end{pmatrix} = \begin{pmatrix} 0 \\ g(S_0 - S_f(x,t)) \end{pmatrix} \tag{2.10}$$

Donde

$$A = \begin{pmatrix} \xi y & \xi u \\ \xi u & \xi g \end{pmatrix}$$

tiene los valores propios

$$\lambda = \xi u \pm \sqrt{\xi^2 yg}$$

Como la profundidad de flujo es un valor positivo, los valores propios son reales.

Un método numérico comúnmente usado para aproximar una solución en PDEs hiperbólicas o en sistemas hiperbólicos de ecuaciones es el método numérico de Lax-Friedrichs. El método consiste en primero discretizar el dominio tanto en la parte espacial como temporal [27]

$$\begin{aligned} x_j &= x_0 + j\Delta x \qquad & j &= 0, 1, \ldots, J \\ t_n &= t_0 + n\Delta t \qquad & n &= 0, 1, \ldots, N \end{aligned}$$

Luego, se usa aproximaciones para las derivadas basadas en la expansión de Taylor de una función llamadas diferencias finitas [15] que son necesarias definir.

Definición 2.2.2. Sea $f : \mathbb{R} \to \mathbb{R}$ una función suave (infinitamente diferenciable). Entonces $f(x)$ puede ser representada en su **serie de Taylor** alrededor de un punto $x_0 \in \mathbb{R}$ si f y sus derivadas son continuas alrededor ($\Delta x = x - x_0$) de este punto como

$$f(x) = f(x_0) + \cdots + \frac{(\Delta x)^n}{n!} f^{(n)}(x_0) + \ldots \tag{2.11}$$

o puede ser representada en su **series de Taylor con residuo** de la forma

$$f(x) = f(x_0) + \cdots + \frac{(\Delta x)^n}{n!} f^{(n)}(x_0) + R_n \tag{2.12}$$

$$R_n = \frac{(\Delta x)^{n+1}}{(n+1)!} f^{(n+1)}(\xi_{n+1}) \quad \text{donde } x_0 < \xi_{n+1} < x = x_0 + \Delta x \tag{2.13}$$

Con la expansión de Taylor se puede ver que para $x_0 \in \mathbb{R}$

$$\frac{df}{dx}(x_0) = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} - \frac{\Delta x}{2}\frac{d^2 f}{dx^2}(\xi_2) \tag{2.14}$$

De donde se obtiene la aproximación de diferencia forward (Euler forward):

$$\frac{df}{dx}(x_0) \approx \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \tag{2.15}$$

Y con una diferencia $-\Delta x$, se obtiene la aproximación de diferencia backward

$$\frac{df}{dx}(x_0) \approx \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x} \tag{2.16}$$

Si se suman ambas aproximaciones se tiene la diferencia centrada de la primera derivada.

$$\frac{df}{dx}(x_0) \approx \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} \tag{2.17}$$

Sea v una variable que puede representar a y o a u . Se va usar la notación $v(x_j, t_n) = v_j^n$. El método consiste en aproximar las derivadas parciales de forma similar a como se definieron las aproximaciones de diferencias finitas, para cada variable independiente. Para la derivada parcial respecto al tiempo se aproxima como Euler forward, mientras que para el espacio se aproxima la derivada parcial como centrada. Es decir,

$$v_t \approx \frac{v_j^{n+1} - v_j^n}{\Delta t} \tag{2.18}$$

$$v_x \approx \frac{v_{j+1}^n - v_{j-1}^n}{2\Delta x} \tag{2.19}$$

Esta aproximación se conoce como FTCS (Forward Time Centered Space). Hasta aquí, el método resulta ser inestable incondicionalmente. Lax y Friedrichs optaron por cambiar el término v_j^n de la derivada temporal con el promedio de los términos espaciales

$$v_j^n \approx \frac{v_{j+1}^n + v_{j-1}^n}{2}$$

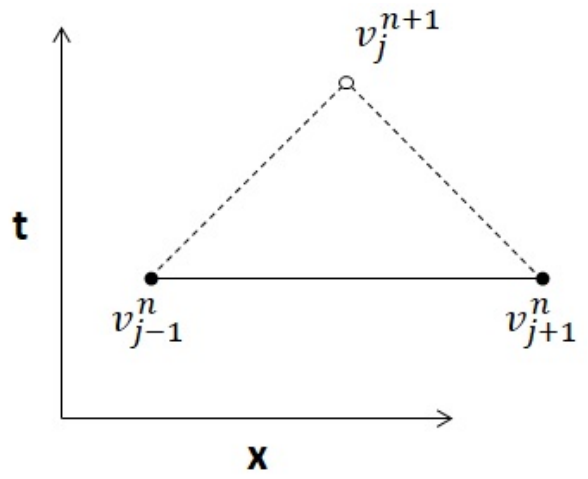


Figura 2.1: Representación del esquema de Lax-Friedrich

Aplicando todas estas aproximaciones en las Ecuaciones de Saint-Venant y despejando el término temporal $n+1$ se tiene:

$$y_j^{n+1} = \frac{y_{j+1}^n + y_{j-1}^n}{2} - u_j^n \frac{\Delta t}{2\Delta x}(y_{j+1}^n - y_{j-1}^n) - y_j^n \frac{\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) \tag{2.20}$$

$$\begin{aligned} u_j^{n+1} = \frac{u_{j+1}^n + u_{j-1}^n}{2} - g\frac{\Delta t}{2\Delta x}(y_{j+1}^n - y_{j-1}^n) - u_j^n \frac{\Delta t}{2\Delta x}(u_{j+1}^n - u_{j-1}^n) \\ +g\Delta t\left(S_0 - c^2(u_j^n)^2\left(\frac{T(y_j^n)}{T+2y_j^n}\right)^{-4/3}\right) \end{aligned} \tag{2.21}$$

Cabe aclarar que en los puntos iniciales y finales de la variable espacial no se tiene el elemento $j-1$ o el elemento $j+1$, así que se tiene que adaptar para $j=0$.

$$y_j^{n+1} = y_j^n - u_j^n \frac{\Delta t}{\Delta x}(y_{j+1}^n - y_j^n) - y_j^n \frac{\Delta t}{\Delta x}(u_{j+1}^n - u_j^n) \tag{2.22}$$

$$u_j^{n+1} = u_j^n - g\frac{\Delta t}{\Delta x}(y_{j+1}^n - y_j^n) - u_j^n \frac{\Delta t}{\Delta x}(u_{j+1}^n - u_j^n) + g\Delta t\left(S_0 - c^2(u_j^n)^2\left(\frac{T(y_j^n)}{T+2y_j^n}\right)^{-4/3}\right) \tag{2.23}$$

Y para $j=N$

$$y_j^{n+1} = y_j^n - u_j^n \frac{\Delta t}{\Delta x}(y_j^n - y_{j-1}^n) - y_j^n \frac{\Delta t}{\Delta x}(u_j^n - u_{j-1}^n) \tag{2.24}$$

$$u_j^{n+1} = u_j^n - g\frac{\Delta t}{\Delta x}(y_{j+1}^n - y_j^n) - u_j^n\frac{\Delta t}{\Delta x}(u_{j+1}^n - u_j^n) + g\Delta t\left(S_0 - c^2(u_j^n)^2\left(\frac{T(y_j^n)}{T+2y_j^n}\right)^{-4/3}\right) \tag{2.25}$$

Todo este sistema de ecuaciones se les llamará a partir de aquí el esquema discreto, esquema numérico o esquema de diferencias finitas de las Ecuaciones de Saint Venant.

2.3. Análisis de Estabilidad

2.3.1. Análisis de Estabilidad de von Neumann

Uno de los criterios más importantes cuando se trata de usar un método numérico para resolver problemas es el de estabilidad. Los métodos numéricos llevan consigo un error debido a que su solución es una aproximación del valor verdadero que se quiere encontrar y el análisis de este es clave porque brinda información para predecir cuando el método puede fallar. Se dice que un esquema de diferencias finitas es estable (en el sentido de Lax) si el error producido por cada iteración temporal no se magnifica a lo largo del cómputo del método.

Para problemas de valor inicial, el análisis de estabilidad de von Neumann es uno de los análisis más populares debido a que el criterio de estabilidad de von Neumann es necesario y suficiente para la estabilidad en el sentido de Lax el cual es sumamente útil debido al Teorema de Equivalencia de Lax.

Teorema 6 (**Teorema de Equivalencia de Lax**)**.** Un método lineal consistente de diferencias finitas de un problema de valores iniciales bien planteado es estable si y sólo si es convergente

En este caso que un método sea convergente significa que la secuencia de soluciones aproximadas se pueden acercar sin interrupción a las soluciones verdaderas conforme ocurren las iteraciones.

El análisis de von Neumann solo puede ser realizado en PDEs y esquemas de diferencias lineales donde la PDE tiene coeficientes constantes, hasta dos variables independientes y las condiciones

de borde son periódicas (PBC). En este caso la linealidad se puede ver en términos de un operador [17].

Definición 2.3.1. Un operador $T : X \to Y$ es lineal si

- El dominio $\mathcal{D}(T)$ de T es un espacio vectorial y el rango $\mathcal{R}(T)$ yace en un espacio vectorial sobre el mismo campo
- $\forall x, y \in \mathcal{D}(T)$ y escalares $\alpha \in K$:

$$T(x + y) = Tx + Ty$$

$$T(\alpha x) = \alpha T x$$

De esta manera, se puede resumir al tipo de PDEs a los que el análisis de von Neumann es aplicable como:

Sea $u(x, t)$ la función objetivo de una ecuación diferencial descrita como

$$u_t = \mathcal{L}u \tag{2.26}$$

tal que $x_L < x < x_R$ y $t > 0$ que satisface $u(x, 0) = u_0(x)$ y $u(x_L, t) = u(x_R, t)$ o similares para la derivada de u. $\mathcal{L}$ es un operador lineal diferencial que por términos prácticos se puede considerar un operador que es una combinación lineal entre operadores diferenciables ($\frac{d}{dx}, \frac{\partial}{\partial t}, \dots$) y operadores que multiplican la función por un escalar (αu) u otra función ($u(x, t)v(x, t)$).

Originalmente, la ecuación que se estudió en primer lugar por von Neumann, Crank, Nicholson y Richtmyer con estas características fue la ecuación de calor con condiciones perdiódicas de borde definida cuya función objetivo estuviera definida en un intervalo espacial de tamaño L.

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} \tag{2.27}$$

Que puede ser resuelta por separación de variables. Se asume que la solución u está dada por

$$u(x,t) = f(t)g(x) \tag{2.28}$$

De esta forma la ecuación se puede expresar como

$$\frac{df(t)}{dt}\frac{1}{\alpha f(t)} = \frac{d^2g(x))}{dx^2}\frac{1}{g(x)} = -\beta^2 \tag{2.29}$$

Cuyas soluciones son

$$f(t) = e^{-\alpha\beta^2 t} = e^{-\phi t} \tag{2.30}$$

$$g(x) = A\sin(\beta x) + B\cos(\beta x) \tag{2.31}$$

$$\rightarrow u(x,t) = e^{-\phi t}(A\sin(\beta x) + B\cos(\beta x)) \tag{2.32}$$

Que de una manera más simple se puede escribir como.

$$u(x,t) = e^{-\phi t}e^{i\beta x} \tag{2.33}$$

Debido a la Fórmula de Euler

$$e^{i\theta} = \cos(\theta) + i\sin(\theta) \tag{2.34}$$

También se sabe que por el principio de superposición que establece que si funciones $u_1, u_2, \ldots u_M$ satisfacen $\mathcal{L}(u_j) = 0$, para $j = \{1, \ldots, M\}$, entonces una combinación lineal de la forma $\sum_{j=1}^{M} c_j u_j$ también es solución. Este concepto será importante más adelante, pero se puede deducir que otra solución será de la forma

$$u(x,t) = \sum_{-M}^{M} e^{-\phi_m t}e^{i\beta_m x} = \sum_{-M}^{M} e^{-\phi_m t_{n+1}}(A_m\sin(\beta_m x) + B_m\cos(\beta_m x)) \tag{2.35}$$

Donde, $M = \frac{L}{\Delta x}$. La clave de esta solución es precisamente por la periodicidad que se tiene en la dimensión espacial.

Por otro lado, si se discretiza el dominio, en este caso rectangular, con pasos Δx y Δt, tal que $x_j = j\Delta x$ y $t_n = n\Delta t$ para $j = 1, \ldots, J$ y $n = 1, \ldots, N$. Entonces,

$$u(x_j, t_n) = e^{-\phi t_n} e^{i\beta x_j} \tag{2.36}$$

$$u(x_j, t_{n+1}) \tag{2.37}$$

Donde, se puede definir

$$G_a = \frac{u(x_j, t_{n+1})}{u(x_j, t_n)} = e^{-\phi \Delta t} \tag{2.38}$$

G_a es conocido como el factor de amplificación analítica. Esto es útil ya que $u(x_j, t_n) = {G_a}^n e^{i\beta x_j}$ muestra un comportamiento claro respecto a la solución siguiendo las iteraciones. Siguiendo esta idea, se aplicó un esquema FTCS a la ecuación de calor tal que

$$u(x_j, t_n) \approx u_j^n \tag{2.39}$$

$$\rightarrow \frac{u_j^{n+1} - u_j^n}{\Delta t} - \alpha \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} \tag{2.40}$$

$$\rightarrow u_j^{n+1} = u_j^n + r(u_{j+1}^n - 2u_j^n + u_{j-1}^n) \tag{2.41}$$

Donde $r = \frac{\alpha \Delta t}{(\Delta x)^2}$. Es muy importante observar que ahora u_j^n representa la variable en el esquema numérico, ya no es parte de la solución analítica (no necesariamente) como en la parte de arriba. Es más, el esquema de diferencias finitas de la Ecuación 2.40 ahora representa un operador aparte que es lineal.

$$\mathcal{F}(u_j^n) = \frac{u_j^{n+1} - u_j^n}{\Delta t} - \alpha \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} = 0 \tag{2.42}$$

Entre estudios, publicaciones y refutaciones, los precursores de este análisis de estabilidad establecieron que la fuente de inestabilidad del esquema de diferencias finitas estaba dada por la evolución del error de redondeo ε_j^n que se puede definir en este caso como la diferencia entre la solución del esquema numérico u_j^n y la solución del esquema numérico N_j^n pero sin precisión aritmética.

$$\varepsilon_j^n = u_j^n - N_j^n \tag{2.43}$$

Como el operador del esquema numérico es lineal, el error también satisface el esquema numérico por el principio de superposición. Esto se verá en más detalle en la siguiente sección.

$$\varepsilon_j^{n+1} = \varepsilon_j^n + r(\varepsilon_{j+1}^n - 2\varepsilon_j^n + \varepsilon_{j-1}^n) \tag{2.44}$$

Como el error tiene el mismo comportamiento que la aproximación numérica, se puede considerar que tiene características similares y cumple con la periodicidad espacial. Esto es fundamental debido al siguiente concepto.

Definición 2.3.2. Sea $f : \mathbb{R} \to \mathbb{R}$ una función integrable en el sentido de Riemann en un intervalo $[-L, L]$, entonces se puede obtener su serie de Fourier, cuya periodicidad puede verse extendida fuera del intervalo, como

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{\pi i \frac{n}{L} x} \tag{2.45}$$

Así se puede aproximar al error como una expansión finita de Fourier expresada en cada valor de tiempo. De esta manera los coeficientes de cada serie de Fourier se vuelven dependientes del elemento del tiempo en la discretización del dominio.

$$\varepsilon(x_j, t_n) \sim \sum_{m=-M}^{M} E_m(t_n) e^{i k_m x_j} \tag{2.46}$$

Aquí k_m se denomina el número de onda y está definido como $k_m = \frac{\pi m}{L}$ y nuevamente $M = \frac{L}{\Delta x}$. La ventaja de la expansión de serie de Fourier es que tiene una base ortogonal, que precisamente por la linealidad del esquema numérico, hace que cada modo de la serie sea una solución también. Esto se puede ver en el Anexo A. Así que para cada modo m se escribe

$$\varepsilon_j^n = E_m(t_n)e^{ik_m x_j} \tag{2.47}$$

$$\varepsilon_j^{n+1} = E_m(t_{n+1})e^{ik_m x_j} \tag{2.48}$$

$$\varepsilon_{j+1}^n = E_m(t_n)e^{ik_m x_{j+1}} \tag{2.49}$$

$$\varepsilon_{j-1}^n = E_m(t_n)e^{ik_m x_{j-1}} \tag{2.50}$$

Si se aplica esto en el esquema numérico se tiene que

$$\frac{\varepsilon_j^{n+1}}{\varepsilon_j^n} = \frac{E_m(t^{n+1}}{E_m(t^n)} = 1 + r(e^{ik_m\Delta x} + e^{-ik_m\Delta x} - 2) = 1 - 4r\sin^2(\frac{k_m\Delta x}{2}) \tag{2.51}$$

De forma analoga, se puede definir el factor de amplificación numérico como

$$G = \frac{E_m(t^{n+1})}{E_m(t^n)} = \frac{\varepsilon_j^{n+1}}{\varepsilon_j^n} \tag{2.52}$$

De donde se puede obtener una relación de recurrencia

$$\varepsilon_j^n = G^n\varepsilon_j^0 = G^n E_m(t_0)e^{ik_m x_j} = E_0 G^n e^{ik_m x_j} \tag{2.53}$$

Es decir, que para poder tener una condición de estabilidad a medida que $n \to \infty$, se requiere que

$$|G| \leq 1 \tag{2.54}$$

Para que $\varepsilon_j^n \to 0$ se pueda dar. Finalmente, en la ecuación de calor se tiene que

$$\left|1 - 4r\sin^2\left(\frac{k_m\Delta x}{2}\right)\right| \leq 1 \tag{2.55}$$

Y esto solo se da si $r \leq \frac{1}{2}$.

El método de von Neumann consiste entonces en asumir que el error tendrá una forma $\varepsilon_j^n = A_0 G^n e^{ikj\Delta x}$ para poder obtener una expresión para G una vez que se introduce esta noción en el

esquema numérico y determinar qué condición se requiere para que $|G| \leq 1$. Como el término u_j^n también cumple con el esquema, es común usar una expresión definida en la variable en vez de escribir el término de error como tal ($u_j^n = A_0 \xi^n e^{ikj\Delta x}$). En este trabajo el enfoque estará en los errores.

2.3.2. Estabilidad para el método de Lax-Friedrichs

Para este método se ha visto que el análisis de estabilidad de von Neumann o Fourier es muy útil, así que se va a presentar una idea de su funcionamiento [27]. Para esto, se va asumir que se cumplen condiciones periódicas tanto para u como para y sin pérdida de generalidad. Ya se revisó en la sección anterior cómo se puede aproximar la solución analítica a un esquema numérico y a este se le puede asociar un error de redondeo. Suponer que se va a usar la misma discretización de dominio rectangular y sea v la representación para y o u de aquí en adelante, entonces $v(x_j, t_n) \approx v_j^n$. Entonces para el error de redondeo se puede escribir

$$v_j^n = N_v(x_j, t_n) + \varepsilon_v(x_j, t_n) \tag{2.56}$$

Donde $N_v(x_j, t_n)$ representa la solución del esquema numérico pero sin precisión aritmética para cada una de las variables dependientes. Desafortunadamente, el esquema numérico de las Ecuaciones de Saint-Venant no es un esquema lineal y el análisis de Von Neumann no puede ser aplicado. Sin embargo, se puede linealizar las ecuaciones asumiendo que los coeficientes de las ecuaciones diferenciales sean constantes en cada sección que engloba un paso del método de Lax (Figura 2.1). Es decir, se puede escribir una aproximación de las ecuaciones para dar un comportamiento local de la siguiente forma

$$\begin{cases} \frac{\partial y}{\partial t} + \bar{u}\frac{\partial y}{\partial x} + \bar{y}\frac{\partial u}{\partial x} = 0 \\ \frac{\partial u}{\partial t} + g\frac{\partial y}{\partial x} + \bar{u}\frac{\partial u}{\partial x} = g(S_0 - S_f(\bar{u}, \bar{y})) \end{cases} \tag{2.57}$$

Así que en un paso de método de Lax, se puede expresar estas ecuaciones como

$$\begin{cases} \frac{y_j^{n+1}}{\Delta t} - \frac{y_{j+1}^n + y_{j-1}^n}{2\Delta t} + \bar{u}\frac{y_{j+1}^n - y_{j-1}^n}{2\Delta x} + \bar{y}\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0 \\ \frac{u_j^{n+1}}{\Delta t} - \frac{u_{j+1}^n + u_{j-1}^n}{2\Delta t} + g\frac{y_{j+1}^n - y_{j-1}^n}{2\Delta x} + \bar{u}\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = g(S_0 - S_f(\bar{u}, \bar{y})) \end{cases} \tag{2.58}$$

La idea es reemplazar la expresión de 2.56. En este caso, solo se hará para la segunda ecuación y para la primera la metodología se puede suponer similar.

$$\begin{aligned}\frac{N_u(x_j,t_{n+1})}{\Delta t}-\frac{N_u(x_{j+1},t_n)+N_u(x_{j-1},t_n)}{2\Delta t}+g\frac{N_y(x_{j+1},t_n)-N_y(x_{j-1},t_n)}{2\Delta x}+\\ \bar{u}\frac{N_u(x_{j+1},t_n)-N_u(x_{j-1},t_n)}{2\Delta x}+\frac{\varepsilon_u(x_j,t_{n+1})}{\Delta t}-\frac{\varepsilon_u(x_{j+1},t_n)+\varepsilon_u(x_{j-1},t_n)}{2\Delta t}+\\ g\frac{\varepsilon_y(x_{j+1},t_n)-\varepsilon_y(x_{j-1},t_n)}{2\Delta x}+\bar{u}\frac{\varepsilon_u(x_{j+1},t_n)-\varepsilon_u(x_{j-1},t_n)}{2\Delta x}=g(S_0-S_f(\bar{u},\bar{y}))\end{aligned} \tag{2.59}$$

Como $N_v(x_j,t_n)$ es solución del esquema lineal, entonces se tiene que

$$\begin{aligned}\frac{\varepsilon_u(x_j,t_{n+1})}{\Delta t}-\frac{\varepsilon_u(x_{j+1},t_n)+\varepsilon_u(x_{j-1},t_n)}{2\Delta t}+g\frac{\varepsilon_y(x_{j+1},t_n)-\varepsilon_y(x_{j-1},t_n)}{2\Delta x}\\ +\bar{u}\frac{\varepsilon_u(x_{j+1},t_n)-\varepsilon_u(x_{j-1},t_n)}{2\Delta x}=0\end{aligned} \tag{2.60}$$

Es decir, que los errores satisfacen el esquema numérico de la ecuación homogénea. Se puede escribir el sistema de ecuaciones como

$$\begin{aligned}\varepsilon_y(x_j,t_{n+1})=\frac{\varepsilon_y(x_{j+1},t_n)+\varepsilon_y(x_{j-1},t_n)}{2}-\bar{u}\frac{\Delta t}{2\Delta x}\left(\varepsilon_y(x_{j+1},t_n)-\varepsilon_y(x_{j-1},t_n)\right)\\ -\bar{y}\frac{\Delta t}{2\Delta x}\left(\varepsilon_u(x_{j+1},t_n)-\varepsilon_u(x_{j-1},t_n)\right)=0\end{aligned} \tag{2.61}$$

$$\begin{aligned}\varepsilon_u(x_j,t_{n+1})=\frac{\varepsilon_u(x_{j+1},t_n)+\varepsilon_u(x_{j-1},t_n)}{2}-g\frac{\Delta t}{2\Delta x}\left(\varepsilon_y(x_{j+1},t_n)-\varepsilon_y(x_{j-1},t_n)\right)\\ -\bar{u}\frac{\Delta t}{2\Delta x}\left(\varepsilon_u(x_{j+1},t_n)-\varepsilon_u(x_{j-1},t_n)\right)=0\end{aligned} \tag{2.62}$$

Se prosigue a hacer el análisis de estabilidad de von Neumann.

$$\varepsilon_y(x_j,t_n)+\varepsilon_u(x_j,t_n)=\varepsilon(x_j,t_n)=K\xi^n e^{ik_m x_j}=K_y\xi^n e^{ik_m x_j}+K_u\xi^n e^{ik_m x_j} \tag{2.63}$$

O de forma vectorial se puede aproximar los errores como

$$\begin{pmatrix}\varepsilon_y\\ \varepsilon_u\end{pmatrix}=\xi^n e^{ikj\Delta x}\begin{pmatrix}K_y\\ K_u\end{pmatrix} \tag{2.64}$$

Reemplazando este tipo de solución y aplicando la Fórmula de Euler, se obtiene una sistema de ecuaciones que puede representarse matricialmente como

$$\begin{pmatrix} \xi - cos(k\Delta x) + i\bar{u}\frac{\Delta t}{\Delta x}\,\mathrm{sen}(k\Delta x) & i\bar{h}\frac{\Delta t}{\Delta x}\,\mathrm{sin}(k\Delta x) \\ ig\frac{\Delta t}{\Delta x}\,\mathrm{sen}(k\Delta x) & \xi - cos(k\Delta x) + i\bar{u}\frac{\Delta t}{\Delta x}\,\mathrm{sen}(k\Delta x) \end{pmatrix} \begin{pmatrix} K_y \\ K_u \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{2.65}$$

En este caso, se tiene una sistema matricial y se puede aplicar los siguiente teoremas

Teorema 7. Sea una ecuación matricial $Ax = 0$ donde A es una matriz $n \times n$ y $x \in \mathbb{R}^n$. Si la solución a la ecuación sólamente corresponde a $x = 0$, entonces A no es invertible.

Teorema 8. Una matriz es invertible si y sólo si su determinante no es cero.

Claramente, se busca una solución tal que $K_y, K_v \neq 0$, ya que sino las soluciones de la función incógnita serían simplemente la solución homogénea. Para que haya otra respuesta, es necesario que la matriz del lado izquierdo no sea invertible y por lo tanto su determinante sea igual a cero, lo que lleva a la expresión para la estabilidad

$$|\xi|^2 = \cos^2(k\Delta x) + \left(\sqrt{g\bar{y}} + \bar{u}\right)^2 \left(\frac{\Delta t}{\Delta x}\right)^2 \mathrm{sen}^2(k\Delta x) < 1 \tag{2.66}$$

Que solo es cierta si

$$\left|\sqrt{g\bar{y}} + \bar{u}\right| \frac{\Delta t}{\Delta x} < 1 \tag{2.67}$$

El factor $|\sqrt{g\bar{y}} + \bar{u}|\,\frac{\Delta t}{\Delta x}$ va a ser referido como el factor de estabilidad. Cabe recalcar que el análisis presente se está dando de forma local con ciertas asunciones, así que el factor está dando información de la estabilidad pero para cada $\bar{y}$ y cada $\bar{u}$ definidos en un paso de Lax. Es por esto, que para tener una idea general de la estabilidad en todo el dominio se establece que si los valores máximos $\bar{y}_{max}$ y $\bar{u}_{max}$ en el dominio cumplen la condición, entonces necesariamente todos los demás valores dentro del dominio también deben cumplirla.

Por la forma de la condición inicial del problema, se puede enunciar que $\bar{y}_{max} = 2$. Sin embargo,

no se puede hacer un análisis similar de la velocidad, por lo que bajo un entendimiento físico, se puede sugerir que la velocidad máxima que se puede alcanzar dentro del dominio ocurre si es que toda la energía potencial desde $\bar{y}_{max}$ se convierte en energía cinética.

$$\frac{1}{2}m\bar{u}_{max}^2 = mg\bar{y}_{max} \tag{2.68}$$

Por lo que $\bar{u}_{max} = \sqrt{2g\bar{y}_{max}}$. De esta forma, se puede tener un factor de estabilidad descrito como

$$(2+\sqrt{2})\sqrt{g}\frac{\Delta t}{\Delta x} < 1 \tag{2.69}$$

Este factor será probado en un estudio dónde se variará los parámetros Δt y Δx hasta observar un valor en el que la solución de las Ecuaciones de Saint Venant presenten algún símbolo de inestabilidad. Además, este factor también será la base para construir el dominio discretizado para el método de la red neuronal.

2.4. Método con Red Neuronal

Este método fue recomendado en [38]. Antes se mencionó que la base de las redes son los pesos y los sesgos. Precisamente estas variables son conocidas como los parámetros de la red, y para generalizarlos se les va a representar dentro de un vector θ, similar a la notación usada en la sección 1.3.1. La idea es que uno puede representar una función incógnita (aplicado al caso presentado en este trabajo) de una ecuación diferencial como

$$v(x,t,\theta_v) = A_v(x,t) + f_v(x,t,N_v(x,t,\theta_v)) \tag{2.70}$$

Donde A_v es una función que se enfoca en las condiciones iniciales o de borde y no depende de parámetros de alguna red neuronal. Por otro lado, f_v representa una función que además de depender de las variables independientes, también depende de una salida de red neuronal que se va a encargar del comportamiento general de la función incógnita.

Toda red neuronal tiene una función de pérdida o de costo. Su cometido es que al ser minimizada,

mediante algoritmos de optimización, esta influya en los parámetros de la red neuronal para que se cumpla dicho cometido. A medida que la red neuronal va actualizando sus parámetros, se esperaría que el modelo, en este caso N_v , mejore su comportamiento para obtener un valor aceptable de v. Sea la discretización usada en 2.2 , es decir con J términos espaciales y N temporales. Sean

$$F_1\left(x_j, t_n, y(x_j, t_n), u(x_j, t_n), \ldots, \frac{\partial}{\partial x}(u(x_j, t_n))\right) = 0 \tag{2.71}$$

$$F_2\left(x_j, t_n, y(x_j, t_n), u(x_j, t_n), \ldots, \frac{\partial}{\partial x}(u(x_j, t_n), S_f(x_j, t_n)\right) = 0 \tag{2.72}$$

representaciones operacionales de las dos ecuaciones de Saint-Venant. Se puede obtener una función de pérdida ya que se conoce que ambas ecuaciones deben ser igual a 0 al mismo tiempo. Esto se traduce a que se puede usar una medida que permita ver si la ecuación diferencial se está acercando a cero. Una función de pérdida común es el del error de medias cuadradas, que puede adaptarse a este caso como

$$E(x_j, t_n, \theta_u, \theta_v) = \frac{1}{N \times J} \sum_{(x_j, t_n) \in \mathcal{D}} \left[F_1(x_j, t_n, \ldots)^2 + F_2(x_j, t_n, \ldots)^2\right] \tag{2.73}$$

Donde $\mathcal{D}$ representa el dominio rectangular previamente mencionado. Así que si se aumenta esta restricción, se tiene todo el modelo.

Como se menciona en [26], un lenguaje y librería muy utilizados para redes neuronales son Python y PyTorch, además de que un optimizador especializado para estos casos es ADAM, cuyo funcionamiento se explicó anteriormente. Se usará todo junto con las sugerencias de reescalamiento de variables que también se usó en dicho trabajo, además de las funciones A_v y f_v manejadas, pero adaptadas a este problema en particular. Se usará el modelo de ANN propuesto para trabajar las redes neuronales de la profundidad de flujo y la velocidad de forma paralela.

Cabe mencionar, que una de las peculiaridades que tiene este modelo es que tiene una sola

entrada y la salida tiene un número total de valores equivalente al número de puntos en el dominio ($N \times J$). Además, se toma $N = J$ en la discretización del dominio.

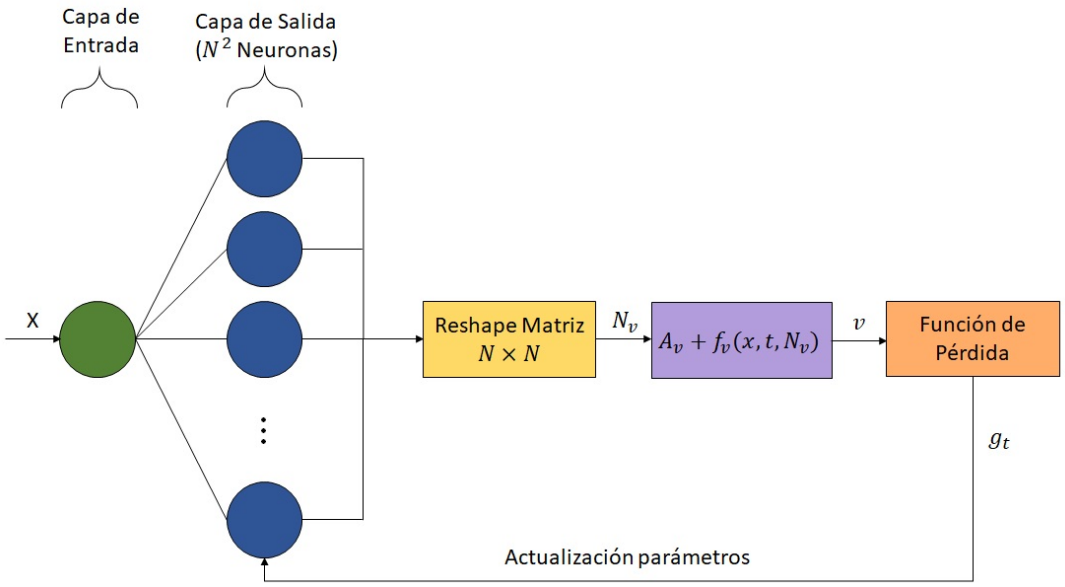


Figura 2.2: Modelo Método de Red Neuronal

Como primer paso, se va a escoger un número N que defina el número de elementos en el dominio ($N \times N$) para poder trabajar en él contínuamente. Para esto se efectuará un estudio en el que se varía el número de elementos en la partición (31, 51, 101, 151, 251, 501) con un modelo que tiene como parámetro de tasa de aprendizaje del optimizador de 0.01 y un valor de convergencia al que la imagen de la función de pérdida debe alcanzar de 1×10^{-2}. Para ver qué tan bueno es el modelo, se ha optado por tomar como medidas de calidad el número de iteraciones que requiere la red para optimizarse, el tiempo que se demoran esas iteraciones, y una comparación con la solución con el método de Lax dado dos pautas.

La primera pauta es analizar el error de medias cuadradas entre ambas soluciones. v_{NN} representará aquí la solución de la funciones incógnitas y o u del modelo de red neuronal, mientras que v_{Lax} representa la solución del método de Lax-Friedrichs.

$$M_v^1 = \sum_{(x_j,t_n)\in\mathcal{D}} \left(v_{NN}(x_j,t_n) - v_{Lax}(x_j,t_n)\right)^2 \tag{2.74}$$

La segunda es usar una métrica comúnmente usada entre funciones

$$M_v^2 = \max_{(x_j,t_n)\in\mathcal{D}} ||v_{NN}(x_j,t_n) - v_{Lax}(x_j,t_n)|| \tag{2.75}$$

La clave está en que M^1 da un vistazo a qué tan parecido es el modelo en promedio con la solución del método numérico, mientras que M^2 revelará si es que hay algún punto que está demasiado alejado de la solución de Lax-Friedrichs.

Luego, una vez escogido el número de particiones, se procede a hacer un estudio de estabilidad de la red, tomando en cuenta el factor de estabilidad. Después, se presentarán soluciones obtenidas variando el valor de convergencia y la tasa de aprendizaje. Finalmente, dados los resultados, también se decidió analizar variaciones del modelo de la red incluso con un modelo mucho más profundo.

Todos los estudios se efectuaron en una computadora con procesador de cuatro núcleos Intel(R) Core(TM) i7-5500U de 2.40 GHz y RAM de 12 GB.

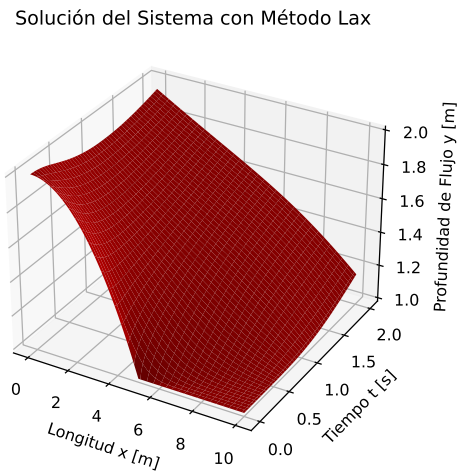
Capítulo 3

Resultados

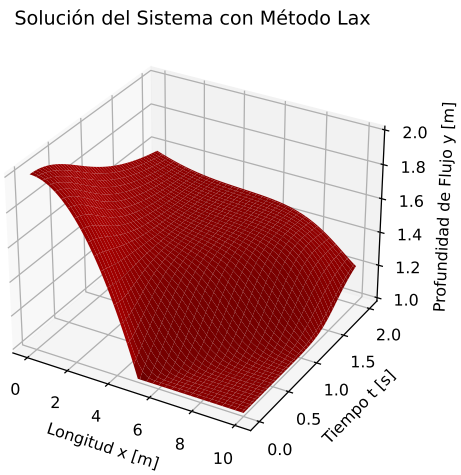
3.1. Método de Lax y Estabilidad

En esta sección lo que se presentará es ver cómo afecta los términos del factor de estabilidad Δt y Δx en las soluciones de las Ecuaciones de Saint Venant Para esto, se hizo un estudio donde se fijó el dominio en el rectángulo $0 \leq x \leq 10$ y $0 \leq t \leq 2$ y se fijó Δx=0.1. Luego, se varió Δt desde un valor de 0.001 hasta llegar a un valor donde se presentaran signos de inestabilidad en la solución para ver hasta qué valor podía alcanzar el factor de estabilidad. El indicio sucedió cuando Δt =0.021 con un factor de estabilidad aproximado de 2.246. En las Figuras 3.1 y 3.2 se presentan las soluciones de la profundidad de flujo y de velocidad para cada caso respectivamente.

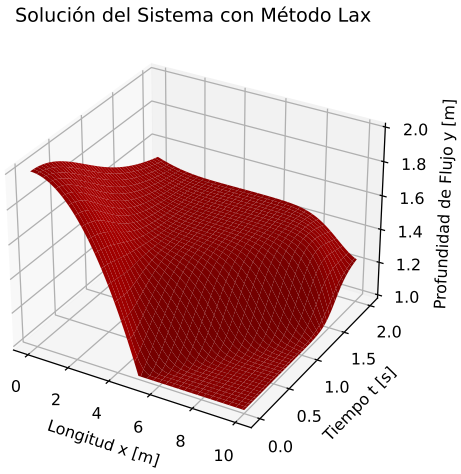
Como era de esperarse, si es que el factor de estabilidad es menor que 1, se tienen soluciones estables como en las Figuras 3.1(a), 3.1(b). 3.2(a) y 3.2(b). Sin embargo, en las Figuras 3.1(d) y 3.2(d) se puede ver que empiezan a haber oscilaciones en la imagen del extremo del dominio. Ya que el modelo empezó a presentar inestabilidad solo cuando el factor fue de 2.246 aproximadamente, parece que la condición de que sea menor a uno brinda un gran intervalo de protección,



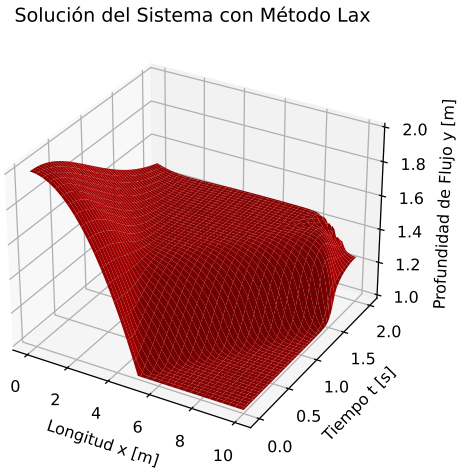
(a) Δt =0.001, Factor $\approx$ 0.107



(b) Δt =0.005, Factor $\approx$ 0.535



(c) Δt =0.01, Factor $\approx$ 1.069



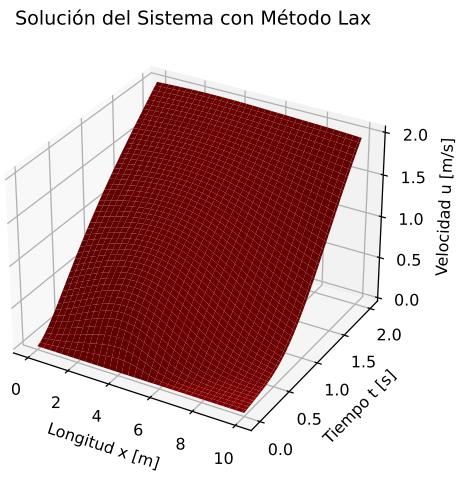
(d) Δt =0.021, Factor $\approx$ 2.246

Figura 3.1: Análisis de Estabilidad para Profundidad con Método de Lax

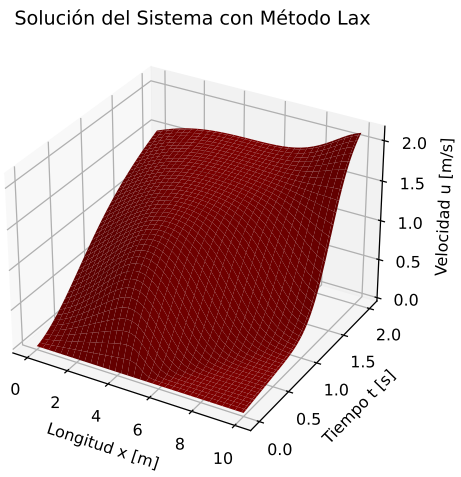
lo que se busca precisamente para la red neuronal.

3.2. Número de Elementos en el Dominio y Estabilidad

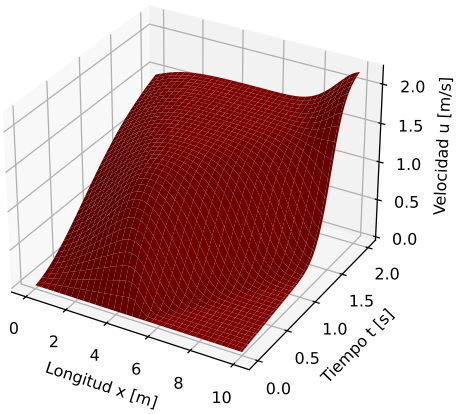
Primero, antes que nada se tiene que definir las funciones A_v y f_v. Aunque las condiciones iniciales no son las mismas, se optó por mantener la idea de [26], pero modificada al problema



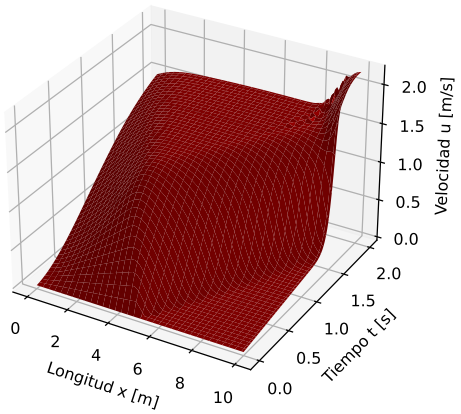
(a) Δt =0.001, Factor $\approx$ 0.107



(b) Δt =0.005, Factor $\approx$ 0.535



(c) Δt =0.01, Factor $\approx$ 1.069



(d) Δt =0.021, Factor $\approx$ 2.246

Figura 3.2: Análisis de Estabilidad para Velocidad con Método de Lax

de este trabajo para que se cumpla la condición inicial, cuando $t = 0$.

$$A_y(x,t) = \begin{cases} (-\frac{1}{25}x^2 + 1)(\exp(-0{,}7t)) + 1 & \text{si } 0 \leq x < 5 \\ 1 & \text{si } x \geq 5 \end{cases} \tag{3.1}$$

$$A_u(x,t) = 0 \tag{3.2}$$

Tabla 3.1: Comparación de soluciones de Modelo de Red Neuronal con las soluciones del Método de Lax-Friedrichs mediante MAX y MSE para distintos dominios.

#	N	Número de Iteraciones	Tiempo [min]	MAX Profundidad	MAX Velocidad	MSE Profundidad	MSE Velocidad
1	31	17000	01:22.1	0.5973	1.4969	0.0091	0.0637
2	51	16500	01:37.4	0.4190	1.0264	0.0096	0.0602
3	101	16900	01:59.1	0.3412	0.8216	0.0086	0.0497
4	151	17600	02:32.6	0.3348	0.8009	0.0083	0.0464
5	251	21200	04:55.6	0.3417	0.7612	0.0085	0.0451
6	501	49600	36:55.9	0.3351	0.7244	0.0094	0.0493

Por otro lado, las funciones f_v serán exactamente las mismas que en dicho trabajo.

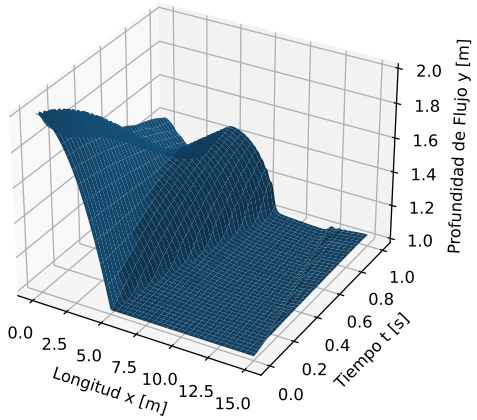
$$f_y(x,t) = tN_y(x,t) \tag{3.3}$$

$$f_u(x,t) = tN_u(x,t) \tag{3.4}$$

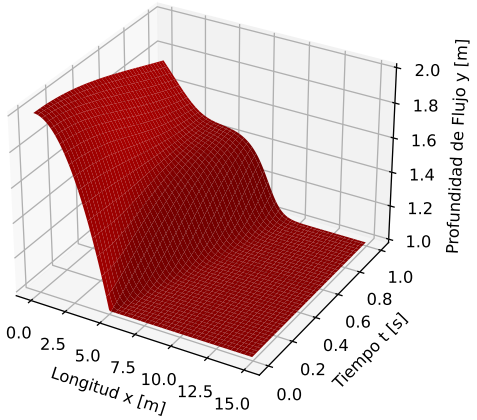
Donde N_y y N_u serán la salida de las redes neuronales descritas en Metodología. Ambas tendrán la misma estructura, sin embargo nada asegura que sean optimizadas de igual forma.

Ahora, se procederá a hacer el estudio de cómo difiere el número de elementos en las soluciones. El error de medias cuadradas se representa como MSE y la métrica asociada al máximo se representa como MAX. Debido a que la parte espacial y temporal tienen mismo número de partes, se escogió un dominio $0 \leq x \leq 15$ y $0 \leq t \leq 1$ para que se permitiera tener un factor de estabilidad menor a 1 sin problema ($\approx$0.713). (Solo se presentará la gráfica del experimento que tenga mejores cualidades)

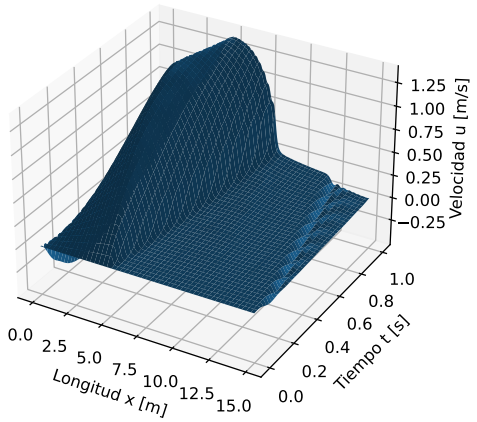
Los resultados muestran que el número de iteraciones es relativamente similar hasta el experimento de 151 elementos. El tiempo de cómputo sin embargo, se ve claramente afectado en el experimento de 501 elementos. Finalmente, los valores de los errores se ven similares a partir del experimento de 101 elementos. Cabe recalcar que todos las soluciones del método de redes neuronales se están comparando con su análogo del método de Lax-Friedrichs, por lo que el error es relativo al modelo de Lax. Por esta razón, ya que una buena discretización da más



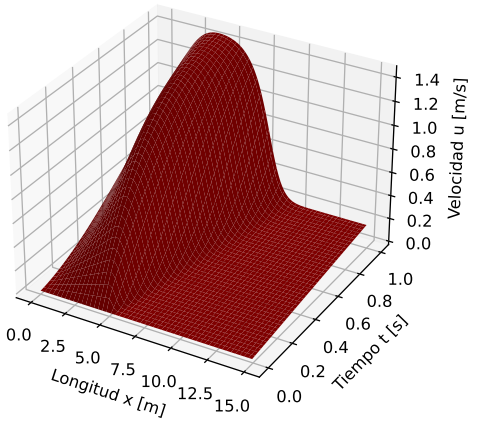
(a) Modelo Red Neuronal para profundidad de flujo, N =251



(b) Modelo Lax-Friedrichs para profundidad de flujo, N =251



(c) Modelo Red Neuronal para velocidad, N =251

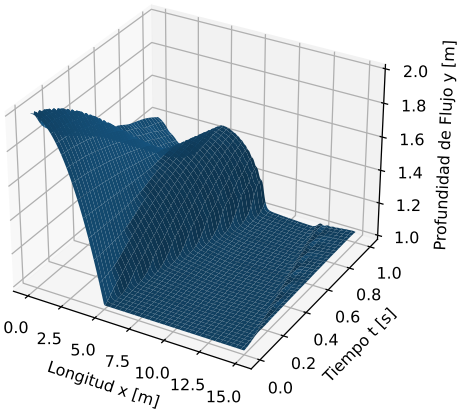


(d) Modelo Lax-Friedrichs para velocidad, N =251

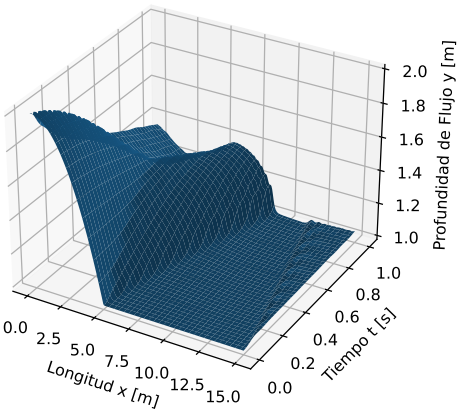
Figura 3.3: Gráficas de Modelos para dominio de 251×251 puntos

información, se ha optado por dividir las dimensiones espacial y temporal en 250 partes, ya que el tiempo de cómputo aún no es tan alto.

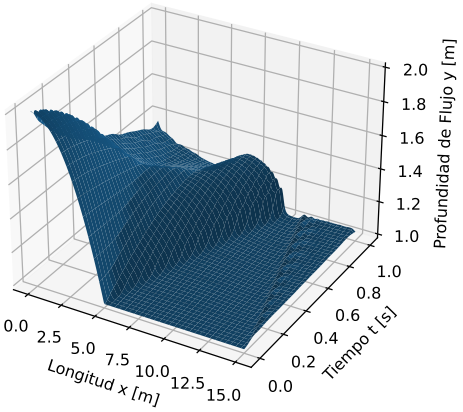
Por otro lado, para analizar la estabilidad, se hizo un mallado de 251×251 puntos en un dominio donde el intervalo $0 \leq x \leq 15$ queda fijo, mientras que se varió el valor máximo del dominio del tiempo para afectar directamente en el valor del factor de estabilidad. Para esto, se varió el último término temporal en pasos de 0.2 y se volvió a analizar los parámetros de calidad. En la



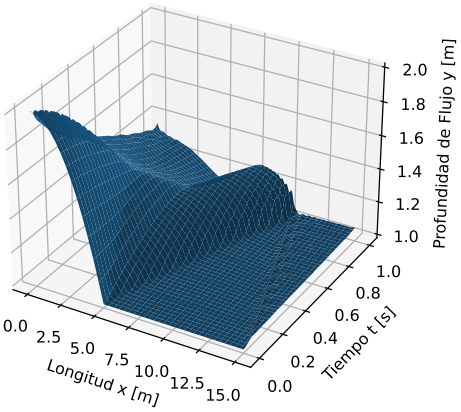
(a) Modelo Red Neuronal para factor de estabilidad 0.713



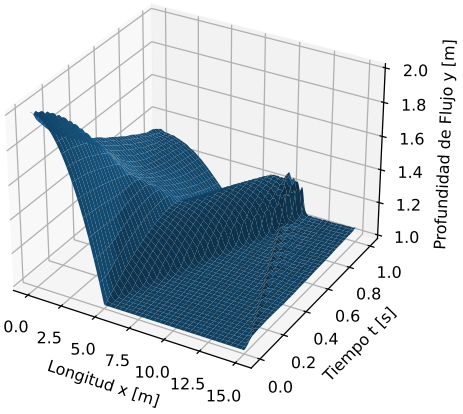
(b) Modelo Red Neuronal para factor de estabilidad 0.855



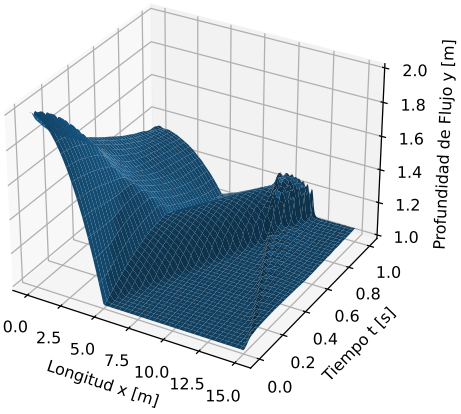
(c) Modelo Red Neuronal para factor de estabilidad 0.998



(d) Modelo Red Neuronal para factor de estabilidad 1.140



(e) Modelo Red Neuronal para factor de estabilidad 1.283



(f) Modelo Red Neuronal para factor de estabilidad 1.425

Figura 3.4: Análisis de Estabilidad para Modelo de Red Neuronal

Figura 3.4 se encuentra la solución de estos experimentos solo para la profundidad de flujo.

Se puede ver que cuando $0 \leq t \leq 1$ y cuando $0 \leq t \leq$1.8, el modelo mostró inestabilidad. El factor de estabilidad con $0 \leq t \leq$1.8 es alrededor de 1.28, lo que nos indica que el criterio del factor de estabilidad resultó ser conveniente para estas condiciones de la red neuronal. A medida que se siga haciendo más estudios, se observará que el método no se desestabiliza.

Aquí cabe mencionar que aunque los valores de tasa de aprendizaje de la red y valor de convergencia parece que se tomaron de forma aleatoria, realmente tienen un sentido. Uno de los mayores problemas de las redes neuronales es que su función de pérdida puede no ser convexa y los algoritmos de optimización están planeados para funciones convexas y bien condicionadas. Así que bajar mucho la tolerancia del algoritmo puede provocar que este se estanque, lo que supondría que tendríamos que disminuir la tasa de aprendizaje. Sin embargo, bajar una magnitud a la tasa de aprendizaje puede provocar que el tiempo de cómputo se quintuplique. Es decir, que no se tiene tanta libertad tampoco para que la función de pérdida converja a un valor pequeño.

3.3. Cambio de funciones A_v

Como se pudo observar en la Figura 3.3 y en la Tabla 3.1, está habiendo un problema con valores que están muy distanciados de los del modelo de Lax especialmente respecto a los de velocidad. Se ve ligeramente que hay algunos puntos tal que $x = 0$, donde la velocidad se está haciendo negativa. Es por esta razón que se ha decidido hacer un estudio para ver si las funciones A_v afectan considerablemente en los resultados. Para esto, se ha decidido usar funciones que crecen o decrecen a partir de la condición inicial en la dirección temporal tanto para la velocidad como la profundidad de flujo. Estas serán las siguientes

$$A_{y,1}(x,t) = \begin{cases} (-\frac{1}{25}x^2 + 1)(\exp(-0{,}7t)) + 1 & \text{si } 0 \leq x < 5 \\ 1 & \text{si } x \geq 5 \end{cases} \tag{3.5}$$

Tabla 3.2: Comparación de soluciones de Modelo de Red Neuronal con las soluciones del Método de Lax-Friedrichs mediante MAX y MSE para distintos A_y y A_u.

Exp.	Número de Iteraciones	Tiempo [min]	MAX Profundidad	MAX Velocidad	MSE Profundidad	MSE Velocidad
11	17100	04:02.8	0.2325	0.7261	0.0025	0.0191
12	26600	06:10.1	0.6338	1.5297	0.0243	0.1549
21	17300	04:04.2	0.4289	0.9054	0.0050	0.0328
22	28200	06:16.8	0.3408	0.9900	0.0112	0.0776

$$A_{y,2}(x,t) = \begin{cases} (-\frac{1}{25}x^2 + 1)(\exp(0{,}7t)) + 1 & \text{si } 0 \leq x < 5 \\ 1 & \text{si } x \geq 5 \end{cases} \tag{3.6}$$

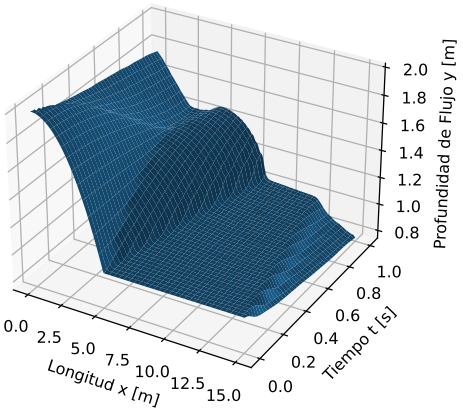
$$A_{u,1} = t \tag{3.7}$$

$$A_{u,2} = -t \tag{3.8}$$

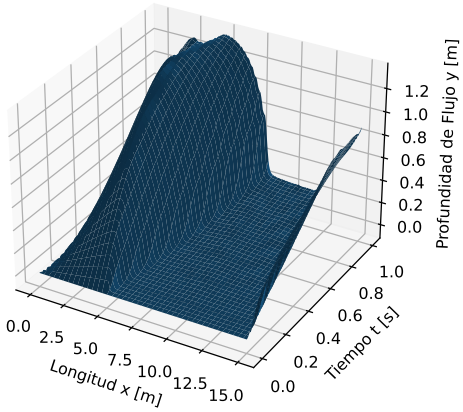
Así, se hará un estudio pequeño de 4 experimentos que se etiquetarán ij para $A_{y,i}$ y $A_{u,j}$. Por ejemplo, el experimento 12 será el que haya usado a $A_{y,1}$ y $A_{y,2}$. Los resultados se encuentran en la Tabla .

Se puede ver que en efecto, la función A_v es muy importante ya que los valores de los parámetros de calidad son muy distintos entre experimentos. Como era de esperarse por las soluciones de las Figuras 3.5(a) y 3.6(a), hay mejores resultados si es que la parte temporal tiene una función A_u creciente y una función A_y decreciente. Sin duda, tomar el crecimiento como punto de partida ha sido un acierto.

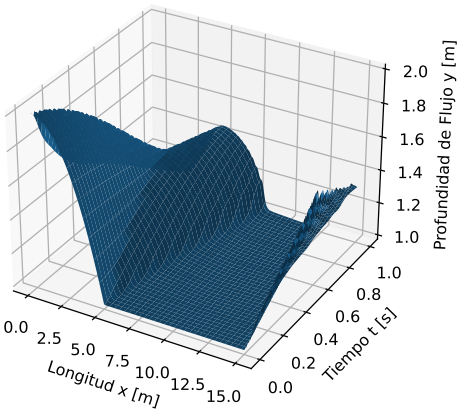
Aun así, se puede ver que los valores de MAX siguen siendo muy altos, especialmente en la velocidad. En la Tabla 3.2 se puede ver que la solución del experimento 11 indica mejores parámetros de calidad. Aparentemente esto se da porque se han perdido los valores negativos en la gráfica de la velocidad, pero ha aparecido un nuevo problema al otro extremo del dominio. Fuera de estas fallas, el modelo parece estarse adaptando mejor.



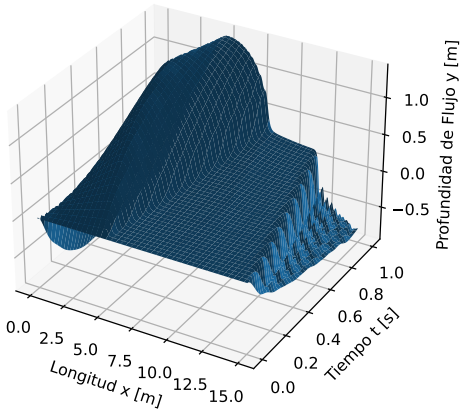
(a) Modelo Red Neuronal para profundidad de flujo y experimento 11



(b) Modelo Red Neuronal para velocidad y experimento 11



(c) Modelo Red Neuronal para profundidad de flujo y experimento 12

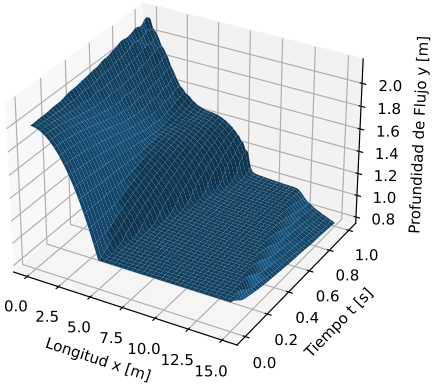


(d) Modelo Red Neuronal para velocidad y experimento 12

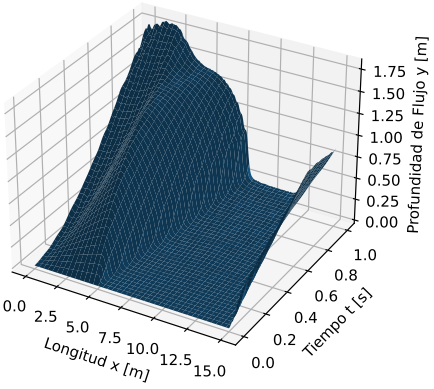
Figura 3.5: Gráficas de Modelos de Red Neuronal para $A_{y,1}$

3.4. Modelo más profundo

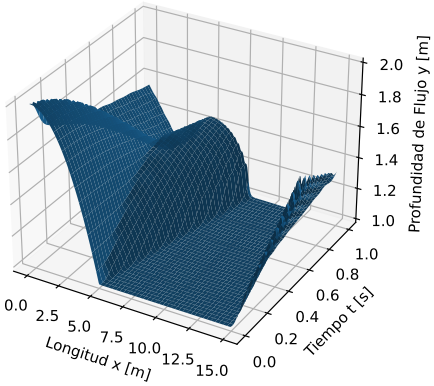
Profundizando mucho más en los datos del anterior estudio, hubo cómo darse cuenta de algo peculiar. Al momento de revisar los valores finales de los parámetros de la red neuronal, se podía notar que habían ciertos puntos en los que los parámetros variaban demasiado entre puntos del dominio. Esto llevó a pensar a que estos saltos debían estarse dando porque precisamente algún término de la función de pérdida estaba provocándolo, y eso se traducía a un valor extraño de



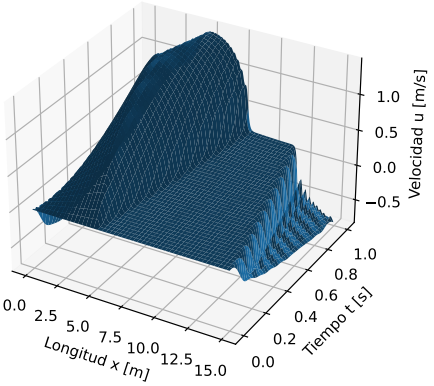
(a) Modelo Red Neuronal para profundidad de flujo y experimento 21



(b) Modelo Red Neuronal para velocidad y experimento 21



(c) Modelo Red Neuronal para profundidad de flujo y experimento 22



(d) Modelo Red Neuronal para velocidad y experimento 22

Figura 3.6: Gráficas de Modelos de Red Neuronal para $A_{y,2}$

gradiente que afectaba a la optimización de los parámetros. Resulta que algo que no se está tomando en cuenta y es muy delicado, es que en el término de las pendientes de la segunda ecuación de Saint-Venant está habiendo un término en un denominador que si se acerca a cero, hace crecer el valor de la función de pérdida notablemente. Ese término es la profundidad de flujo y está siendo una fuente de inestabilidad.

Es por esta razón, que se optó por cambiar la expresión de f_y de forma que la profundidad de flujo nunca pueda hacerse cero, y en efecto, la profundidad de flujo tampoco tiene sentido como

una variable negativa.

$$f_y = (tN_y)^2 \tag{3.9}$$

Una vez que se probó este nuevo f_y ocurrió un problema ya mencionado. El modelo no convergió con los parámetros y si es que se ajustaba la tasa de aprendizaje, la función de pérdida decrecía sumamente lento. Así que para mejorar las condiciones de la red se optó por hacerla más profunda (el código de la clase se puede ver en el Anexo B).

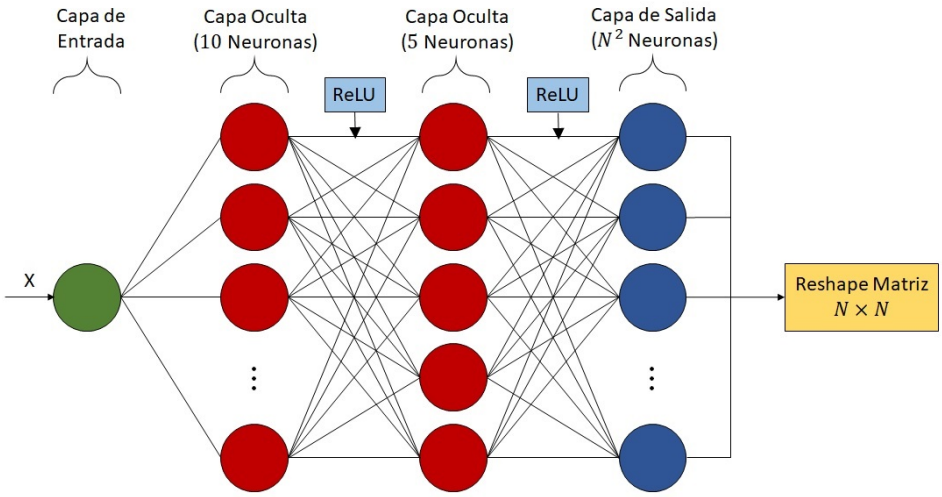


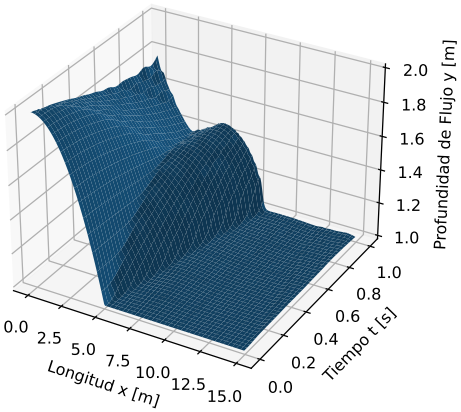
Figura 3.7: Modelo de Red Neuronal Profunda

El cambio consistió en algo simple, aumentar una capa intermedia para poder usar una función de activación y así mejorar el algoritmo de optimización como se ve en la Figura 3.8. El problema está en que si el número de neuronas de esta capa es muy alto, entonces el algoritmo podría requerir demasiada capacidad computacional ya que el número de parámetros depende de la cantidad de neuronas de cada capa. Además, se observó que para que la función de pérdida llegara al menos a un valor de 0.001, se requería una tasa de aprendizaje más pequeña, así que se usó γ =0.001. Bajo este concepto se hizo un modelo que usara pocas neuronas en las capas ocultas. La función de activación que se usó fue ReLU que se define como

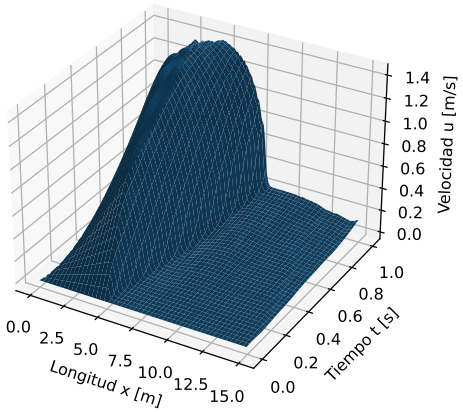
$$ReLU(x) = \text{máx}\{0, x\} \tag{3.10}$$

Los resultados se encuentran en la Figura 3.8. Como se puede notar, las fallas que estaban apareciendo anteriormente fueron eliminadas. Lo único que se diferencia esta solución con la

del Método de Lax (Figras 3.3(b) y 3.3(d)) son esos valles pronunciados en la solución de la profundidad de flujo que se están produciendo alrededor de $x = 4$m y $t = 0{,}8$s. El número de iteraciones fue de 9000, el tiempo de cómputo de 3 minutos y 18.2 segundos. Los valores de MAX fueron de 0.1128 y 0.2178 para profundidad de flujo y velocidad respectivamente. El MSE para la profundidad de flujo fue de 0.0008 y el MSE para la velocidad fue de 0.0038. Es decir, que este modelo no solo mejoró las soluciones, sino que incluso requirió menos gasto computacional.



(a) Solución con Modelo de Red Neuronal profunda para profundidad de flujo



(b) Solución con Modelo de Red Neuronal profunda para velocidad

Figura 3.8: Solución con Modelo de Red Neuronal profunda

Capítulo 4

Conclusiones y Trabajo Futuro

El criterio de estabilidad del análisis de von Neumman resultó ser útil en la mayoría de los casos analizados en este trabajo, y especialmente en la aplicación del método numérico. Sin embargo, los modelos de redes neuronales dependen de muchos factores fuera de los que contiene el factor de estabilidad por lo que este factor requiere también de agudeza en el modelo de ANN para no colocar condiciones que se conocen desestabilizarán el modelo.

Dentro de este estudio se encontró que una de las consideraciones más importantes en el método con ANN es definir correctamente f_v y A_v, pues son clave para mejorar tanto el rendimiento como el tiempo de computación. Pero, sin duda, la consideración más importante que hay que mencionar que mejoró el modelo considerablemente fue la profundidad.

El modelo final obtuvo un rendimiento computacional excelente y MSEs bajos de 0.0008 en la profundidad de flujo y 0.0038 en la velocidad, lo que se puede considerar como un método que lleva a su modelo bastante cerca del modelo del método numérico tradicional en promedio. Sin embargo, desde un punto de vista matemático, y dados los valores de MAX que no son del todo bajos, se tiene una zona donde no se capta todavía todo el comportamiento de la solución por lo

que es necesario seguir mejorando el modelo.

Como trabajo futuro existen muchas cosas que se pueden hacer desde varios ámbitos. Desde la parte ingenieril se puede buscar adaptar las Ecuaciones de Saint-Venant a canales de pendiente alta y a considerar flujo rápidamente variado. También se puede añadir parámetros que consideren viscosidad o analizar el caso de fluidos no newtonianos.

Desde el ámbito computacional se puede seguir mejorando el rendimiento a través del estudio de optimización, además de estudiar a fondo lo que podrían hacer modelos más profundos junto a sus çajas negrasz usar más variedad en el tipo de capas de neuronas. Se puede llevar a cabo estudios para conocer análisis de consistencia, precisión y estabilidad directos.

Por último, en el ámbito matemático se puede buscar que el modelo no sea tan dependiente de varios parámetros para así poder generalizar el algoritmo hacia otras ecuaciones diferenciales. Asímismo, se puede estudiar metodologías que permitan tratar con problemas con condiciones de borde.

Bibliografía

[1] Ram Agashe et al. «Secure Socket Layer in the Network and Web Security». En: *International Journal of Advanced Research in Computer and Communication Engineering* 5 (2022), págs. 638-642. DOI: 10.17148/IJARCCE.2022.115115.

[2] Shikha Agrawal y Jitendra Agrawal. «Neural Network Techniques for Cancer Prediction: A Survey». En: *Procedia Computer Science* (2015), págs. 769-774. DOI: https://doi.org/10.1016/j.procs.2015.08.234.

[3] Kaushik Bhattacharya et al. «Model Reduction And Neural Networks For Parametric PDEs». En: *Journal of Computational Mathematics* (2020). https://arxiv.org/abs/2005.03180. DOI: https://doi.org/10.48550/arXiv.2005.03180.

[4] Alex Bihlo y Roman O. Popovych. «Physics-informed neural networks for the shallow-water equations on the sphere». En: *Journal of Computational Physics* (2022), pág. 111024. DOI: https://doi.org/10.1016/j.jcp.2022.111024.

[5] Alexandra Butoi et al. «TRAINING NEURAL NETWORKS AS RECOGNIZERS OF FORMAL LANGUAGES». En: *arXiv* (2025). https://arxiv.org/abs/2411.07107. DOI: https://doi.org/10.48550/arXiv.2411.07107.

[6] Zhiquiang Cai, Jingshuan Chen y Min Liu. «Least-squares ReLU neural network (LSNN) method for linear advection-reaction equation». En: *Journal of Computational Physics* (2021), pág. 110514. DOI: https://doi.org/10.1016/j.jcp.2021.110514.

[7] Jingrun Chen et al. «Quasi-Monte Carlo sampling for machine-learning partial differential equations». En: *arXiv* (2019). https://arxiv.org/abs/1911.01612. DOI: https://doi.org/10.48550/arXiv.1911.01612.

[8] Ven Te Chow. *Hidráulica de Canales Abiertos*. Ed. por Martha Edna Suáres. Toronto: McGRAW-HILL, 2004. ISBN: 958-600-228-4.

[9] F. H. Coutinho et al. «Modelling the influence of environmental parameters over marine planktonic microbial communities using artificial neural networks». En: *Science of the Total Environment* (2019), págs. 205-214. DOI: https://doi.org/10.1016/j.scitotenv.2019.04.009.

[10] Roza Dastres y Mohsen Soori. «Artificial Neural Network Systems». En: *International Journal of Imaging and Robotics* 2 (2021), págs. 12-25. ISSN: 2231–525X.

[11] Adam Eck. «Neural Networks for Survey Researchers». En: *Survey Practice* 1 (2018). DOI: https://doi.org/10.29115/SP-2018-0002..

[12] Ammar H. Elsheikh et al. «Modeling of solar energy systems using artificial neural network: A comprehensive review». En: *Solar Energy* 1 (2019), págs. 622-639. DOI: https://doi.org/10.1016/j.solener.2019.01.037.

[13] Lawrence C. Evans. *Partial Differential Equations*. Second Edition. Rhode Island: American Mathematical Society, 2010. ISBN: 78-0-8218-4974-3.

[14] Dongyu Feng, Zeli Tan y QiZhi He. «Physics-informed neural networks of the Saint-Venant equations for downscaling a large-scale river model». En: *Water Resources Research* 2 (2023). DOI: https://doi.org/10.1029/2022WR033168.

[15] Richard Haberman. *Applied Differential Equations with Fourier Series and Boundary Value Problems*. Ed. por Deirdre Lynch. Fifth Edition. New Jersey: Pearson, 2013. ISBN: 0-321-79705-1.

[16] Diederick Kingma y Jimmy Ba. «Adam: A Method for Stochastic Optimization». En: *arXiv* (2015). https://arxiv.org/abs/1412.6980. DOI: https://doi.org/10.48550/arXiv.1412.6980.

[17] Erwin Kreyszig. *Introductory Funcional Analysis with Applications*. First. University of Windsor: John Wiley & Sons, 1978. ISBN: 0-471-50731-8.

[18] Alex Krizhevsky, Ilya Sutskever y Geoffrey E Hinton. «ImageNet Classification with Deep Convolutional Neural Networks». En: *Advances in Neural Information Processing Systems*. Ed. por F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[19] Vishal Lalwani et al. «Response Surface Methodology and Artificial Neural Network-Based Models for Predicting Performance of Wire Electrical Discharge Machining of Inconel 718 Alloy». En: *Journal of Manufacturing and Materials Processing* 2 (2020), pág. 44. DOI: https://doi.org/10.3390/jmmp4020044.

[20] Siwei Ma et al. «Image and Video Compression with Neural Networks: A Review». En: *IEEE Transactions on Circuits and Systems for Video Technology* 6 (2019), págs. 1683-1698. DOI: 10.1109/TCSVT.2019.2910119.

[21] Sonali B. Maind y Priyanka Wankar. «Research Paper on Basic of Artificial Neural Network». En: *International Journal on Recent and Innovation Trends in Computing and Communication* 1 (2014), págs. 96-100. ISSN: 2321-8169. DOI: https://doi.org/10.17762/ijritcc.v2i1.2920.

[22] Deena Babu Mandru et al. «Assessing Deep Neural Network and Shallow for Network Intrusion Detection Systems in Cyber Security». En: *Computer Networks and Inventive Communication Technologies*. 2021. DOI: https://doi.org/10.1007/978-981-16-3728-5_67.

[23] Wei-Bo Mao et al. «Application of artificial neural networks in detection and diagnosis of gastrointestinal and liver tumors». En: *World Journal of Clinical Cases* 18 (2020), págs. 3971-3977. DOI: 10.12998/wjcc.v8.i18.3971.

[24] Ibrahim M. Nasser y Samy Abu-Naser. «Predicting Tumor Category Using Artificial Neural Networks». En: *International Journal of Academic Health and Medical Research* 2 (2019), págs. 1-7.

[25] Ali Bou Nassif et al. «Speech Recognition Using Deep Neural Networks: A Systematic Review». En: *IEEE Access* (2019), págs. 19143-19165. DOI: 10.1109/ACCESS.2019.2896880.

[26] José Palacios-García, Julio Ibarra-Fiallo y Servando Espín-Torres. «A Novel Deep Learning Method for Solving PDE's Applied to a Shallow Water Problem». En: *Computer Networks and Inventive Communication Technologies*. 2025. DOI: https://doi.org/10.1007/978-3-031-85902-1_15.

[27] William H. Press et al. *Numerical Recipes*. Third Edition. New York: Cambridge University Press. ISBN: 978-0-511-33555-6.

[28] Roheen Qamar y Baqar Ali Zardari. «Artificial Neural Networks: An Overview». En: *Mesopotamian Journal of Computer Science* (2023), págs. 124-133. ISSN: 2958-6631. DOI: https://doi.org/10.58496/MJCSC/2023/015.

[29] Sebastian Ruder. «An Overview of Gradient Descent Optimization Algorithms». En: *arXiv* (2017). https://arxiv.org/abs/1609.04747. DOI: https://doi.org/10.48550/arXiv.1609.04747.

[30] Walter Rudin. *Principios de Análisis Matemático*. Tercera Edición. México: McGraw Hill, 1980. ISBN: 968-6046-82-8.

[31] Ramzi M. Sadek et al. «ANN for Parkinson's Disease Prediction». En: *International Journal of Academic Health and Medical Research* 1 (2019).

[32] Saeed Seraj et al. «HamDroid: Permission-based harmful Android anti-malware detection using neural networks». En: *Nueral Computing and Applications* (2022), págs. 15165-15174. DOI: 10.1007/s00521-021-06755-4.

[33] S. Sheikhi, M. T. Kheirabadi y A. Bazzazi. «An Effective Model for SMS Spam Detection Using Content-based Features and Averaged Neural Network». En: *International Journal of Engineering* 2 (2020), págs. 221-228. DOI: 10.5829/ije.2020.33.02b.06.

[34] Dengliang Shi. «A Study on Neural Network Language Modeling». En: *arXiv* (2017). https://arxiv.org/abs/1708.07252. DOI: https://doi.org/10.48550/arXiv.1708.07252.

[35] D. K. Shreyas et al. «A Review on Neural Networks and its Applications». En: *Journal of Computer Technology & Applications* 2 (2023), págs. 59-70. DOI: 10.37591/JoCTA.

[36] Michael Spivak. *Calculus*. Tercera Edición. Barcelona: Reverté, 2014. ISBN: 978-84-291-5182-4.

[37] Dang Duy Thang y Toshihiro Matsui. «Image Transformation can make Neural Networks more robust against Adversarial Examples». En: *arXiv* (2019). https://arxiv.org/abs/1901.03037. DOI: https://doi.org/10.48550/arXiv.1901.03037.

[38] Neha Yadav, Anupam Yadav y Manoj Kumar. *An Introduction to Neural Network Method for Differential Equations*. London: Springer, 2015. ISBN: 978-94-017-9815-0. DOI: 10.1007/978-94-017-9816-7.

ANEXO A: MODOS DE FOURIER COMO SOLUCIÓN

Como se mencionó en la sección 2.3.1, aquí se va a mostrar cómo el hecho de que una serie de Fourier es solución implica que sus modos individuales también lo son en la ecuación de calor. Se tiene que el error debe satisfacer el esquema numérico lineal y que se le puede aproximar con su serie finita de Fourier. Lo que se quiere probar es

$$\varepsilon_j^{n+1} = \varepsilon_j^n + r(\varepsilon_{j+1}^n - 2\varepsilon_j^n + \varepsilon_{j-1}^n)$$

Dado que

$$\varepsilon(x_j, t_n) \sim \sum_{m=-M}^{M} E_m(t_n) e^{ik_m x_j}$$

es una solución. Reemplazando $\varepsilon(x_j, t_n)$ en el esquema se tiene

$$\begin{aligned} \rightarrow \sum_{m=-M}^{M} E_m(t_{n+1}) e^{ik_m x_j} = \sum_{m=-M}^{M} E_m(t_n) e^{ik_m x_j} + r \Bigg(\sum_{m=-M}^{M} E_m(t_n) e^{ik_m x_{j+1}} \\ -2 \sum_{m=-M}^{M} E_m(t_n) e^{ik_m x_j} + \sum_{m=-M}^{M} E_m(t_n) e^{ik_m x_{j-1}} \Bigg) \end{aligned}$$

$$\begin{aligned} \rightarrow \sum_{m=-M}^{M} E_m(t_{n+1}) e^{ik_m j\Delta x} = \sum_{m=-M}^{M} E_m(t_n) e^{ik_m j\Delta x} + r \Bigg(\sum_{m=-M}^{M} E_m(t_n) e^{ik_m (j+1)\Delta x} \\ -2 \sum_{m=-M}^{M} E_m(t_n) e^{ik_m j\Delta x} + \sum_{m=-M}^{M} E_m(t_n) e^{ik_m (j-1)\Delta x} \Bigg) \end{aligned}$$

$$\begin{aligned} \rightarrow \sum_{m=-M}^{M} E_m(t_{n+1}) e^{ik_m j\Delta x} = \sum_{m=-M}^{M} \big(E_m(t_n) + r E_m(t_n) e^{ik_m \Delta x} - 2r E_m(t_n) \\ + r E_m(t_n) e^{-ik_m \Delta x} \big) e^{ik_m j\Delta x} \end{aligned}$$

Donde finalmente, se tiene la expresión

$$\sum_{m=-M}^{M} \left(E_m(t_n) + rE_m(t_n)e^{ik_m\Delta x} - 2rE_m(t_n) + rE_m(t_n)e^{-ik_m\Delta x} - E_m(t_{n+1})\right) e^{ik_m j\Delta x} = 0$$

Por la fórmula de Euler se conoce que para $m = -M, \ldots, M$

$$e^{ik_m x_j} = \cos(k_m x_j) + i\sin(k_m x_j) = \cos\left(\frac{\pi m}{L}x_j\right) + i\sin\left(\frac{\pi m}{L}x_j\right)$$

Definición A1. Sea el espacio de funciones continuas real-valuadas en un intervalo $[a, b]$ y f y g dos elementos de dicho espacio. Se puede definir un producto interno como

$$\langle f, g\rangle = \int_a^b f(x)g(x)dx$$

Definición A2. Sean f y g dos funciones continuas real-valuadas en un intervalo $[a, b]$. Se dice que f y g son ortogonales si

$$\langle f, g\rangle = \int_a^b f(x)g(x)dx = 0$$

Teorema A1. $f(x) = \sin\left(\frac{n\pi x}{L}\right)$ y $g(x) = \cos\left(\frac{m\pi x}{L}\right)$ son ortogonales para todo $m, n \in \mathbb{Z}$ y $x \in [-L, L]$ o $[0, 2L]$.

Teorema A2. $f(x) = \sin\left(\frac{n\pi x}{L}\right)$ y $g(x) = \sin\left(\frac{m\pi x}{L}\right)$ son ortogonales para todo $m, n \in \mathbb{Z}$ tal que $m \neq n$ y $x \in [-L, L]$ o $[0, 2L]$.

Teorema A3. $f(x) = \cos\left(\frac{n\pi x}{L}\right)$ y $g(x) = \cos\left(\frac{m\pi x}{L}\right)$ son ortogonales para todo $m, n \in \mathbb{Z}$ tal que $m \neq n$ y $x \in [-L, L]$ o $[0, 2L]$.

La demostración de todos los teoremas anteriores son bastante conocidos por lo que no se los presenta.

Teorema A4. Sea $f_m = \cos\left(\frac{\pi m}{L}x\right) + i\sin\left(\frac{\pi m}{L}x\right)$ definida para $x \in [-L, L]$ o $[0, 2L]$. Si

$$\sum_{m=-M}^{M} c_m f_m = 0$$

para $c_m \in \mathbb{C}$, entonces $c_m = 0$ para todo $m = -M, \ldots, M$.

Demostración. Sea $l \in \{-M, \ldots, M\}$, entonces

$$\left\langle \sum_{m=-M}^{M} c_m f_m, \cos\left(\frac{\pi l}{L}x\right) \right\rangle = \left\langle 0, \cos\left(\frac{\pi l}{L}x\right) \right\rangle = 0$$

Donde, por la ortogonalidad de todas las funciones f_m, se tiene

$$\left\langle \sum_{m=-M}^{M} c_m f_m, \cos\left(\frac{\pi l}{L}x\right) \right\rangle = c_l \left\langle f_l, \cos\left(\frac{\pi m}{L}x\right) \right\rangle$$

Por propiedades del producto interior se puede obtener

$$c_l \left\langle f_l, \cos\left(\frac{\pi m}{L}x\right) \right\rangle = c_l \left\langle \cos\left(\frac{\pi m}{L}x\right), \cos\left(\frac{\pi m}{L}x\right) \right\rangle + ic_l \left\langle \sin\left(\frac{\pi m}{L}x\right), \cos\left(\frac{\pi m}{L}x\right) \right\rangle$$

Para cualquiera de los dos intervalos $[-L, L]$ o $[0, 2L]$, el valor de $\left\langle \sin\left(\frac{\pi m}{L}x\right), \cos\left(\frac{\pi m}{L}x\right) \right\rangle = 0$ y $\left\langle \cos\left(\frac{\pi m}{L}x\right), \cos\left(\frac{\pi m}{L}x\right) \right\rangle \neq 0$. Es decir,

$$\left\langle \sum_{m=-M}^{M} c_m f_m, \cos\left(\frac{\pi l}{L}x\right) \right\rangle = c_l \left\langle \cos\left(\frac{\pi m}{L}x\right), \cos\left(\frac{\pi m}{L}x\right) \right\rangle = 0$$

Implica que $c_l = 0$. Como l se escogió arbitrariamente se prueba el teorema $\square$.

Es decir, por este teorema se puede afirmar que

$$\sum_{m=-M}^{M} \left(E_m(t_n) + rE_m(t_n)e^{ik_m\Delta x} - 2rE_m(t_n) + rE_m(t_n)e^{-ik_m\Delta x} - E_m(t_{n+1})\right) e^{ik_m j\Delta x} = 0$$

implica que para todo m

$$E_m(t_n) + rE_m(t_n)e^{ik_m\Delta x} - 2rE_m(t_n) + rE_m(t_n)e^{-ik_m\Delta x} - E_m(t_{n+1}) = 0$$

$$\rightarrow E_m(t_{n+1}) = E_m(t_n) + r\left(E_m(t_n)e^{ik_m\Delta x} - 2E_m(t_n) + E_m(t_n)e^{-ik_m\Delta x}\right)$$

Ahora, para un modo m, se puede usar esta expresión para el término ε_j^{n+1}.

$$\begin{aligned}\varepsilon_j^{n+1} &= E_m(t_{n+1})e^{ik_m x_j} = E_m(t_n) + r\left(E_m(t_n)e^{ik_m\Delta x} - 2E_m(t_n) + E_m(t_n)e^{-ik_m\Delta x}\right)e^{ik_m x_j} \\ &= E_m(t_n)e^{ik_m j\Delta x} + r\left(E_m(t_n)e^{ik_m(j+1)\Delta x} - 2E_m(t_n)e^{ik_m j\Delta x} + E_m(t_n)e^{ik_m(j-1)\Delta x}\right) \\ &= \varepsilon_j^n + r(\varepsilon_{j+1}^n - 2\varepsilon_j^n + \varepsilon_{j-1}^n)\end{aligned}$$

Y se prueba que cada modo también es una solución del esquema lineal

ANEXO B: CLASE DE RED NEURONAL PARA MODELO PROFUNDO

```
class NeuralNetwork(nn.Module):
    def __init__(self):
        super().__init__()
        self.flatten = nn.Flatten()
        self.Layer10 = nn.Linear(1, 10)
        self.Layer11 = nn.Linear(1, 10)
        self.act10=nn.ReLU()
        self.act11=nn.ReLU()
        self.Layer20 = nn.Linear(10, 5)
        self.Layer21 = nn.Linear(10, 5)
        self.act20=nn.ReLU()
        self.act21=nn.ReLU()
        self.Layer30 = nn.Linear(5, Num * Num)
        self.Layer31 = nn.Linear(5, Num * Num)
        nn.init.zeros_(self.Layer10.weight)
        nn.init.zeros_(self.Layer11.weight)
        nn.init.zeros_(self.Layer20.weight)
        nn.init.zeros_(self.Layer21.weight)
        nn.init.zeros_(self.Layer30.weight)
        nn.init.zeros_(self.Layer31.weight)

    def forward(self, x):
        x = self.flatten(x)
        output10 = self.Layer10(x)
        output11 = self.Layer11(x)
        output20 = self.Layer20(self.act10(output10))
        output21 = self.Layer21(self.act11(output11))
        output30 = self.Layer30(self.act20(output20)).reshape(Num, Num)
    / 100
        output31 = self.Layer31(self.act21(output21)).reshape(Num, Num)
    / 100
        return output30, output31
```