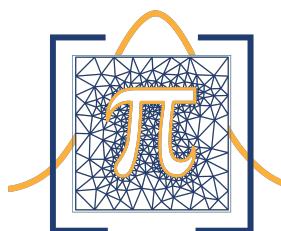


UNIVERSIDAD NACIONAL AUTÓNOMA DE HONDURAS
FACULTAD DE CIENCIAS
ESCUELA DE MATEMÁTICA Y CIENCIAS DE LA COMPUTACIÓN
MAESTRÍA EN MATEMÁTICA



Aprendizaje profundo para la resolución de EDPs: El enfoque del método Deep Galerkin

Tesis presentada para optar al título de máster en matemática con orientación en ingeniería matemática



Tesis presentada por:
Ángel Roberto Rivera Muñoz

Asesor tesis:
Luis Alonzo Zerón Moncada

Tegucigalpa, Honduras

2025

Aprendizaje profundo para la resolución de EDPs: El enfoque del método Deep Galerkin

Presentada por: Ángel Roberto Rivera Muñoz

**Tesis presentada para optar al título de máster en matemática
con orientación en ingeniería matemática**

Asesor:

Luis Alonzo Zerón Moncada
UNAH

TERNA EXAMINADORA

Ph.D Leandro Jesús Galo Mendoza
UNAH

M.Sc Darwin Edisniel Quiroz Betanco
UNAH

M.Sc José Mauricio Alvarenga Rodríguez
UNAH

TEGUCIGALPA M.D.C 31 DE JULIO DE 2025

A mis dos amores
Anahí y Angel Fernando

Resumen

Esta tesis investiga la aplicación del método Deep Galerkin (DGM), un enfoque de aprendizaje profundo libre de mallas, para la resolución de ecuaciones diferenciales parciales (EDPs) complejas. El método se fundamenta en entrenar una red neuronal profunda para que su salida aproxime la solución de la propia ecuación diferencial, respetando sus condiciones iniciales y de frontera., formulando el problema como la minimización de una función de pérdida sobre puntos muestrados aleatoriamente en el dominio. Se evalúa sistemáticamente la eficacia y versatilidad de esta técnica en un amplio espectro de modelos físicos, incluyendo las ecuaciones de Euler-Bernoulli, Timoshenko y Kirchhoff-Love en mecánica estructural; la ecuación de Burgers viscosa en dinámica de fluidos; y la ecuación de Schrödinger en mecánica cuántica.

Se abordaron desafíos como las derivadas de alto orden, mediante su reformulación a sistemas acoplados, y problemas en dominios de alta dimensión, como el caso del átomo de hidrógeno. Los resultados experimentales, validados con soluciones analíticas, demuestran que el DGM proporciona aproximaciones precisas y estables, capturando las propiedades físicas fundamentales de los sistemas estudiados.

Palabras clave: (*Aprendizaje profundo, Ecuaciones diferenciales parciales, Deep Galerkin Method, Redes neuronales, Métodos sin malla*)

Abstract

This thesis investigates the application of the Deep Galerkin Method (DGM), a mesh-free deep learning approach, for the solution of complex partial differential equations (PDEs). The method is based on training a deep neural network so that its output approximates the solution of the differential equation itself, while respecting its initial and boundary conditions, by formulating the problem as the minimization of a loss function over points randomly sampled in the domain.

The effectiveness and versatility of this technique are systematically evaluated across a broad spectrum of physical models, including the Euler-Bernoulli, Timoshenko, and Kirchhoff-Love equations in structural mechanics; the viscous Burgers' equation in fluid dynamics; and the Schrödinger equation in quantum mechanics. Challenges such as high-order derivatives were addressed through their reformulation into coupled systems, as well as problems in high-dimensional domains, such as the case of the hydrogen atom. The experimental results, validated with analytical solutions, demonstrate that the DGM provides accurate and stable approximations, capturing the fundamental physical properties of the systems studied.

Keywords: (*Deep learning, Partial differential equations, Deep Galerkin Method, Neural networks, Mesh-free methods*).

Agradecimientos

Deseo expresar mi más sincero agradecimiento a todas aquellas personas que hicieron posible la realización de esta tesis.

En primer lugar, agradezco profundamente al Dr. Luis Zerón, mi asesor de tesis, por su guía constante, sus valiosas observaciones y su paciencia a lo largo de este proceso. Su intervención fue crucial al asignarme un tema de investigación tan desafiante, así como su orientación en el proceso de selección del contexto de aplicación, lo cual definió el rumbo y la profundidad de este trabajo.

A los profesores de la Maestría en Matemática con orientación en Ingeniería Matemática de la Universidad Nacional Autónoma de Honduras (UNAH), por su dedicación y por compartir generosamente sus conocimientos durante toda mi formación académica.

A mis compañeros más cercanos de la Escuela de Matemática, por su apoyo constante y por motivarme —con exigencia y afecto— a dar este último paso necesario para alcanzar la meta de graduarme.

A mis padres, Ángel Rivera y Esmeralda Muñoz, por su ejemplo de esfuerzo y perseverancia, y por haberme inculcado desde siempre el valor del conocimiento y la superación personal.

Y, de manera muy especial, a mi esposa Anahi Cruz y a nuestro hijo. Su amor, su apoyo incondicional y su confianza en mí han sido el motor fundamental que me ha sostenido en cada etapa de este camino. Esta meta no habría sido posible sin su presencia constante, su paciencia y su fe en mis capacidades.

A todos ellos, mi más profunda gratitud.

Contenido

Resumen	v
1. Introducción	2
2. Ecuaciones diferenciales parciales y sus aplicaciones	5
2.1. El modelo de Euler-Bernoulli	6
2.1.1. Formulación como sistema de ecuaciones diferenciales	9
2.2. El modelo de Rayleigh	11
2.3. El modelo de Timoshenko	12
2.3.1. Formulación como sistema acoplado con fuerza externa	14
2.4. El modelo de Kirchhoff–Love	15
2.4.1. Formulación como sistema acoplado	16
2.5. Ecuación de Burgers viscosa en 2D	17
2.6. La ecuación de Schrödinger	18
2.6.1. Solución por separación de variables	19
2.6.2. Estados fundamentales y excitados	20
2.6.3. Interpretación física	20
2.6.4. Ecuación de Schrödinger para el átomo de hidrógeno	20
2.6.5. Ecuación de Schrödinger para el átomo de hidrógeno con el protón fijo en el origen	22
3. Aprendizaje profundo	24
3.1. Descripción matemática de una red neuronal profunda	24
3.2. Funciones de activación	26
3.3. Inicialización de parámetros: esquema Xavier	30
3.4. Métodos de optimización	31
3.4.1. Descenso del gradiente	31

3.4.2. Método Adam	33
3.5. Función de pérdida y entrenamiento de redes neuronales	34
3.6. Redes neuronales con compuertas LSTM	35
4. Método Deep Galerkin	37
4.1. Fundamentos del método Deep Galerkin	38
4.1.1. Formulación matemática	39
4.1.2. Arquitectura de la red neuronal	40
4.1.3. Minimización y entrenamiento	41
4.1.4. Algoritmo	43
4.2. Adaptación del método Deep Galerkin	45
4.2.1. Ecuación de Euler–Bernoulli	46
4.2.2. Ecuación de Timoshenko	47
4.2.3. Ecuación de Kirchhoff–Love en placas delgadas	48
4.2.4. Ecuación de Burgers viscosa en dos dimensiones	49
4.2.5. Ecuación de Schrödinger unidimensional	50
4.2.6. Ecuación de Schrödinger para el átomo de hidrógeno	51
4.2.7. Ecuación de Schrödinger para el átomo de hidrógeno con protón fijo	53
5. Experimentos numéricos	55
5.1. Ecuación de Euler-Bernoulli como sistema acoplado	57
5.2. Ecuación de Timoshenko	63
5.3. Dinámica de placas delgadas de Kirchhoff–Love en 2D	66
5.4. Ecuación de Burgers viscosa en 2D	72
5.5. Ecuación de Schrödinger	84
5.5.1. Caso unidimensional con potencial cero	84
5.5.2. Caso átomo de hidrógeno con protón y electrón móviles	90
5.5.3. Caso átomo de hidrógeno con protón fijo y electrón móvil	93
5.5.4. Caso: átomo de hidrógeno con protón fijo y electrón móvil en coordenadas esféricas	97
6. Conclusiones y trabajos futuros	101
6.1. Conclusiones	101
6.2. Trabajos futuros	102

A. Algoritmo de Metropolis–Hastings	103
B. Teorema de aproximación universal	105
Bibliografía	107

Listado de Figuras

2.1. Representación esquemática de una viga de Euler-Bernoulli. La línea central corresponde al eje neutro, cuya deformación vertical se describe mediante la función $w(x, t)$. La inclinación de la sección transversal se aproxima por el ángulo $\theta(x, t)$, el cual, bajo la hipótesis de pequeños giros, satisface $\theta(x, t) \approx \frac{\partial w}{\partial x}$	7
3.1. Ilustración de una red neuronal profunda <i>fully connected</i> . La arquitectura se denota por (l_0, l_1, \dots, l_L) : capa de entrada con l_0 neuronas, capas ocultas intermedias con l_1, \dots, l_{L-1} neuronas y capa de salida con l_L neuronas.	26
3.2. Función sigmoide $\sigma(x) = \frac{1}{1+e^{-x}}$	27
3.3. Función tangente hiperbólica $\tanh(x)$	27
3.4. Función ReLU $\text{ReLU}(x) = \max(0, x)$	28
3.5. Función Leaky ReLU con $\alpha = 0.1$	28
3.6. Función SiLU (Swish): $\text{SiLU}(x) = x \sigma(x)$	29
3.7. Función GELU	30
5.1. Comparación de la evolución de la pérdida de entrenamiento en función del número de épocas para la viga de Euler–Bernoulli en 1D.	59
5.2. Ejemplo de la selección de puntos de muestreo aleatorios en el dominio espacio-temporal. Los puntos azules corresponden al interior, los puntos rojos a la condición inicial en $t = 0$, y los puntos negros a las condiciones de frontera en $x = 0$ y $x = \pi$	60
5.3. Ecuación de Euler-Bernoulli; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $).	61

5.4. Ecuación de Euler-Bernoulli - 8k épocoas con decaimiento exponencial; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $)	62
5.5. Comparación de la evolución de la pérdida de entrenamiento en función del número de épocas para la viga de Timoshenko en 1D.	64
5.6. Ecuación de Timoshenko; Izquierda: Solución aproximada θ^* y w^* por DGM Centro: Solución analítica θ y w Derecha: Error absoluto en la predicción ($ \theta - \theta^* $) y ($ w - w^* $)	65
5.7. Evolución de la pérdida de entrenamiento en función del número de épocas - Ecuación de Kirchhoff-Love en 2D - Tiempo de entrenamiento 73 minutos	68
5.8. Ecuación de Kirchhoff-Love en $t = 0.0$; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $)	69
5.9. Ecuación de Kirchhoff-Love en $t = 0.25$; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $)	70
5.10. Ecuación de Kirchhoff-Love en $t = 0.50$; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $)	71
5.11. Ecuación de Kirchhoff-Love en $t = 0.75$; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $)	72
5.12. Ecuación de Kirchhoff-Love en $t = 1.0$; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $)	73
5.13. Evolución de la pérdida de entrenamiento en función del número de épocas - Ecuación de Burgers viscosa en 2D - Tiempo de entrenamiento 11 minutos	74
5.14. Ecuación de Burgers en $t = 0.00$; Izquierda: Solución aproximada u^* y v^* por DGM Centro: Solución analítica u y v Derecha: Error absoluto en la predicción ($ u - u^* $) y ($ v - v^* $)	74

5.15. Ecuación de Burgers en $t = 0.25$; Izquierda: Solución aproximada u^* y v^* por DGM Centro: Solución analítica u y v Derecha: Error absoluto en la predicción ($ u - u^* $) y ($ v - v^* $)	75
5.16. Ecuación de Burgers en $t = 0.50$; Izquierda: Solución aproximada u^* y v^* por DGM Centro: Solución analítica u y v Derecha: Error absoluto en la predicción ($ u - u^* $) y ($ v - v^* $)	76
5.17. Ecuación de Burgers en $t = 0.75$; Izquierda: Solución aproximada u^* y v^* por DGM Centro: Solución analítica u y v Derecha: Error absoluto en la predicción ($ u - u^* $) y ($ v - v^* $)	77
5.18. Ecuación de Burgers en $t = 1.00$; Izquierda: Solución aproximada u^* y v^* por DGM Centro: Solución analítica u y v Derecha: Error absoluto en la predicción ($ u - u^* $) y ($ v - v^* $)	78
5.19. Ecuación de Burgers viscosa; evolución del error relativo L_2 en el tiempo $[0, 1]$	79
5.20. Ecuación de Burgers viscosa; evolución del error relativo L_2 en el tiempo $[0, 1]$	80
5.21. Ecuación de Burgers viscosa en $t = 0.0$; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $)	80
5.22. Ecuación de Burgers viscosa en $t = 0.25$; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $)	81
5.23. Ecuación de Burgers viscosa en $t = 0.50$; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $)	81
5.24. Ecuación de Burgers viscosa en $t = 0.75$; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $)	82
5.25. Ecuación de Burgers viscosa en $t = 1.00$; Izquierda: Solución aproximada w^* y M^* por DGM Centro: Solución analítica w y M Derecha: Error absoluto en la predicción ($ w - w^* $) y ($ M - M^* $)	82

5.26. Evolución de la pérdida de entrenamiento en función del número de épocas - Ecuación de Schrödinger unidimensional - Tiempo de entrenamiento 24 minutos	85
5.27. Ecuación de Schrödinger unidimensional; evolución del error relativo L_2 en el tiempo $[0, 1]$	86
5.28. Evolución de la pérdida de entrenamiento en función del número de épocas - Ecuación de Schrödinger unidimensional - Tiempo de entrenamiento 88 minutos	86
5.29. Ecuación de Schrödinger; Izquierda: Solución exacta parte real e imaginaria u y v Derecha: Solución aproximada por DGM u^* y v^* . Abajo: Errores absolutos parte real e imaginaria	87
5.30. Ecuación de Schrödinger unidimensional; evolución del error relativo L_2 en el tiempo $[0, 1]$	88
5.31. Ecuación de Schrödinger; Izquierda: Solución exacta parte real e imaginaria u y v Derecha: Solución aproximada por DGM u^* y v^* . Abajo: Errores absolutos parte real e imaginaria	89
5.32. Evolución de la pérdida de entrenamiento - Ecuación de Schrödinger para el átomo de hidrógeno con protón y electrón móviles - Tiempo de entrenamiento 40 minutos.	91
5.33. Visualización de la densidad electrónica del átomo de hidrógeno estimada mediante DGM	92
5.34. Perfil radial de la función de onda $ \psi(r) ^2$ en función de la distancia $r = \mathbf{r}_e - \mathbf{r}_p $	93
5.35. Evolución de la pérdida de entrenamiento - Ecuación de Schrödinger para el átomo de hidrógeno con protón fijo	94
5.36. Densidad de probabilidad electrónica estimada mediante DGM para el caso de protón fijo	95
5.37. Comparación del perfil radial $ \psi(r) ^2$ obtenido por DGM con la solución analítica $R_{1s}(r) = 2e^{-r}$	96
5.38. Evolución de la pérdida de entrenamiento para el caso de protón fijo (DGM en coordenadas esféricas).	98
5.39. Densidad de probabilidad electrónica estimada mediante DGM (protón fijo).	99

5.40. Perfil radial en coordenadas esféricas $ \psi(r) ^2$ obtenido por DGM vs. solución analítica $R_{1s}(r) = 2e^{-r}$	99
--	----

Listado de Tablas

5-1. Hiperparámetros utilizados para la resolución de la ecuación de Euler-Bernoulli	58
5-2. Errores L_2 absolutos y relativos de la validación en ecuación de Euler-Bernoulli	60
5-3. Hiperparámetros de la Ecuación de Timoshenko	64
5-4. Errores L_2 absolutos y relativos de la validación en ecuación de Timoshenko	64
5-5. Hiperparámetros para la Placa de Kirchhoff-Love 2D	68
5-6. Errores L_2 absolutos y relativos de la validación en la ecuación de Kirchhoff-Love usando la primera configuración de la tabla 5-5	68
5-7. Errores L_2 absolutos y relativos de la validación en la ecuación de Kirchhoff-Love usando la segunda configuración de la tabla 5-5	69
5-8. Hiperparámetros para la Ecuación de Burgers 2D	73
5-9. Errores L_2 absolutos y relativos de la validación en la ecuación de Burgers viscosa bidimensional	75
5-10. Errores L_2 absolutos y relativos de la validación en la ecuación de Burgers viscosa bidimensional - Afinamiento de hiperparámetros	79
5-11. Hiperparámetros: Ecuación de Schrödinger 1D	85
5-12. Hiperparámetros: Ecuación de Schrödinger - Átomo de hidrógeno con protón y electrón móviles	90
5-13. Hiperparámetros: Ecuación de Schrödinger - Átomo de hidrógeno con protón fijo y electrón móvil	94
5-14. Hiperparámetros: Ecuación de Schrödinger (protón fijo, electrón móvil en coordenadas esféricas).	98

Capítulo 1

Introducción

Las ecuaciones diferenciales parciales (EDPs) son el lenguaje matemático con las cuales la ciencia y la ingeniería describen el universo. Desde la flexión de una viga en un puente hasta la propagación de una onda cuántica, pasando por la dinámica de los fluidos y los mercados financieros, las EDPs constituyen la herramienta fundamental para modelar sistemas cuya evolución depende de múltiples variables, como el tiempo y el espacio. La resolución de estas ecuaciones, que implica hallar funciones que satisfagan tanto la dinámica interna del sistema como sus condiciones de frontera e iniciales, ha sido uno de los motores centrales del desarrollo científico y computacional durante siglos.

Históricamente, la resolución de EDPs se ha apoyado en métodos numéricos tradicionales como el método de elementos finitos (FEM), diferencias finitas (FDM) y volúmenes finitos (FVM). Si bien estos enfoques han demostrado ser increíblemente exitosos, enfrentan una barrera formidable: la *maldición de la dimensionalidad* [1, 2]. Este fenómeno se manifiesta en el crecimiento exponencial del coste computacional al generar una malla o discretización del dominio a medida que aumenta el número de variables independientes (d). Es decir, si para discretizar una dimensión se necesitan N puntos, para dos dimensiones se requerirían N^2 , para tres N^3 , y así sucesivamente hasta N^d . Este crecimiento explosivo hace que los métodos tradicionales basados en mallas se vuelvan computacionalmente intratables en problemas de alta dimensión. Esta limitación es particularmente severa en campos como la mecánica cuántica, donde la función de onda de un sistema de N partículas reside en un espacio de $3N$ -dimensional.

En la última década, ha surgido un nuevo paradigma con el potencial de superar estas limitaciones: el aprendizaje profundo (Deep Learning). Las redes neuronales profundas (DNNs), gracias a su probada capacidad como **aproximadores universales** de funciones

complejas [3, 4] (ver apéndice B), ofrecen un marco alternativo para la resolución de EDPs. En lugar de discretizar el dominio, estos métodos reformulan el problema como una tarea de optimización no supervisada guiada por restricciones físicas. La idea central es entrenar una red neuronal para que su salida no solo se ajuste a las condiciones iniciales y de frontera, sino que también minimice el residuo de la propia ecuación diferencial en un conjunto de puntos muestrado aleatoriamente dentro del dominio.

Dentro de este emergente campo, el Método Deep Galerkin (DGM), propuesto por Sirignano y Spiliopoulos [5], se destaca por su elegancia y robustez. El DGM es un método intrínsecamente libre de mallas que utiliza una arquitectura de red neuronal profunda para aproximar la solución de la EDP en todo su dominio espacio-temporal. Su enfoque, que se basa en la diferenciación automática para calcular las derivadas requeridas por el operador diferencial, permite abordar de manera unificada ecuaciones de alta dimensión y con estructuras no lineales complejas.

El objetivo principal de esta tesis es realizar una validación sistemática y exhaustiva de la eficacia y versatilidad del Deep Galerkin Method como herramienta para la resolución de EDPs representativas de distintos dominios de la física y la ingeniería. Para ello, se plantean los siguientes objetivos específicos:

- Implementar el método DGM para resolver ecuaciones de segundo y cuarto orden, abordando los desafíos que presentan las derivadas espaciales de orden superior.
- Analizar el rendimiento y la precisión del método en problemas lineales y no lineales. La precisión se mide comparando las soluciones con resultados analíticos conocidos, mientras que el rendimiento del entrenamiento se evalúa mediante el análisis de la convergencia de la función de pérdida.
- Investigar la capacidad del DGM para resolver sistemas de EDPs acopladas y problemas en dominios de alta dimensión, donde los métodos clásicos son computacionalmente inviables.
- Analizar el impacto de la arquitectura de la red, los hiperparámetros de entrenamiento y las estrategias de aprendizaje en el rendimiento de la aproximación.

Para cumplir estos objetivos, el alcance de este trabajo abarca un abanico diverso y representativo de modelos matemáticos. En el campo de la mecánica estructural, se abordarán las ecuaciones de Euler-Bernoulli, Timoshenko y Kirchhoff-Love, que modelan el comportamiento dinámico de vigas y placas. Se explorará la estrategia de reformular estas EDPs de cuarto orden en sistemas acoplados de segundo orden para facilitar su

implementación. En la dinámica de fluidos, se estudiará la ecuación de Burgers viscosa en dos dimensiones, un modelo canónico para fenómenos de advección-difusión no lineales. Finalmente, en el ámbito de la mecánica cuántica, se resolverá la ecuación de Schrödinger dependiente del tiempo para tres casos de complejidad creciente: una partícula libre en una dimensión, el átomo de hidrógeno con protón fijo, y el problema completo de dos cuerpos (protón y electrón móviles) en un dominio de seis dimensiones espaciales más el tiempo.

La contribución de esta investigación es doble. Por un lado, proporciona una validación empírica rigurosa del método DGM en un conjunto de problemas más amplio y variado de lo que usualmente se reporta en la literatura, destacando su aplicabilidad tanto en la ingeniería clásica como en la física fundamental. Por otro lado, ofrece un análisis práctico y detallado de las estrategias de implementación y entrenamiento que resultan más efectivas para cada clase de problema, sirviendo como una guía valiosa para futuros investigadores y profesionales que deseen aplicar estas técnicas. Este trabajo demuestra que el aprendizaje profundo no es solo una herramienta para el análisis de datos, sino también una poderosa alternativa para la simulación de sistemas físicos complejos.

La tesis se estructura de la siguiente manera. El Capítulo 2 presenta el marco teórico de las ecuaciones diferenciales parciales estudiadas. El Capítulo 3 introduce los fundamentos del aprendizaje profundo y las redes neuronales. El Capítulo 4 detalla la formulación matemática y algorítmica del método Deep Galerkin. El Capítulo 5 presenta los resultados de los experimentos numéricos para cada una de las ecuaciones, evaluando la precisión y el rendimiento del método. Finalmente, el Capítulo 6 expone las conclusiones del trabajo y delinea posibles líneas de investigación futura.

Capítulo 2

Ecuaciones diferenciales parciales y sus aplicaciones

Las ecuaciones diferenciales parciales (EDP) constituyen una herramienta fundamental en la modelación matemática de fenómenos físicos, biológicos, financieros e ingenieriles. Estas ecuaciones permiten describir la evolución de sistemas que dependen simultáneamente de múltiples variables independientes, tales como el tiempo y el espacio. Resolver una EDP implica encontrar funciones que satisfagan no solo relaciones diferenciales internas, sino también condiciones iniciales y de frontera que reflejen el comportamiento real del sistema modelado.

En el ámbito de la ingeniería estructural, las EDP gobiernan el comportamiento dinámico de elementos sometidos a carga, como las vigas. Modelos clásicos como las ecuaciones de Euler–Bernoulli, Rayleigh y Timoshenko describen la deflexión transversal de estos elementos bajo cargas distribuidas, capturando distintos niveles de complejidad física: desde la flexión pura en estructuras esbeltas hasta efectos significativos de corte y rotación en estructuras más robustas [6, 7, 8]. Asimismo, la teoría de Kirchhoff–Love [9, 10] extiende este análisis al caso bidimensional de placas delgadas, dando lugar a sistemas acoplados de cuarto orden que plantean mayores retos computacionales.

Avanzando en el espectro de aplicaciones, nos encontramos con la ecuación de Burgers viscosa [11], la cual surge como un modelo simplificado de las ecuaciones de Navier–Stokes. Esta EDP no lineal combina advección y difusión, y resulta fundamental para el estudio de la dinámica de fluidos. Su estructura representa un punto de inflexión conceptual en esta tesis: nos alejamos del mundo estructural esencialmente lineal y entramos en el terreno de la no linealidad intrínseca y los fenómenos de transporte, donde surgen fenómenos como

ondas de choque, capas límite y turbulencia moderada.

Finalmente, en la cúspide de esta progresión se encuentra la ecuación de Schrödinger [12], pilar de la mecánica cuántica. Esta ecuación describe la evolución temporal del estado cuántico de una partícula en presencia de un potencial, involucrando funciones de onda complejas y dominios de alta dimensión. A diferencia de los modelos anteriores, aquí la solución no representa una magnitud física directa como desplazamiento o velocidad, sino una amplitud de probabilidad. La resolución de esta ecuación, en una o más dimensiones espaciales —incluyendo casos de interés como el átomo de hidrógeno—, demanda enfoques capaces de capturar estructuras oscilatorias finas y propiedades físicas como la conservación de la norma y la energía.

En las secciones posteriores se presentará, para cada una de las ecuaciones diferenciales parciales estudiadas, una exposición detallada que incluye su deducción a partir de principios físicos fundamentales, así como su reformulación matemática en una forma adecuada para su resolución mediante métodos numéricos. Este proceso abarca la identificación del dominio, las condiciones iniciales y de frontera, la transformación en sistemas equivalentes de menor orden cuando sea posible realizar dicha transformación.

2.1. El modelo de Euler-Bernoulli

La **ecuación de Euler-Bernoulli**, la cual fue formulada inicialmente por Leonhard Euler y Daniel Bernoulli en el siglo XVIII y sigue siendo fundamental en el análisis de estructuras esbeltas [7, 13], constituye una aproximación clásica para describir la flexión de vigas bajo condiciones de deformaciones y ángulos de giro relativamente pequeños. A fin de derivar la EDP correspondiente, se formulan las siguientes hipótesis simplificadoras:

1. Los desplazamientos verticales de cualquier punto en una sección transversal de la viga son lo bastante reducidos como para considerar que siguen la misma dirección longitudinal x .
2. El desplazamiento lateral (en dirección y) se desestima, de modo que no se incluyen grados de libertad en esa dirección.
3. Las secciones transversales, inicialmente perpendiculares al eje neutro de la viga, permanecen planas y esencialmente ortogonales al eje luego de la deformación; con ello se ignora la rotación significativa y se descartan los efectos apreciables de corte.

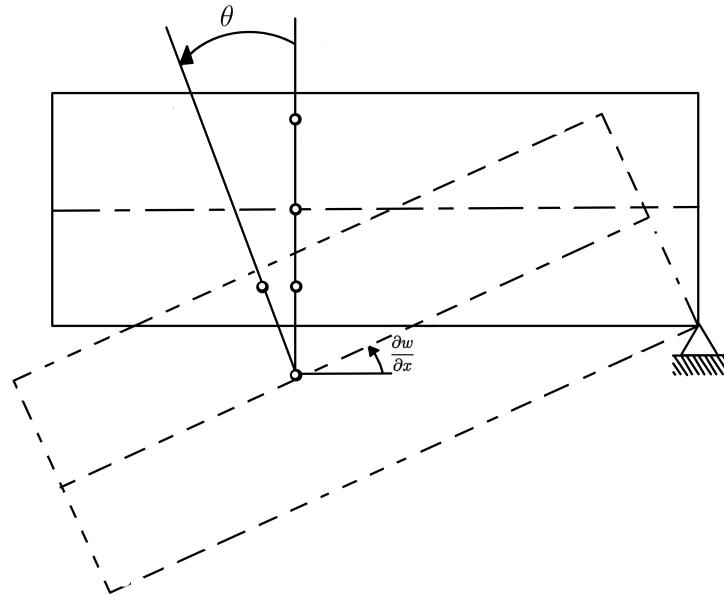


Figura 2.1: Representación esquemática de una viga de Euler-Bernoulli. La línea central corresponde al eje neutro, cuya deformación vertical se describe mediante la función $w(x, t)$. La inclinación de la sección transversal se aproxima por el ángulo $\theta(x, t)$, el cual, bajo la hipótesis de pequeños giros, satisface $\theta(x, t) \approx \frac{\partial w}{\partial x}$.

Sea $w(x, t)$ la función que describe la deflexión de la viga en el tiempo t sobre la abscisa x , con $0 \leq x \leq L$. Como se ilustra en la figura 2.1, al considerar pequeños ángulos de giro, la pendiente de la línea elástica resulta una buena aproximación del ángulo de rotación de la sección transversal, esto es:

$$\theta(x, t) \approx \frac{\partial w}{\partial x}.$$

Además, si suponemos que la deformación se produce de forma predominante en la dirección vertical, la cinemática indica que

$$u(x, y, z, t) = -z \frac{\partial w}{\partial x}, \quad v(x, y, z, t) = 0, \quad w(x, y, z, t) = w(x, t),$$

donde (u, v, w) representa los desplazamientos en cada eje.

Para caracterizar la energía total en la viga, consideremos que $\rho(x)$ es la densidad de masa (lineal) y $EI(x)$ es el producto del módulo de Young E y el momento de inercia I . Se define la energía

$$E = E(w, \frac{\partial w}{\partial t}),$$

que, conforme a la teoría de Euler-Bernoulli, puede modelarse mediante la integral

$$\mathbb{E} = \frac{1}{2} \int_0^L \left[\rho \left(\frac{\partial w}{\partial t} \right)^2 + EI \left(\frac{\partial^2 w}{\partial x^2} \right)^2 \right] dx, \quad (2.1)$$

donde el término $\rho \left(\frac{\partial w}{\partial t} \right)^2$ alude a la energía cinética, y $EI \left(\frac{\partial^2 w}{\partial x^2} \right)^2$ a la energía potencial de flexión.

Asumiendo que $w(x, t)$ es suficientemente suave, derivamos con respecto al tiempo la ecuación (2.1):

$$\frac{d\mathbb{E}}{dt} = \int_0^L \left[\rho \left(\frac{\partial w}{\partial t} \right) \left(\frac{\partial^2 w}{\partial t^2} \right) + EI \left(\frac{\partial^2 w}{\partial x^2} \right) \left(\frac{\partial^3 w}{\partial t \partial x^2} \right) \right] dx.$$

La segunda parte de la integral anterior se trata mediante integración por partes dos veces:

$$\begin{aligned} \int_0^L \left(\frac{\partial^2 w}{\partial x^2} \right) \left(\frac{\partial^3 w}{\partial t \partial x^2} \right) dx &= \left[\left(\frac{\partial^2 w}{\partial x^2} \right) \left(\frac{\partial^2 w}{\partial t \partial x} \right) \right]_0^L - \int_0^L \left(\frac{\partial^2 w}{\partial t \partial x} \right) \left(\frac{\partial^3 w}{\partial x^3} \right) dx, \\ &= \left(\frac{\partial^2 w}{\partial x^2} \right) \left(\frac{\partial^2 w}{\partial t \partial x} \right) \Big|_0^L - \left[\left(\frac{\partial^3 w}{\partial x^3} \right) \left(\frac{\partial w}{\partial t} \right) \right]_0^L + \int_0^L \left(\frac{\partial w}{\partial t} \right) \left(\frac{\partial^4 w}{\partial x^4} \right) dx. \end{aligned}$$

Así, la derivada de la energía en el tiempo cobra la forma

$$\frac{d\mathbb{E}}{dt} = \int_0^L \left(\frac{\partial w}{\partial t} \right) \left(\rho \frac{\partial^2 w}{\partial t^2} + EI \frac{\partial^4 w}{\partial x^4} \right) dx + EI \left(\frac{\partial^2 w}{\partial x^2} \right) \left(\frac{\partial^2 w}{\partial t \partial x} \right) \Big|_0^L - EI \left(\frac{\partial^3 w}{\partial x^3} \right) \left(\frac{\partial w}{\partial t} \right) \Big|_0^L.$$

En el caso de una viga empotrada en ambos extremos ($x = 0$ y $x = L$), se imponen

$$w(0) = w'(0) = 0, \quad w(L) = w'(L) = 0,$$

lo que anula los términos en los límites. Suponiendo, además, que no existe aporte ni disipación externa de energía dentro del cuerpo, el factor que multiplica $\frac{\partial w}{\partial t}$ se fuerza a ser nulo en el interior. De esta forma, se llega a la ecuación de Euler-Bernoulli:

$$\rho \frac{\partial^2 w}{\partial t^2} + EI \frac{\partial^4 w}{\partial x^4} = 0 \quad (2.2)$$

la cual rige la dinámica de flexión de la viga con las restricciones de contorno anteriores. Este modelo resulta fundamental en el análisis de estructuras esbeltas presentes en puentes, máquinas y edificios, proporcionando una aproximación confiable siempre que las hipótesis iniciales de la teoría sean válidas.

2.1.1. Formulación como sistema de ecuaciones diferenciales

La ecuación de Euler-Bernoulli en su forma (2.2) es una ecuación en derivadas parciales de cuarto orden en el espacio. Para ciertos análisis y, en particular, para la aplicación de métodos numéricos, es conveniente reformularla como un sistema de ecuaciones de menor orden.

Una forma de lograr esto es introduciendo una variable auxiliar $M(x, t)$ que represente la curvatura de la viga. Si definimos:

$$M(x, t) = \frac{\partial^2 w(x, t)}{\partial x^2}$$

podemos observar que esta nueva variable está directamente relacionada con el momento flector, donde el momento flector M en un punto a lo largo de una viga es una medida de las fuerzas de flexión internas que actúan sobre la sección transversal de la viga en ese punto. Se define a través de su relación con la curvatura de la viga..

Sustituyendo esta definición en la ecuación de Euler-Bernoulli (2.2), y considerando además la presencia de una fuerza externa $q(x, t)$ distribuida transversalmente, obtenemos un sistema de dos ecuaciones en derivadas parciales de segundo orden acopladas:

$$\begin{cases} \frac{\partial^2 w(x, t)}{\partial x^2} = M(x, t), \\ \rho \frac{\partial^2 w(x, t)}{\partial t^2} + EI \frac{\partial^2 M(x, t)}{\partial x^2} = q(x, t). \end{cases}$$

Este sistema es matemáticamente equivalente a la ecuación original y describe la dinámica de la viga cuando está sujeta a una excitación externa $q(x, t)$.

La función $q(x, t)$ puede representar distintas situaciones físicas, tales como:

- una carga uniforme constante en el tiempo: $q(x, t) = q_0$,
- una carga espacialmente variable: $q(x, t) = q_0 \sin\left(\frac{\pi x}{L}\right)$,
- una carga pulsante o de impacto: $q(x, t) = q_0 \delta(t - t_0)$,
- o una carga oscilante en el tiempo: $q(x, t) = q_0 \sin(\omega t)$.

Este término permite simular fenómenos más complejos como vibraciones inducidas, cargas dinámicas o interacción con el entorno.

La deducción de la ecuación de Euler-Bernoulli se basó en el caso específico de una viga empotrada en ambos extremos. Sin embargo, el modelo es aplicable a una variedad de configuraciones de soporte. Las condiciones de frontera se obtienen al analizar los términos evaluados en los límites ($x = 0$ y $x = L$) que surgieron durante la integración por partes

de la derivada de la energía:

$$\frac{dE}{dt} = \int_0^L \left(\frac{\partial w}{\partial t} \right) \left(\rho \frac{\partial^2 w}{\partial t^2} + EI \frac{\partial^4 w}{\partial x^4} \right) dx + EI \left(\frac{\partial^2 w}{\partial x^2} \right) \Big|_0^L - EI \left(\frac{\partial^3 w}{\partial x^3} \right) \left(\frac{\partial w}{\partial t} \right) \Big|_0^L.$$

Para un sistema conservativo sin trabajo externo en los extremos, la suma de estos términos debe anularse. Esto se logra imponiendo dos condiciones en cada extremo de la viga ($x = 0$ y $x = L$). Estas condiciones pueden ser de carácter cinemático (desplazamiento y giro) o estático (fuerza cortante y momento flector). Las condiciones de frontera más comunes son:

- **Extremo empotrado:** El desplazamiento y el giro son nulos:

$$w = 0, \quad \frac{\partial w}{\partial x} = 0.$$

- **Extremo articulado o con apoyo simple:** El desplazamiento es nulo y no hay momento flector, permitiendo el giro libre:

$$w = 0, \quad EI \frac{\partial^2 w}{\partial x^2} = 0.$$

- **Extremo libre:** No hay momento flector ni fuerza cortante:

$$EI \frac{\partial^2 w}{\partial x^2} = 0, \quad \frac{\partial}{\partial x} \left(EI \frac{\partial^2 w}{\partial x^2} \right) = 0.$$

La combinación de estas condiciones en $x = 0$ y $x = L$ define el comportamiento estructural de la viga.

Dado que la ecuación de Euler-Bernoulli es de segundo orden con respecto al tiempo, su solución única requiere la especificación de dos condiciones iniciales a lo largo de toda la viga. Estas condiciones definen el estado de la viga en el instante $t = 0$:

- **Deflexión inicial:** Se debe prescribir la forma o deflexión inicial de la viga para todo $x \in [0, L]$:

$$w(x, 0) = f(x).$$

- **Velocidad inicial:** Se debe prescribir la velocidad transversal inicial de cada punto de la viga para todo $x \in [0, L]$:

$$\frac{\partial w}{\partial t}(x, 0) = g(x).$$

Las funciones $f(x)$ y $g(x)$ representan perfiles conocidos y deben ser consistentes con las condiciones de frontera impuestas en el problema.

2.2. El modelo de Rayleigh

El modelo fue introducido por Lord Rayleigh y constituye una extensión del modelo de Euler-Bernoulli al incorporar el efecto de la inercia rotacional en los movimientos transversales de la viga [14, 15] y se remonta a inicios del siglo XX. Para derivarlo, se asume que el elemento de viga, de ancho δ y centrado sobre el eje neutro, dispone adicionalmente de un momento de masa de inercia alrededor de dicho eje, igual a $\delta I_\rho(x)$.

En consecuencia, la energía total de la viga queda descrita por la integral

$$\mathsf{E} = \frac{1}{2} \int_0^L \left[\rho \left(\frac{\partial w}{\partial t} \right)^2 + I_\rho \left(\frac{\partial^2 w}{\partial t \partial x} \right)^2 + EI \left(\frac{\partial^2 w}{\partial x^2} \right)^2 \right] dx \quad (2.3)$$

donde I_ρ es una constante de inercia rotatoria, que con frecuencia se expresa como $I_\rho = \rho I/A$, y EI denota, como antes, el producto entre el módulo de Young E y el momento de inercia I . Al igual que en el modelo de Euler-Bernoulli, sea $w(x, t)$ una función de desplazamiento suficientemente suave en el intervalo $0 \leq x \leq L$ y tiempo $t \geq 0$.

Procediendo a la diferenciación de (2.3) con respecto a t , obtenemos:

$$\frac{d\mathsf{E}}{dt} = \int_0^L \left[\rho \left(\frac{\partial w}{\partial t} \right) \left(\frac{\partial^2 w}{\partial t^2} \right) + I_\rho \left(\frac{\partial^2 w}{\partial t \partial x} \right) \left(\frac{\partial^3 w}{\partial t^2 \partial x} \right) + EI \left(\frac{\partial^2 w}{\partial x^2} \right) \left(\frac{\partial^3 w}{\partial t \partial x^2} \right) \right] dx.$$

El segundo término de la integral se maneja mediante integración por partes:

$$\int_0^L \left(\frac{\partial^2 w}{\partial x^2} \right) \left(\frac{\partial^3 w}{\partial t \partial x^2} \right) dx = \left[\left(\frac{\partial^2 w}{\partial x^2} \right) \left(\frac{\partial^2 w}{\partial t \partial x} \right) \right]_0^L - \int_0^L \left(\frac{\partial^2 w}{\partial t \partial x} \right) \left(\frac{\partial^3 w}{\partial x^3} \right) dx,$$

y análogamente para el tercer término, con un procedimiento de integración por partes reiterado.

Tras reagrupar y cancelar términos en los extremos mediante las condiciones de frontera de la viga, se halla que

$$\begin{aligned} \frac{d\mathsf{E}}{dt} &= \int_0^L \left[\left(\frac{\partial w}{\partial t} \right) \left(\rho \frac{\partial^2 w}{\partial t^2} - I_\rho \frac{\partial^4 w}{\partial t^2 \partial x^2} + EI \frac{\partial^4 w}{\partial x^4} \right) \right] dx \\ &\quad + I_\rho \left(\frac{\partial^3 w}{\partial t^2 \partial x} \right) \left(\frac{\partial w}{\partial t} \right) \Big|_0^L + EI \left(\frac{\partial^2 w}{\partial x^2} \right) \left(\frac{\partial^2 w}{\partial x \partial t} \right) \Big|_0^L - EI \left(\frac{\partial^3 w}{\partial x^3} \right) \left(\frac{\partial w}{\partial t} \right) \Big|_0^L. \end{aligned}$$

donde los términos evaluados en $x = 0$ y $x = L$, que tienden a anularse bajo condiciones de contorno apropiadas (por ejemplo, para una viga empotrada con $w(0) = w'(0) = 0$ y $w(L) = w'(L) = 0$).

Suponiendo la ausencia de intercambio neto de energía con el entorno (no hay trabajo externo realizado sobre la viga), se obtiene finalmente la llamada *ecuación de Rayleigh*:

$$\rho \frac{\partial^2 w}{\partial t^2} - I_p \frac{\partial^4 w}{\partial t^2 \partial x^2} + EI \frac{\partial^4 w}{\partial x^4} = 0 \quad (2.4)$$

que refleja el efecto adicional de la inercia rotatoria, en contraste con la clásica ecuación de Euler–Bernoulli. Este modelo es útil para describir vigas que requieran mayor fidelidad en su comportamiento dinámico, especialmente cuando la rotación y la inercia transversal no pueden ignorarse.

2.3. El modelo de Timoshenko

El modelo fue desarrollado por Stephen P. Timoshenko en la década de 1920 y representa una mejora sustancial, en la predicción del comportamiento dinámico de vigas cortas y gruesas, al incluir efectos de corte y rotación [8, 16]. El modelo de Timoshenko extiende el análisis de vigas más allá del modelo de Euler-Bernoulli al incluir la deformación por corte y la rotación de la sección transversal, lo que lo hace adecuado para analizar vigas más gruesas o de materiales compuestos. Este modelo se basa en tres hipótesis:

1. Comparte las hipótesis 1 y 2 del modelo de Euler-Bernoulli: los desplazamientos verticales de los puntos de una sección transversal son pequeños y no hay desplazamientos laterales.
2. A diferencia de Euler-Bernoulli, el modelo de Timoshenko introduce una hipótesis adicional: las secciones transversales no permanecen necesariamente normales al eje de la viga después de la deformación, lo que permite una mayor precisión en la modelización de vigas gruesas.

En el modelo de Timoshenko, la rotación total del elemento de la viga está descrita por una variable independiente, el ángulo θ , que no necesariamente coincide con la pendiente de la deformada del eje de la viga. La deformación de corte se define como:

$$\gamma = \frac{\partial w}{\partial x} - \theta,$$

y la curvatura de flexión es

$$\kappa = \frac{\partial \theta}{\partial x}.$$

La energía total del sistema incluye términos para la energía cinética asociada a las velocidades traslacionales y rotacionales, la deformación por corte y la flexión de la viga. La expresión para la energía total E es:

$$E = \frac{1}{2} \int_0^L \left[\rho A \left(\frac{\partial w}{\partial t} \right)^2 + I_\rho \left(\frac{\partial \theta}{\partial t} \right)^2 + GA \left(\frac{\partial w}{\partial x} - \theta \right)^2 + EI \left(\frac{\partial \theta}{\partial x} \right)^2 \right] dx$$

donde:

- ρA es la masa por unidad de longitud,
- I_ρ es el momento de inercia rotacional por unidad de longitud,
- GA es la rigidez a corte (donde G es el módulo de rigidez y A es el área de la sección transversal),
- EI es la rigidez flexional (con E el módulo de Young y I el momento de inercia).

Al aplicar el principio de Hamilton o las ecuaciones de Euler–Lagrange al funcional energético, se obtienen las ecuaciones diferenciales parciales acopladas que describen el desplazamiento y la rotación en función de la posición y el tiempo:

$$\rho A \frac{\partial^2 w}{\partial t^2} - \frac{\partial}{\partial x} \left(GA \left(\frac{\partial w}{\partial x} - \theta \right) \right) = 0, \quad (2.5)$$

$$I_\rho \frac{\partial^2 \theta}{\partial t^2} - \frac{\partial}{\partial x} \left(EI \frac{\partial \theta}{\partial x} \right) + GA \left(\frac{\partial w}{\partial x} - \theta \right) = 0. \quad (2.6)$$

Estas ecuaciones representan el equilibrio dinámico de la viga bajo flexión y corte. La primera describe la dinámica transversal del eje de la viga, mientras que la segunda caracteriza la evolución de la rotación de la sección transversal.

Para vigas muy delgadas, el término de corte $GA(w_x - \theta)$ tiende a ser grande, forzando la condición $\theta \approx w_x$. En este caso, el modelo de Timoshenko se reduce al modelo de Euler–Bernoulli. Asimismo, al eliminar la variable de rotación θ de las ecuaciones anteriores y suponer propiedades constantes, se obtiene una única ecuación diferencial parcial en w , conocida como la ecuación de Timoshenko simplificada:

$$\rho \frac{\partial^2 w}{\partial t^2} - I_\rho \frac{\partial^4 w}{\partial t^2 \partial x^2} + EI \frac{\partial^4 w}{\partial x^4} + \frac{\rho}{GA} \left(I_\rho \frac{\partial^4 w}{\partial t^4} - EI \frac{\partial^4 w}{\partial t^2 \partial x^2} \right) = 0. \quad (2.7)$$

Esta ecuación puede considerarse una extensión directa de la ecuación de Rayleigh, que incorpora tanto la inercia rotacional como los efectos de corte en una sola formulación. El modelo de Timoshenko, por tanto, unifica los enfoques previos al permitir el análisis de vigas gruesas o cortas con mayor precisión.

2.3.1. Formulación como sistema acoplado con fuerza externa

Para facilitar la aplicación de métodos numéricos modernos, como los basados en aprendizaje profundo, es útil trabajar con el modelo de Timoshenko como un sistema de ecuaciones diferenciales de segundo orden, en forma acoplada. Considerando una fuerza externa $q(x, t)$ distribuida transversalmente, el sistema se expresa como:

$$\begin{cases} \rho A \frac{\partial^2 w}{\partial t^2} = \frac{\partial}{\partial x} \left(GA \left(\theta - \frac{\partial w}{\partial x} \right) \right) + q(x, t), \\ I_\rho \frac{\partial^2 \theta}{\partial t^2} = \frac{\partial}{\partial x} \left(EI \frac{\partial \theta}{\partial x} \right) - GA \left(\theta - \frac{\partial w}{\partial x} \right). \end{cases} \quad (2.8)$$

Este sistema describe de manera acoplada la evolución temporal del desplazamiento transversal $w(x, t)$ y de la rotación de la sección transversal $\theta(x, t)$, incorporando los efectos de flexión, corte y carga externa distribuida. Su estructura lo hace especialmente adecuada para su tratamiento computacional.

Las condiciones de frontera necesarias para garantizar una solución bien planteadas dependen del tipo de soporte en los extremos $x = 0$ y $x = L$. Cada extremo debe tener dos condiciones: una sobre el desplazamiento w y otra sobre la rotación θ o sus derivadas, según el tipo de apoyo.

Condiciones de frontera generales:

- **Empotrado:** El desplazamiento y la rotación son nulos:

$$w = 0, \quad \theta = 0.$$

- **Articulado o apoyado simple:** El desplazamiento es nulo y el momento flector se anula:

$$w = 0, \quad \frac{\partial \theta}{\partial x} = 0.$$

- **Libre:** No hay cortante ni momento flector:

$$GA \left(\theta - \frac{\partial w}{\partial x} \right) = 0, \quad EI \frac{\partial \theta}{\partial x} = 0.$$

La combinación de estas condiciones en $x = 0$ y $x = L$ define completamente el comportamiento estructural de la viga. Por ejemplo, una viga empotrada en ambos extremos debe satisfacer:

$$w(0, t) = 0, \quad \theta(0, t) = 0, \quad w(L, t) = 0, \quad \theta(L, t) = 0.$$

Condiciones iniciales: Dado que el sistema es de segundo orden respecto al tiempo, se requieren dos condiciones iniciales para cada variable dependiente: $w(x, t)$ y $\theta(x, t)$. Estas condiciones se prescriben para todo $x \in [0, L]$ en el instante $t = 0$:

- **Deflexión inicial:**

$$w(x, 0) = f_1(x), \quad \frac{\partial w}{\partial t}(x, 0) = f_2(x).$$

- **Rotación inicial:**

$$\theta(x, 0) = g_1(x), \quad \frac{\partial \theta}{\partial t}(x, 0) = g_2(x).$$

Las funciones $f_1(x), f_2(x), g_1(x), g_2(x)$ deben ser compatibles con las condiciones de frontera impuestas. Por ejemplo, si la viga está empotrada en $x = 0$, entonces debe cumplirse:

$$f_1(0) = 0, \quad f_2(0) = 0, \quad g_1(0) = 0, \quad g_2(0) = 0.$$

Este conjunto de condiciones garantiza la correcta formulación del problema de valor inicial y de frontera (PVIB) para el sistema acoplado de Timoshenko.

2.4. El modelo de Kirchhoff–Love

La teoría de placas delgadas de **Kirchhoff–Love** es una extensión bidimensional de la teoría de vigas de Euler–Bernoulli. En esta teoría se considera una placa delgada y plana, de espesor h , en la cual las secciones normales a la superficie media permanecen planas y normales después de la deformación, despreciando efectos de cortante transversal [10, 9]. Sea $w(x, y, t)$ el desplazamiento transversal de la placa rectangular delgada sobre el plano XY , y $q(x, y, t)$ una carga externa distribuida. La ecuación de movimiento de Kirchhoff–Love es [17, 18]:

$$\rho h \frac{\partial^2 w}{\partial t^2}(x, y, t) + D \nabla^4 w(x, y, t) = q(x, y, t), \quad (2.9)$$

donde ρ es la densidad del material, y D es la rigidez flexional:

$$D = \frac{Eh^3}{12(1 - \nu^2)},$$

con E el módulo de Young y ν el coeficiente de Poisson. El operador ∇^4 es el bi-Laplaciano:

$$\nabla^4 w = \frac{\partial^4 w}{\partial x^4} + 2 \frac{\partial^4 w}{\partial x^2 \partial y^2} + \frac{\partial^4 w}{\partial y^4}.$$

2.4.1. Formulación como sistema acoplado

Para reducir el orden espacial de la ecuación (2.9), se introduce una variable auxiliar $M(x, y, t) = \Delta w(x, y, t)$, donde Δ es el operador Laplaciano del cual se obtiene el siguiente sistema:

$$\begin{cases} \rho h \frac{\partial^2 w}{\partial t^2}(x, y, t) + D \Delta M(x, y, t) = q(x, y, t) \\ \Delta w(x, y, t) = M(x, y, t), \end{cases} \quad (2.10)$$

Este sistema acoplado es equivalente a la ecuación original y es útil para métodos numéricos.

Condiciones de frontera En el contorno $\partial\Omega$ de la placa rectangular se pueden aplicar diversas condiciones según el tipo de soporte. Cada condición involucra al desplazamiento transversal $w(x, y, t)$ y sus derivadas espaciales normales:

- **Empotrado:**

$$w(x, y, t) = 0, \quad \frac{\partial w}{\partial n}(x, y, t) = 0.$$

- **Simplemente apoyado:**

$$w(x, y, t) = 0, \quad M_{nn}(x, y, t) = 0.$$

- **Libre:**

$$M_{nn}(x, y, t) = 0, \quad Q_n(x, y, t) = 0.$$

Aquí, $M_{nn}(x, y, t)$ representa el momento flector normal al borde, y $Q_n(x, y, t)$ la fuerza cortante normal. Estas condiciones están bien establecidas en la literatura clásica sobre teoría de placas [10, 18].

Condiciones iniciales Para el análisis dinámico, se requieren condiciones iniciales sobre la deflexión y la velocidad inicial:

$$w(x, y, 0) = w_0(x, y), \quad \frac{\partial w}{\partial t}(x, y, 0) = v_0(x, y),$$

donde w_0 y v_0 son funciones definidas en todo el dominio Ω que describen la configuración inicial. Estas condiciones permiten resolver el problema de valor inicial y de frontera para la placa de Kirchhoff–Love.

2.5. Ecuación de Burgers viscosa en 2D

La **ecuación de Burgers viscosa** es una ecuación diferencial parcial no lineal que combina los efectos de la advección no lineal con la difusión lineal. Fue introducida por el físico holandés Johannes Martinus Burgers en la década de 1940 como un modelo simplificado para el estudio de la turbulencia y la dinámica de fluidos, constituyendo una aproximación unidimensional de las ecuaciones de Navier–Stokes [11, 19].

La forma general en una dimensión es:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad (2.11)$$

donde $u = u(x, t)$ representa, por ejemplo, la velocidad de un fluido, y $\nu > 0$ es el coeficiente de viscosidad cinemática. Esta ecuación sirve como prototipo para el estudio de fenómenos no lineales en medios disipativos y permite capturar procesos como ondas de choque suaves, capas límite y disipación estructural [20]. A pesar de su simplicidad, su versatilidad la ha hecho aplicable en diversos campos de la física e ingeniería, desde la dinámica de fluidos [21], hasta modelos de tráfico vehicular [22, 23], y análisis conceptual de la turbulencia.

En dos dimensiones, la ecuación de Burgers con viscosidad $\nu > 0$ se generaliza a un sistema vectorial donde el campo de velocidades es $\mathbf{u}(x, y, t) = (u(x, y, t), v(x, y, t))$. El sistema asociado es:

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \end{aligned} \quad (x, y) \in [0, 1]^2, \quad t > 0 \quad (2.12)$$

Este sistema representa la evolución de un campo de velocidad bidimensional en un medio viscoso, conservando la estructura no lineal de advección junto con los efectos de disipación difusiva. Gracias a su estructura matemática bien comprendida y a la existencia de soluciones exactas bajo ciertas condiciones (como las derivadas mediante la transformación de Cole–Hopf en 1D [24, 25]), también es posible construir soluciones cerradas para ciertos casos en 2D.

Una de estas soluciones exactas corresponde a una configuración autosemejante y con simetría radial centrada en el punto $(1/2, 1/2)$. Esta solución está dada por:

$$\begin{aligned} u(x, y, t) &= \frac{x - \frac{1}{2}}{1 + 4\nu t + (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2} \\ v(x, y, t) &= \frac{y - \frac{1}{2}}{1 + 4\nu t + (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2} \end{aligned} \quad (2.13)$$

Esta solución es suave, divergencia libre, y se disipa con el tiempo debido al efecto de la viscosidad, lo que la convierte en un excelente caso de validación para esquemas numéricos.

La condición inicial correspondiente a $t = 0$ es:

$$\begin{aligned} u(x, y, 0) &= \frac{x - \frac{1}{2}}{1 + (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2} & (x, y) \in [0, 1]^2. \\ v(x, y, 0) &= \frac{y - \frac{1}{2}}{1 + (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2} \end{aligned} \quad (2.14)$$

Asimismo, como la solución exacta es conocida en todo el dominio, las condiciones de frontera pueden imponerse directamente evaluando la solución en el borde del dominio:

$$\begin{aligned} u(x, y, t) &= \frac{x - \frac{1}{2}}{1 + 4\nu t + (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2} & (x, y) \in \partial[0, 1]^2, t > 0 \\ v(x, y, t) &= \frac{y - \frac{1}{2}}{1 + 4\nu t + (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2} \end{aligned} \quad (2.15)$$

Estas condiciones garantizan consistencia con la solución exacta y permiten evaluar de manera precisa el rendimiento de métodos numéricos para ecuaciones no lineales con difusión.

2.6. La ecuación de Schrödinger

La **ecuación de Schrödinger** constituye el eje central de la mecánica cuántica no relativista, al proporcionar una descripción matemática precisa de cómo evoluciona en el tiempo el estado cuántico de una partícula sometida a un potencial externo [12]. Propuesta en 1926 por Erwin Schrödinger, esta ecuación representa un cambio paradigmático respecto a la mecánica clásica, al introducir el concepto de función de onda como portadora de la información probabilística del sistema físico.

La forma general de la ecuación de Schrödinger dependiente del tiempo en una dimensión espacial es:

$$i\hbar \frac{\partial \psi}{\partial t}(x, t) = -\frac{\hbar^2}{2m} \frac{\partial^2 \psi}{\partial x^2}(x, t) + V(x)\psi(x, t), \quad (2.16)$$

donde i es la unidad imaginaria, $\psi(x, t)$ es la función de onda, \hbar es la constante de Planck reducida, m es la masa de la partícula y $V(x)$ representa el potencial externo que actúa sobre el sistema. En [12], se formula la interpretación probabilística postulada por Born el cual establece que $|\psi(x, t)|^2$ corresponde a la densidad de probabilidad de localizar la partícula en la posición x en el instante t .

La ecuación (2.16) es una ecuación diferencial parcial compleja de tipo parabólico. Si se descompone la función de onda en sus partes real e imaginaria, es decir, $\psi(x, t) = u(x, t) + iv(x, t)$, y se sustituye en la ecuación original, se obtiene un sistema acoplado de dos ecuaciones reales:

$$\begin{cases} \frac{\partial u}{\partial t} = -\frac{\hbar}{2m} \frac{\partial^2 v}{\partial x^2} + \frac{1}{\hbar} V(x)v, \\ \frac{\partial v}{\partial t} = \frac{\hbar}{2m} \frac{\partial^2 u}{\partial x^2} - \frac{1}{\hbar} V(x)u, \end{cases} \quad (2.17)$$

lo cual permite, si se desea, abordar la resolución del problema desde un enfoque puramente real. Esta reformulación es especialmente útil en métodos numéricos y en implementaciones computacionales donde se desea evitar directamente el uso de variables complejas.

2.6.1. Solución por separación de variables

Una de las estrategias más importantes para resolver la ecuación de Schrödinger es el método de separación de variables, aplicable cuando el potencial es independiente del tiempo, es decir, $V = V(x)$. En este caso, se busca una solución de la forma:

$$\psi(x, t) = \phi(x)T(t), \quad (2.18)$$

donde $\phi(x)$ es la parte espacial y $T(t)$ es la parte temporal de la función de onda. Al sustituir (2.18) en la ecuación (2.16) y dividir entre $\phi(x)T(t)$, se obtiene:

$$i\hbar \frac{1}{T(t)} \frac{dT}{dt} = -\frac{\hbar^2}{2m} \frac{1}{\phi(x)} \frac{d^2\phi}{dx^2} + V(x). \quad (2.19)$$

Como el lado izquierdo depende sólo de t y el derecho sólo de x , ambos deben ser iguales a una constante, denotada por E , que corresponde a la *energía del sistema*. Esto genera dos ecuaciones:

$$i\hbar \frac{dT}{dt} = \mathsf{E} T(t), \quad (2.20)$$

$$-\frac{\hbar^2}{2m} \frac{d^2\phi}{dx^2} + V(x)\phi(x) = \mathsf{E} \phi(x). \quad (2.21)$$

La ecuación (2.20) tiene solución inmediata:

$$T(t) = e^{-i\mathsf{E}t/\hbar}, \quad (2.22)$$

y la ecuación (2.21) es conocida como la *ecuación de Schrödinger independiente del tiempo*. Su solución $\phi(x)$, junto con las condiciones de frontera del problema, determina los *niveles de energía permitidos* E_n y sus correspondientes funciones propias $\phi_n(x)$, llamadas *estados estacionarios*.

2.6.2. Estados fundamentales y excitados

Cada solución $\psi_n(x, t) = \phi_n(x)e^{-i\mathsf{E}_n t/\hbar}$ representa un *estado estacionario* del sistema, cuya densidad de probabilidad $|\psi_n(x, t)|^2 = |\phi_n(x)|^2$ es independiente del tiempo. El estado con menor energía E_0 se denomina *estado fundamental*, mientras que los demás, con $\mathsf{E}_n > \mathsf{E}_0$ para $n \geq 1$, se llaman *estados excitados*.

En problemas típicos como el pozo de potencial infinito o el oscilador armónico cuántico, los niveles de energía están cuantizados, es decir, sólo ciertos valores discretos de E están permitidos. Este fenómeno, característico de la mecánica cuántica, refleja la naturaleza ondulatoria y las restricciones impuestas por las condiciones de contorno.

2.6.3. Interpretación física

De acuerdo con la *interpretación de Copenhague*, la función de onda $\psi(x, t)$ contiene toda la información accesible del sistema. El cuadrado de su módulo, $|\psi(x, t)|^2$, se interpreta como una densidad de probabilidad. Así, por ejemplo, en el estado fundamental de una partícula en un pozo infinito, la probabilidad de hallarla cerca del centro del pozo es máxima, y nula en las paredes.

2.6.4. Ecuación de Schrödinger para el átomo de hidrógeno

El sistema del átomo de hidrógeno consiste en dos partículas: un electrón de masa m_e y carga $-e$, y un protón de masa m_p y carga $+e$. La función de onda total depende de las

coordenadas espaciales de ambas partículas y del tiempo:

$$\psi = \psi(\mathbf{r}_e, \mathbf{r}_p, t), \quad (2.23)$$

donde $\mathbf{r}_e = (x_e, y_e, z_e)$ y $\mathbf{r}_p = (x_p, y_p, z_p)$ posición del electrón y protón respectivamente.

La ecuación de Schrödinger dependiente del tiempo para este sistema es:

$$i\hbar \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m_e} \Delta_{\mathbf{r}_e} \psi - \frac{\hbar^2}{2m_p} \Delta_{\mathbf{r}_p} \psi - \frac{e^2}{|\mathbf{r}_p - \mathbf{r}_e|} \psi, \quad (2.24)$$

donde el término de interacción corresponde al potencial de Coulomb entre el electrón y el protón con e el valor de la carga. Adicional, $\Delta_{\mathbf{r}_p} = (\partial/\partial x_p + \partial/\partial y_p + \partial/\partial z_p)$ y $\Delta_{\mathbf{r}_e} = (\partial/\partial x_e + \partial/\partial y_e + \partial/\partial z_e)$

Usando los parámetros descritos en [26] y sustituyendo en (2.24), la ecuación adimensional resultante es:

$$i \frac{\partial \psi}{\partial t} = -\frac{\mu}{2} \Delta_{\mathbf{r}_p} \psi - \frac{1}{2} \Delta_{\mathbf{r}_e} \psi - \frac{1}{|\mathbf{r}_e - \mathbf{r}_p|} \psi \quad (2.25)$$

donde $\mu = m_e/m_p \approx 5.4462 \times 10^{-4}$

Al descomponer la función de onda $\psi(\mathbf{r}_e, \mathbf{r}_p, t) = u(\mathbf{r}_e, \mathbf{r}_p, t) + iv(\mathbf{r}_e, \mathbf{r}_p, t)$ se obtiene el sistema acoplado de dos ecuaciones reales:

$$\begin{cases} \frac{\partial u}{\partial t} = -\frac{\mu}{2} \Delta_{\mathbf{r}_p} v - \frac{1}{2} \Delta_{\mathbf{r}_e} v - \frac{1}{|\mathbf{r}_p - \mathbf{r}_e|} v \\ \frac{\partial v}{\partial t} = \frac{\mu}{2} \Delta_{\mathbf{r}_p} u + \frac{1}{2} \Delta_{\mathbf{r}_e} u + \frac{1}{|\mathbf{r}_p - \mathbf{r}_e|} u \end{cases} \quad (2.26)$$

Este sistema de ecuaciones acopladas (2.26) describe completamente la evolución temporal del estado cuántico del átomo de hidrógeno en función de las coordenadas absolutas de sus dos partículas constituyentes. La formulación adimensional permite estudiar el comportamiento del sistema sin depender de unidades físicas específicas, facilitando su implementación numérica.

Cabe destacar que la interpretación probabilística de la mecánica cuántica permanece vigente en este contexto: el módulo al cuadrado de la función de onda, $|\psi(\mathbf{r}_e, \mathbf{r}_p, t)|^2$, representa la densidad de probabilidad conjunta de encontrar simultáneamente al electrón en la posición \mathbf{r}_e y al protón en \mathbf{r}_p en el instante t .

Si bien en muchos estudios se realiza un cambio a coordenadas del centro de masa para reducir el problema a una sola partícula con masa reducida como se muestra en la

siguiente subsección, la representación explícita en coordenadas absolutas es fundamental para enfoques computacionales modernos que buscan resolver el sistema completo, especialmente cuando se consideran efectos externos o generalizaciones a sistemas de muchos cuerpos.

2.6.5. Ecuación de Schrödinger para el átomo de hidrógeno con el protón fijo en el origen

En este modelo, se considera que el protón de masa m_p y carga $+e$ se encuentra inmóvil en el origen del sistema de coordenadas. Bajo esta suposición, se estudia únicamente la dinámica del electrón de masa m_e y carga $-e$, el cual se mueve bajo la acción del campo electrostático generado por el protón. La función de onda depende así de la posición del electrón $\mathbf{r} = (x, y, z)$ y del tiempo t , por lo que se escribe como $\psi = \psi(\mathbf{r}, t)$.

La ecuación de Schrödinger dependiente del tiempo que describe este sistema es:

$$i\hbar \frac{\partial \psi}{\partial t}(\mathbf{r}, t) = -\frac{\hbar^2}{2m_e} \Delta \psi(\mathbf{r}, t) - \frac{e^2}{4\pi\epsilon_0|\mathbf{r}|} \psi(\mathbf{r}, t), \quad (2.27)$$

donde Δ denota el operador Laplaciano tridimensional y $|\mathbf{r}| = \sqrt{x^2 + y^2 + z^2}$ es la distancia entre el electrón y el protón; y el parámetro ϵ_0 representa la *permitividad eléctrica del vacío*, una constante fundamental del electromagnetismo que cuantifica la capacidad del espacio vacío para permitir la formación de campos eléctricos.

Para simplificar la ecuación y eliminar las constantes físicas, se introduce una formulación adimensional utilizando las unidades atómicas: el radio de Bohr a_0 , la energía del estado fundamental E_0 y la unidad natural de tiempo t_0 . La ecuación se describe a continuación

$$i \frac{\partial \psi}{\partial t}(\mathbf{r}, t) = -\frac{1}{2} \Delta \psi(\mathbf{r}, t) - \frac{1}{|\mathbf{r}|} \psi(\mathbf{r}, t) \quad (2.28)$$

Al descomponer la función de onda en sus partes real e imaginaria conduce al siguiente sistema acoplado de ecuaciones reales:

$$\begin{cases} \frac{\partial u}{\partial t} = -\frac{1}{2} \Delta v - \frac{1}{|\mathbf{r}|} v \\ \frac{\partial v}{\partial t} = \frac{1}{2} \Delta u + \frac{1}{|\mathbf{r}|} u \end{cases} \quad (2.29)$$

Este sistema describe de forma completa la evolución temporal del electrón en el potencial coulombiano generado por un protón fijo. Su formulación en unidades adimensionales simplifica tanto la implementación computacional como el análisis de los niveles de energía.

Capítulo 3

Aprendizaje profundo

El **aprendizaje profundo** (Deep Learning) es una rama del aprendizaje automático que se basa en redes neuronales profundas (DNN, por sus siglas en inglés) para extraer representaciones complejas y jerárquicas a partir de grandes volúmenes de datos. Estas redes se componen de múltiples capas de neuronas artificiales, lo que les permite modelar relaciones altamente no lineales entre entradas y salidas, superando las capacidades de técnicas tradicionales de aprendizaje superficial.

En los últimos años, el Deep Learning ha revolucionado múltiples áreas de la ciencia y la ingeniería, desde el reconocimiento de imágenes y procesamiento del lenguaje natural hasta el modelado de sistemas físicos complejos. En particular, ha surgido como una herramienta prometedora para la resolución de ecuaciones diferenciales parciales, especialmente en contextos donde los métodos numéricos clásicos enfrentan limitaciones computacionales o de escalabilidad.

3.1. Descripción matemática de una red neuronal profunda

En términos formales, una **red neuronal profunda** es una función parametrizada construida mediante la composición de múltiples transformaciones no lineales, denominadas *capas*. El objetivo de esta estructura es aproximar funciones complejas, mapeando un vector de entrada $\mathbf{x} \in \mathbb{R}^{l_0}$ a un vector de salida $\mathbf{y} \in \mathbb{R}^{l_L}$.

Cada capa está compuesta por un conjunto de **neuronas artificiales**, que son las unidades básicas de procesamiento de la red. Una neurona artificial toma como entrada

un conjunto de valores numéricos, los pondera mediante un conjunto de *pesos sinápticos*, suma un *sesgo* o término independiente y aplica una *función de activación* no lineal que determina su salida. Matemáticamente, la salida $a_j^{(l)}$ de la neurona j en la capa l se define como:

$$a_j^{(l)} = \sigma_l \left(\sum_{i=1}^{l_{l-1}} w_{ji}^{(l)} x_i^{(l-1)} + b_j^{(l)} \right),$$

donde $w_{ji}^{(l)}$ representa el peso de la conexión entre la neurona i de la capa anterior y la neurona j de la capa actual, $b_j^{(l)}$ es el sesgo asociado, y $\sigma_l(\cdot)$ es la función de activación que introduce la no linealidad del modelo.

Consideremos una red *feed-forward* con $L \in \{2, 3, \dots\}$ capas y una arquitectura definida por la tupla de dimensiones (l_0, l_1, \dots, l_L) , donde l_0 es la dimensión del espacio de entrada, l_L la del espacio de salida, y l_1, \dots, l_{L-1} son las dimensiones de las capas ocultas intermedias.

La transformación de los datos se realiza de forma secuencial a través de las capas. Denotando la entrada a la red como $\mathbf{x}^{(0)} = \mathbf{x}$, la salida de cada capa $l \in \{1, 2, \dots, L\}$, denotada por $\mathbf{x}^{(l)} \in \mathbb{R}^{l_l}$, se calcula a partir de la salida de la capa anterior $\mathbf{x}^{(l-1)} \in \mathbb{R}^{l_{l-1}}$ mediante la siguiente operación:

$$\mathbf{x}^{(l)} = \sigma_l (\mathbf{W}^{(l)} \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}) \quad (3.1)$$

donde

- $\mathbf{W}^{(l)} \in \mathbb{R}^{l_l \times l_{l-1}}$ es la **matriz de pesos** de la capa l .
- $\mathbf{b}^{(l)} \in \mathbb{R}^{l_l}$ es el **vector de sesgos** (*bias*) de la capa l .
- $\sigma_l : \mathbb{R}^{l_l} \rightarrow \mathbb{R}^{l_l}$ es la **función de activación** no lineal, que se aplica elemento a elemento.

El conjunto de todos los parámetros entrenables de la red se agrupa en $\boldsymbol{\theta} = \{\mathbf{W}^{(1)}, \mathbf{b}^{(1)}, \dots, \mathbf{W}^{(L)}, \mathbf{b}^{(L)}\}$.

La red neuronal completa, $f : \mathbb{R}^{l_0} \rightarrow \mathbb{R}^{l_L}$, se puede expresar como una función $f(\mathbf{x}; \boldsymbol{\theta})$ que representa la composición de todas las transformaciones de capa. Si definimos cada transformación de capa como $f^{(l)}(\cdot; \boldsymbol{\theta}_l) = \sigma_l(\mathbf{W}^{(l)} \cdot + \mathbf{b}^{(l)})$, la red completa es:

$$f(\mathbf{x}; \boldsymbol{\theta}) = (f^{(L)} \circ f^{(L-1)} \circ \dots \circ f^{(1)}) (\mathbf{x}) \quad (3.2)$$

Esta estructura permite modelar funciones de alta complejidad mediante el ajuste de los parámetros $\boldsymbol{\theta}$ durante un proceso de entrenamiento, ofreciendo así un marco robusto para la aproximación numérica. Un ejemplo ilustrativo se muestra en la figura 3.1

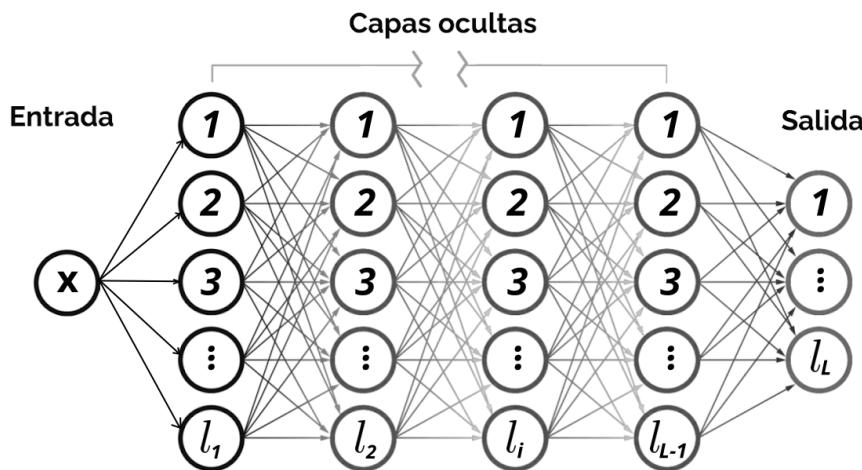


Figura 3.1: Ilustración de una red neuronal profunda *fully connected*. La arquitectura se denota por (l_0, l_1, \dots, l_L) : capa de entrada con l_0 neuronas, capas ocultas intermedias con l_1, \dots, l_{L-1} neuronas y capa de salida con l_L neuronas.

3.2. Funciones de activación

Las funciones de activación introducen no linealidad a las redes neuronales, permitiendo que éstas puedan aproximar funciones complejas. A continuación se presentan las funciones más utilizadas, junto con su formulación matemática y representación gráfica.

1. Función sigmoide

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.3)$$

Rango: $(0, 1)$

Derivada: $\sigma'(x) = \sigma(x)(1 - \sigma(x))$

Aunque fue ampliamente utilizada en las primeras redes neuronales [27, 28], actualmente su uso es limitado debido al problema del *desvanecimiento del gradiente*, especialmente en redes profundas. Enriquecer **2. Tangente hiperbólica (tanh)**

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.4)$$

Rango: $(-1, 1)$

Derivada: $\tanh'(x) = 1 - \tanh^2(x)$

Es otra función clásica utilizada en redes neuronales, con ventajas sobre la sigmoide debido a su centrado en cero [28].

3. ReLU (Rectified Linear Unit)

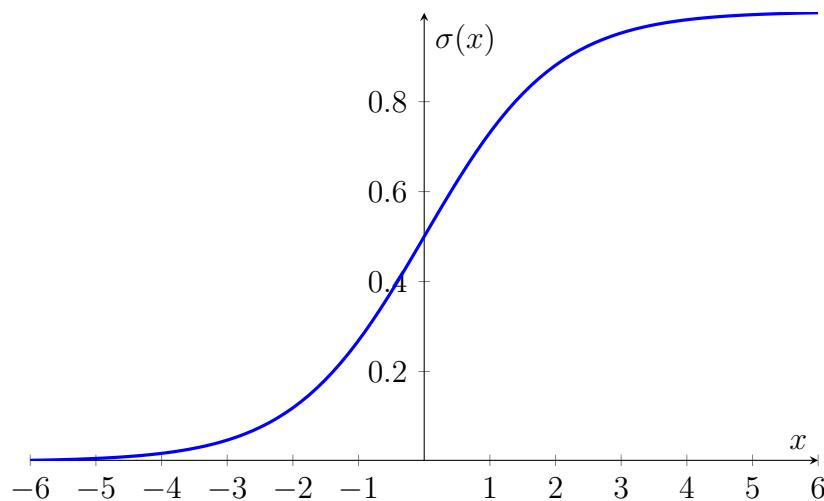


Figura 3.2: Función sigmoide $\sigma(x) = \frac{1}{1+e^{-x}}$

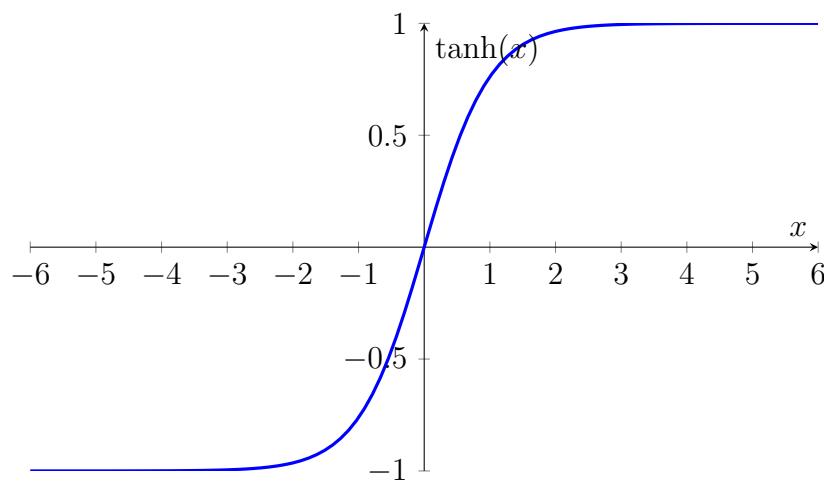


Figura 3.3: Función tangente hiperbólica $\tanh(x)$

$$\text{ReLU}(x) = \max(0, x) \quad (3.5)$$

Rango: $[0, \infty)$

Derivada:

$$\text{ReLU}'(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x > 0 \end{cases}$$

La ReLU fue introducida por Glorot et al. [29] y se convirtió rápidamente en la función más utilizada en redes profundas debido a su simplicidad computacional y a que evita el problema del desvanecimiento del gradiente.

4. Leaky ReLU

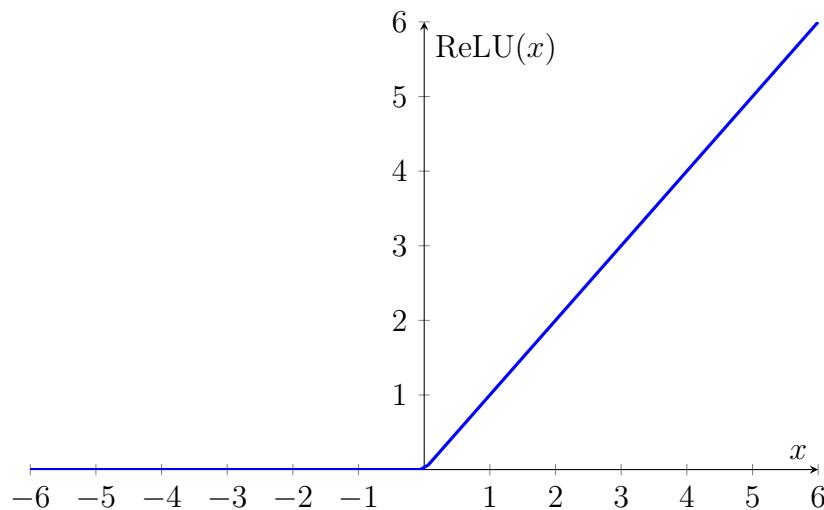


Figura 3.4: Función ReLU $\text{ReLU}(x) = \max(0, x)$

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{si } x \geq 0 \\ \alpha x & \text{si } x < 0 \end{cases}, \quad \alpha = 0.1 \quad (3.6)$$

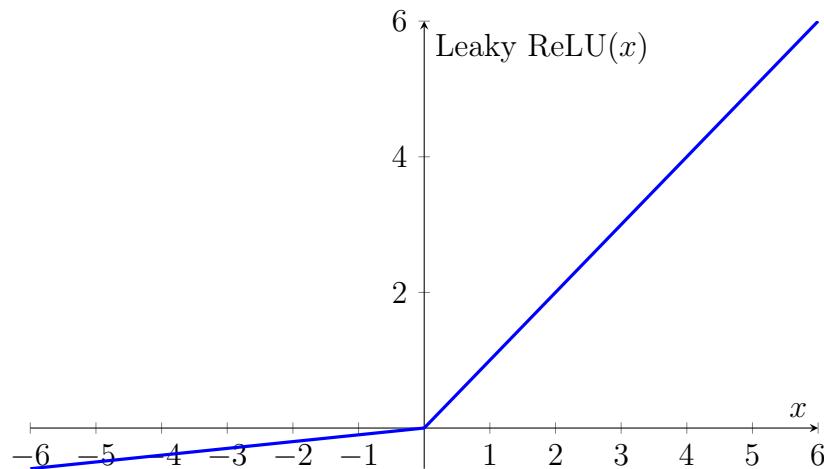


Figura 3.5: Función Leaky ReLU con $\alpha = 0.1$

La Leaky ReLU fue introducida para corregir el problema de las neuronas muertas de la ReLU clásica [30], permitiendo un pequeño gradiente en la región negativa.

5. SiLU (Sigmoid Linear Unit)

La función **SiLU** (*Sigmoid Linear Unit*), también conocida como **Swish**, fue introducida por Hendrycks y Gimpel en 2016 como una activación suave y autorregulada que combina propiedades de la función sigmoide y la ReLU. Se define como:

$$\text{SiLU}(x) = x \sigma(x) = \frac{x}{1 + e^{-x}}. \quad (3.7)$$

Rango: $(-0.28, \infty)$

Derivada:

$$\text{SiLU}'(x) = \sigma(x) + x \sigma(x) (1 - \sigma(x)).$$

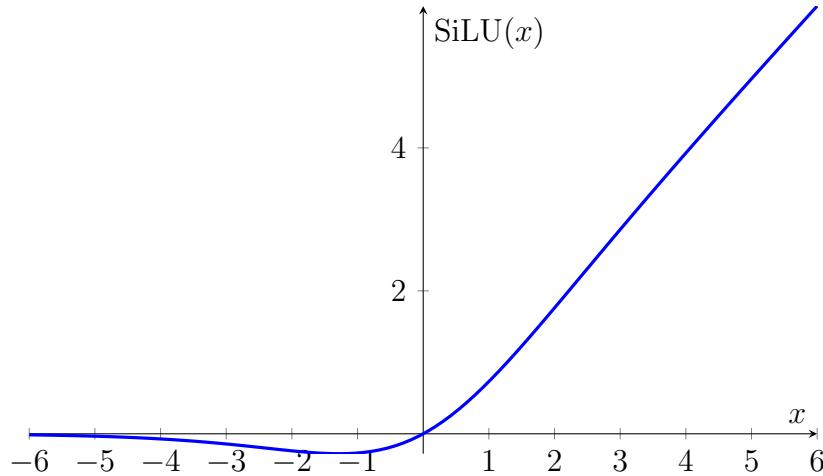


Figura 3.6: Función SiLU (Swish): $\text{SiLU}(x) = x \sigma(x)$

A diferencia de ReLU, la función SiLU es suave y no presenta puntos de discontinuidad en la derivada. Su componente sigmoidal atenúa los valores negativos de x de manera gradual, en lugar de anularlos completamente, lo que permite conservar información en regiones donde ReLU produciría gradiente nulo. En la práctica, esta característica mejora la propagación del gradiente y favorece una convergencia más estable en redes profundas [31].

6. GELU

La función GELU, o Gaussian Error Linear Unit, fue propuesta por Dan Hendrycks y Kevin Gimpel en 2016 en su artículo introductorio [31]. Se define como:

$$\text{GELU}(x) = x \Phi(x), \quad (3.8)$$

donde $\Phi(x)$ es la función de distribución acumulada de una distribución normal estándar. En forma equivalente, se puede escribir como:

$$\text{GELU}(x) = \frac{x}{2} \left[1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right) \right]. \quad (3.9)$$

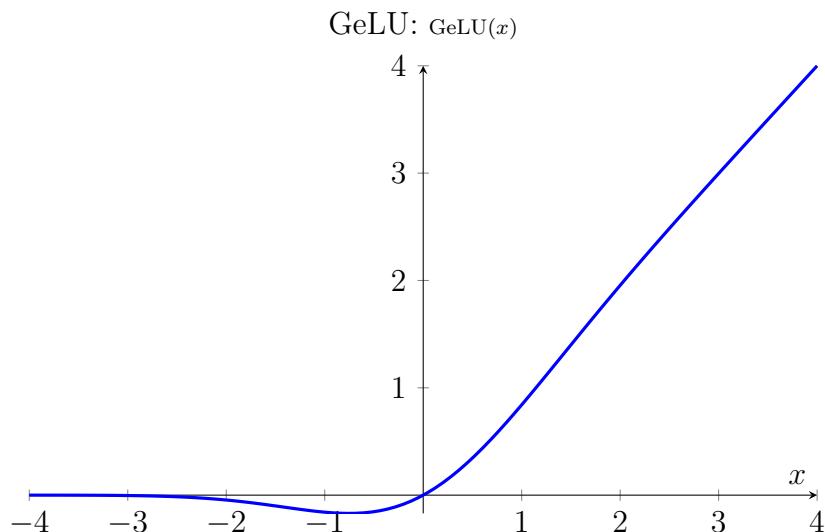


Figura 3.7: Función GELU

A diferencia de funciones como ReLU, que activan el valor basado únicamente en el signo de x , GELU pondera la activación de x por la probabilidad $\Phi(x)$, introduciendo así una transición suave entre la región activa y la apagada. Esta suavidad favorece la propagación de gradientes en todo el dominio, evitando regiones muertas donde la derivada sea nula o indeseable. Además, los mismos autores [31] observaron mejoras en el rendimiento en diversas aplicaciones de visión por computador, procesamiento del lenguaje natural y reconocimiento de voz al comparar GELU con ReLU o ELU.

3.3. Inicialización de parámetros: esquema Xavier

Una etapa esencial antes de comenzar el entrenamiento de una red neuronal profunda es la inicialización adecuada de los pesos y sesgos. Si los valores iniciales no se eligen con cuidado, la propagación hacia adelante o hacia atrás puede sufrir de gradientes demasiado pequeños o demasiado grandes, lo que impide una convergencia estable.

El esquema de inicialización conocido como Xavier (o Glorot) fue propuesto por Xavier Glorot y Yoshua Bengio en [32] para redes feed-forward profundas. La idea central es elegir la distribución inicial de los pesos de modo que la varianza de las activaciones y la de los gradientes permanezcan aproximadamente constantes a lo largo de las capas, mitigando la degradación del flujo de señal durante el entrenamiento.

Concretamente, para una capa l con n_{in} unidades de entrada (fan-in) y n_{out} unidades de

salida (fan-out), el peso $W_{ij}^{(l)}$ se inicializa aleatoriamente de acuerdo con:

$$W_{ij}^{(l)} \sim \mathcal{U}(-a, a), \quad a = \sqrt{\frac{6}{n_{\text{in}} + n_{\text{out}}}}.$$

Alternativamente, se puede usar una distribución normal con media cero y varianza

$$\text{Var}(W_{ij}^{(l)}) = \frac{2}{n_{\text{in}} + n_{\text{out}}}$$

que conduce a un comportamiento similar en cuanto al control de varianzas. Las constantes n_{in} y n_{out} corresponden a los tamaños de la capa anterior y la capa posterior, respectivamente.

Los sesgos $\mathbf{b}^{(l)}$ suelen inicializarse en cero o con valores pequeños constantes.

Este mecanismo de inicialización permite que, inicialmente, las señales de activación y los gradientes no se atenúen ni exploten dramáticamente al propagarse a través de muchas capas, favoreciendo un entrenamiento más eficiente y estable incluso en redes profundas.

3.4. Métodos de optimización

Los métodos de optimización son algoritmos que ajustan los pesos de una red neuronal para minimizar la función de costo o pérdida. Son esenciales para garantizar una convergencia rápida y estable durante el entrenamiento de modelos. A continuación se presentan dos de los métodos más representativos.

- **Descenso del Gradiente:** método base para la optimización, que actualiza los parámetros en dirección opuesta al gradiente de la función de pérdida con respecto a los parámetros. Fue ampliamente descrito en la literatura clásica de redes neuronales [28].
- **Adam (Adaptive Moment Estimation):** optimizador adaptativo ampliamente utilizado en redes neuronales profundas, propuesto por Kingma y Ba [33]. Combina ideas de los métodos Momentum y RMSProp, adaptando la tasa de aprendizaje individual para cada parámetro.

3.4.1. Descenso del gradiente

El método de descenso del gradiente tiene como objetivo minimizar una función de costo $\mathcal{J}(\boldsymbol{\theta})$ encontrando el conjunto de parámetros $\boldsymbol{\theta}$ que produce el valor mínimo. Al

vectorizar todos los pesos y sesgos de la red neuronal, el conjunto total de parámetros puede representarse como un único vector en un espacio euclíadiano de alta dimensión:

$$\boldsymbol{\theta} \in \mathbb{R}^P, \quad P = \sum_{k=1}^L (l_k l_{k-1} + l_k). \quad (3.10)$$

De forma equivalente, el espacio de parámetros puede expresarse como:

$$\mathcal{P} = \prod_{k=1}^L (\mathbb{R}^{l_k \times l_{k-1}} \times \mathbb{R}^{l_k}), \quad (3.11)$$

donde cada punto $\boldsymbol{\theta} \in \mathcal{P}$ representa una configuración específica de la red. Este espacio constituye el dominio sobre el cual se define la función de pérdida y se lleva a cabo el proceso de optimización durante el entrenamiento.

En su forma más simple, la actualización de los parámetros se realiza mediante la regla del descenso del gradiente:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}_t), \quad (3.12)$$

donde:

- η es la **tasa de aprendizaje**, que controla la magnitud del paso en cada iteración.
- $\nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}_t)$ es el **gradiente** de la función de pérdida respecto a los parámetros.

La **tasa de aprendizaje** (η) es uno de los hiperparámetros más críticos en el proceso de optimización de una red neuronal. Controla el tamaño del paso con el que se actualizan los parámetros $\boldsymbol{\theta}$ en la dirección del gradiente negativo de la función de pérdida.

Una tasa de aprendizaje demasiado grande puede provocar oscilaciones o divergencia, impidiendo la convergencia hacia un mínimo. Por el contrario, una tasa demasiado pequeña puede hacer que el proceso de entrenamiento sea extremadamente lento o que la red quede atrapada en mínimos locales poco profundos.

El ajuste adecuado de η equilibra la velocidad y estabilidad de la convergencia, siendo frecuente el uso de estrategias adaptativas (como Adam) o de programación de tasas de aprendizaje decrecientes para mejorar la eficiencia del entrenamiento.

Existen varias variantes:

- **Batch Gradient Descent:** utiliza todos los datos de entrenamiento para calcular el gradiente.
- **Stochastic Gradient Descent (SGD):** calcula el gradiente usando una sola muestra por iteración, lo que introduce ruido pero permite una convergencia más rápida.

- **Mini-batch Gradient Descent:** equilibrio entre los dos anteriores, utiliza pequeños subconjuntos del conjunto de entrenamiento.

Una desventaja de este método es que utiliza una tasa de aprendizaje constante para todos los parámetros, lo que puede dificultar la convergencia en problemas mal condicionados.

3.4.2. Método Adam

El optimizador **Adam** (*Adaptive Moment Estimation*), propuesto por Kingma y Ba en 2015, se ha consolidado como uno de los métodos de optimización estocástica más eficientes para el entrenamiento de redes neuronales profundas [34]. Su popularidad radica en que combina las ventajas de dos algoritmos predecesores: **Momentum** y **RMSProp**. En esencia, Adam calcula tasas de aprendizaje adaptativas para cada parámetro del modelo. Para ello, mantiene un promedio móvil exponencial tanto de los gradientes pasados (el primer momento) como de los gradientes al cuadrado (el segundo momento).

El algoritmo procede de la siguiente manera en cada iteración t :

1. **Actualización de los momentos:** Se calculan las estimaciones del primer momento (m_t , la media) y del segundo momento (v_t , la varianza no centrada) de los gradientes.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} \mathcal{J}(\theta_t) \quad (3.13)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) [\nabla_{\theta} \mathcal{J}(\theta_t)]^2 \quad (3.14)$$

Aquí, la operación $[\nabla_{\theta} \mathcal{J}(\theta_t)]^2$ es un cuadrado elemento a elemento (producto de Hadamard). Los hiperparámetros β_1 y β_2 controlan las tasas de decaimiento exponencial de estos promedios.

2. **Corrección de sesgo:** Dado que m_t y v_t se inicializan en cero, están sesgados hacia el origen, especialmente durante las primeras iteraciones. Adam contrarresta este sesgo calculando estimaciones corregidas:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (3.15)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}. \quad (3.16)$$

3. **Actualización de parámetros:** Finalmente, los parámetros de la red θ se actualizan utilizando las estimaciones corregidas. La tasa de aprendizaje η se escala

de forma adaptativa por la raíz cuadrada del segundo momento:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon} \quad (3.17)$$

Los hiperparámetros del algoritmo suelen fijarse a valores recomendados en la literatura [33]:

$\beta_1 \approx 0.9$ Controla el promedio del gradiente. Un valor alto suaviza la trayectoria de la actualización.

$\beta_2 \approx 0.999$ Controla el promedio del gradiente al cuadrado. Un valor alto permite que la información de gradientes pasados persista por más tiempo.

$\epsilon \approx 10^{-8}$ Es una constante de estabilidad numérica para evitar la división por cero.

Ventajas de Adam El método ha demostrado ser especialmente robusto y eficiente por varias razones:

- Logra una rápida convergencia en una amplia variedad de problemas y arquitecturas de redes profundas.
- Es robusto a la escala de los gradientes, ya que la actualización depende de una normalización por el segundo momento.
- Sus hiperparámetros predeterminados funcionan bien en la mayoría de los casos prácticos, lo que *reduce la necesidad de un ajuste fino*.

Gracias a estas características, Adam es particularmente eficaz en problemas con gradientes ruidosos o dispersos y se ha convertido en el método de optimización predeterminado en bibliotecas modernas de aprendizaje profundo como TensorFlow y PyTorch [34].

3.5. Función de pérdida y entrenamiento de redes neuronales

El proceso de entrenamiento de una red neuronal consiste en ajustar los parámetros $\boldsymbol{\theta}$ para minimizar una función de costo o **función de pérdida**, la cual cuantifica la discrepancia entre las salidas predichas por la red y los valores reales o deseados. Formalmente, si el conjunto de entrenamiento está compuesto por N pares de datos $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, donde

$\mathbf{x}_i \in \mathbb{R}^{l_0}$ es la entrada y $\mathbf{y}_i \in \mathbb{R}^{l_L}$ la salida esperada, la función de pérdida general se define como:

$$\mathcal{J}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N \mathcal{L}(f(\mathbf{x}_i; \boldsymbol{\theta}), \mathbf{y}_i),$$

donde \mathcal{L} representa la pérdida individual asociada a cada muestra.

El objetivo del aprendizaje profundo es encontrar un conjunto de parámetros $\boldsymbol{\theta}^*$ que minimice la función de pérdida:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta} \in \mathcal{P}} \mathcal{J}(\boldsymbol{\theta}),$$

donde \mathcal{P} es el espacio de parámetros de la red, $\mathcal{P} = \mathbb{R}^P$.

El proceso de minimización se lleva a cabo mediante algoritmos iterativos de optimización, como el **descenso del gradiente** o sus variantes adaptativas. Durante cada iteración, los gradientes de la función de pérdida con respecto a los parámetros se calculan mediante la técnica de **retropropagación del error** (backpropagation), permitiendo actualizar eficientemente los pesos y sesgos en todas las capas de la red.

El entrenamiento, por tanto, puede interpretarse como un problema de búsqueda en un espacio de alta dimensión, donde cada punto $\boldsymbol{\theta}$ define una posible configuración de la red, y el gradiente indica la dirección de mayor descenso del error. La convergencia hacia un mínimo adecuado depende tanto de la topología de la función de pérdida como de la tasa de aprendizaje y del método de optimización empleado.

3.6. Redes neuronales con compuertas LSTM

Las redes neuronales con compuertas **LSTM** (*Long Short-Term Memory*) son una extensión de las redes recurrentes clásicas, que fueron introducidas en [35] e implementadas en [36], y están diseñadas para capturar dependencias a largo plazo en secuencias temporales o espaciales. Su estructura interna incorpora *compuertas* que regulan el flujo de información, permitiendo conservar o descartar datos relevantes durante la propagación temporal o a través de capas profundas.

Matemáticamente, una red neuronal con compuertas LSTM puede describirse como una función parametrizada

$$u_{\boldsymbol{\theta}}(t, \mathbf{x}) : \mathbb{R}^{d+1} \longrightarrow \mathbb{R}, \quad \mathbf{z} = (t, \mathbf{x}) \in \mathbb{R}^{d+1},$$

donde los parámetros $\boldsymbol{\theta}$ incluyen los pesos y sesgos de cada bloque recurrente.

La red está compuesta por L aplicaciones sucesivas de **bloques LSTM**. Cada bloque recibe un estado de entrada $\mathbf{S}_l \in \mathbb{R}^m$ y produce un nuevo estado $\mathbf{S}_{l+1} \in \mathbb{R}^m$ mediante un conjunto de compuertas que controlan la información retenida, olvidada y actualizada. Dicho mecanismo se modela como:

$$\begin{aligned}\mathbf{Z} &= \sigma(W_z \mathbf{S}_l + \mathbf{b}_z), \\ \mathbf{G} &= \sigma(W_g \mathbf{S}_l + \mathbf{b}_g), \\ \mathbf{R} &= \sigma(W_r \mathbf{S}_l + \mathbf{b}_r), \\ \mathbf{H} &= \tanh(W_h (\mathbf{R} \odot \mathbf{S}_l) + \mathbf{b}_h),\end{aligned}$$

donde \odot denota el producto elemento a elemento, y $\sigma(\cdot)$ es la función sigmoide.

En este contexto:

- **Z** es la **compuerta de actualización**, que determina la fracción del estado anterior que se conserva.
- **G** es la **compuerta de olvido**, que controla cuánto de la nueva información sustituye al estado previo.
- **R** es la **compuerta de entrada**, que regula el flujo de información hacia la activación intermedia.
- **H** representa la **información candidata**, generada mediante una transformación no lineal de los datos filtrados.

Capítulo 4

Método Deep Galerkin

En este capítulo se describe detalladamente el enfoque numérico basado en redes neuronales profundas método Deep Galerkin (DGM), para la resolución de ecuaciones en derivadas parciales, propuesto inicialmente por Sirignano y Spiliopoulos en 2018 [5]. Este método se distingue por su capacidad para aproximar soluciones de EDP en espacios de alta dimensión sin necesidad de generar mallas espaciales, lo que lo convierte en una herramienta versátil y poderosa en el contexto del aprendizaje profundo aplicado a problemas físicos y de ingeniería.

El capítulo se estructura en varias secciones que abordan tanto los fundamentos teóricos del método como su implementación práctica para distintas ecuaciones de interés. Inicialmente, se presenta la formulación matemática general del método, en la cual se aproxima la solución de una EDP mediante una red neuronal profunda entrenada a partir del residuo de la ecuación, así como de las condiciones iniciales y de frontera. A continuación, se describe la arquitectura específica utilizada en esta tesis: una red basada en bloques DGM con compuertas inspiradas en las redes LSTM, que permite una mayor capacidad de modelado para soluciones con comportamiento no lineal y dinámico.

La función de pérdida que se optimiza durante el entrenamiento incorpora múltiples componentes: el residuo de la ecuación diferencial parcial, los errores respecto a las condiciones iniciales y de frontera, y posibles técnicas de regularización. El entrenamiento se realiza mediante algoritmos de optimización como Adam o SGD, utilizando muestreo aleatorio de puntos en el dominio, y diferenciación automática para el cálculo preciso de derivadas parciales de orden elevado.

Una parte central de este capítulo está dedicada a la *aplicación práctica del método DGM* a diferentes modelos de ecuaciones. Cada uno de estos casos incluye su propia

formulación matemática, una reformulación si es necesario (como sistemas acoplados de segundo orden), el diseño de la función de pérdida, la estrategia de entrenamiento y la justificación del enfoque. Las ecuaciones tratadas en esta tesis comprenden:

- **Ecuación de Euler–Bernoulli**, para modelar la dinámica transversal de vigas delgadas.
- **Ecuación de Timoshenko**, que introduce la rotación del eje neutro en la dinámica de vigas.
- **Ecuación de Kirchhoff–Love**, que modela la dinámica de placas delgadas en dominios bidimensionales.
- **Ecuación de Burgers viscosa en dos dimensiones**, como modelo no lineal clásico de la dinámica de fluidos.
- **Ecuación de Schrödinger dependiente del tiempo**, en su forma compleja, para fenómenos cuánticos.
- **Ecuación de Schrödinger para el átomo de hidrógeno**, con formulación acoplada en variables reales.

Cada una de estas ecuaciones se aborda utilizando la misma filosofía: reformular la ecuación, si es necesario, para reducir el orden de las derivadas; construir una o más redes neuronales para predecir simultáneamente las variables relevantes; y entrenar dichas redes sobre muestras aleatorias del dominio físico utilizando técnicas de diferenciación automática y optimización adaptativa.

El objetivo de este capítulo es doble. Por un lado, proporcionar una descripción clara y sistemática del método y de su implementación en problemas con distintas características físicas y matemáticas. Por otro, sentar las bases teóricas y computacionales para los experimentos numéricos que se presentan en el siguiente capítulo, donde se evaluará la capacidad de este método para aproximar soluciones.

4.1. Fundamentos del método Deep Galerkin

El **método Deep Galerkin**, propuesto por Sirignano y Spiliopoulos [5] en 2018, es un método basado en aprendizaje profundo diseñado para resolver ecuaciones diferenciales parciales de alta dimensión sin requerir el uso de mallas espaciales. A diferencia del método de Galerkin tradicional, que proyecta la solución sobre una base de funciones previamente seleccionadas, el DGM utiliza una red neuronal profunda para aproximar directamente la

solución de la ecuación en todo el dominio espacio-temporal, bajo condiciones iniciales y de frontera.

4.1.1. Formulación matemática

En [5] se plantea la ecuación diferencial parcial general de tipo parabólico con solución $u(t, x)$:

$$\begin{cases} \frac{\partial u}{\partial t}(t, x) + \mathcal{L}u(t, x) = 0, & (t, x) \in [0, T] \times \Omega, \\ u(0, x) = u_0(x), & x \in \Omega, \\ u(t, x) = g(t, x), & (t, x) \in [0, T] \times \partial\Omega, \end{cases} \quad (4.1)$$

donde \mathcal{L} es un operador diferencial en el espacio. El método DGM aproxima u por una red neuronal profunda $f(t, x; \boldsymbol{\theta})$, con vector de parámetros entrenables $\boldsymbol{\theta}$, que se ajustan minimizando la siguiente función de pérdida:

$$\mathcal{J}(f) = \left\| \frac{\partial f}{\partial t} + \mathcal{L}f \right\|_{[0, T] \times \Omega}^2 + \|f - g\|_{[0, T] \times \partial\Omega}^2 + \|f(0, x) - u_0(x)\|_{L^2, \Omega}^2, \quad (4.2)$$

donde cada término se estima sobre muestras aleatorias en el dominio, sin formar una malla espacial.

La minimización del funcional o de la función de pérdida $\mathcal{J}(f)$ busca construir una red neuronal $f(t, x; \boldsymbol{\theta})$ que se aproxime a la solución de la EDP bajo una métrica integral compuesta por tres contribuciones: el residuo de la ecuación diferencial, el error en las condiciones de frontera y el desajuste con la condición inicial. Cada término es evaluado mediante muestreo aleatorio en el dominio correspondiente, lo cual elimina la necesidad de construir mallas espaciales o temporales.

Este enfoque se fundamenta en dos principios clave. Primero, las redes neuronales profundas poseen la capacidad universal de aproximar funciones suaves en dominios compactos, por lo que existe, en teoría, un conjunto de parámetros $\boldsymbol{\theta}$ que permite a la red representar la solución verdadera de la EDP con precisión arbitraria. Segundo, al minimizar el residuo en norma L^2 , se garantiza que la red no solo aproxima la función solución, sino también sus derivadas involucradas en el operador diferencial \mathcal{L} , lo que asegura una convergencia estructural hacia la solución del problema.

En particular, Sirignano y Spiliopoulos demuestran empíricamente que este procedimiento es eficaz incluso en espacios de alta dimensión, donde los métodos clásicos basados

en mallas se vuelven inviables. La combinación de muestreo aleatorio, diferenciación automática y optimización estocástica permite aproximar soluciones de EDPs evolutivas y elípticas sin recurrir a discretizaciones estructuradas del dominio.

Este marco de minimización puede interpretarse como una generalización del método de Galerkin, en la que, en lugar de proyectar el residuo sobre una base funcional fija, se optimiza directamente la red neuronal para minimizar la energía del residuo a lo largo del dominio, respetando las condiciones del problema.

4.1.2. Arquitectura de la red neuronal

La arquitectura propuesta para el DGM se inspira en redes tipo *Long Short-Term Memory* (LSTM), integrando compuertas que permiten manejar interacciones complejas y dependencias espaciales intensas. Este tipo de estructuras de memoria han demostrado ser especialmente eficaces para capturar dependencias de largo alcance en datos secuenciales, debido a su capacidad para mitigar el problema del desvanecimiento del gradiente.

Al trasladar esta idea al contexto de EDPs se logra una arquitectura que no solo aprende la dinámica local del sistema físico, sino que también puede retener y propagar información relevante a través de múltiples capas, facilitando la representación de soluciones con discontinuidades suaves (esto es, regiones donde la función es continua pero presenta cambios abruptos en su derivada), transiciones rápidas o estructuras oscilatorias complejas.

En este trabajo, se utiliza una arquitectura de red basada en bloques DGM con compuertas adaptadas, que combinan conceptos de las redes LSTM con arquitecturas *feedforward* profundas. Estas compuertas permiten una adaptación selectiva de la información que fluye entre capas, controlando qué se conserva, qué se actualiza y qué se descarta.

Bloques con compuertas LSTM

Sea una red neuronal con parámetros θ :

$$u_{\theta}(t, \mathbf{x}) : \mathbb{R}^{d+1} \longrightarrow \mathbb{R}, \quad \mathbf{z} = (t, \mathbf{x}) \in \mathbb{R}^{d+1}.$$

La red está formada por L aplicaciones sucesivas de **bloques con compuertas LSTM**. Cada bloque recibe un estado $\mathbf{S}_{\ell} \in \mathbb{R}^m$ y produce un nuevo estado $\mathbf{S}_{\ell+1} \in \mathbb{R}^m$ mediante

compuertas (sigmoideas) y una no linealidad interna:

$$\begin{aligned}\mathbf{Z}_\ell &= \sigma(W_z^{(\ell)} \mathbf{S}_\ell + \mathbf{b}_z^{(\ell)}), \\ \mathbf{G}_\ell &= \sigma(W_g^{(\ell)} \mathbf{S}_\ell + \mathbf{b}_g^{(\ell)}), \\ \mathbf{R}_\ell &= \sigma(W_r^{(\ell)} \mathbf{S}_\ell + \mathbf{b}_r^{(\ell)}), \\ \mathbf{H}_\ell &= \tanh(W_h^{(\ell)} (\mathbf{R}_\ell \odot \mathbf{S}_\ell) + \mathbf{b}_h^{(\ell)}),\end{aligned}\quad \mathbf{S}_{\ell+1} = (1 - \mathbf{G}_\ell) \odot \mathbf{H}_\ell + \mathbf{Z}_\ell \odot \mathbf{S}_\ell,$$

donde \odot denota el producto elemento a elemento y $\sigma(\cdot)$ es la función sigmoide.

Encadenando los L bloques, el estado final se obtiene mediante la composición:

$$\mathbf{S}^{(L)} = \underbrace{B_{\text{LSTM}}^{(L)} \circ \cdots \circ B_{\text{LSTM}}^{(1)}}_{\text{composición de bloques con compuertas}} (\mathbf{z}),$$

y la salida escalar de la red se calcula con una proyección lineal:

$$u_{\boldsymbol{\theta}}(t, \mathbf{x}) = W^{(L+1)} \mathbf{S}^{(L)} + b^{(L+1)}.$$

Implementación adaptada en el DGM. Para una entrada $\mathbf{x} = (t, x)$, la red neuronal realiza las siguientes operaciones a nivel de capa:

$$\mathbf{S}_1 = \sigma(W^{(1)} \mathbf{x} + \mathbf{b}^{(1)}), \tag{4.3}$$

$$\mathbf{Z}_\ell = \sigma(U_z^{(\ell)} \mathbf{x} + W_z^{(\ell)} \mathbf{S}_\ell + \mathbf{b}_z^{(\ell)}), \tag{4.4}$$

$$\mathbf{G}_\ell = \sigma(U_g^{(\ell)} \mathbf{x} + W_g^{(\ell)} \mathbf{S}_\ell + \mathbf{b}_g^{(\ell)}), \tag{4.5}$$

$$\mathbf{R}_\ell = \sigma(U_r^{(\ell)} \mathbf{x} + W_r^{(\ell)} \mathbf{S}_\ell + \mathbf{b}_r^{(\ell)}), \tag{4.6}$$

$$\mathbf{H}_\ell = \sigma(U_h^{(\ell)} \mathbf{x} + W_h^{(\ell)} (\mathbf{S}_\ell \odot \mathbf{R}_\ell) + \mathbf{b}_h^{(\ell)}), \tag{4.7}$$

$$\mathbf{S}_{\ell+1} = (1 - \mathbf{G}_\ell) \odot \mathbf{H}_\ell + \mathbf{Z}_\ell \odot \mathbf{S}_\ell, \tag{4.8}$$

para $\ell = 1, \dots, L$. La salida final se expresa como:

$$f(t, x; \boldsymbol{\theta}) = W^{(L+1)} \mathbf{S}_{L+1} + b^{(L+1)}. \tag{4.9}$$

Esta red utiliza compuertas de actualización, reseteo y fusión para preservar la estabilidad del gradiente y la capacidad de modelar soluciones altamente no lineales, como ha sido demostrado en aplicaciones recientes de redes neuronales profundas para resolver EDPs [5, 37].

4.1.3. Minimización y entrenamiento

Como se explicó en el capítulo anterior, el entrenamiento de redes neuronales profundas se basa en métodos de optimización como el descenso del gradiente y sus variantes modernas,

entre las cuales destaca el algoritmo Adam por su eficiencia y estabilidad en entornos de alta dimensión y funciones de pérdida complejas. En el contexto del método, este proceso tiene como objetivo minimizar la funcional de error $\mathcal{J}(f)$, definida en términos del residuo de la EDP, las condiciones iniciales y de frontera:

$$\mathcal{J}(f) = \left\| \frac{\partial f}{\partial t} + \mathcal{L}f \right\|_{[0,T] \times \Omega}^2 + \|f - g\|_{[0,T] \times \partial\Omega}^2 + \|f(0, x) - u_0(x)\|_\Omega^2, \quad (4.10)$$

donde cada término se evalúa mediante muestreo aleatorio en el dominio correspondiente, lo que evita la necesidad de construir una malla estructurada.

Para ello, se seleccionan puntos (t, x) al azar en el interior del dominio espacio-temporal, en su frontera y sobre la superficie de condición inicial. En cada paso de entrenamiento, se calcula el valor de la función de pérdida y sus derivadas con respecto a los parámetros θ de la red neuronal utilizando diferenciación automática. La actualización de los parámetros se realiza típicamente mediante:

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta \mathcal{J}(f), \quad (4.11)$$

siguiendo el esquema de descenso del gradiente estocástico (SGD) o su versión adaptativa Adam, ya descrita en detalle en el capítulo anterior. Recordando esa formulación, Adam introduce mecanismos de acumulación de momentos de primer y segundo orden, lo que permite ajustar dinámicamente la magnitud del paso para cada parámetro individual, evitando inestabilidades derivadas de gradientes mal condicionados.

La aplicación de Adam en el entrenamiento de redes DGM ha demostrado ser particularmente efectiva en la literatura [5], ya que mejora la velocidad de convergencia y proporciona robustez en dominios de alta dimensión o con geometría compleja. Además, este esquema se puede enriquecer mediante técnicas adicionales, como el **muestreo adaptativo**, que prioriza regiones del dominio donde el residuo de la EDP es elevado y concentra el esfuerzo de entrenamiento donde más se necesita; la **regularización L_2** , que penaliza soluciones con pesos excesivamente grandes y reduce el riesgo de sobreajuste; y el **promedio móvil exponencial**, que suaviza la evolución de los parámetros acumulando versiones anteriores del modelo entrenado.

Estas estrategias combinadas permiten que el entrenamiento de redes neuronales profundas dentro del marco DGM logre aproximar con alta fidelidad soluciones de ecuaciones diferenciales parciales sin mallas.

4.1.4. Algoritmo

A continuación se resume el procedimiento general del método Deep Galerkin (DGM) con arquitectura basada en compuertas, integrando los elementos presentados en las subsecciones anteriores: la formulación matemática de la EDP, la estructura de red neuronal inspirada en LSTM y el proceso de entrenamiento mediante optimización estocástica. Este esquema ofrece una guía operativa para implementar el método de forma sistemática:

Algoritmo 1 Aplicación del método Deep Galerkin con arquitectura de compuertas

Entrada: EDP gobernante con operador \mathcal{L} , condiciones iniciales y de frontera

Salida: Aproximación $u(t, x)$ mediante red neuronal profunda

- 1: Definir la EDP: $\partial_t u + \mathcal{L}u = 0$, con condiciones iniciales y de frontera
- 2: Seleccionar muestras aleatorias en $(t, x) \in \Omega \times [0, T]$
- 3: Construir red neuronal profunda con arquitectura tipo compuertas LSTM
- 4: Definir pérdida total:

$$\mathcal{J}(f) = \|\partial_t f + \mathcal{L}f\|^2 + \text{condiciones iniciales} + \text{condiciones de frontera}$$

- 5: Minimizar $\mathcal{J}(f)$ usando optimizador como Adam o SGD

- 6: Usar red entrenada para predecir solución aproximada $u(t, x)$
-

El algoritmo 1 resume la metodología general del método Deep Galerkin utilizando una arquitectura con compuertas tipo LSTM. Este esquema describe los pasos fundamentales para construir, entrenar y utilizar una red neuronal profunda como aproximador de la solución de una ecuación diferencial parcial, integrando tanto el modelo matemático como el procedimiento de optimización.

Sin embargo, en problemas de mayor complejidad y dimensionalidad, es necesario incorporar técnicas adicionales que mejoren la estabilidad y la eficiencia del entrenamiento. El algoritmo 2 presenta una versión optimizada del método, en la cual se incluyen estrategias clave como:

- **Inicialización de Xavier**, que mejora la propagación del gradiente al comienzo del entrenamiento.
- **Uso de mini-lotes**, lo cual reduce el costo computacional por iteración y permite un entrenamiento estocástico más eficiente.
- **Decaimiento exponencial de la tasa de aprendizaje**, que guía a la red neuronal

hacia mínimos más estables conforme avanza el entrenamiento.

- **Remuestreo adaptativo**, diseñado para evitar un muestreo uniforme del dominio y enfocarse en regiones críticas del espacio-tiempo donde los errores o los gradientes del residuo son mayores.

Algoritmo 2 DGM optimizado con inicialización Xavier, mini-lotes, decaimiento exponencial y remuestreo adaptativo

Entrada: EDP gobernante en 2D, condiciones iniciales y de frontera, región crítica en el tiempo

Salida: Aproximación $u(x, y, t)$ y $v(x, y, t)$ mediante red DGM optimizada

1: **Inicialización:**

- Construir red neuronal DGM con arquitectura tipo compuertas
- Aplicar *inicialización Xavier* a todas las capas lineales
- Definir optimizador Adam y *scheduler* con decaimiento exponencial
- Configurar hiperparámetros

2: **Generación de datos:**

- Muestrear puntos interiores, de frontera e iniciales
- Asignar *peso elevado a región crítica en el tiempo*

3: **para** fase = 1, ..., N **hacer**

- 4: **para** step = 1, ..., STEPS **hacer**
 - 5: Submuestrear puntos interiores ponderados (mini-lote)
 - 6: Habilitar gradientes en los puntos seleccionados
 - 7: Evaluar la red neuronal para obtener u, v
 - 8: Calcular derivadas espaciales y temporales de u y v con *autograd*
 - 9: Construir residuos de la ecuación
 - 10: Calcular pérdida interior: \mathcal{L}_P
 - 11: Calcular pérdida de frontera e inicial: $\mathcal{L}_{BC/IC}$
 - 12: Calcular pérdida total: $\mathcal{J} = \lambda_P \mathcal{L}_P + \lambda_B \mathcal{L}_{BC/IC}$
 - 13: Propagación hacia atrás: $\nabla \mathcal{J} \rightarrow \text{backward}()$
 - 14: Actualizar pesos: *optimizer.step()*
 - 15: Actualizar tasa de aprendizaje: *scheduler.step()*
 - 16: Registrar \mathcal{J} en el historial
 - 17: **fin para**
 - 18: **fin para**
-

Estas técnicas, en conjunto, permiten acelerar la convergencia, mejorar la precisión de la solución aprendida y adaptarse mejor a estructuras locales de la solución, como capas límite o singularidades suaves. El algoritmo resultante representa una implementación robusta del DGM para problemas como la ecuación de Burgers 2D o la ecuación de Schrödinger en dominios tridimensionales.

En el algoritmo DGM optimizado, los términos λ_P y λ_B de la pérdida total son hiperparámetros de ponderación que juegan un papel crucial en el éxito del entrenamiento. Su función es equilibrar la contribución de los dos componentes principales de la función de pérdida total \mathcal{J} . En particular, la elección adecuada de estos parámetros de penalización influye directamente en la capacidad de la red neuronal para aprender de manera efectiva las condiciones iniciales y de frontera, evitando que el modelo priorice en exceso la minimización del residuo de la EDP a costa del cumplimiento de dichas condiciones físicas.

4.2. Adaptación del método Deep Galerkin

En esta sección se detalla la aplicación del método a un conjunto diverso de ecuaciones diferenciales parciales que modelan fenómenos representativos de distintas áreas de la física y la ingeniería. El objetivo es demostrar la versatilidad y robustez del enfoque en problemas con características matemáticas y físicas variadas.

Primero, se aborda la mecánica estructural clásica a través de las ecuaciones de Euler-Bernoulli, Timoshenko y Kirchhoff-Love. Estos modelos, que describen la dinámica transversal de vigas y placas delgadas, se caracterizan por la presencia de derivadas espaciales de alto orden. Se mostrará cómo su reformulación a sistemas de orden reducido es una estrategia clave para facilitar la implementación mediante redes neuronales profundas y el uso eficiente de la diferenciación automática.

A continuación, el análisis se extiende a la dinámica de fluidos con la ecuación de Burgers viscosa. Este modelo no lineal canónico, que combina efectos de advección y difusión, sirve como un importante caso de prueba para evaluar la capacidad del método de manejar no linealidades intrínsecas en el sistema.

Finalmente, el enfoque se aplica a la mecánica cuántica mediante la resolución de la ecuación de Schrödinger dependiente del tiempo, un pilar de la física moderna. Este caso introduce desafíos únicos, como el manejo de funciones de onda con valores complejos y la aplicación en dominios de alta dimensionalidad, donde los métodos numéricos

tradicionales suelen ser computacionalmente prohibitivos.

Para cada uno de estos modelos, se detallan las funciones de pérdida específicas, las condiciones de contorno e iniciales, y las estrategias de entrenamiento aplicadas.

4.2.1. Ecuación de Euler–Bernoulli

La ecuación de Euler–Bernoulli que modela la dinámica transversal de una viga elástica homogénea sin amortiguamiento, bajo condiciones iniciales y de frontera apropiadas, se expresa como:

$$\frac{\partial^2 w(x, t)}{\partial t^2} + \lambda \frac{\partial^4 w(x, t)}{\partial x^4} = 0, \quad (x, t) \in (0, L) \times (0, T), \quad (4.12)$$

donde $w(x, t)$ representa el desplazamiento transversal en el punto x y tiempo t , y $\lambda = \frac{EI}{\rho A}$ es una constante física que depende del módulo de elasticidad E , el momento de inercia I , la densidad ρ y el área de la sección transversal A .

Para facilitar la implementación en redes neuronales profundas mediante el método DGM,

se introduce la variable auxiliar $M(x, t) = \frac{\partial^2 w}{\partial x^2}(x, t)$, lo que permite reformular la ecuación original como un sistema acoplado de ecuaciones de segundo orden:

$$\begin{cases} \frac{\partial^2 w(x, t)}{\partial t^2} + \lambda \frac{\partial^2 M(x, t)}{\partial x^2} = 0, \\ M(x, t) = \frac{\partial^2 w(x, t)}{\partial x^2}. \end{cases} \quad (4.13)$$

Este sistema permite una evaluación más estable de las derivadas espaciales de alto orden, facilitando la diferenciación automática durante el entrenamiento.

Para aplicar el método Deep Galerkin, se construye una red neuronal $f(x, t; \boldsymbol{\theta})$ que approxima la solución $w(x, t)$ y, opcionalmente, otra red $m(x, t; \boldsymbol{\theta})$ para $M(x, t)$. La función de pérdida a minimizar incorpora los residuos de ambas ecuaciones, junto con las condiciones iniciales y de frontera:

$$\begin{aligned} \mathcal{J}_{EB}(f, m) = & \left\| \frac{\partial^2 f}{\partial t^2} + \lambda \frac{\partial^2 m}{\partial x^2} \right\|_{(0, L) \times (0, T)}^2 + \left\| m - \frac{\partial^2 f}{\partial x^2} \right\|_{(0, L) \times (0, T)}^2 \\ & + \|f(x, 0) - w_0(x)\|_{(0, L)}^2 + \left\| \frac{\partial f}{\partial t}(x, 0) - v_0(x) \right\|_{(0, L)}^2 + \mathcal{B}(f) \end{aligned} \quad (4.14)$$

donde $w_0(x)$ y $v_0(x)$ son las condiciones iniciales de desplazamiento y velocidad, respectivamente. El término $\mathcal{B}(f)$ representa las condiciones de frontera. Por ejemplo, para una viga empotrada en ambos extremos se tiene:

$$\mathcal{B}(f) = \|f(0, t)\|^2 + \left\| \frac{\partial f}{\partial x}(0, t) \right\|^2 + \|f(L, t)\|^2 + \left\| \frac{\partial f}{\partial x}(L, t) \right\|^2. \quad (4.15)$$

4.2.2. Ecuación de Timoshenko

La ecuación de Timoshenko modela la dinámica transversal de una viga considerando tanto la flexión como la deformación por cortante. En su forma adimensional, el sistema de ecuaciones para el desplazamiento transversal $w(x, t)$ y la rotación de la sección transversal $\theta(x, t)$ se expresa como:

$$\begin{cases} \frac{\partial^2 \theta}{\partial t^2} - \frac{\partial^2 \theta}{\partial x^2} + \left(\theta - \frac{\partial w}{\partial x} \right) = 0, \\ \frac{\partial^2 w}{\partial t^2} + \frac{\partial}{\partial x} \left(\theta - \frac{\partial w}{\partial x} \right) = g(x, t), \end{cases} \quad (4.16)$$

donde $g(x, t)$ representa una fuente transversal externa. Este sistema corresponde a una versión adimensional simplificada del modelo clásico de Timoshenko, donde los parámetros físicos han sido reescalados para facilitar la implementación computacional.

Para resolver este sistema mediante el método Deep Galerkin, se introducen dos redes neuronales $f(x, t; \boldsymbol{\theta}_w)$ y $g(x, t; \boldsymbol{\theta}_\theta)$, que aproximan las soluciones $w(x, t)$ y $\theta(x, t)$, respectivamente. La función de pérdida a minimizar incorpora los residuos de ambas ecuaciones del sistema, las condiciones iniciales y las condiciones de frontera:

$$\begin{aligned} \mathcal{J}_T(f, g) = & \left\| \frac{\partial^2 g}{\partial t^2} - \frac{\partial^2 g}{\partial x^2} + \left(g - \frac{\partial f}{\partial x} \right) \right\|_{(0,1) \times (0,T)}^2 \\ & + \left\| \frac{\partial^2 f}{\partial t^2} + \frac{\partial}{\partial x} \left(g - \frac{\partial f}{\partial x} \right) - g(x, t) \right\|_{(0,1) \times (0,T)}^2 \\ & + \|f(x, 0) - w_0(x)\|_{(0,1)}^2 + \left\| \frac{\partial f}{\partial t}(x, 0) - v_0(x) \right\|_{(0,1)}^2 \\ & + \|g(x, 0) - \theta_0(x)\|_{(0,1)}^2 + \left\| \frac{\partial g}{\partial t}(x, 0) - \omega_0(x) \right\|_{(0,1)}^2 + \mathcal{B}(f, g), \end{aligned} \quad (4.17)$$

donde $w_0(x)$, $v_0(x)$, $\theta_0(x)$ y $\omega_0(x)$ representan las condiciones iniciales del desplazamiento, velocidad, rotación y velocidad angular, respectivamente.

El término de condiciones de frontera $\mathcal{B}(f, g)$ depende del tipo de apoyo considerado. Por ejemplo, para una viga empotrada en ambos extremos, se impone:

$$\mathcal{B}(f, g) = \|f(0, t)\|^2 + \left\| \frac{\partial f}{\partial x}(0, t) \right\|^2 + \|f(1, t)\|^2 + \left\| \frac{\partial f}{\partial x}(1, t) \right\|^2 + \|g(0, t)\|^2 + \|g(1, t)\|^2. \quad (4.18)$$

4.2.3. Ecuación de Kirchhoff–Love en placas delgadas

La teoría de Kirchhoff–Love describe la dinámica de placas delgadas considerando que las secciones normales permanecen perpendiculares a la superficie media. En su forma adimensional y reformulada para implementación con el método Deep Galerkin, la ecuación de cuarto orden (ecuación 2.9) se descompone en un sistema acoplado de segundo orden utilizando una variable auxiliar $M(x, y, t)$ y tomando parámetros adimensionales, de modo que la ecuación queda en su forma simplificada por

$$\begin{cases} \frac{\partial^2 w}{\partial t^2} + \Delta M = q \\ M = \Delta w, \end{cases} \quad (4.19)$$

donde $w(x, y, t)$ representa el desplazamiento transversal de la placa, $q(x, y, t)$ es una carga transversal, y Δ denota el operador Laplaciano en las variables espaciales x e y :

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}.$$

Este sistema acoplado permite evitar la evaluación directa de derivadas de cuarto orden mediante el uso de diferenciación automática, lo cual es especialmente útil en redes neuronales profundas.

Se introducen dos redes neuronales $f(x, y, t; \boldsymbol{\theta}_w)$ y $m(x, y, t; \boldsymbol{\theta}_M)$, que aproximan las funciones w y M , respectivamente. La función de pérdida total incorpora los residuos de ambas ecuaciones, junto con las condiciones iniciales y de frontera:

$$\begin{aligned} \mathcal{J}_{KL}(f, m) &= \left\| \frac{\partial^2 f}{\partial t^2} + \Delta m - q(x, y, t) \right\|_{\Omega \times (0, T)}^2 + \|m - \Delta f\|_{\Omega \times (0, T)}^2 \\ &\quad + \|f(x, y, 0) - w_0(x, y)\|_{\Omega}^2 + \left\| \frac{\partial f}{\partial t}(x, y, 0) \right\|_{\Omega}^2 \\ &\quad + \|m(x, y, 0) - M_0(x, y)\|_{\Omega}^2 + \mathcal{B}(f, m) \end{aligned} \quad (4.20)$$

donde $w_0(x, y)$ y $M_0(x, y)$ representan las condiciones iniciales del desplazamiento y del momento auxiliar, respectivamente. En el caso de placas simplemente apoyadas o empotradas, se consideran condiciones homogéneas del tipo:

$$\mathcal{B}(f, m) = \|f\|_{\partial\Omega \times (0, T)}^2 + \|m\|_{\partial\Omega \times (0, T)}^2. \quad (4.21)$$

4.2.4. Ecuación de Burgers viscosa en dos dimensiones

La ecuación de Burgers viscosa en dos dimensiones modela la evolución de un campo de velocidades $\mathbf{u}(x, y, t) = (u, v)$ bajo la acción combinada de advección no lineal y difusión viscoso-lineal. Esta EDP constituye un sistema no lineal de segundo orden que conserva propiedades fundamentales de las ecuaciones de Navier–Stokes, y es frecuentemente utilizada como banco de pruebas para esquemas numéricos avanzados. Su estructura simétrica y la existencia de soluciones exactas autosemejantes la convierten en un caso ideal para validar la precisión del método. El sistema de ecuaciones gobernante es:

$$\begin{cases} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right), \end{cases} \quad (x, y) \in \Omega \subset \mathbb{R}^2, \quad t > 0 \quad (4.22)$$

Para aproximar la solución con el método se introducen dos redes neuronales profundas $f(x, y, t; \boldsymbol{\theta}_u)$ y $g(x, y, t; \boldsymbol{\theta}_v)$, que representan las componentes u y v del campo de velocidades, respectivamente. Cada red es entrenada de manera conjunta sobre una función de pérdida que incluye los residuos de ambas ecuaciones, así como las condiciones iniciales y de frontera impuestas por la solución exacta conocida.

La función de pérdida total se define como:

$$\begin{aligned} \mathcal{J}_{\text{B2D}}(f, g) = & \left\| \frac{\partial f}{\partial t} + f \frac{\partial f}{\partial x} + g \frac{\partial f}{\partial y} - \nu \Delta f \right\|_{\Omega \times (0, T)}^2 \\ & + \left\| \frac{\partial g}{\partial t} + f \frac{\partial g}{\partial x} + g \frac{\partial g}{\partial y} - \nu \Delta g \right\|_{\Omega \times (0, T)}^2 \\ & + \|f(x, y, 0) - u_0(x, y)\|_{\Omega}^2 + \|g(x, y, 0) - v_0(x, y)\|_{\Omega}^2 + \mathcal{B}(f, g) \end{aligned} \quad (4.23)$$

donde

$$\mathcal{B}(f, g) = \|f(x, y, t) - u_r(x, y, t)\|_{\partial\Omega \times (0, T)}^2 + \|g(x, y, t) - v_r(x, y, t)\|_{\partial\Omega \times (0, T)}^2, \quad (4.24)$$

Además, $u_0(x, y)$, $v_0(x, y)$, $u_r(x, y, t)$ y $v_r(x, y, t)$ se obtienen evaluando la solución exacta autosemejante del problema. El operador Δ corresponde al Laplaciano bidimensional. Durante el entrenamiento, los puntos de evaluación se muestran aleatoriamente en el dominio tridimensional $(x, y, t) \in [0, 1]^2 \times (0, T]$. La diferenciación automática se emplea para obtener los gradientes necesarios, incluyendo derivadas no lineales mixtas. Como en los casos anteriores, se utiliza un algoritmo de optimización como Adam para ajustar los parámetros de ambas redes neuronales.

4.2.5. Ecuación de Schrödinger unidimensional

Partiendo de la forma general de la ecuación de Schrödinger dependiente del tiempo en una dimensión espacial, dada en la ecuación (2.16), se considera el caso particular sin potencial externo $V(x) = 0$, lo cual conduce a una simplificación significativa del modelo. Además, para facilitar el análisis y su implementación computacional mediante el método Deep Galerkin (DGM), se realiza un cambio de variables hacia una formulación adimensional, en la que se normalizan las constantes físicas como $\hbar = 1$ y $m = 1$, resultando en la siguiente ecuación:

$$i \frac{\partial \psi}{\partial t} + \frac{1}{2} \frac{\partial^2 \psi}{\partial x^2} = 0, \quad (x, t) \in (0, 1) \times (0, 1] \quad (4.25)$$

donde $\psi(x, t)$ representa la función de onda compleja adimensionalizada. Este caso específico ha sido estudiado en [38] donde se propone una solución exacta con características gaussianas oscilantes que se propaga en el tiempo. Dicha solución sirve como referencia para evaluar la precisión del método DGM bajo condiciones iniciales y de frontera conocidas.

Esta ecuación se complementa con la condición inicial:

$$\psi(x, 0) = \psi_0(x),$$

y condiciones de frontera:

$$\psi(0, t) = g(0, t), \quad \psi(1, t) = g(1, t).$$

La solución exacta utilizada como referencia es:

$$\psi(x, t) = \sqrt{\frac{i}{i - 200t}} \exp\left(\frac{50k_0^2 t - (10x - 5)k_0 - i(10x - 5)^2}{i - 200t}\right), \quad \text{con } k_0 = 1, \quad (4.26)$$

proporcionando tanto el valor inicial como las condiciones de frontera exactas. Esta solución exhibe características oscilatorias en el dominio y se propaga en el tiempo manteniendo su estructura gaussiana con fase compleja.

Para resolver el problema mediante el método Deep Galerkin (DGM), se propone una red neuronal que toma como entrada $(x, t) \in [0, 1]^2$ y produce como salida la parte real $u(x, t)$ y la parte imaginaria $v(x, t)$, tales que:

$$u(x, t) = \text{Re}[\psi(x, t)], \quad v(x, t) = \text{Im}[\psi(x, t)].$$

Al sustituir esta representación en la ecuación de Schrödinger y separar partes reales e imaginarias, se obtiene el sistema acoplado:

$$\begin{cases} \frac{\partial u}{\partial t} = -\frac{1}{2} \frac{\partial^2 v}{\partial x^2}, \\ \frac{\partial v}{\partial t} = \frac{1}{2} \frac{\partial^2 u}{\partial x^2}. \end{cases} \quad (4.27)$$

Durante el entrenamiento, se emplean puntos de muestreo aleatorios en el dominio $(x, t) \in [0, 1]^2$. A partir de estos, se calcula la pérdida total como:

$$\begin{aligned} \mathcal{J}_{\text{Sch}}(u, v) = & \left\| \frac{\partial u}{\partial t} + \frac{1}{2} \frac{\partial^2 v}{\partial x^2} \right\|_{[0,1]^2}^2 + \left\| \frac{\partial v}{\partial t} - \frac{1}{2} \frac{\partial^2 u}{\partial x^2} \right\|_{[0,1]^2}^2 \\ & + \|u(x, 0) - \text{Re}[\psi_0(x)]\|_{[0,1]}^2 + \|v(x, 0) - \text{Im}[\psi_0(x)]\|_{[0,1]}^2 \\ & + \|u(x, t) - \text{Re}[g(x, t)]\|_{\partial[0,1] \times (0,1]}^2 + \|v(x, t) - \text{Im}[g(x, t)]\|_{\partial[0,1] \times (0,1]}^2 \end{aligned} \quad (4.28)$$

donde los términos de condición inicial y de frontera se extraen directamente de la solución exacta (4.26).

4.2.6. Ecuación de Schrödinger para el átomo de hidrógeno

La resolución numérica del sistema de ecuaciones acoplado (2.26) mediante el DGM, ahora se basa en aproximar la función de onda compleja $\psi(\mathbf{r}_e, \mathbf{r}_p, t)$ a través de una red neuronal profunda que toma como entrada el vector adimensionalizado $(x_e, y_e, z_e, x_p, y_p, z_p, t)$ y produce como salida las componentes reales u e imaginarias v de la función de onda, donde $\mathbf{r}_e = (x_e, y_e, z_e)$ y $\mathbf{r}_p = (x_p, y_p, z_p)$ representan las posiciones del electrón y del protón, respectivamente, en el espacio tridimensional.

El entrenamiento de la red se realiza minimizando una función de pérdida que incorpora los residuos del sistema (2.26), junto con penalizaciones por desviaciones respecto a las condiciones iniciales y de frontera.

La función de pérdida total se define como:

$$\begin{aligned} \mathcal{J}_H(u, v) = & \left\| \frac{\partial u}{\partial t} + \frac{\mu}{2} \Delta_{\mathbf{r}_p} v + \frac{1}{2} \Delta_{\mathbf{r}_e} v + \frac{1}{|\mathbf{r}_p - \mathbf{r}_e|} v \right\|^2 + \left\| \frac{\partial v}{\partial t} - \frac{\mu}{2} \Delta_{\mathbf{r}_p} u - \frac{1}{2} \Delta_{\mathbf{r}_e} u - \frac{1}{|\mathbf{r}_p - \mathbf{r}_e|} u \right\|^2 \\ & + \|u(\mathbf{r}_e, \mathbf{r}_p, 0) - \text{Re}[\psi_0]\|^2 + \|v(\mathbf{r}_e, \mathbf{r}_p, 0) - \text{Im}[\psi_0]\|^2 + \mathcal{B}(u, v) \end{aligned} \quad (4.29)$$

donde el operador Δ denota el Laplaciano tridimensional respecto a las coordenadas correspondientes, y $\mathcal{B}(u, v)$ representa los términos de penalización por condiciones de frontera. Estos términos se obtienen de la solución exacta si está disponible, o se imponen conforme al comportamiento esperado del sistema.

La condición inicial utilizada para entrenar el modelo consiste en una distribución gaussiana tridimensional normalizada, centrada en el origen del sistema la cual se muestra en la ecuación siguiente

$$\psi_0(\mathbf{r}) = A e^{-\alpha|\mathbf{r}|^2}, \quad A = \left(\frac{2\alpha}{\pi} \right)^{3/4}, \quad \text{Im}[\psi_0] = 0, \quad (4.30)$$

donde $\alpha = 0.75$ controla la concentración espacial inicial del electrón en torno al origen (posición del protón). Esta forma asegura que $\int |\psi_0|^2 d\mathbf{r} = 1$, cumpliendo así con la normalización de la función de onda.

Una vez entrenada, la red permite evaluar de forma eficiente la evolución de la función de onda en cualquier punto del dominio, y por tanto, calcular cantidades físicas de interés como la densidad de probabilidad conjunta:

$$|\psi(\mathbf{r}_e, \mathbf{r}_p, t)|^2 = u^2 + v^2.$$

Para visualizar esta densidad y simular el comportamiento del sistema cuántico, se puede aplicar un algoritmo de Metropolis–Hastings 3 con función de aceptación basada en la distribución de probabilidad anterior, lo cual permite generar nubes de puntos representativas de la configuración electrónica.

Este enfoque, además de evitar la discretización en malla y ser escalable a dimensiones mayores, proporciona una base sólida para la extensión a problemas más complejos, como sistemas de múltiples partículas o interacciones externas variables en el tiempo.

En la parte experimental, se estableció un dominio de simulación adimensional para el sistema de dos cuerpos correspondiente al átomo de hidrógeno. El espacio total de configuración se restringió al hipercubo:

$$(\mathbf{r}_e, \mathbf{r}_p) \in [-4, 4]^6, \quad t \in [0, 1],$$

donde $\mathbf{r}_e = (x_e, y_e, z_e)$ y $\mathbf{r}_p = (x_p, y_p, z_p)$ corresponden a las coordenadas del electrón y del protón, respectivamente. En las seis variables espaciales se impusieron condiciones de frontera homogéneas tipo Dirichlet para la función de onda:

$$\psi(\mathbf{r}_e, \mathbf{r}_p, t) = 0 \quad \text{en} \quad \partial\Omega \times [0, 1],$$

las cuales son compatibles con la rápida decaída del estado inicial gaussiano y con la naturaleza local de la función de onda en este tipo de sistemas enlazados.

4.2.7. Ecuación de Schrödinger para el átomo de hidrógeno con protón fijo

La resolución numérica del sistema acoplado (2.29) mediante el método se basa en aproximar la función de onda compleja adimensional $\psi(\mathbf{r}, t)$ donde $\mathbf{r} = (x, y, z)$ mediante la red neuronal profunda que recibe como entrada el vector (\mathbf{r}, t) , correspondiente a las coordenadas espaciales del electrón y el tiempo, y devuelve como salida las funciones reales $u(\mathbf{r}, t)$ y $v(\mathbf{r}, t)$ que representan las partes real e imaginaria de la función de onda, respectivamente:

$$\psi(\mathbf{r}, t) = u(\mathbf{r}, t) + i v(\mathbf{r}, t).$$

La función de pérdida total se construye a partir del sistema de ecuaciones (2.29), integrando residuos en el dominio interior junto con las condiciones iniciales y de frontera extraídas de una solución exacta de referencia o diseñada para el experimento. Su forma general es:

$$\begin{aligned} \mathcal{J}_{H0}(u, v) = & \left\| \frac{\partial u}{\partial t} + \frac{1}{2} \Delta v + \frac{1}{|\mathbf{r}|} v \right\|^2 + \left\| \frac{\partial v}{\partial t} - \frac{1}{2} \Delta u - \frac{1}{|\mathbf{r}|} u \right\|^2 \\ & + \|u(\mathbf{r}, 0) - \text{Re}[\psi_0(\mathbf{r})]\|^2 + \|v(\mathbf{r}, 0) - \text{Im}[\psi_0(\mathbf{r})]\|^2 + \mathcal{B}(u, v) \end{aligned} \quad (4.31)$$

donde $\mathcal{B}(u, v)$ agrupa los términos correspondientes a las condiciones de frontera impuestas sobre la superficie del dominio tridimensional considerado. Estas condiciones pueden derivarse directamente de una solución analítica conocida o establecerse de forma compatible con el problema físico.

Para el caso del protón fijo, se utilizó la misma condición inicial gaussiana normalizada que en el sistema de dos cuerpos, expresada como:

$$\psi_0(\mathbf{r}) = A e^{-\alpha|\mathbf{r}|^2}, \quad A = \left(\frac{2\alpha}{\pi}\right)^{3/4}, \quad \text{Im}[\psi_0] = 0, \quad (4.32)$$

Esta condición inicial garantiza la normalización de la función de onda y representa un estado localizado físicamente plausible para el electrón.

A partir de la solución aprendida, se puede calcular la densidad de probabilidad del electrón como:

$$|\psi(\mathbf{r}, t)|^2 = u(\mathbf{r}, t)^2 + v(\mathbf{r}, t)^2,$$

la cual puede utilizarse para análisis físicos detallados o para la generación de configuraciones representativas mediante muestreo tipo Metropolis–Hastings.

En este caso, se trabajó sobre un dominio tridimensional reducido, correspondiente a las coordenadas del electrón en torno al protón fijo en el origen. El dominio considerado para la simulación fue:

$$\mathbf{r} \in [-4, 4]^3, \quad t \in [0, 1],$$

y se aplicaron condiciones de frontera homogéneas tipo Dirichlet para la función de onda:

$$\psi(\mathbf{r}, t) = 0 \quad \text{en} \quad \partial\Omega \times [0, 1],$$

con el propósito de contener el soporte efectivo de la función de onda inducida por la condición inicial gaussiana. Esta estrategia evita interferencias artificiales en la evolución de la solución y facilita el entrenamiento estable del modelo.

Capítulo 5

Experimentos numéricos

En este capítulo se presentan una serie de experimentos numéricos dirigidos a resolver diferentes ecuaciones en derivadas parciales representativas de fenómenos en mecánica estructural y física matemática, utilizando el *Deep Galerkin Method* explicado en el capítulo anterior. Los modelos abordados comprenden tanto sistemas lineales como no lineales, en una o varias dimensiones espaciales, con soluciones exactas conocidas que permiten validar cuantitativamente la calidad de la aproximación obtenida mediante redes neuronales.

Específicamente, se consideran los siguientes modelos:

1. La ecuación de Euler–Bernoulli en una dimensión espacial, reformulada como sistema acoplado.
2. La ecuación de Timoshenko, que incorpora el acoplamiento entre deflexión y rotación.
3. La ecuación de Kirchhoff–Love en dominio bidimensional, modelando placas delgadas.
4. La ecuación de Burgers viscosa en dos dimensiones, como modelo canónico de dinámica de fluidos.
5. La ecuación de Schrödinger dependiente del tiempo unidimensional, en su formulación compleja.
6. El sistema acoplado derivado de la ecuación de Schrödinger aplicada al átomo de hidrógeno.

Cada una de estas EDP se reformula, cuando es conveniente, en forma de sistemas acoplados de segundo orden para facilitar la implementación automática de derivadas mediante diferenciación automática. Esta reformulación permite evitar el cálculo explícito de derivadas de orden superior, cuya obtención directa mediante la construcción del grafo computacional resulta computacionalmente costosa, tanto en memoria como en tiempo de procesamiento.

La metodología se basa en entrenar redes neuronales profundas tipo DGM, diseñadas para aprender soluciones de EDP directamente a partir del residuo de la ecuación, las condiciones iniciales y de frontera. Estas redes están construidas con función de activación `tanh` o `silu`, y una arquitectura compuesta por múltiples capas ocultas con un número fijo de neuronas por capa. Las configuraciones específicas de cada experimento, incluyendo el número de puntos de muestreo, capas y neuronas, se detallan en las respectivas tablas. Los entrenamientos se han realizado bajo dos esquemas distintos de tasa de aprendizaje (learning rate): uno constante y otro con decaimiento exponencial. En este último caso, la tasa de aprendizaje se actualiza en cada época según la fórmula:

$$\text{LR}_{\text{epoch}} = \text{LR}_0 \cdot \gamma^{\text{epoch}}$$

donde LR_0 es la tasa de aprendizaje inicial, $\gamma \in (0, 1)$ es un factor de decaimiento, y se impone un umbral mínimo LR_{\min} . Esta estrategia permite mejorar la estabilidad y precisión del entrenamiento, especialmente en problemas con soluciones oscilatorias o en dominios de mayor dimensionalidad.

Todos los experimentos fueron ejecutados en la plataforma **Google Colab**, utilizando entornos con acceso a unidades de procesamiento gráfico (GPU). Dependiendo del experimento, se utilizó una **GPU NVIDIA A100** o una **GPU NVIDIA T4**, las cuales proporcionan una aceleración significativa en el proceso de entrenamiento. En particular, la GPU A100 (con arquitectura Ampere) posee 40 GB de memoria HBM2 y un elevado número de núcleos tensoriales, lo que permite entrenamientos más rápidos y eficientes para redes profundas. Por su parte, la GPU T4 (basada en arquitectura Turing) es más modesta, pero aún adecuada para entrenamientos en problemas de menor escala.

Para cada uno de los modelos estudiados, se incluyen las siguientes secciones: formulación del sistema acoplado, solución analítica exacta (cuando está disponible), condiciones de frontera e iniciales, configuración de la red neuronal, evolución de la pérdida durante el entrenamiento, visualización de las soluciones aproximadas y comparación con la solución

exacta, así como una evaluación cuantitativa del error en la norma L_2 , tanto absoluta como relativa.

De esta forma, se busca validar la capacidad del método DGM para aproximar con alta fidelidad soluciones de EDPs de segundo y cuarto orden en problemas estructurales, destacando sus ventajas como método sin mallas y su potencial para extenderse a geometrías y modelos más complejos.

Adicionalmente, el cálculo de los errores absolutos y relativos se realizó bajo un enfoque basado en la norma L_2 discreta, evaluada sobre un conjunto de puntos de validación independientes del entrenamiento, lo que permite una estimación objetiva del desempeño de las soluciones aproximadas generadas por el modelo.

5.1. Ecuación de Euler-Bernoulli como sistema acoplado

En este primer experimento, resolvemos numéricamente la ecuación de Euler-Bernoulli en una dimensión espacial. Con el fin de evitar el uso de derivadas espaciales de cuarto orden, reformulamos la ecuación como se muestra en la ecuación 4.13 como un sistema acoplado de segundo orden en el espacio y el tiempo:

$$\begin{cases} \frac{\partial^2 w}{\partial t^2} + \frac{\partial^2 M}{\partial x^2} = 0, \\ M = \frac{\partial^2 w}{\partial x^2}, \end{cases}$$

donde $w(x, t)$ es la deflexión transversal de la viga y $M(x, t)$ representa el momento flector.

Para este experimento, consideramos la solución exacta:

$$w(x, t) = \frac{\pi}{2} \sin(x) \cos(t),$$

de donde se deducen las siguientes condiciones:

- Condiciones iniciales:

$$w(x, 0) = \frac{\pi}{2} \sin(x), \quad \frac{\partial w}{\partial t}(x, 0) = 0, \quad M(x, 0) = -\frac{\pi}{2} \sin(x) \quad M_t(x, 0) = 0$$

- Condiciones de frontera (viga empotrada en ambos extremos):

$$w(0, t) = w(\pi, t) = 0, \quad M(0, t) = M(\pi, t) = 0$$

Tabla 5-1: Hiperparámetros utilizados para la resolución de la ecuación de Euler–Bernoulli.

No. de Puntos	N_i	N_f	N_{int}	Capas	Neuronas	Épocas	Tasa de Apr.
1300	100	200	1000	3	50	8000	1×10^{-3} (constante)
1300	100	200	1000	3	50	30000	1×10^{-3} (constante)
1300	100	200	1000	3	50	8000	1×10^{-2} (d. exponencial)

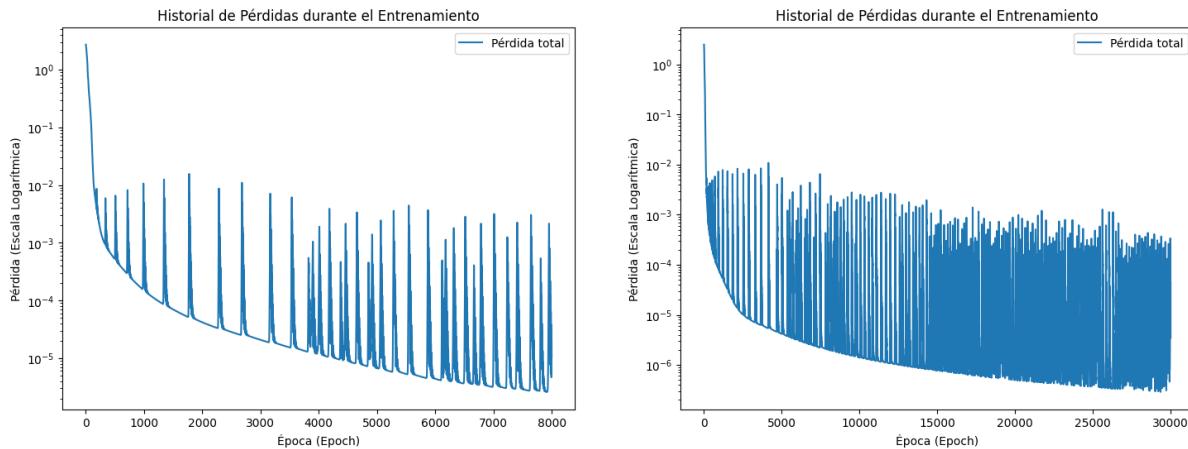
Nota: En la tercera configuración se emplea un decaimiento exponencial de la forma $LR = LR_0 \cdot \gamma^{epoch}$, con $\gamma = 0.999113$ y $LR_{\min} = 1 \times 10^{-5}$.

Se utiliza una red neuronal basada en el algoritmo 1, con activación `tanh`, con distintos hiperparámetros que se detallan en la tabla 5-1. La red predice simultáneamente $w(x, t)$ y $M(x, t)$. El entrenamiento se realiza minimizando una función de pérdida 4.14, cuya evolución se muestra en la figura 5.1.

Durante el entrenamiento del modelo se generaron puntos de muestreo aleatorios en el dominio espacio-temporal $(x, t) \in [0, \pi] \times [0, 1]$, como se muestra en la figura 5.2. Estos puntos se utilizan para imponer el cumplimiento del residuo de la ecuación de Euler–Bernoulli escrita en su forma aumentada, donde el desplazamiento $w(x, t)$ y el momento flector $M(x, t)$ son predichos simultáneamente por la red neuronal. Además, se generan conjuntos independientes de puntos para las condiciones iniciales y de frontera. La aleatoriedad del muestreo en el dominio permite evitar problemas asociados a discretizaciones rígidas y promueve una mejor cobertura del espacio de entrenamiento durante las distintas fases de optimización.

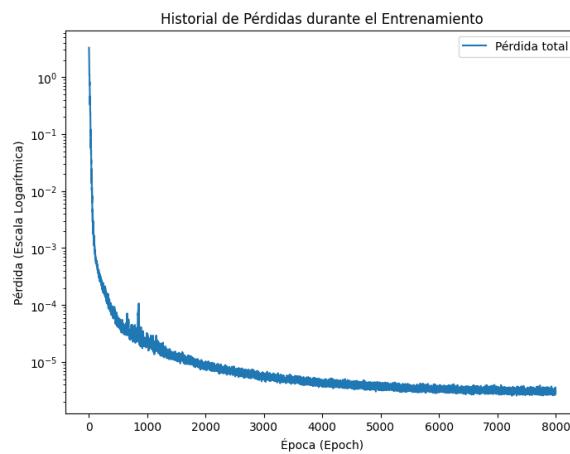
La figura 5.3 presenta la evolución de la aproximación numérica $w^*(x, t)$ en distintos instantes de tiempo. Se muestra también el error absoluto, calculado como la diferencia entre la solución aproximada $w^*(x, t)$ y la solución exacta $w(x, t)$. Esta comparación visual permite evaluar la precisión del modelo en la reconstrucción del comportamiento dinámico de la solución de la ecuación de Euler–Bernoulli.

Los resultados presentados en la tabla 5-1 permiten establecer una comparación clara entre las distintas configuraciones de entrenamiento evaluadas para la resolución de la ecuación de Euler–Bernoulli en su forma acoplada. Se observa que el uso de una tasa de aprendizaje constante igual a 1×10^{-3} , junto con un número moderado de épocas (8K), conduce a una muy baja magnitud del error relativo para la deflexión transversal w



((a)) Pérdida durante 8k épocas - Tiempo de entrenamiento 18 minutos

((b)) Pérdida durante 30k épocas - Tiempo de entrenamiento 40 minutos



((c)) Pérdida durante 8k épocas - Tiempo de entrenamiento 6 minutos

Figura 5.1: Comparación de la evolución de la pérdida de entrenamiento en función del número de épocas para la viga de Euler–Bernoulli en 1D.

(5.80×10^{-4}) , lo cual evidencia una convergencia efectiva en un tiempo de entrenamiento relativamente corto. Sin embargo, al aumentar el número de épocas a 30K, se obtiene una mejora significativa en la predicción del momento fletor M , con una reducción en su error relativo de 2.34×10^{-3} a 1.23×10^{-3} , aunque con un leve incremento en el error de w , posiblemente asociado a sobreajuste localizado en regiones menos relevantes para la dinámica de M .

Por otro lado, la tercera configuración, con una tasa de aprendizaje inicial menos agresiva (1×10^{-2}) y un esquema de decaimiento exponencial, logra una cobertura adecuada del dominio en menor tiempo, aunque incurre en mayores errores absolutos y relativos,

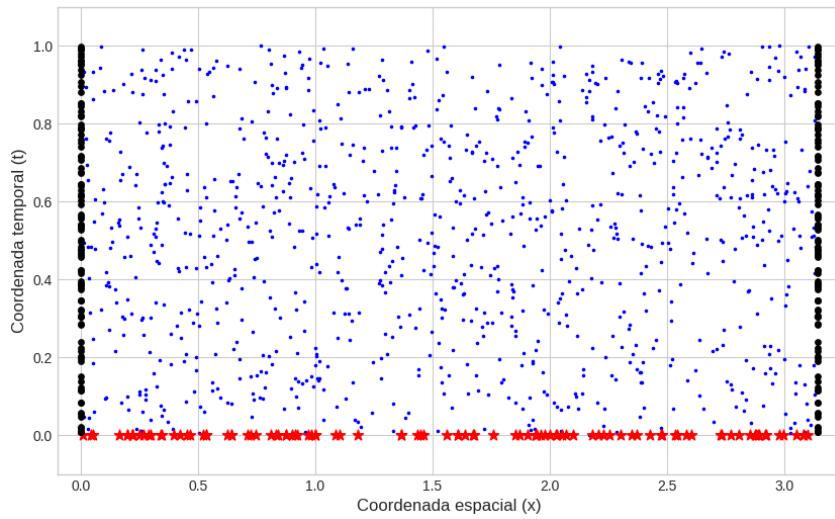


Figura 5.2: Ejemplo de la selección de puntos de muestreo aleatorios en el dominio espacio-temporal. Los puntos azules corresponden al interior, los puntos rojos a la condición inicial en $t = 0$, y los puntos negros a las condiciones de frontera en $x = 0$ y $x = \pi$.

Tabla 5-2: Errores L_2 absolutos y relativos de la validación en ecuación de Euler-Bernoulli

Épocas	Variable	Error L_2 absoluto	Error L_2 relativo
8K	w	5.4663×10^{-2}	5.8030×10^{-4}
	M	2.2050×10^{-1}	2.3408×10^{-3}
30K	w	6.6006×10^{-2}	7.0071×10^{-4}
	M	1.1566×10^{-1}	1.2278×10^{-3}
8K	w	9.3615×10^{-2}	9.9381×10^{-4}
	M	3.3472×10^{-1}	3.5533×10^{-3}

Nota: En la tercera configuración se emplea un decaimiento exponencial como se indica en la tabla 5-1

particularmente para M . Esto se debe a que el decaimiento exponencial permite realizar pasos iniciales más grandes cuando los gradientes son aún significativos, acelerando la exploración del espacio de parámetros y reduciendo rápidamente la función de pérdida en las primeras etapas del entrenamiento. Conforme avanza el proceso, la tasa de aprendizaje decrece gradualmente, estabilizando las actualizaciones y evitando oscilaciones cerca de los mínimos locales. En contraste, una tasa constante mantiene actualizaciones uniformes a lo largo de todo el entrenamiento, lo que puede requerir más iteraciones para alcanzar

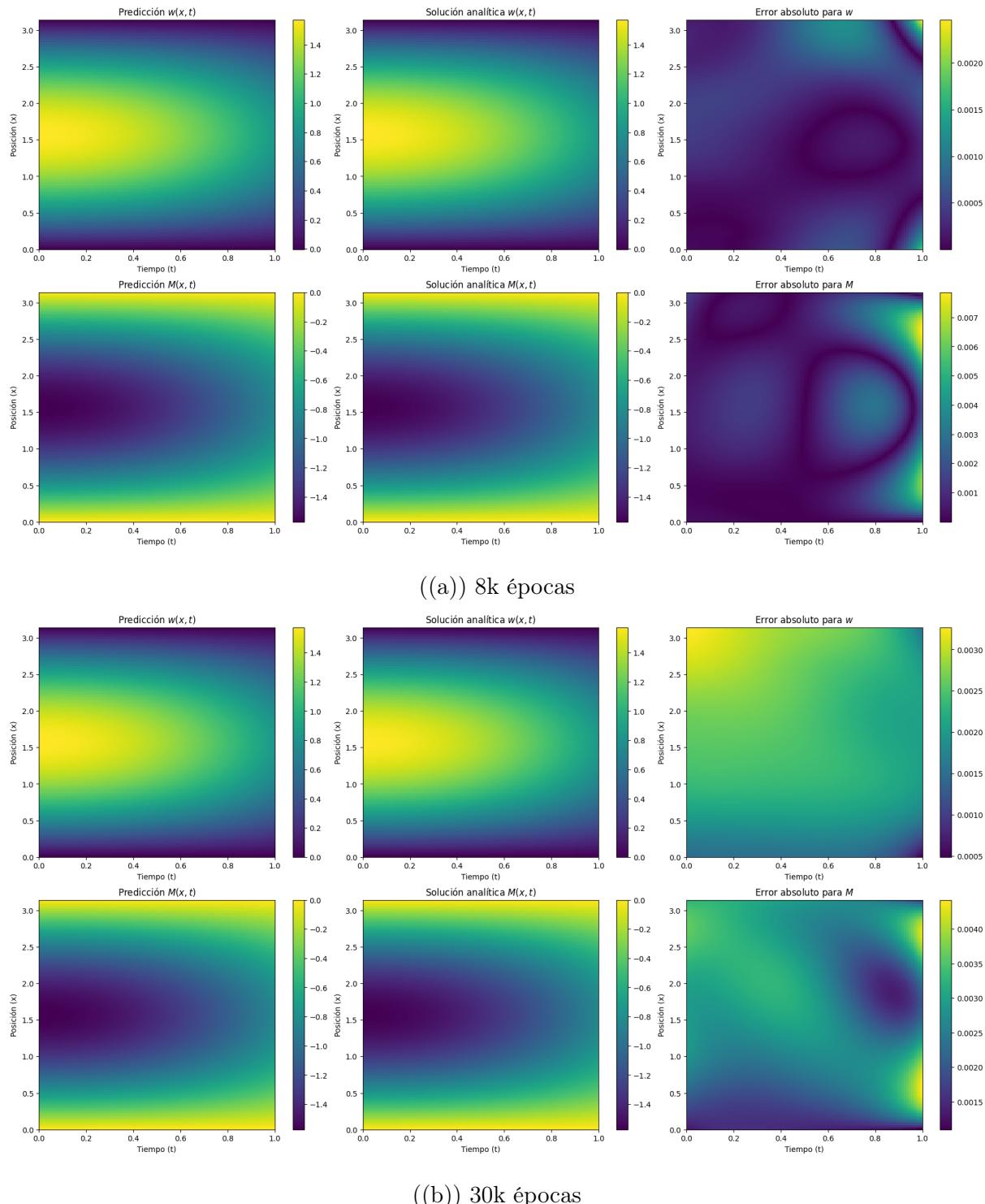


Figura 5.3: Ecuación de Euler-Bernoulli; **Izquierda:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$).

un nivel de pérdida comparable. En consecuencia, una política de aprendizaje demasiado acelerada puede perjudicar la estabilidad del entrenamiento si no se acompaña de una estrategia adaptativa de remuestreo o refinamiento en regiones críticas.

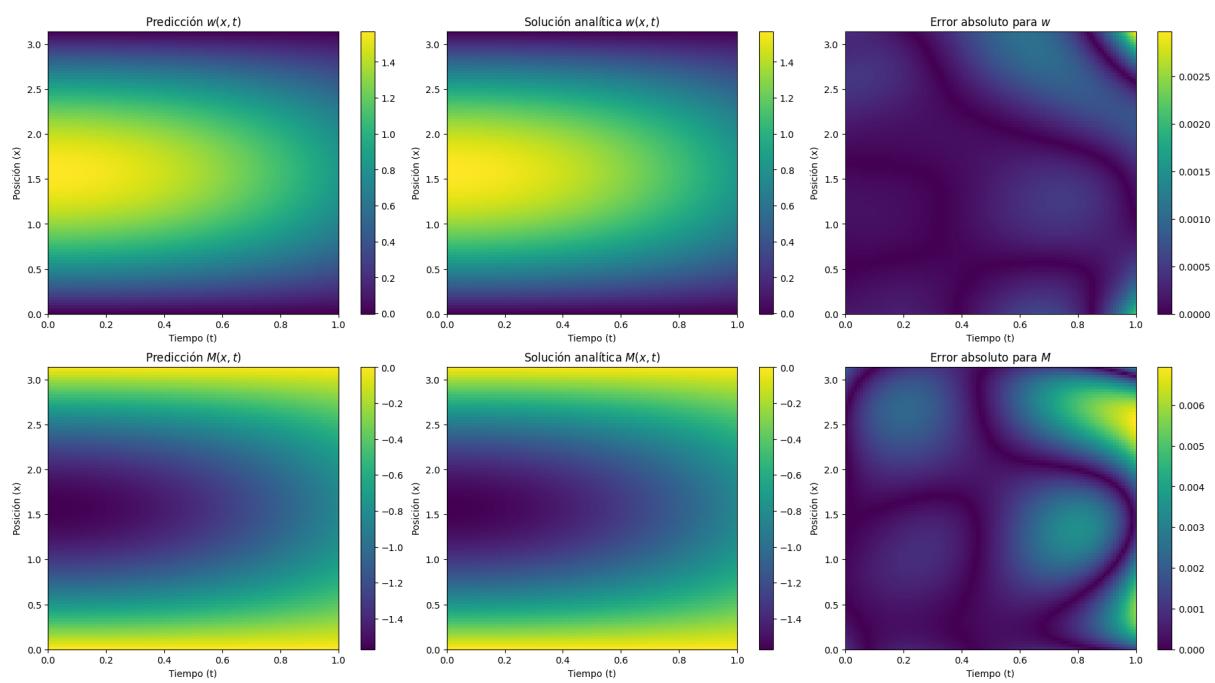


Figura 5.4: Ecuación de Euler-Bernoulli - 8k épocas con decaimiento exponencial; **Izquierda:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$)

En conjunto, los resultados indican que, para este tipo de problemas de dinámica de vigas, una tasa de aprendizaje constante y moderada combinada con un entrenamiento prolongado permite obtener un balance favorable entre precisión, estabilidad y fidelidad física, siendo la segunda configuración la que muestra un mejor compromiso entre ambos campos de predicción (w y M). Estos hallazgos orientan la selección de hiperparámetros en aplicaciones más exigentes o en geometrías extendidas.

5.2. Ecuación de Timoshenko

En este segundo experimento abordamos la resolución numérica de un modelo simplificado de la **dinámica de vigas de Timoshenko**, el cual considera los efectos de flexión y rotación del eje neutro de la viga. El objetivo es verificar la capacidad del método para aproximar soluciones acopladas no triviales que involucran derivadas de segundo orden en el tiempo y el espacio, así como términos de forzamiento externo.

El sistema de ecuaciones que se resuelve numéricamente es el siguiente:

$$\begin{cases} \theta_{tt} - \theta_{xx} + (\theta - w_x) = 0, \\ w_{tt} + \frac{\partial}{\partial x}(\theta - w_x) = g(x, t), \end{cases}$$

donde $\theta(x, t)$ representa la rotación de la sección transversal de la viga, $w(x, t)$ la deflexión transversal, y $g(x, t)$ es una función de forzado definida como:

$$g(x, t) = \cos(t) - \frac{\pi}{2} \sin(x) \cos(t).$$

Este sistema puede derivarse de la teoría de vigas de Timoshenko, que extiende el modelo clásico de Euler-Bernoulli al incorporar la inercia rotacional y la deformación por cortante. En esta implementación se adopta una forma simplificada del modelo, donde se consideran las constantes de rigidez e inercia igual a uno.

Para validar la solución obtenida mediante el método se utiliza una solución analítica exacta dada por:

$$\theta(x, t) = \left(\frac{\pi}{2} \cos(x) + (x - \frac{\pi}{2}) \right) \cos(t), \quad w(x, t) = \frac{\pi}{2} \sin(x) \cos(t).$$

De esta solución se desprenden las condiciones iniciales:

$$\theta(x, 0) = \frac{\pi}{2} \cos(x) + (x - \frac{\pi}{2}), \quad \theta_t(x, 0) = 0, \quad w(x, 0) = \frac{\pi}{2} \sin(x), \quad w_t(x, 0) = 0,$$

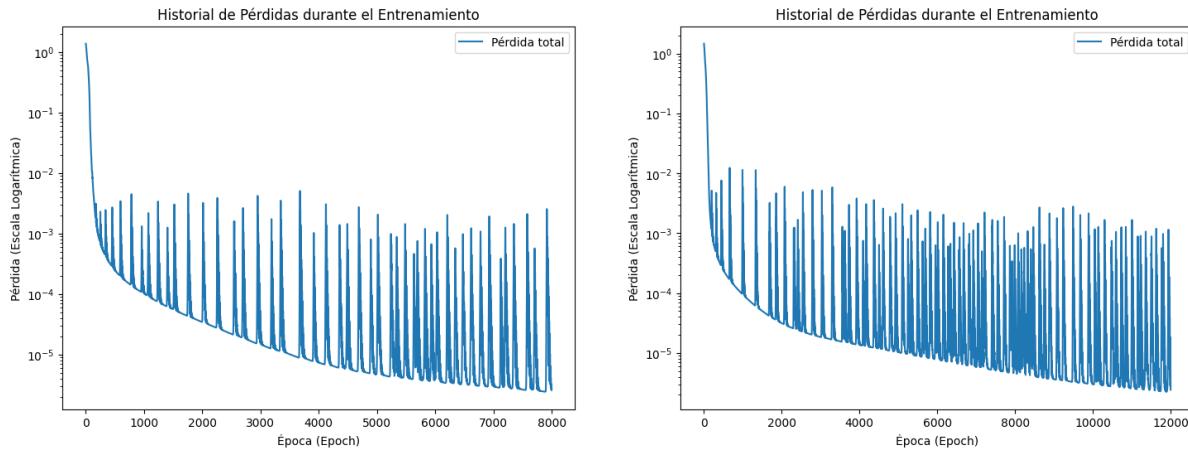
así como las condiciones de frontera homogéneas:

$$\theta(0, t) = \theta(\pi, t) = 0, \quad w(0, t) = w(\pi, t) = 0.$$

La arquitectura utilizada para este experimento tiene los hiperparámetros detallados en la tabla 5-3, utilizando la función de activación `tanh`. La red está diseñada para predecir simultáneamente las dos variables del sistema: θ y w . La función de pérdida total 4.17

Tabla 5-3: Hiperparámetros de la Ecuación de Timoshenko

No. de Puntos	N_i	N_f	N_{int}	Capas	Neuronas	Épocas	Tasa de Apr.
1300	100	200	1000	3	50	8K	1×10^{-3} (constante)
1300	100	200	1000	3	50	12K	1×10^{-3} (constante)



((a)) Pérdida durante 8k épocas - Tiempo de entrenamiento 25 minutos

((b)) Pérdida durante 12k épocas - Tiempo de entrenamiento 37 minutos

Figura 5.5: Comparación de la evolución de la pérdida de entrenamiento en función del número de épocas para la viga de Timoshenko en 1D.

incluye los residuos de las ecuaciones diferenciales del sistema, las condiciones iniciales tanto en valor como en derivada temporal, y las condiciones de frontera.

El modelo fue entrenado durante 8k y 12k épocas con una tasa de aprendizaje de 10^{-3} . La Figura 5.6 presenta las soluciones aproximadas obtenidas para $\theta(x, t)$ y $w(x, t)$, junto con sus errores absolutos con respecto a la solución exacta. Se observa que el error promedio se mantiene por debajo de 10^{-3} en el dominio evaluado, lo que indica una excelente capacidad de aproximación por parte de la red neuronal.

Tabla 5-4: Errores L_2 absolutos y relativos de la validación en ecuación de Timoshenko

Épocas	Variable	Error L_2 Absoluto	Error L_2 Relativo
8K	θ	1.4594×10^{-1}	7.2611×10^{-3}
	w	1.6964×10^{-1}	1.8009×10^{-3}
12K	θ	1.8994×10^{-1}	9.4499×10^{-3}
	w	5.4511×10^{-1}	5.7869×10^{-3}

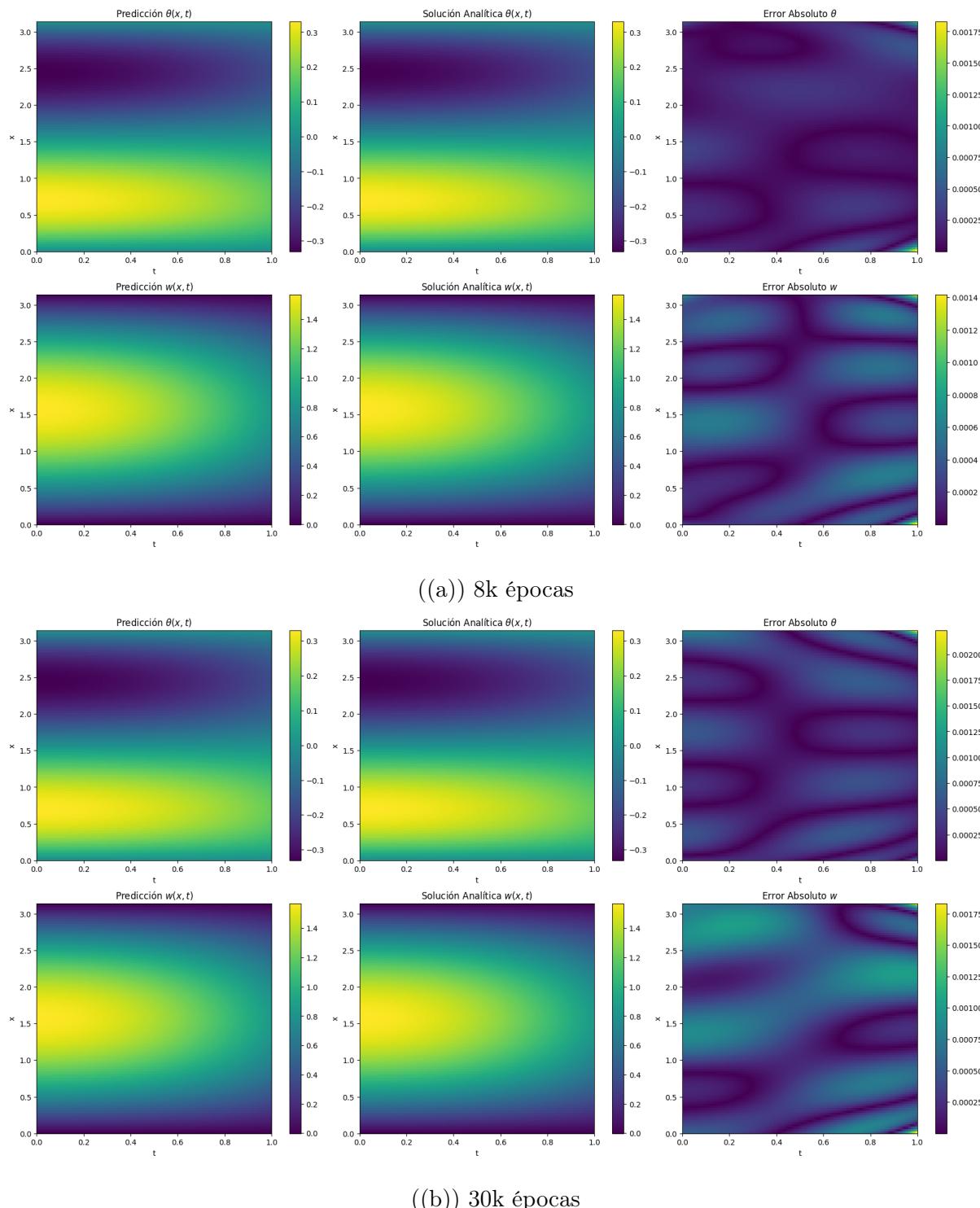


Figura 5.6: Ecuación de Timoshenko; **Izquierda:** Solución aproximada θ^* y w^* por DGM **Centro:** Solución analítica θ y w **Derecha:** Error absoluto en la predicción ($|\theta - \theta^*|$) y ($|w - w^*|$)

Los resultados presentados en la tabla 5-4 permiten comparar el desempeño del modelo DGM al resolver el sistema de Timoshenko bajo diferentes configuraciones de entrenamiento. Para el caso de 8k épocas, se observa que la red neuronal logra una alta

precisión en la aproximación de ambas variables: la rotación θ y la deflexión w , con errores relativos del orden de 10^{-3} . Esto sugiere que la arquitectura propuesta es capaz de capturar eficazmente el acoplamiento entre ambas variables y la dinámica inducida por el forzamiento externo $g(x, t)$.

Sin embargo, al incrementar el número de épocas a 12k, se aprecia una tendencia al aumento del error en ambas variables, especialmente en la predicción de w , cuyo error absoluto y relativo se incrementan considerablemente. Esta degradación del rendimiento podría asociarse a un sobreajuste localizado o a una desestabilización numérica durante las últimas fases del entrenamiento, evidenciando la necesidad de implementar mecanismos de parada temprana o estrategias de remuestreo adaptativo para preservar la generalización del modelo.

Cabe destacar que esta ecuación también ha sido abordada previamente en el trabajo de [39], utilizando el enfoque de redes neuronales informadas por la física (Physics-Informed Neural Networks, PINNs). Esta coincidencia metodológica permite establecer comparaciones relevantes entre distintos esquemas de aprendizaje profundo aplicados a ecuaciones diferenciales parciales, observándose además similitudes en los patrones y magnitudes de los errores absolutos obtenidos.

En general, los resultados indican que el método es capaz de resolver con precisión sistemas acoplados no triviales derivados de modelos físicos como el de Timoshenko. No obstante, la selección cuidadosa de los hiperparámetros y la duración del entrenamiento son factores determinantes para garantizar la estabilidad y la fidelidad de la solución aproximada. Esta evidencia consolida el potencial del enfoque basado en aprendizaje profundo para abordar sistemas hiperbólicos con múltiples variables dependientes y derivadas de segundo orden.

5.3. Dinámica de placas delgadas de Kirchhoff–Love en 2D

En este experimento se aborda la resolución de la ecuación de placas delgadas según la teoría de Kirchhoff–Love, que modela el comportamiento dinámico de una lámina delgada sometida a flexión en dos dimensiones espaciales. Para evitar la evaluación de derivadas de cuarto orden en el espacio, el modelo se reformula como un sistema acoplado de segundo orden que puede ser entrenado de forma eficiente como se mostró en el capítulo anterior

en la ecuación 2.10 y que se muestra a continuación:

$$\begin{cases} w_{tt} + \Delta M = q(x, y, t), \\ M = \Delta w, \end{cases}$$

donde Δ denota el operador de Laplace respecto a las variables espaciales x e y . Esta formulación permite evaluar derivadas de segundo orden únicamente, lo que facilita la implementación automática de derivadas.

Se utiliza como caso de prueba una solución exacta de la forma:

$$w(x, y, t) = \frac{\pi}{2} \sin(x) \sin(y) \cos(t), \quad M(x, y, t) = -\pi \sin(x) \sin(y) \cos(t),$$

lo cual induce una función de forzado dada por:

$$q(x, y, t) = \frac{3\pi}{2} \sin(x) \sin(y) \cos(t).$$

Con esta solución exacta se imponen las siguientes condiciones:

- Condiciones iniciales:

$$w(x, y, 0) = \frac{\pi}{2} \sin(x) \sin(y), \quad w_t(x, y, 0) = 0, \quad M(x, y, 0) = -\pi \sin(x) \sin(y).$$

- Condiciones de frontera homogéneas:

$$w(x, y, t) = 0, \quad M(x, y, t) = 0 \quad \forall x, y \in \partial\Omega.$$

Para este experimento, el dominio espacial es el cuadrado $\Omega = [0, \pi]^2$, y el intervalo temporal considerado es $t \in [0, 1]$. Se implementa el método con hiperparámetros detallados en la tabla 5-5, con función de activación tipo `tanh` minimizando la función de pérdida 4.20. El modelo toma como entrada el triplete (x, y, t) y predice como salida las dos funciones $w(x, y, t)$ y $M(x, y, t)$.

La evolución de la pérdida total a minimizar durante el entrenamiento se muestra en la figura 5.7.

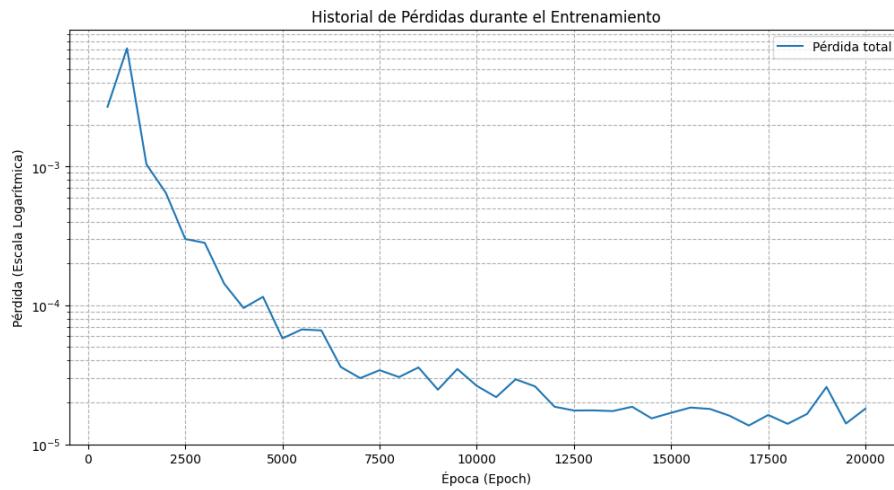


Figura 5.7: Evolución de la pérdida de entrenamiento en función del número de épocas
- Ecuación de Kirchhoff-Love en 2D - Tiempo de entrenamiento 73 minutos

Tabla 5-5: Hiperparámetros para la Placa de Kirchhoff-Love 2D

No. de Puntos	N_i	N_f	N_{int}	Capas	Neuronas	Épocas	Tasa de Apr.
1200	100	100	1000	5	32	20K	1×10^{-3} (constante)
1200	100	100	1000	5	32	20K	1×10^{-2} (d. exponencial)

En la segunda configuración se emplea un decaimiento exponencial de la forma

$$LR = LR_0 \cdot \gamma^{\text{epoch}}, \text{ con } \gamma = 0.9995 \text{ y } LR_{\min} = 1 \times 10^{-5}.$$

Tabla 5-6: Errores L_2 absolutos y relativos de la validación en la ecuación de Kirchhoff-Love usando la primera configuración de la tabla 5-5

Tiempo t	Variable	Error L_2 absoluto	Error L_2 relativo
0.00	w	3.1660×10^{-1}	4.0718×10^{-3}
	M	7.2186×10^{-1}	4.6419×10^{-3}
0.25	w	3.1614×10^{-1}	4.1963×10^{-3}
	M	4.4893×10^{-1}	2.9795×10^{-3}
0.50	w	3.2168×10^{-1}	4.7143×10^{-3}
	M	4.3251×10^{-1}	3.1692×10^{-3}
0.75	w	3.4130×10^{-1}	5.9991×10^{-3}
	M	7.4879×10^{-1}	6.5808×10^{-3}
1.00	w	3.7348×10^{-1}	8.8901×10^{-3}
	M	$1.3905 \times 10^{+0}$	1.6549×10^{-2}

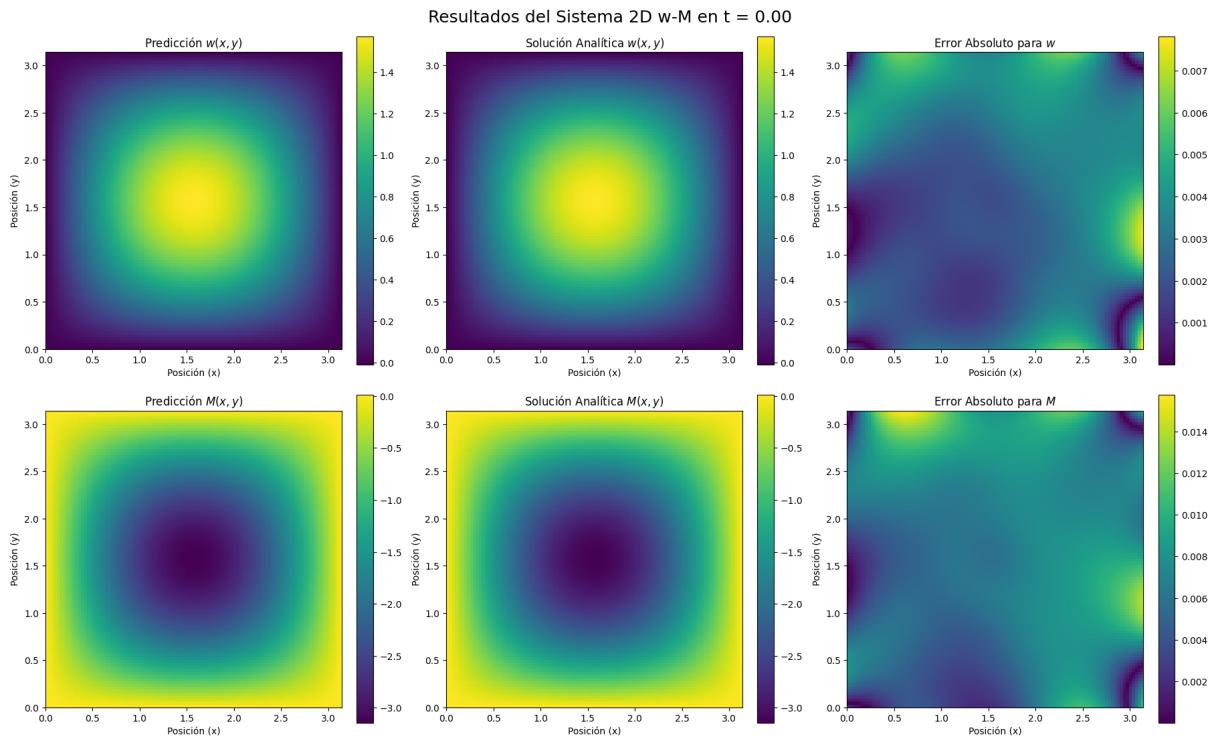


Figura 5.8: Ecuación de Kirchhoff-Love en $t = 0.0$; **Izquierda:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$)

Tabla 5-7: Errores L_2 absolutos y relativos de la validación en la ecuación de Kirchhoff-Love usando la segunda configuración de la tabla 5-5

Tiempo t	Variable	Error L_2 absoluto	Error L_2 relativo
0.00	w	6.0838×10^{-2}	7.8244×10^{-4}
	M	1.1554×10^{-1}	7.7295×10^{-4}
0.25	w	7.6378×10^{-2}	1.0138×10^{-3}
	M	3.1015×10^{-1}	2.0584×10^{-3}
0.50	w	9.4767×10^{-2}	1.3888×10^{-3}
	M	2.7509×10^{-1}	2.0157×10^{-3}
0.75	w	1.0059×10^{-1}	1.7681×10^{-3}
	M	4.0627×10^{-1}	3.5705×10^{-3}
1.00	w	3.3012×10^{-1}	7.8579×10^{-3}
	M	$1.1397 \times 10^{+0}$	1.3564×10^{-2}

Los resultados presentados en las tablas de validación confirman que el método, aplicado a

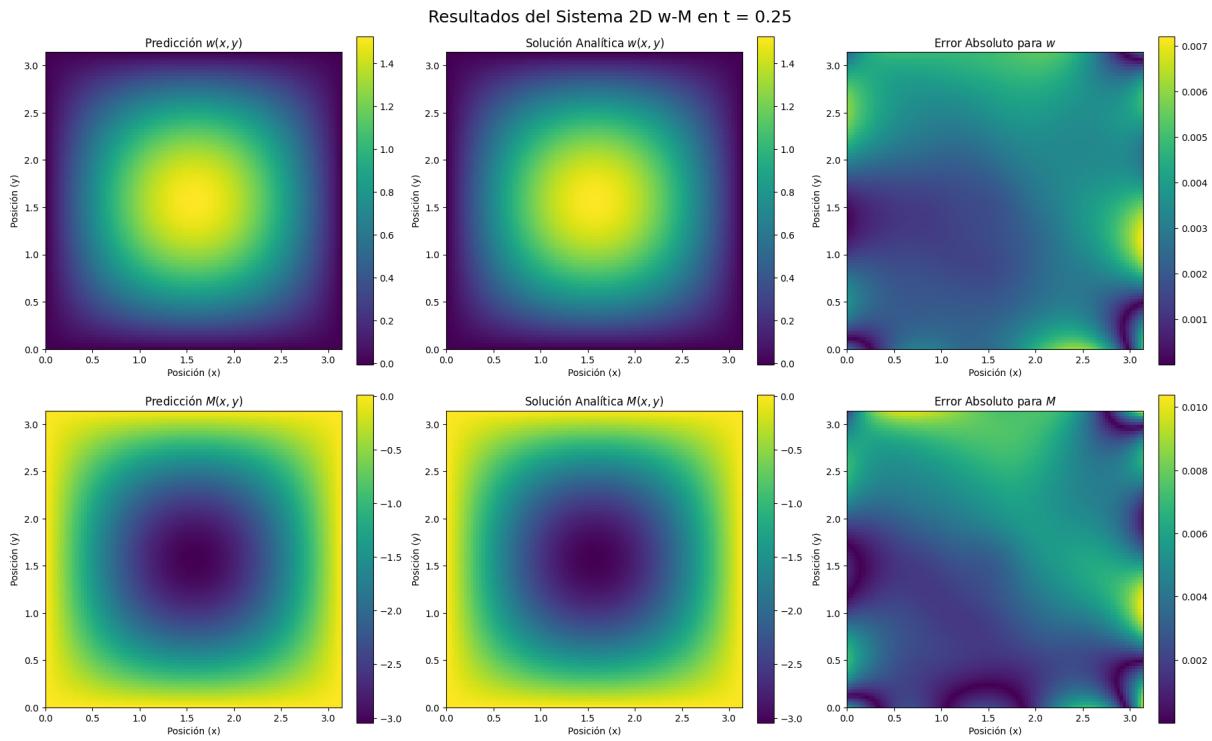


Figura 5.9: Ecuación de Kirchhoff-Love en $t = 0.25$; **Izquierda:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$)

la formulación acoplada de la ecuación de Kirchhoff–Love, logra una aproximación precisa de las variables físicas $w(x, y, t)$ y $M(x, y, t)$ a lo largo del dominio espacio-temporal considerado. En la primera configuración de hiperparámetros, con tasa de aprendizaje constante, se alcanza un error relativo promedio inferior a 10^{-2} , aunque se evidencia un incremento gradual en el error a medida que avanza el tiempo, especialmente en la predicción de M . Esta tendencia sugiere que, si bien la red logra una buena aproximación inicial, el refinamiento temporal podría beneficiarse de una estrategia de entrenamiento más adaptativa.

Por su parte, la segunda configuración, basada en un decaimiento exponencial de la tasa de aprendizaje, muestra una mejora significativa en los errores absolutos y relativos durante las etapas iniciales y medias del intervalo de tiempo. Particularmente, en $t = 0.00$ y $t = 0.25$, los errores relativos para ambas variables se reducen por debajo de 10^{-3} , destacando una mayor capacidad de generalización y fidelidad en la predicción. No obstante, al llegar a $t = 1.0$, el error en la predicción de M aumenta, aunque permanece dentro de límites aceptables para aplicaciones numéricas exigentes.

En conjunto, este experimento demuestra que el modelo DGM, entrenado sobre un sistema

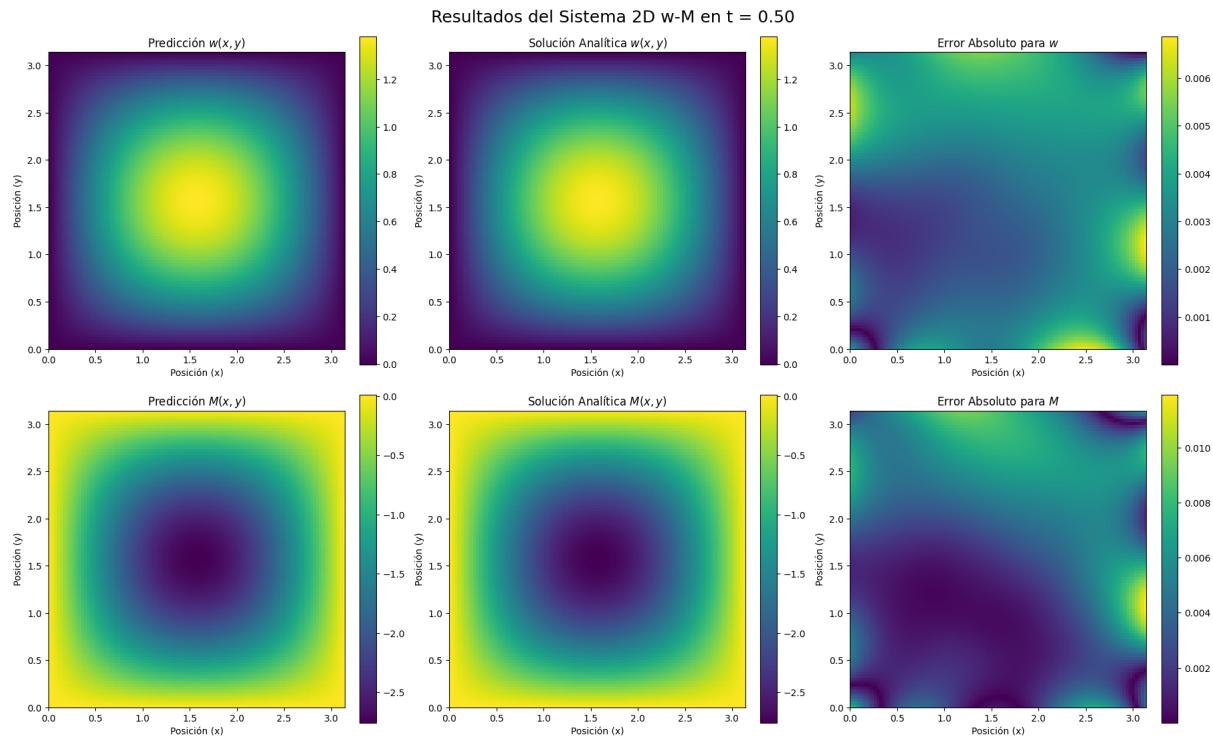


Figura 5.10: Ecuación de Kirchhoff–Love en $t = 0.50$; **Izquierda:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$)

acoplado de segundo orden, es eficaz para resolver problemas dinámicos bidimensionales con estructuras complejas como la ecuación de placas delgadas de Kirchhoff–Love. La formulación acoplada no solo permite una implementación computacional más estable y eficiente, sino que también preserva la exactitud de la solución en tiempo y espacio. Estos hallazgos respaldan la aplicabilidad del enfoque en contextos físicos realistas donde se requiere resolver ecuaciones de orden superior sin recurrir a derivadas explícitas de alto orden.

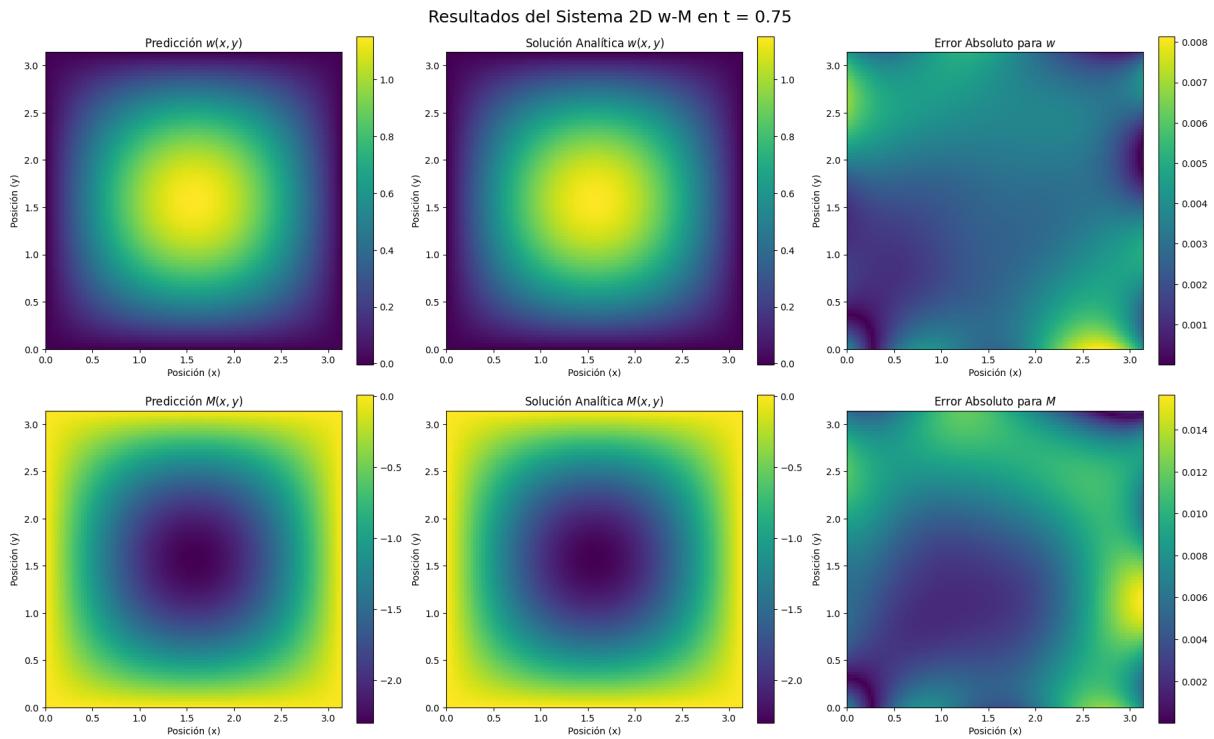


Figura 5.11: Ecuación de Kirchhoff-Love en $t = 0.75$; **Izquierda:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$)

5.4. Ecuación de Burgers viscosa en 2D

En este cuarto experimento se aborda la resolución de la ecuación de Burgers viscosa en dos dimensiones espaciales. Este modelo no lineal combina los efectos de advección y difusión, y es utilizado frecuentemente como una simplificación de las ecuaciones de Navier–Stokes para estudiar fenómenos de dinámica de fluidos.

La forma vectorial de la ecuación de Burgers viscosa 2D se expresa como:

$$\begin{cases} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right), \end{cases} \quad (5.1)$$

donde $u(x, y, t)$ y $v(x, y, t)$ son las componentes del campo de velocidad en las direcciones x e y , respectivamente, y ν es el coeficiente de viscosidad.

Para efectos de validación, se emplea la solución exacta dada en la ecuación 2.13 la cual representa un flujo con simetría radial en torno al punto $(0.5, 0.5)$. Esta solución se utiliza para definir las condiciones de frontera e iniciales del experimento utilizando coeficiente

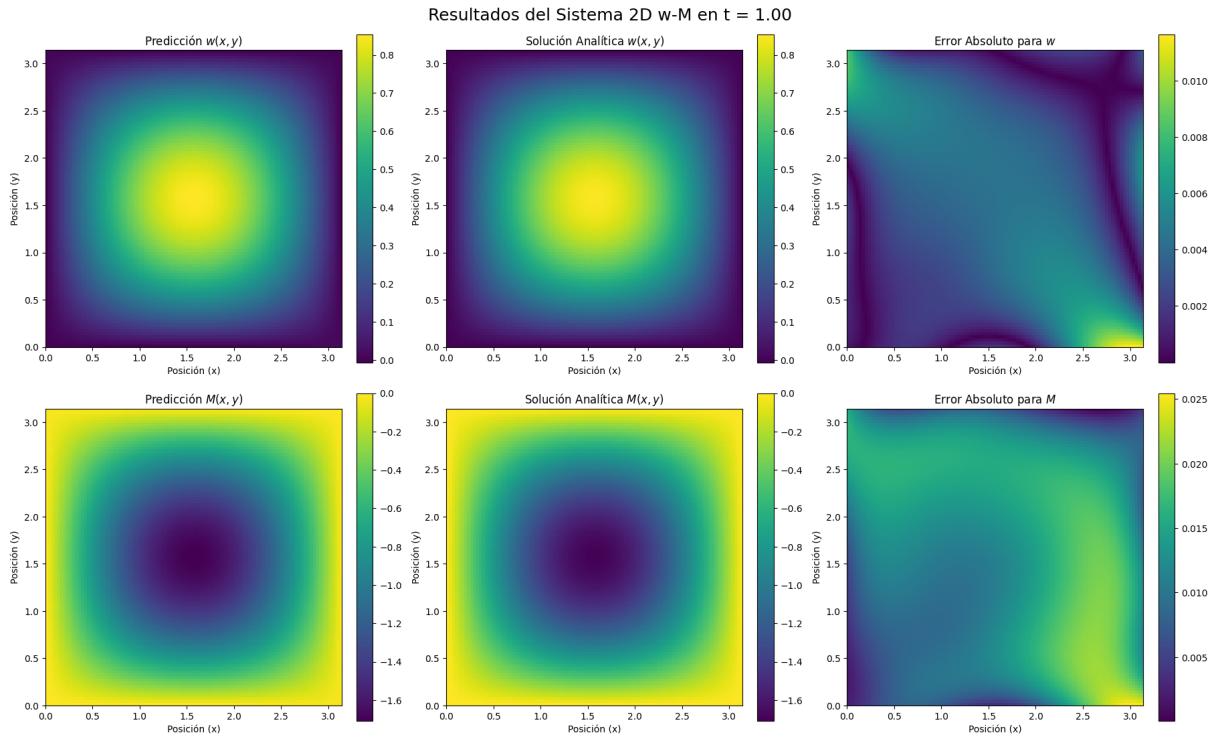


Figura 5.12: Ecuación de Kirchhoff-Love en $t = 1.0$; **Izquierda:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$)

de viscosidad $\nu = 1$.

Para este experimento, el dominio espacial es $\Omega = [0, 1]^2$, y el intervalo temporal considerado es $t \in [0, 1]$. Se implementa una red neuronal con el algoritmo 2 con hiperparámetros detallados en la tabla 5-8, con función de activación tipo SiLu minimizando la función de pérdida 4.23. El modelo toma como entrada el triplete (x, y, t) y predice como salida las dos funciones $u(x, y, t)$ y $v(x, y, t)$.

Tabla 5-8: Hiperparámetros para la Ecuación de Burgers 2D

No. de Puntos	N_i	N_f	N_{int}	Capas	Neuronas	Épocas	Tasa de Apr.
5500	250	250	5000	4	32	10K	1×10^{-2} (d. exponencial)

Nota: Se utiliza un tamaño de lote = 128

La evolución de la pérdida total a minimizar durante el entrenamiento se muestra en la figura 5.13.

Los resultados presentados en la tabla 5-9 y la figura de evolución del error permiten evaluar el desempeño del modelo DGM en la resolución de la ecuación de Burgers viscosa

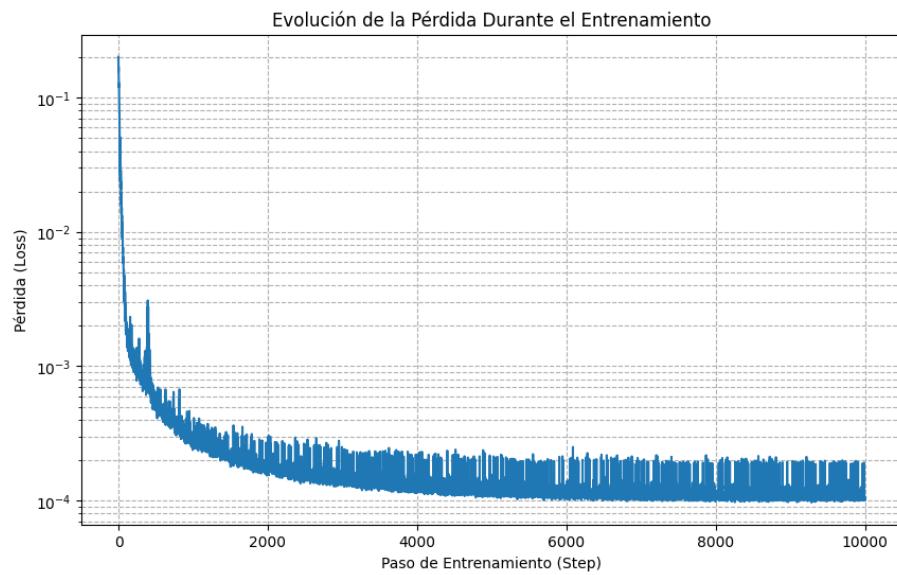


Figura 5.13: Evolución de la pérdida de entrenamiento en función del número de épocas
- Ecuación de Burgers viscosa en 2D - Tiempo de entrenamiento 11 minutos

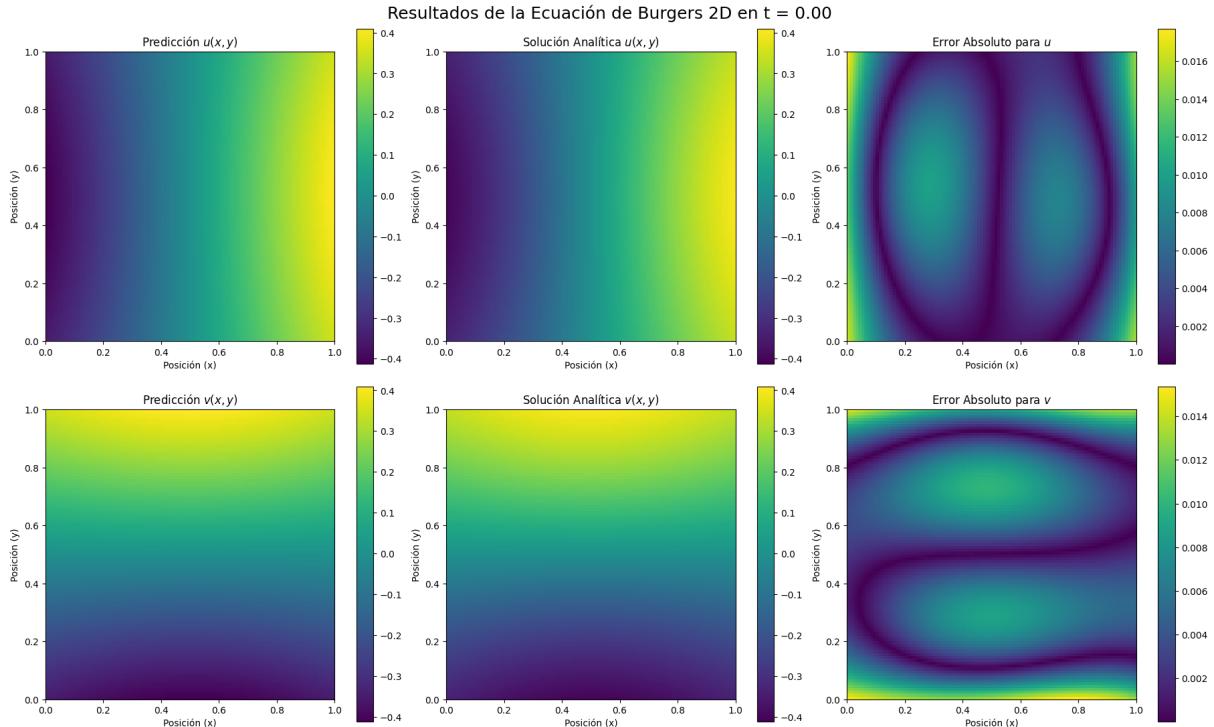


Figura 5.14: Ecuación de Burgers en $t = 0.00$; **Izquierda:** Solución aproximada u^* y v^* por DGM **Centro:** Solución analítica u y v **Derecha:** Error absoluto en la predicción ($|u - u^*|$) y ($|v - v^*|$)

en dos dimensiones. Se observa que el error relativo L_2 para ambas variables, $u(x, y, t)$ y $v(x, y, t)$, se mantiene en rangos aceptables a lo largo del dominio temporal evaluado $t \in [0, 1]$, alcanzando valores mínimos en el entorno de $t \approx 0.75$.

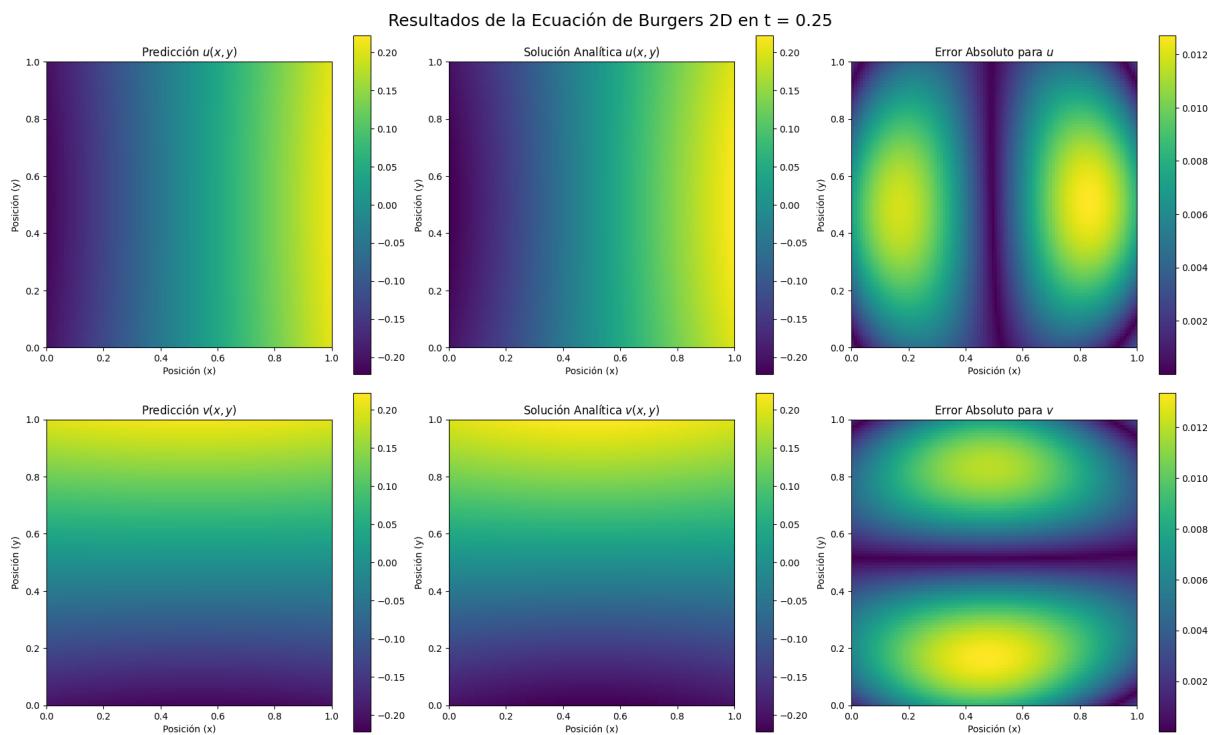


Figura 5.15: Ecuación de Burgers en $t = 0.25$; **Izquierda:** Solución aproximada u^* y v^* por DGM **Centro:** Solución analítica u y v **Derecha:** Error absoluto en la predicción ($|u - u^*|$) y ($|v - v^*|$)

Tabla 5-9: Errores L_2 absolutos y relativos de la validación en la ecuación de Burgers viscosa bidimensional

Tiempo t	Variable	Error L_2 absoluto	Error L_2 relativo
0.00	u	5.7517×10^{-1}	2.4178×10^{-2}
	v	5.7687×10^{-1}	2.4250×10^{-2}
0.25	u	7.3142×10^{-1}	5.5970×10^{-2}
	v	7.6749×10^{-1}	5.8730×10^{-2}
0.50	u	2.5411×10^{-1}	2.8178×10^{-2}
	v	2.3678×10^{-1}	2.6256×10^{-2}
0.75	u	1.0821×10^{-1}	1.5715×10^{-2}
	v	1.0495×10^{-1}	1.5242×10^{-2}
1.00	u	1.0108×10^{-1}	1.8149×10^{-2}
	v	$1.2071 \times 10^{+0}$	2.1673×10^{-2}

En particular, la tabla evidencia que el mayor error relativo ocurre cerca de $t = 0.25$, donde se alcanzan valores del orden de 5.6×10^{-2} para u y 5.8×10^{-2} para v . Esta etapa

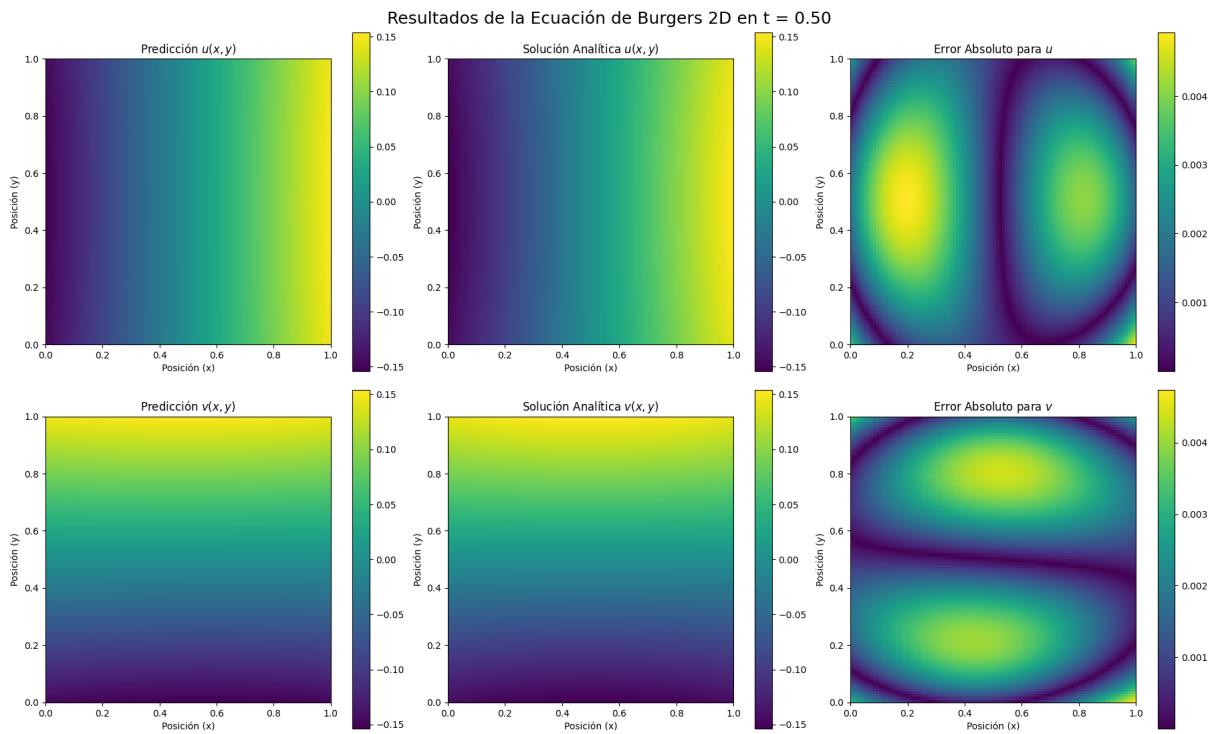


Figura 5.16: Ecuación de Burgers en $t = 0.50$; **Izquierda:** Solución aproximada u^* y v^* por DGM **Centro:** Solución analítica u y v **Derecha:** Error absoluto en la predicción ($|u - u^*|$) y ($|v - v^*|$)

temprana del sistema coincide con la presencia de mayores gradientes y dinámica más intensa en la solución analítica, lo que exige mayor capacidad de generalización por parte del modelo. Conforme avanza el tiempo, el error disminuye progresivamente, mostrando que el modelo logra capturar adecuadamente el comportamiento difusivo dominante en el régimen tardío.

La figura 5.19 ilustra con claridad esta evolución del error relativo. La curva azul sólida representa el error relativo L_2 para $u(x,y,t)$, mientras que la curva roja punteada corresponde a $v(x,y,t)$. Ambas presentan una estructura de campana con un pico en torno a $t = 0.18$, seguido de una fase de decrecimiento hasta alcanzar un mínimo en $t \approx 0.8$, tras el cual el error vuelve a incrementarse ligeramente. Este patrón refleja con fidelidad la dinámica esperada de la ecuación, en la que la solución se torna más suave en el tiempo, facilitando la tarea de aproximación para el modelo neuronal.

A partir del análisis de los resultados preliminares mostrados anteriormente, se llevó a cabo una segunda fase experimental con el objetivo de mejorar la precisión del modelo DGM ante la resolución de la ecuación de Burgers viscosa en 2D, especialmente en los instantes tempranos donde se presentan mayores gradientes y variabilidad en la solución.

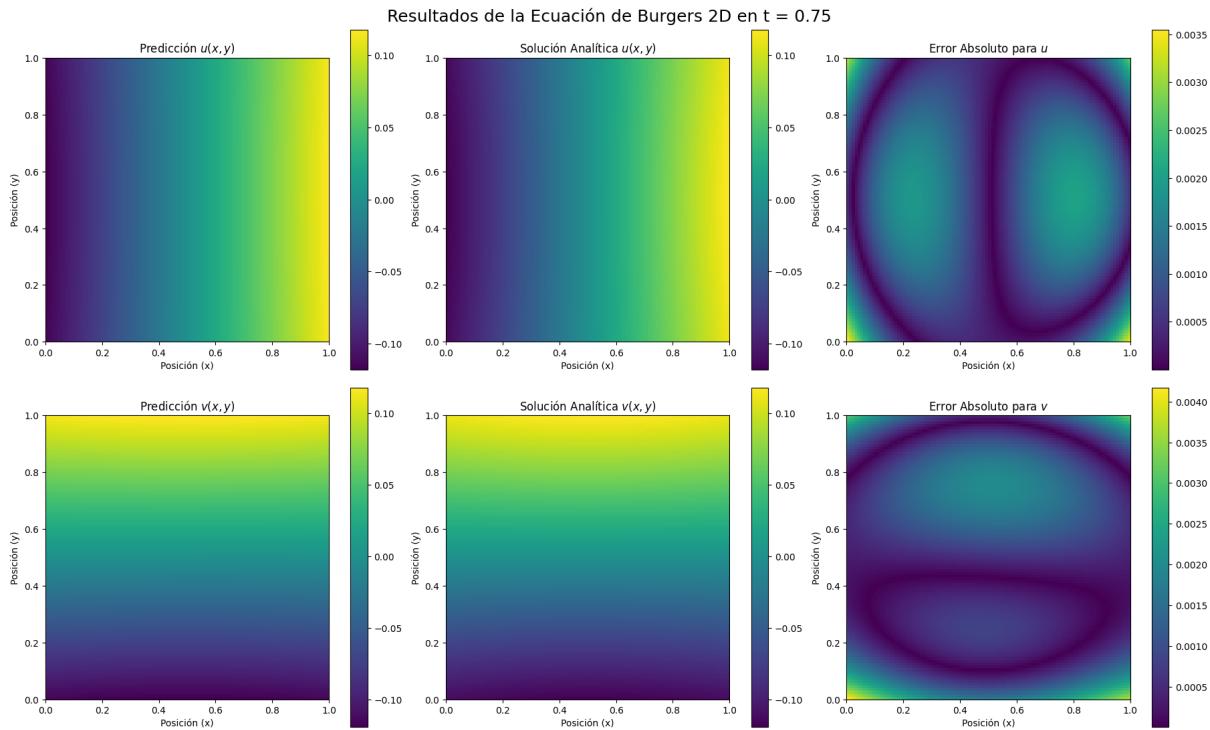


Figura 5.17: Ecuación de Burgers en $t = 0.75$; **Izquierda:** Solución aproximada u^* y v^* por DGM **Centro:** Solución analítica u y v **Derecha:** Error absoluto en la predicción ($|u - u^*|$) y ($|v - v^*|$)

Para ello, se realizó un afinamiento de hiperparámetros, cuyos cambios principales se enumeran a continuación:

- Se incrementó la penalización del término de frontera con un valor $\lambda_B = 100.0$ en la pérdida total, buscando forzar mayor adherencia del modelo a las condiciones de contorno.
- Se aplicó una política de muestreo ponderado en el dominio espacio-temporal, dándole un peso diez veces mayor a los puntos ubicados en la región crítica de tiempo temprano $t \in [0, 0.12]$, donde el error relativo era mayor.
- Se mantuvo la arquitectura de la red con 4 capas ocultas con 32 neuronas, se cambió a activación `tanh`, tasa de aprendizaje inicial 1×10^{-2} , pero con decaimiento exponencial controlado por el factor $\gamma = 0.999113$, y un límite inferior de 1×10^{-7}
- Se inicializaron los pesos con el método de Xavier, lo cual mejora la propagación del gradiente en redes profundas.

Con esta configuración, el entrenamiento se desarrolló a lo largo de un total de 10 000 pasos de optimización, organizados en 10 fases de 1 000 pasos cada una. Al finalizar cada fase, la tasa de aprendizaje fue actualizada conforme a una política de decaimiento exponencial,

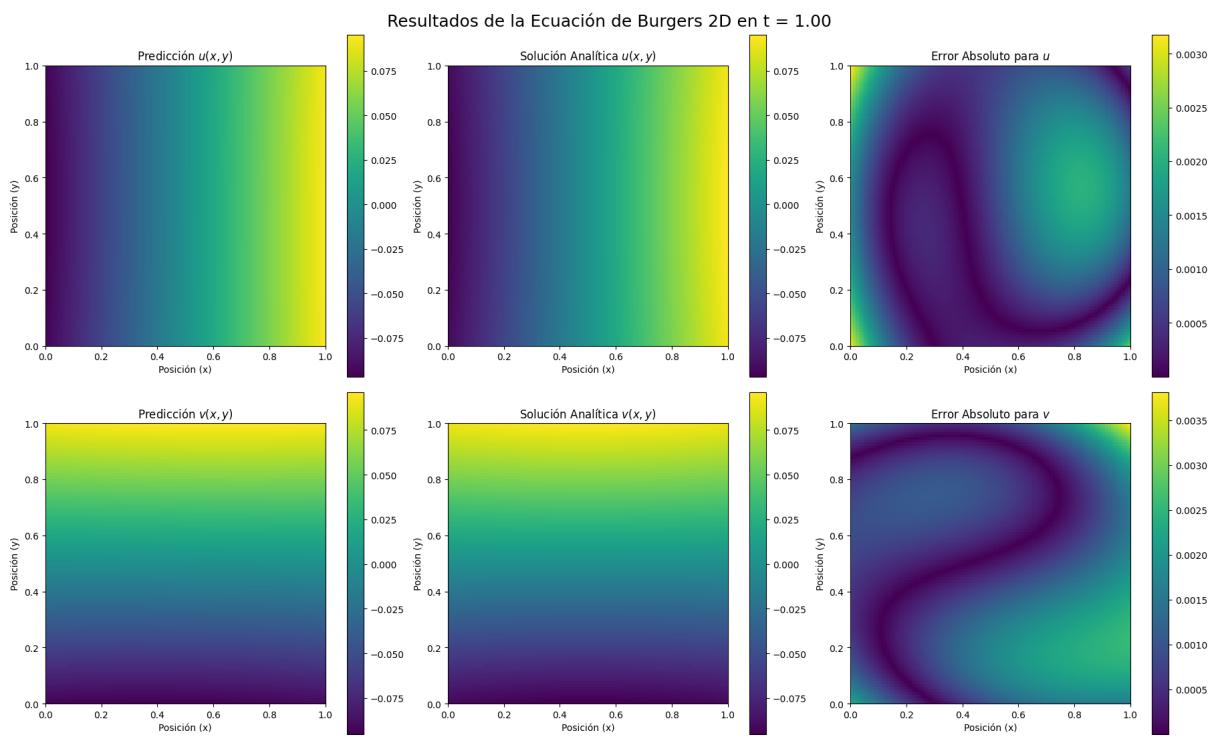


Figura 5.18: Ecuación de Burgers en $t = 1.00$; **Izquierda:** Solución aproximada u^* y v^* por DGM **Centro:** Solución analítica u y v **Derecha:** Error absoluto en la predicción ($|u - u^*|$) y ($|v - v^*|$)

permitiendo reducir progresivamente la magnitud de los ajustes en los parámetros. Este esquema por fases favorece una convergencia más estable y una mejor exploración inicial del espacio de soluciones.

Los resultados cuantitativos se resumen en la tabla 5-10, en la cual se presentan los errores L_2 absoluto y relativo en cinco instantes representativos, así como en las gráficas de las figuras (5.21 a 5.25).

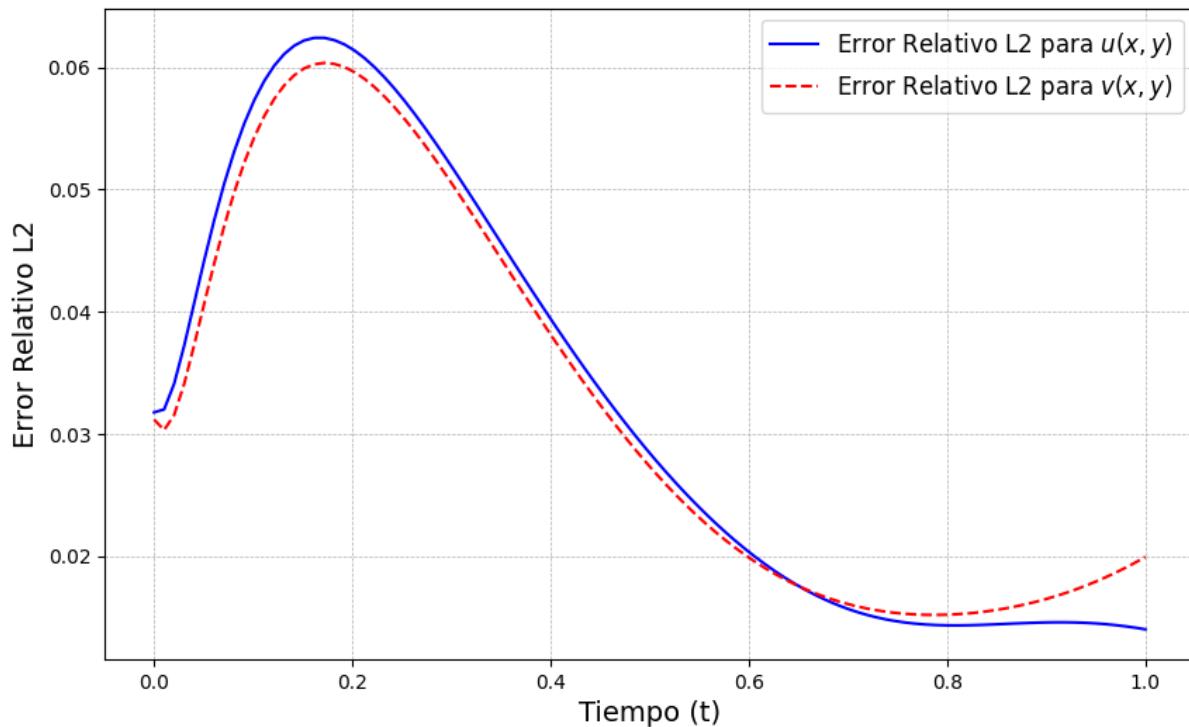


Figura 5.19: Ecuación de Burgers viscosa; evolución del error relativo L_2 en el tiempo $[0, 1]$

Tabla 5-10: Errores L_2 absolutos y relativos de la validación en la ecuación de Burgers viscosa bidimensional - Afinamiento de hiperparámetros

Tiempo t	Variable	Error L_2 absoluto	Error L_2 relativo
0.00	u	1.4431×10^{-1}	6.0663×10^{-3}
	v	1.0647×10^{-1}	4.4758×10^{-3}
0.25	u	4.6355×10^{-1}	3.5472×10^{-2}
	v	4.6858×10^{-1}	3.5857×10^{-2}
0.50	u	2.2824×10^{-1}	2.5309×10^{-2}
	v	2.3018×10^{-1}	2.5524×10^{-2}
0.75	u	1.4541×10^{-1}	2.1117×10^{-2}
	v	1.5015×10^{-1}	2.1805×10^{-2}
1.00	u	8.8164×10^{-2}	1.5829×10^{-2}
	v	8.6007×10^{-2}	1.5442×10^{-2}

Estos resultados permiten concluir que el afinamiento implementado mediante:

- penalización reforzada en las condiciones de frontera,

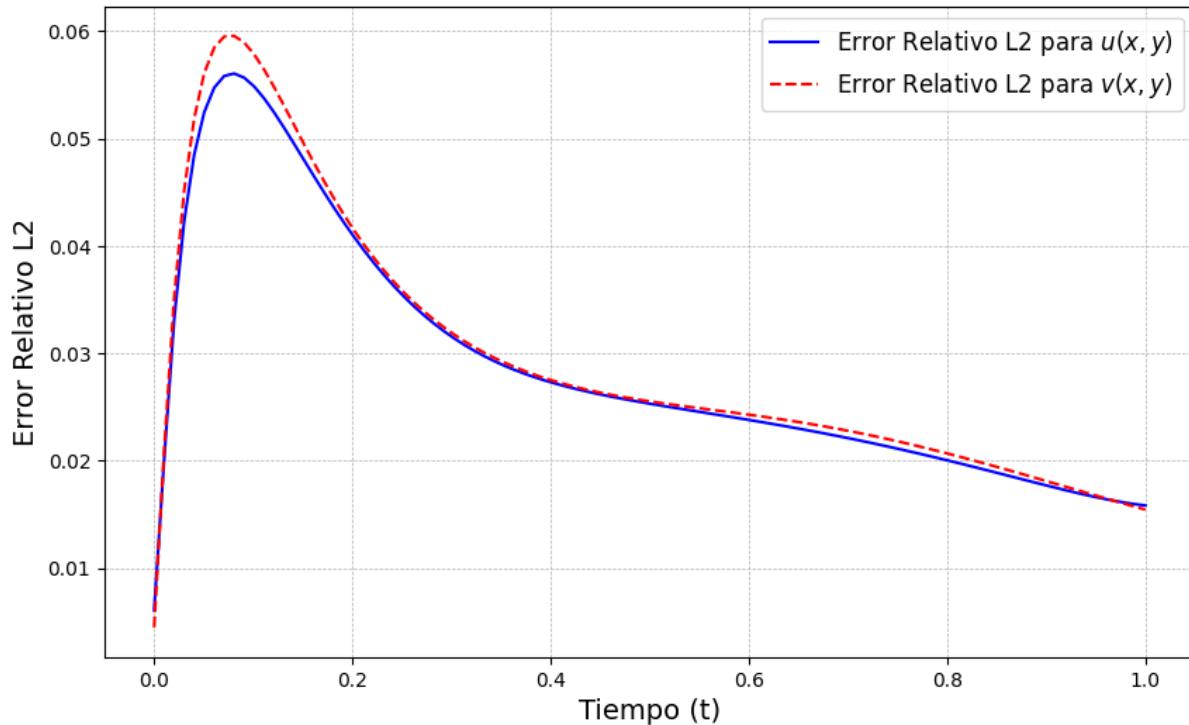


Figura 5.20: Ecuación de Burgers viscosa; evolución del error relativo L_2 en el tiempo $[0, 1]$

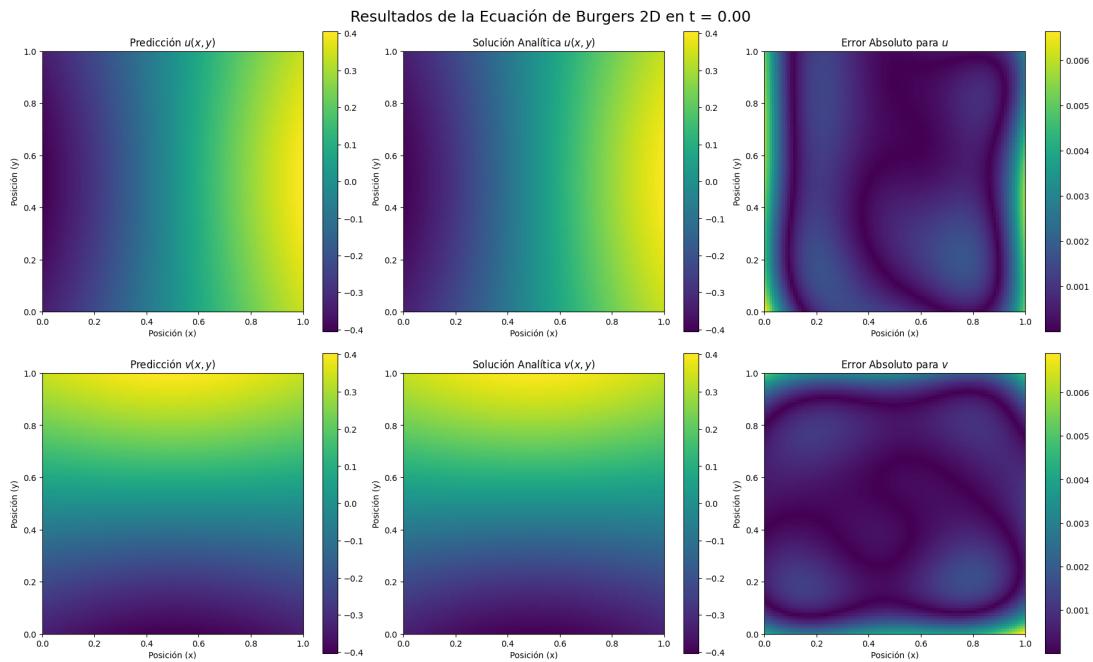


Figura 5.21: Ecuación de Burgers viscosa en $t = 0.0$; **Izquierda:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$)

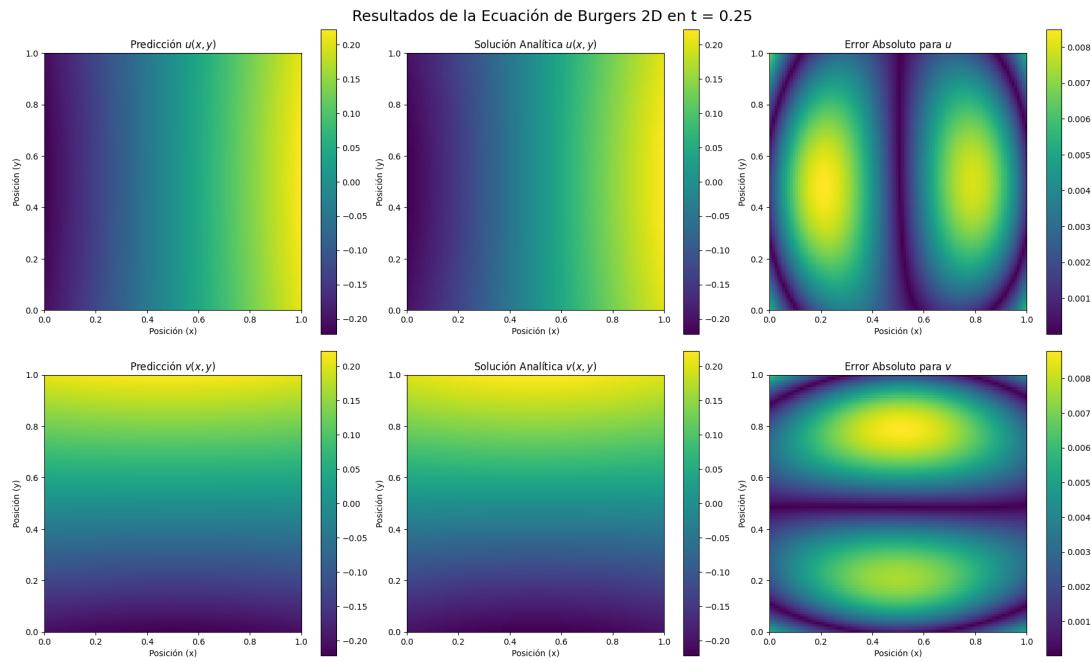


Figura 5.22: Ecuación de Burgers viscosa en $t = 0.25$; **Izquierda:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$)

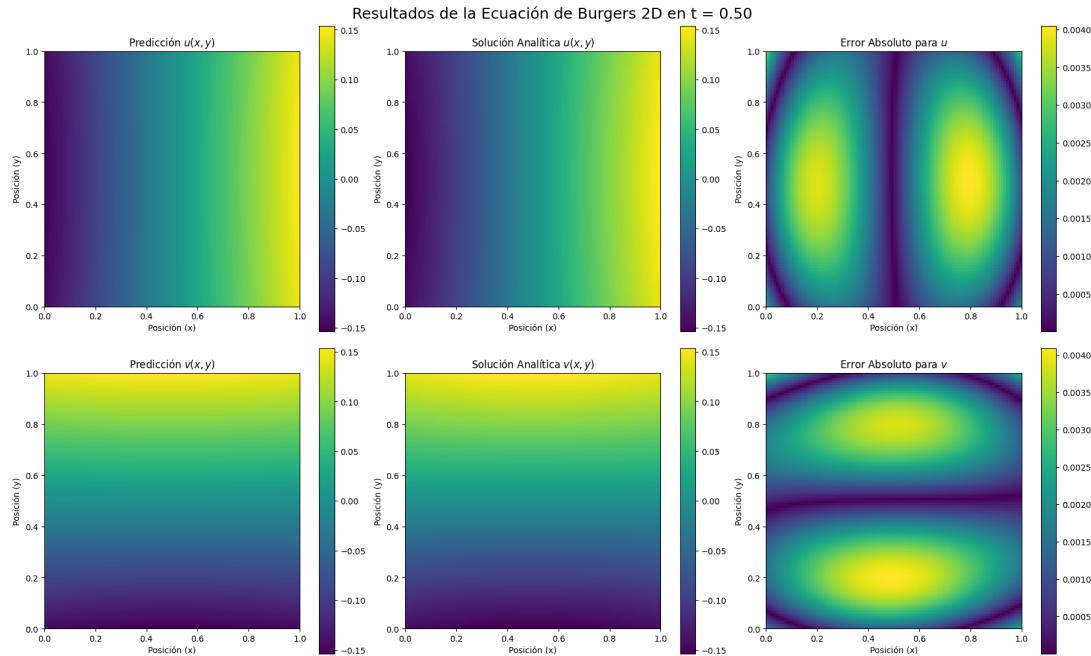


Figura 5.23: Ecuación de Burgers viscosa en $t = 0.50$; **Izquierda:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$)

- reponderación temporal de los datos de entrenamiento,
- y una tasa de aprendizaje con decaimiento suave,

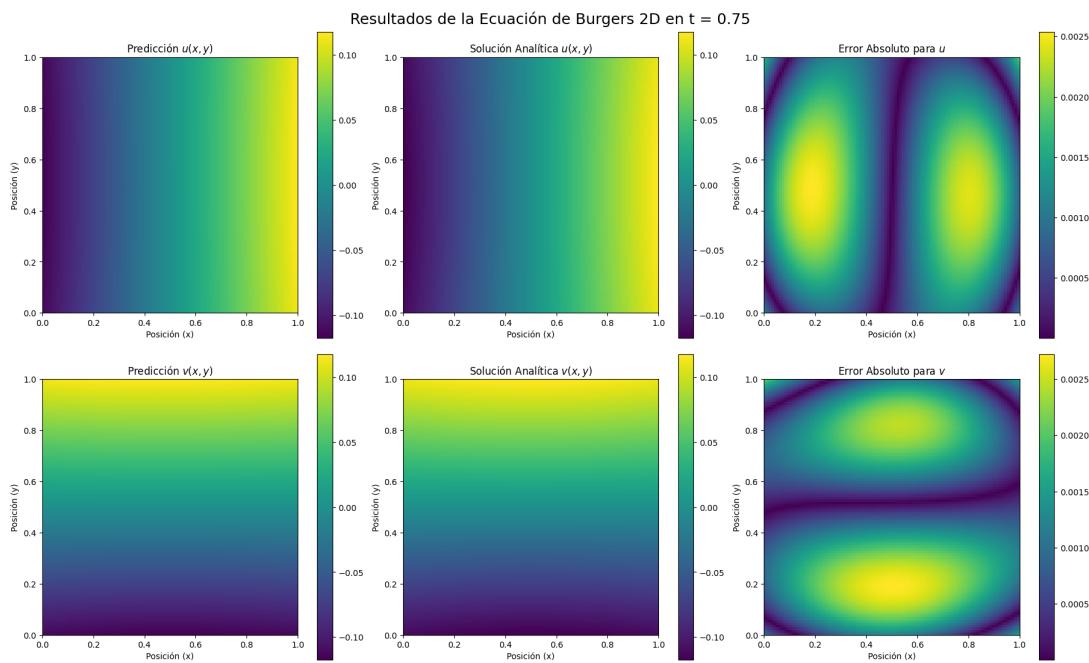


Figura 5.24: Ecuación de Burgers viscosa en $t = 0.75$; **Izquierdo:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$)

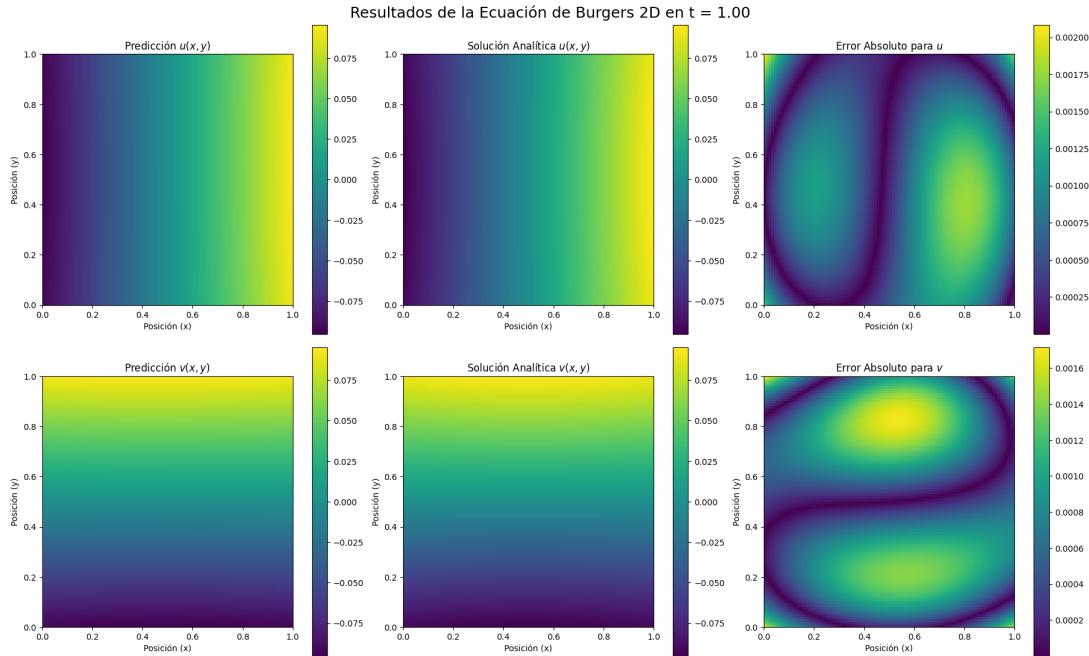


Figura 5.25: Ecuación de Burgers viscosa en $t = 1.00$; **Izquierdo:** Solución aproximada w^* y M^* por DGM **Centro:** Solución analítica w y M **Derecha:** Error absoluto en la predicción ($|w - w^*|$) y ($|M - M^*|$)

mejoró sensiblemente la capacidad de generalización del modelo especialmente en la etapa crítica del sistema.

En conjunto, estos resultados demuestran que el enfoque DGM es adecuado para resolver ecuaciones de Burgers en dos dimensiones, ofreciendo una aproximación confiable incluso en presencia de términos no lineales y estructuras convectivo-difusivas. La red neuronal logra capturar tanto la evolución temporal como los efectos de simetría y suavizado en el flujo, con un error relativo global por debajo del 6.0 % como se observa en la figura 5.20, validando así su aplicabilidad a modelos de dinámica de fluidos simplificados.

5.5. Ecuación de Schrödinger

En esta sección se presenta la resolución de la ecuación de Schrödinger dependiente del tiempo desde tres enfoques complementarios, con el objetivo de validar la capacidad del método para aproximar soluciones complejas en dominios espacio-temporales de distintas dimensiones y niveles de complejidad física:

1. El primer caso considera la evolución de una función de onda unidimensional en ausencia de potencial externo, es decir, con $V(x) = 0$. Este escenario permite analizar la propagación libre de una onda cuántica en una dimensión, sirviendo como caso base con solución exacta conocida.
2. El segundo enfoque aborda el problema tridimensional del átomo de hidrógeno con protón y electrón móviles. Se trata de un sistema de dos cuerpos en interacción coulombiana, donde se evalúa la capacidad del método para aproximar la función de onda conjunta $\psi(\mathbf{r}_e, \mathbf{r}_p, t)$ en un dominio de seis dimensiones espaciales más el tiempo.
3. Finalmente, se analiza el caso del átomo de hidrógeno con protón fijo en el origen y electrón móvil. Esta simplificación reduce el problema a tres dimensiones espaciales, manteniendo la complejidad del potencial singular $\frac{1}{|\mathbf{r}|}$, y permite comparar los resultados con soluciones analíticas conocidas.

Estos tres casos constituyen una progresión natural de dificultad computacional y sirven como banco de pruebas para evaluar la estabilidad, precisión y generalización del método en mecánica cuántica.

5.5.1. Caso unidimensional con potencial cero

Se aborda la resolución de la ecuación de Schrödinger unidimensional sin potencial externo, utilizando la formulación adimensional presentada en la ecuación (4.25). El dominio considerado es el cuadrado unidad $(x, t) \in [0, 1]^2$, y se emplea como referencia la solución exacta (4.26), que permite generar condiciones iniciales y de frontera exactas.

Para aproximar la solución, se utilizó una función activación SiLU, cuya salida corresponde a la parte real $u(x, t)$ y parte imaginaria $v(x, t)$ de la función de onda $\psi(x, t)$. La red se inicializa en base al Algoritmo 2 en donde se emplea entrenamiento por fases, con

mini-lotes de puntos interiores para mejorar la eficiencia computacional. La tasa de aprendizaje sigue un esquema de decaimiento exponencial definido como

$$LR = LR_0 \cdot \gamma^{\text{fase}}, \quad \text{con } \gamma = 0.9, \quad LR_{\min} = 10^{-5}.$$

Tabla 5-11: Hiperparámetros: Ecuación de Schrödinger 1D

N_{total}	N_i	N_f	N_{int}	Capas	Neuronas	Fases	Mini-lote	Tasa de Apr.
32000	1000	1000	30000	3	32	30	32	10^{-3} (d. exponencial)
50000	10000	10000	30000	3	32	20	64	10^{-3} (d. exponencial)

La función de pérdida total (4.28) considera el sistema acoplado (4.27), las condiciones iniciales y de frontera obtenidas a partir de la solución exacta. Durante el entrenamiento, se evalúa el progreso a través del error cuadrático medio total y se monitorea la contribución de cada término de la pérdida.

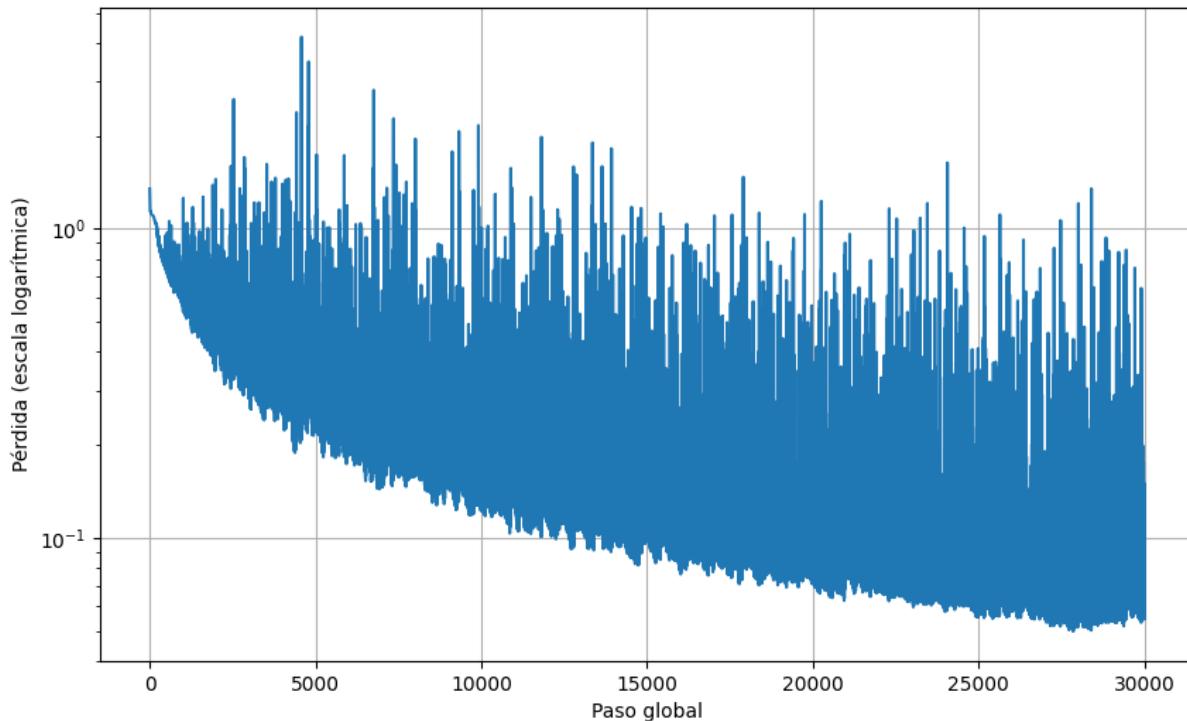


Figura 5.26: Evolución de la pérdida de entrenamiento en función del número de épocas
- Ecuación de Schrödinger unidimensional - Tiempo de entrenamiento 24 minutos

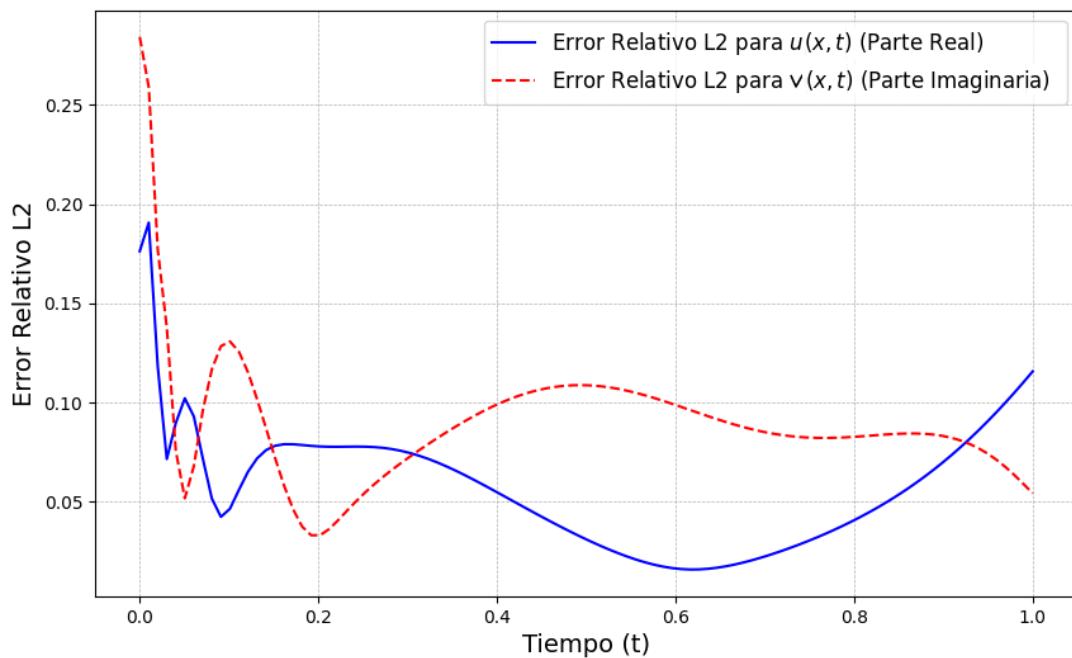


Figura 5.27: Ecuación de Schrödinger unidimensional; evolución del error relativo L_2 en el tiempo $[0, 1]$

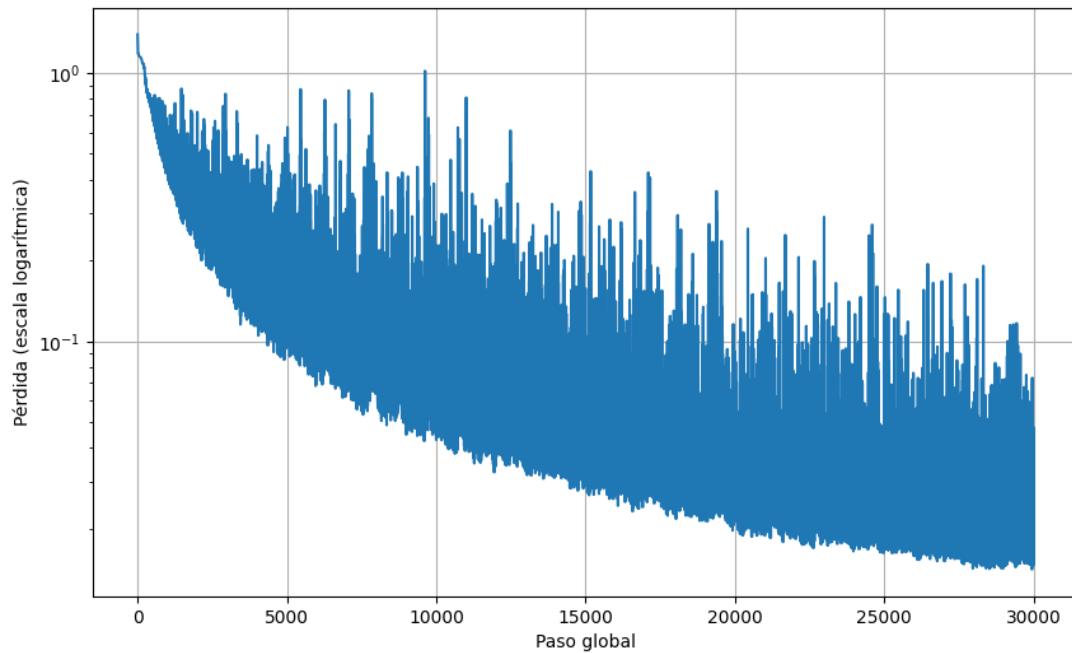


Figura 5.28: Evolución de la pérdida de entrenamiento en función del número de épocas
- Ecuación de Schrödinger unidimensional - Tiempo de entrenamiento 88 minutos

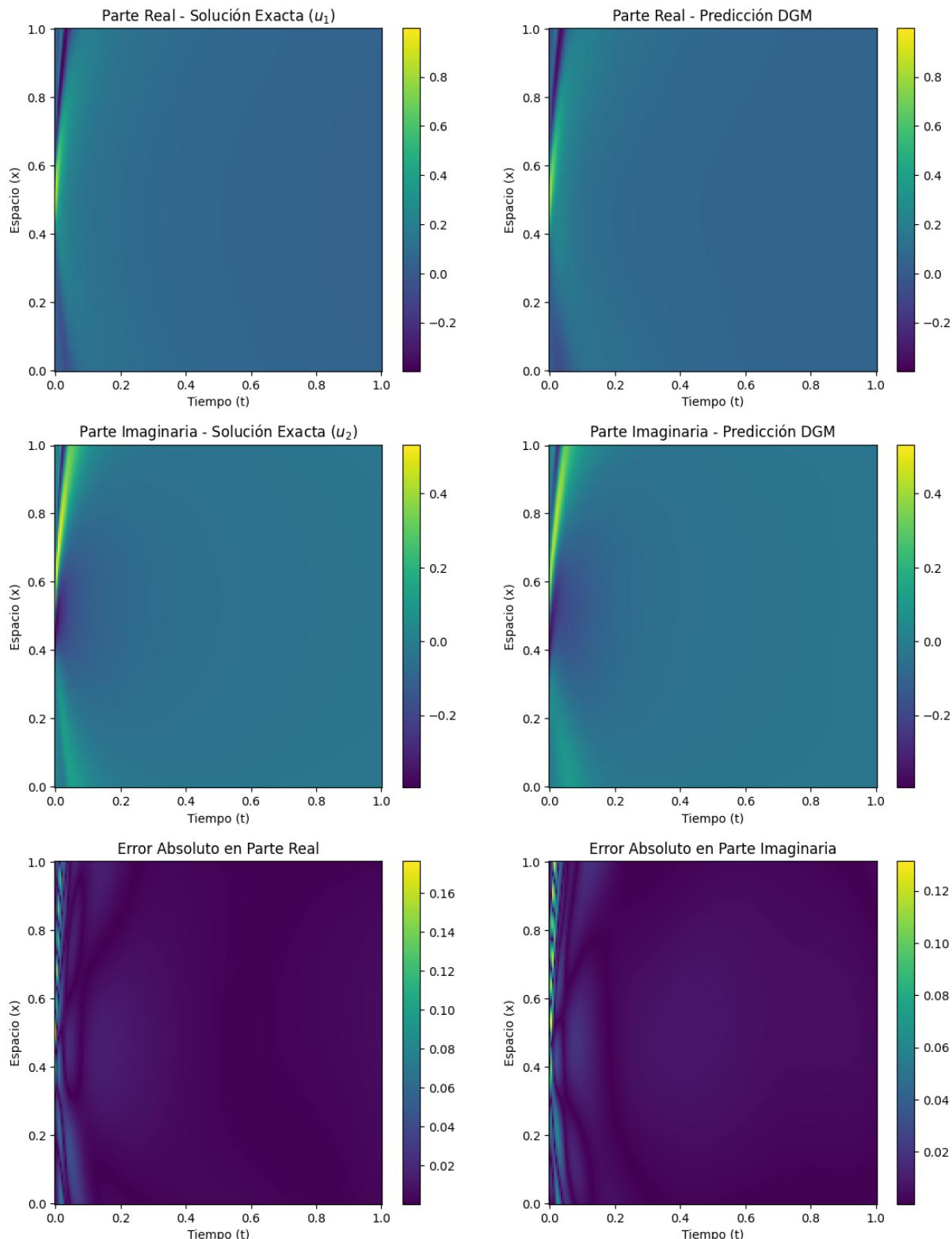


Figura 5.29: Ecuación de Schrödinger; **Izquierdo:** Solución exacta parte real e imaginaria u y v **Derecha:** Solución aproximada por DGM u^* y v^* . **Abajo:** Errores absolutos parte real e imaginaria

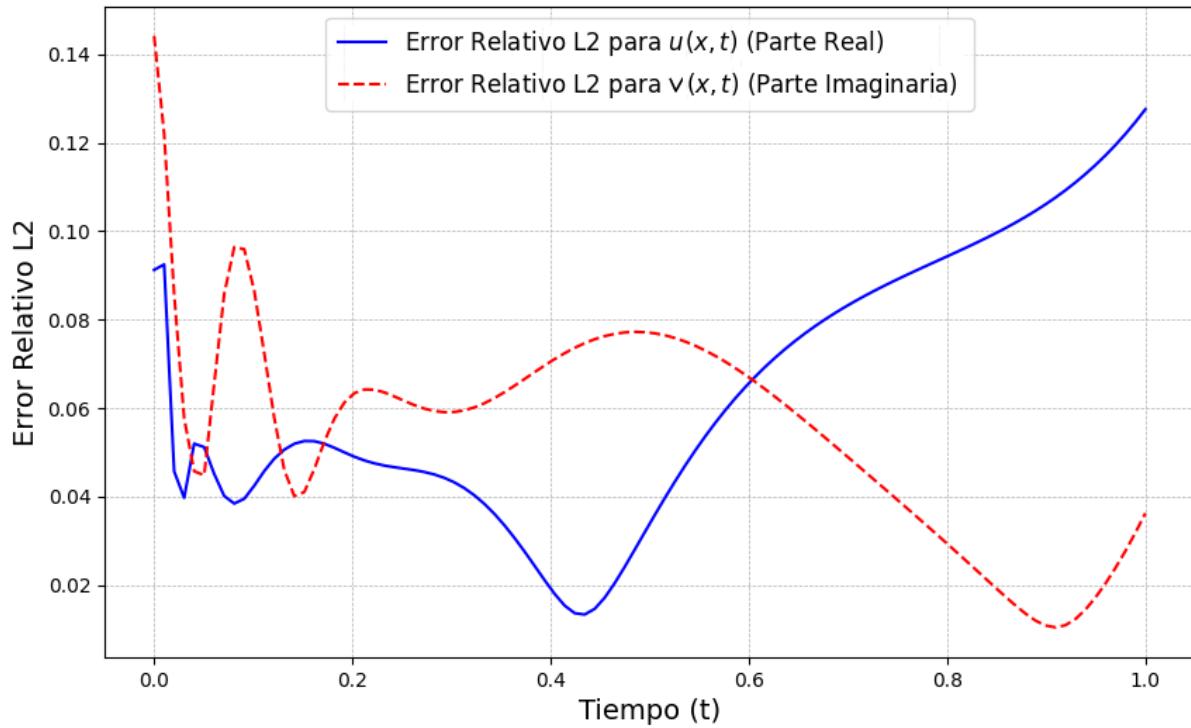


Figura 5.30: Ecuación de Schrödinger unidimensional; evolución del error relativo L_2 en el tiempo $[0, 1]$

Los resultados obtenidos validan la capacidad del método para resolver ecuaciones diferenciales parciales con soluciones complejas, incluso en presencia de comportamiento oscilatorio característico de los sistemas cuánticos. Las figuras 5.27 y 5.30 muestran la evolución del error relativo L_2 en el tiempo para ambas componentes de la función de onda, evaluadas a lo largo del intervalo temporal $t \in [0, 1]$.

En el primer escenario se observaron errores relativos máximos del orden de 0.25 en las etapas iniciales, los cuales disminuyen progresivamente conforme avanza el entrenamiento. En el segundo escenario, se evidencia una mejora sustancial en la precisión, con errores inferiores al 0.12 en la mayor parte del dominio. Esta mejora está directamente asociada a una mayor proporción de puntos en las condiciones iniciales y de frontera, así como a un mini-lote más grande, lo cual estabiliza la actualización de los parámetros de la red.

La figura 5.31 muestra una comparación detallada entre la solución exacta y la aproximación. En dicha figura se visualizan las componentes reales $u(x, t)$ e imaginarias $v(x, t)$ de la función de onda, así como los errores absolutos asociados a cada una.

Se observa que el modelo entrenado es capaz de reproducir con buena fidelidad el comportamiento oscilatorio y la propagación de la solución exacta, tanto en su parte

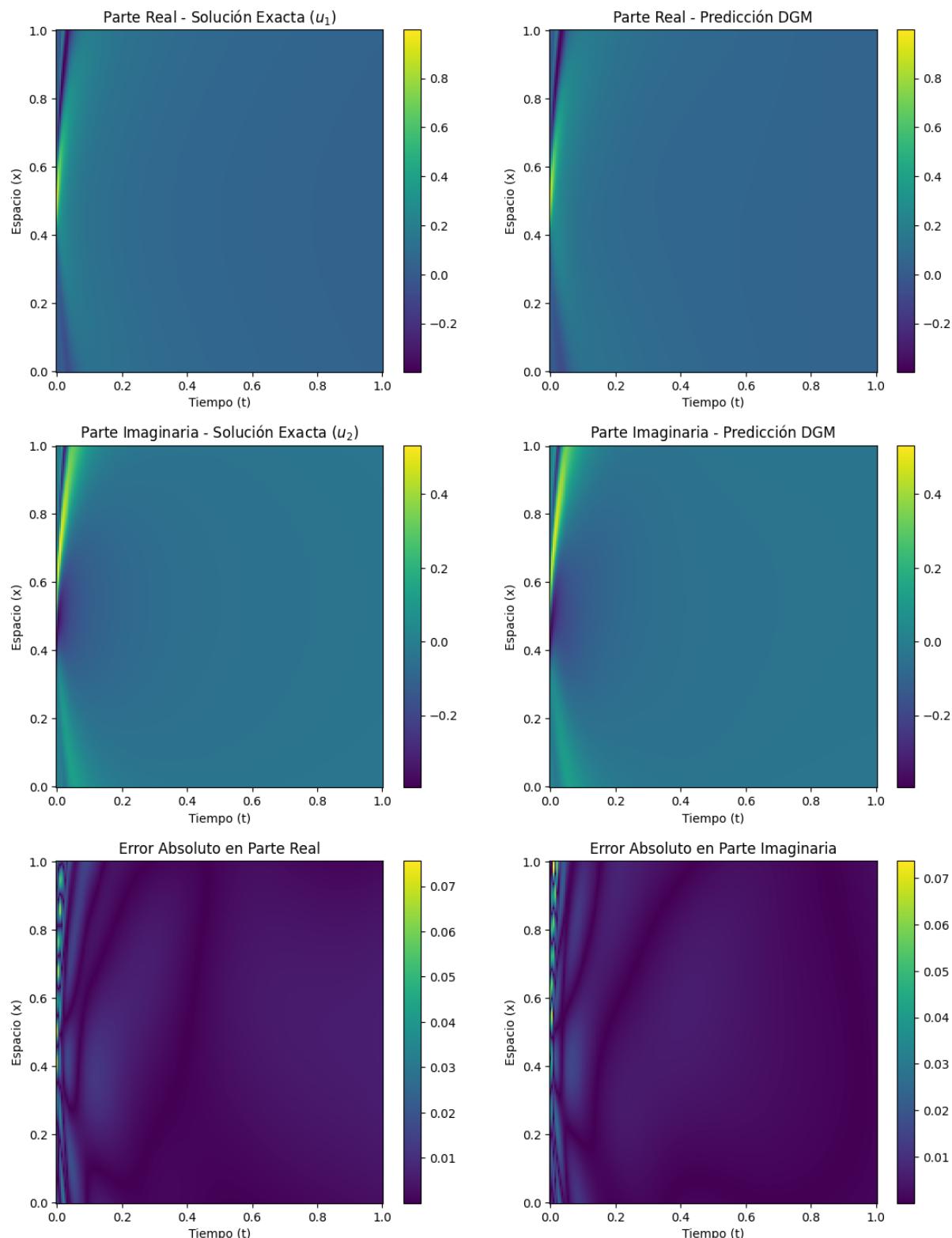


Figura 5.31: Ecuación de Schrödinger; **Izquierdo:** Solución exacta parte real e imaginaria u y v **Derecha:** Solución aproximada por DGM u^* y v^* . **Abajo:** Errores absolutos parte real e imaginaria

real como imaginaria. La mayor parte del dominio presenta una coincidencia visual clara entre la solución analítica y la predicción realizada por la red. No obstante, los errores absolutos se concentran principalmente en las primeras etapas del tiempo ($t < 0.2$) y en las regiones cercanas a los extremos espaciales, donde la solución presenta mayor variación. En cuanto a la magnitud de los errores, la parte real presenta un error absoluto máximo cercano a 0.16, mientras que en la parte imaginaria se mantiene por debajo de 0.13. En las zonas intermedias del dominio, donde la solución tiende a estabilizarse, el modelo conserva una alta precisión para ambas componentes de la función de onda.

En conjunto, estos resultados confirman que el método DGM puede aproximar de manera efectiva soluciones complejas en ecuaciones diferenciales parciales con comportamiento oscilatorio, aun utilizando una arquitectura neuronal moderada y una configuración inicial de hiperparámetros. Lo anterior sugiere que el desempeño del método puede mejorar aún más mediante estrategias de refinamiento adaptativo o el empleo de redes más profundas con mecanismos de memoria interna.

5.5.2. Caso átomo de hidrógeno con protón y electrón móviles

En este experimento partiendo de la formulación descrita anteriormente, donde se definió la función de pérdida (4.29) asociada a la ecuación de Schrödinger para el sistema de dos cuerpos, se implementa ahora el modelo computacional. El objetivo es aproximar la evolución espacio-temporal de la función de onda compleja $\psi(\mathbf{r}_e, \mathbf{r}_p, t)$ en un dominio tridimensional acotado para ambas partículas.

El entrenamiento de la red neuronal se realizó con los hiperparámetros indicados en la tabla 5-12. La arquitectura utiliza función de activación `tanh`, y aprendizaje con mini-lotes. La tasa de aprendizaje decrece de forma exponencial a lo largo de 10 fases de entrenamiento.

Tabla 5-12: Hiperparámetros: Ecuación de Schrödinger - Átomo de hidrógeno con protón y electrón móviles

N_{total}	N_i	N_f	N_{int}	Capas	Neuronas	Fases	Mini-lote	Tasa de Apr.
12000	1000	1000	10000	4	64	10	256	10^{-3} (d. exponencial)

La figura 5.32 muestra la evolución de la pérdida total durante el entrenamiento, indicando un descenso progresivo y estable a lo largo de las épocas.

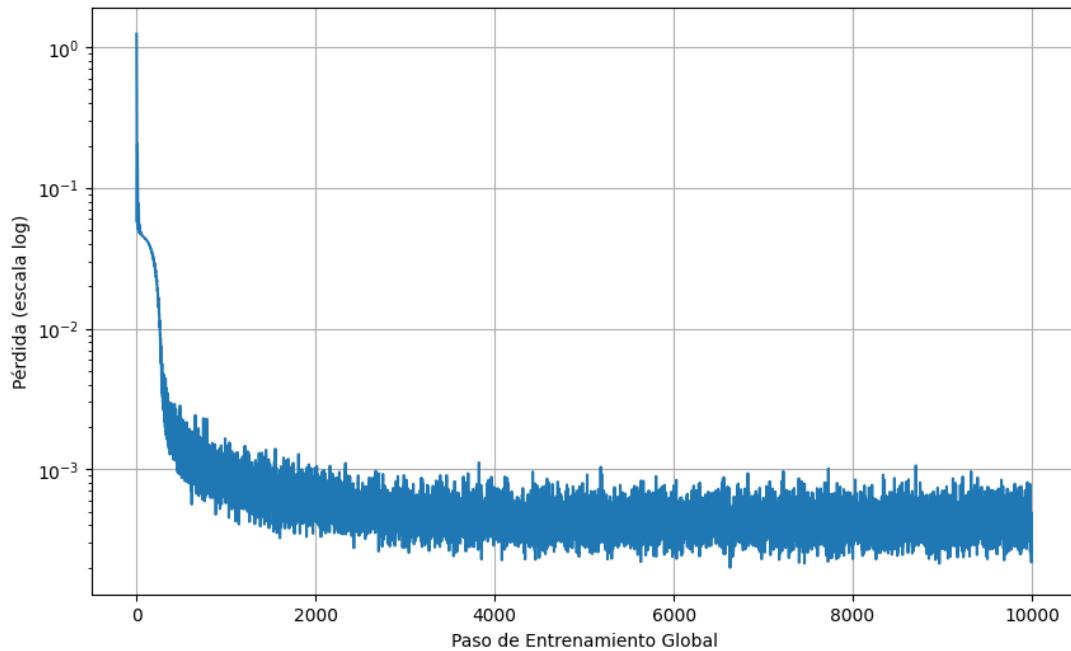


Figura 5.32: Evolución de la pérdida de entrenamiento - Ecuación de Schrödinger para el átomo de hidrógeno con protón y electrón móviles - Tiempo de entrenamiento 40 minutos.

Una vez entrenado el modelo, se realiza la evaluación sobre el dominio adimensionalizado para generar visualizaciones de la densidad de probabilidad conjunta. En la figura 5.33 se muestra la nube electrónica resultante del muestreo tipo Metropolis–Hastings (algoritmo 3), mientras que la figura 5.34 presenta el perfil radial promedio extraído de la solución. Se evaluaron las componentes de energía esperadas a partir de la función de onda entrenada. El valor promedio del potencial fue -1.2371 Hartree, mientras que las energías cinéticas promedio de las partículas fueron:

$$\langle T_e \rangle \approx 0.2164 \quad \text{y} \quad \langle T_p \rangle \approx 0.0005 \quad (\text{Hartree}),$$

consistentes con la diferencia de masas entre electrón y protón. El intervalo energético observado sin filtrado fue $(-34.65, 0.86)$ Hartree.

En cuanto a la energía total estimada, se obtuvo un valor promedio filtrado de -0.1762 Hartree. Si bien esta estimación presenta una desviación respecto al valor teórico exacto del estado fundamental (-0.5 Hartree), el modelo es capaz de capturar la forma general de la solución, así como la simetría esférica del sistema.

Este resultado demuestra la aplicabilidad del método para resolver ecuaciones de Schrödinger de múltiples cuerpos en geometrías tridimensionales, incluso en presencia de interacciones singulares como el potencial coulombiano. La aproximación sin mallas y

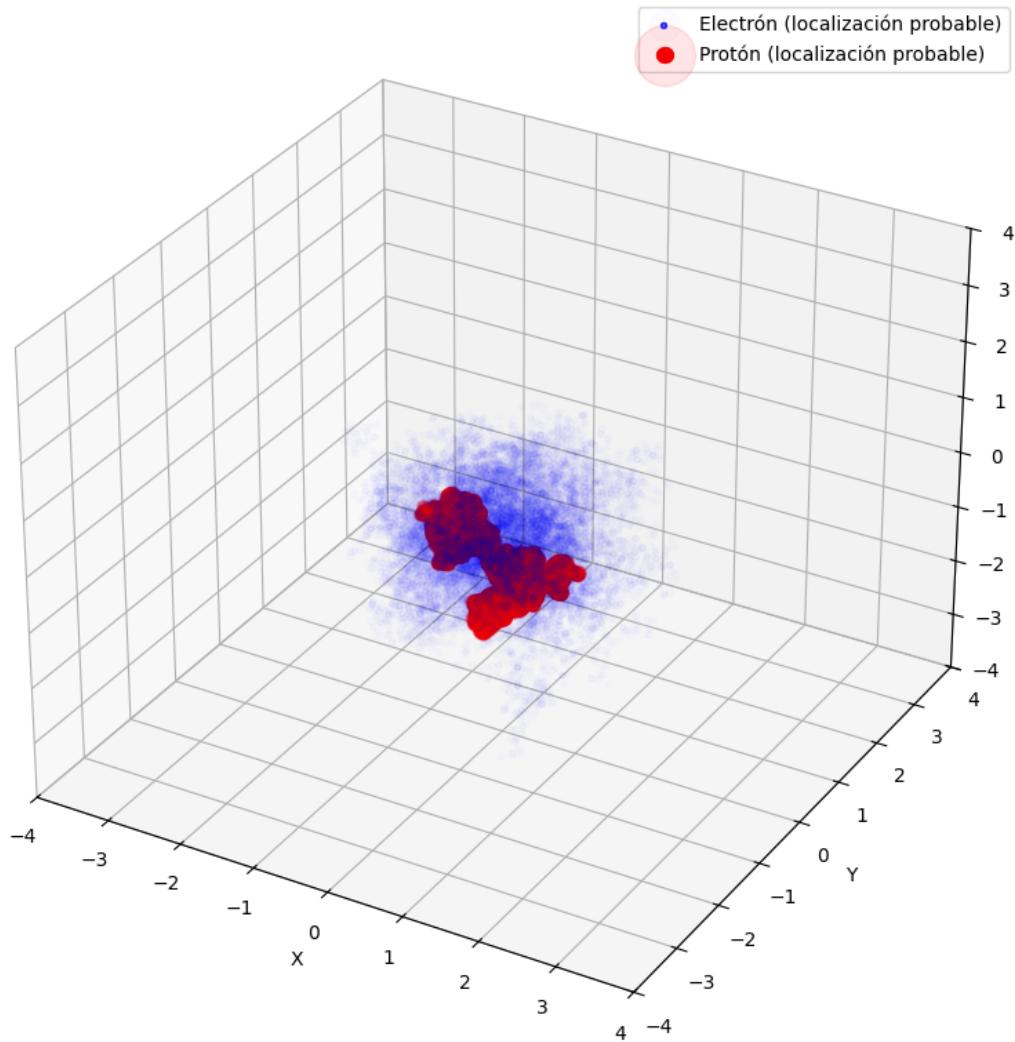


Figura 5.33: Visualización de la densidad electrónica del átomo de hidrógeno estimada mediante DGM

la capacidad de evaluar la solución en cualquier punto del dominio hacen del enfoque una herramienta prometedora para problemas cuánticos más complejos.

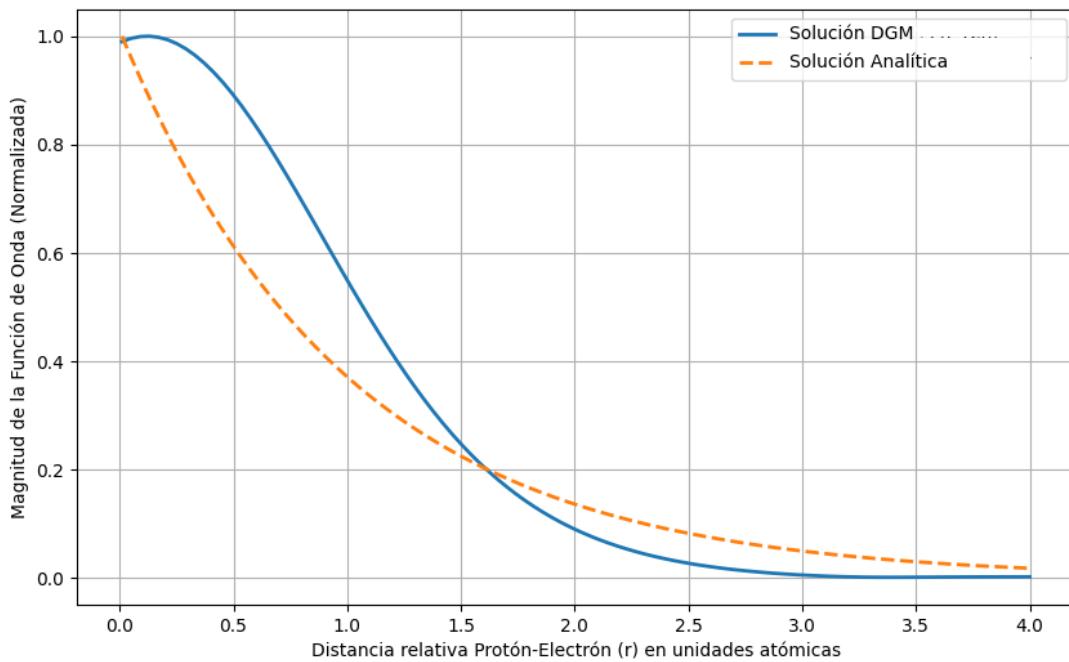


Figura 5.34: Perfil radial de la función de onda $|\psi(r)|^2$ en función de la distancia $r = |\mathbf{r}_e - \mathbf{r}_p|$

5.5.3. Caso átomo de hidrógeno con protón fijo y electrón móvil

En este experimento la arquitectura y metodología empleadas siguen la formulación teórica donde se definió la función de pérdida (4.31). En este caso, se considera que el protón permanece fijo en el origen del sistema de coordenadas, lo que simplifica el sistema de Schrödinger a una única partícula (el electrón) en un potencial central coulombiano.

El modelo recibe como entrada el cuarteto (x, y, z, t) y devuelve las componentes reales u e imaginarias v de la función de onda. A partir de estas componentes se calcula la densidad de probabilidad y se evalúan magnitudes físicas relevantes del sistema.

El entrenamiento de la red neuronal se realizó con los hiperparámetros, similares a los seleccionados con protón y electrón móvil como medio de comparación, que se encuentran en la tabla 5-13. La arquitectura utiliza función de activación `tanh`, y aprendizaje con mini-lotes. La tasa de aprendizaje decrece de forma exponencial a lo largo de 10 fases de entrenamiento.

La evolución de la pérdida total durante el entrenamiento del modelo se muestra en la figura 5.38, con una disminución sostenida que indica convergencia adecuada del método.

La figura 5.36 muestra la densidad de probabilidad electrónica obtenida a partir de la solución aprendida, mientras que la figura 5.37 presenta el perfil radial $|\psi(r)|^2$ en función

Tabla 5-13: Hiperparámetros: Ecuación de Schrödinger - Átomo de hidrógeno con protón fijo y electrón móvil

N_{total}	N_i	N_f	N_{int}	Capas	Neuronas	Fases	Mini-lote	Tasa de Apr.
12000	1000	1000	10000	4	64	10	256	10^{-3} (d. exponencial)

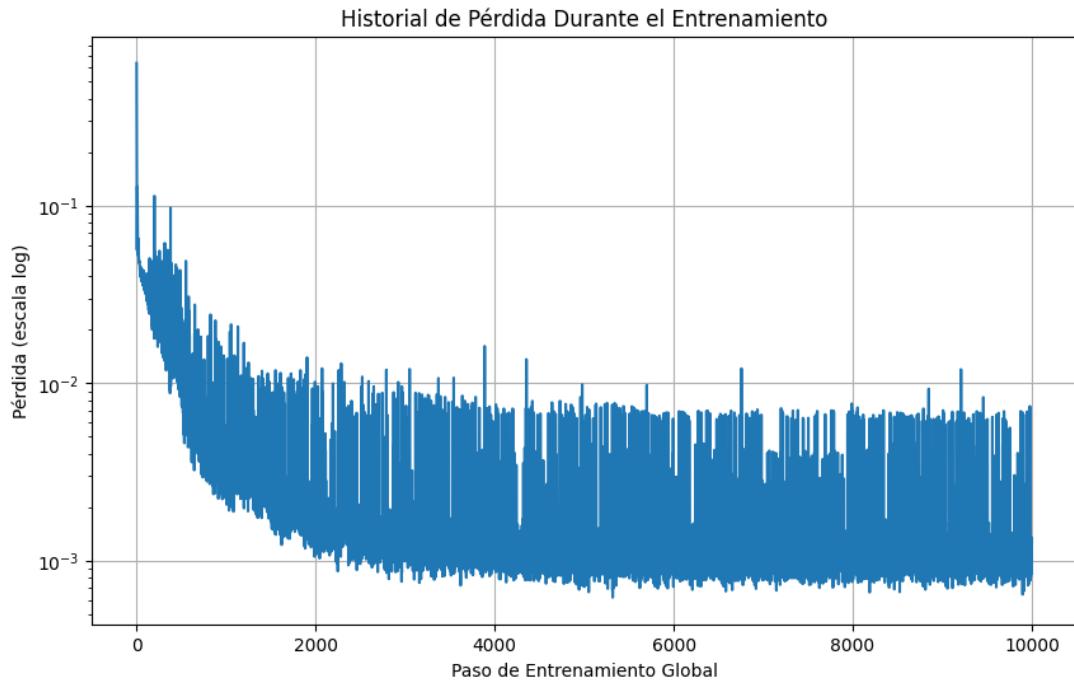


Figura 5.35: Evolución de la pérdida de entrenamiento - Ecuación de Schrödinger para el átomo de hidrógeno con protón fijo

de la distancia al núcleo, comparado con la solución analítica del estado fundamental.

Se obtuvo una energía total promedio de -0.6836 Hartree, la cual se aproxima de forma notable al valor teórico del estado fundamental del hidrógeno, que es -0.5 Hartree. La diferencia observada puede atribuirse a:

- Limitaciones del dominio espacial truncado en la simulación.
- Aproximación de la derivada temporal en una región finita del intervalo.
- Discretización implícita en el muestreo del entrenamiento sin malla.

Aun con estas limitaciones, el modelo demuestra ser capaz de capturar con alta precisión la estructura de la solución cuántica, confirmando que el método DGM constituye una herramienta robusta y flexible para la resolución de la ecuación de Schrödinger en sistemas con simetría esférica y potenciales singulares. Cabe mencionar que, además del presente enfoque basado en aprendizaje profundo, la ecuación de Schrödinger ha sido resuelta

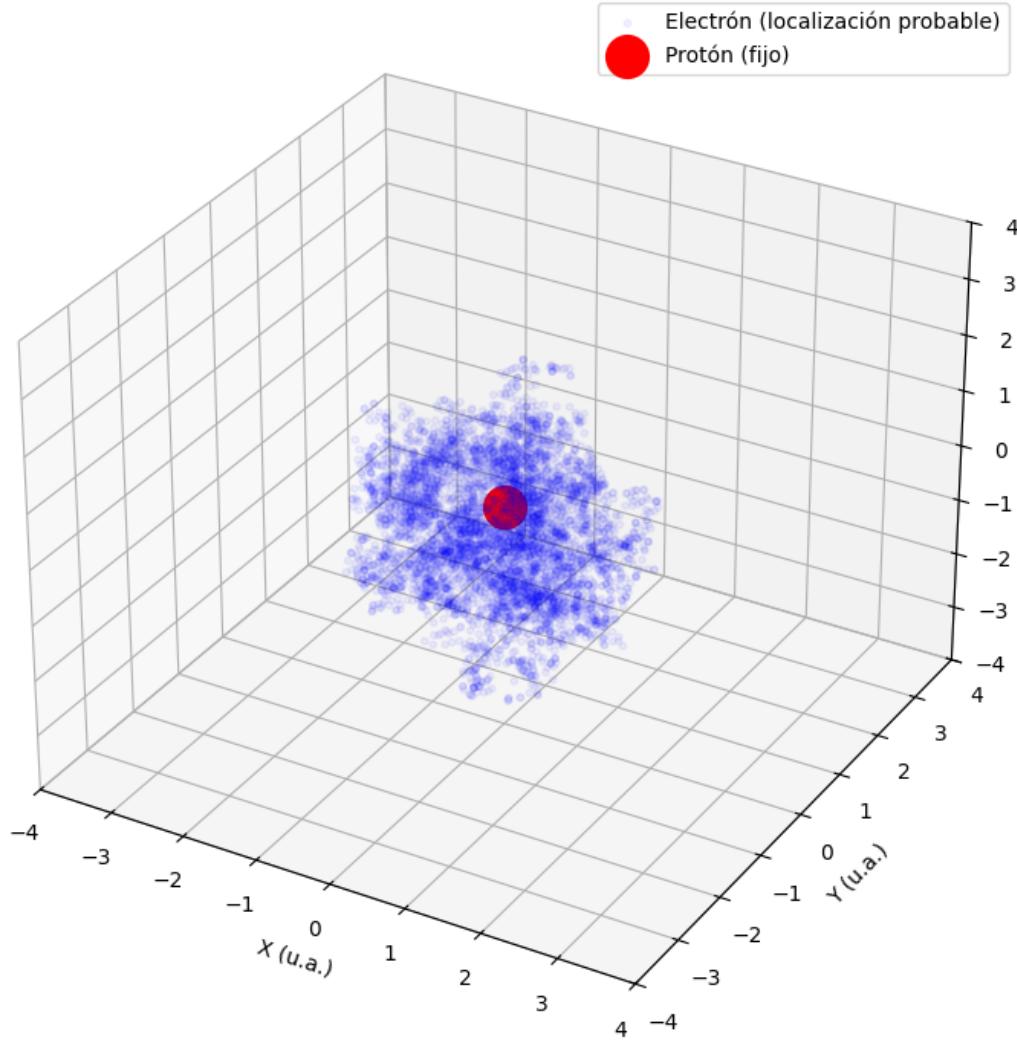


Figura 5.36: Densidad de probabilidad electrónica estimada mediante DGM para el caso de protón fijo

en trabajos anteriores empleando métodos numéricos híbridos. En particular, en [26] se propone un algoritmo que combina técnicas de diferencias finitas con métodos Monte Carlo, el cual fue aplicado exitosamente a sistemas unidimensionales y multidimensionales como el oscilador lineal, el átomo de hidrógeno, el ion de hidrógeno, la molécula de agua, y otros sistemas moleculares complejos. Dicho enfoque, aunque distinto en su formulación, permite obtener soluciones eficientes y económicas, y ha sido formalizado para manejar un número arbitrario de partículas cuánticas.

Lo notable es que la visualización obtenida en este trabajo para el caso del átomo de hidrógeno con protón fijo muestra una configuración espacial de la densidad electrónica cualitativamente similar a la reportada en [26]. En ambas metodologías se evidencia la

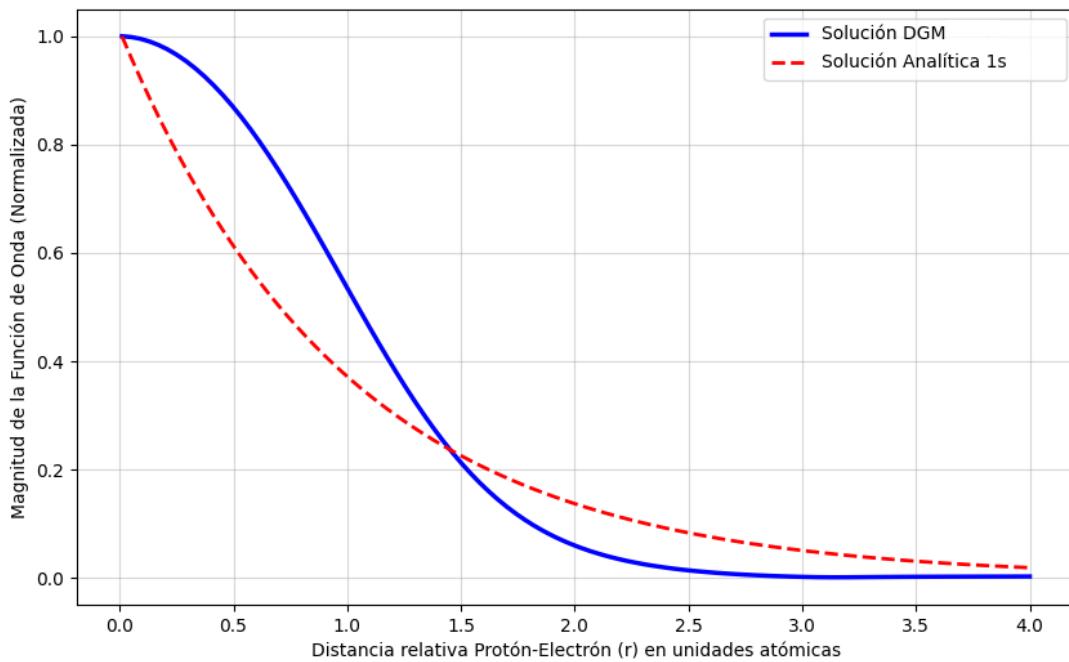


Figura 5.37: Comparación del perfil radial $|\psi(r)|^2$ obtenido por DGM con la solución analítica $R_{1s}(r) = 2e^{-r}$

simetría esférica esperada y la concentración de la densidad en torno al origen, validando la forma de la solución incluso a través de técnicas computacionales disímiles. Esta coincidencia refuerza la confiabilidad del resultado obtenido mediante DGM y su potencial como alternativa moderna para el tratamiento de sistemas cuánticos de interés físico y químico.

5.5.4. Caso: átomo de hidrógeno con protón fijo y electrón móvil en coordenadas esféricas

En este último experimento, la arquitectura y la metodología sigue una similitud a la formulación teórica donde se definió la función de pérdida (4.31) pero ahora se trabaja en coordenadas esféricas para aprovechar la simetría del potencial coulombiano y así favorecer el aprendizaje de la red neuronal. En este caso, la ecuación de Schrödinger dependiente del tiempo en unidades atómicas ($\hbar = m_e = e = 1$) para un electrón en el campo de un protón fijo se escribe como

$$i \frac{\partial \psi}{\partial t}(r, \theta, \phi, t) = \left[-\frac{1}{2} \nabla^2 - \frac{1}{r} \right] \psi(r, \theta, \phi, t), \quad (5.2)$$

donde el operador Laplaciano en coordenadas esféricas es

$$\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{2}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \Delta_{S^2}, \quad (5.3)$$

con Δ_{S^2} el Laplaciano angular sobre la esfera unitaria.

El modelo recibe como entrada el quinteto $(r, \theta, \phi, t, e^{-r})$ y devuelve las componentes real u e imaginaria v de la función de onda $\psi = u + iv$. La inclusión de e^{-r} como característica de entrada y su factorización en la salida responden al conocimiento previo de que las funciones de onda de sistemas ligados presentan colas exponenciales en la región clásicamente prohibida. Este comportamiento ha sido confirmado experimentalmente, como en la observación directa del decaimiento exponencial en el estado halo del dímero de helio [40], demostrado teóricamente en la asintótica de la densidad electrónica en DFT de Kohn–Sham [41] y formalizado matemáticamente en la localización exponencial por debajo del umbral de ionización [42]. Incorporar este sesgo inductivo reduce el rango dinámico de la función a aprender y mejora el acondicionamiento del entrenamiento.

El entrenamiento de la red neuronal se realizó con hiperparámetros similares al caso de protón y electrón móviles, con función de activación `gelu`, mini-lotes y una tasa de aprendizaje que decrece exponencialmente a lo largo de 10 fases. Los valores específicos se muestran en la Tabla 5-14.

La evolución de la pérdida total durante el entrenamiento se muestra en la Figura 5.38, donde se aprecia una disminución sostenida que indica convergencia adecuada del método. La Figura 5.39 muestra la densidad de probabilidad electrónica estimada mediante DGM, mientras que la Figura 5.37 presenta el perfil radial $|\psi(r)|^2$ comparado con la solución analítica del estado fundamental $R_{1s}(r) = 2e^{-r}$.

Tabla 5-14: Hiperparámetros: Ecuación de Schrödinger (protón fijo, electrón móvil en coordenadas esféricas).

N_{total}	N_i	N_f	N_{int}	Capas	Neuronas	Fases	Mini-lote	Tasa de Apr.
12000	1000	1000	10000	4	64	10	256	10^{-3} (d. exponencial)

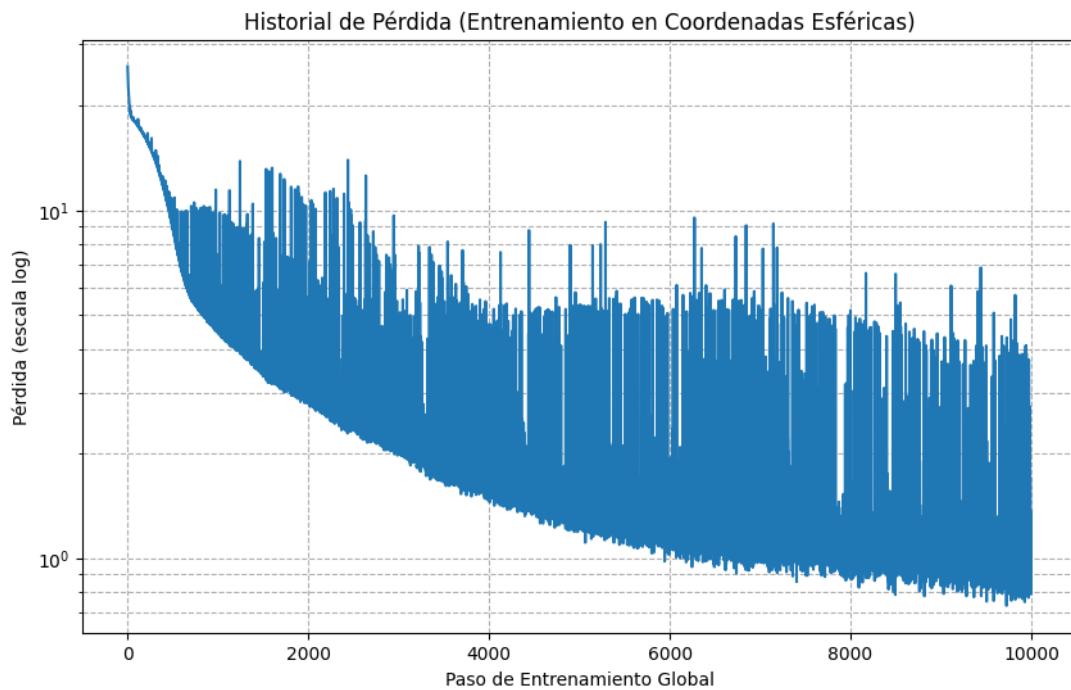


Figura 5.38: Evolución de la pérdida de entrenamiento para el caso de protón fijo (DGM en coordenadas esféricas).

En la figura 5.40 se observa el perfil radial $|\phi(r)|$ obtenido con el modelo DGM en coordenadas esféricas, comparado con la solución analítica del estado fundamental del hidrógeno $1s$. La concordancia entre ambas curvas es notable a lo largo de todo el dominio radial, mostrando que el método es capaz de reproducir con alta fidelidad tanto la pendiente exponencial en la región lejana al núcleo como la amplitud en las proximidades de $r = 0$. Esta coincidencia sugiere que la arquitectura y el sesgo inductivo implementados, incluyendo la factorización por e^{-r} permiten capturar con precisión el comportamiento asintótico y la estructura interna de la función de onda.

En contraste, la figura 5.37 corresponde a un caso previo resuelto en coordenadas cartesianas sin la misma optimización para la simetría esférica. Allí se aprecia una mayor discrepancia entre la solución DGM y la solución analítica, especialmente en la cola exponencial y en la región intermedia $1.0 < r < 3.0$ Bohr, donde la función aprendida

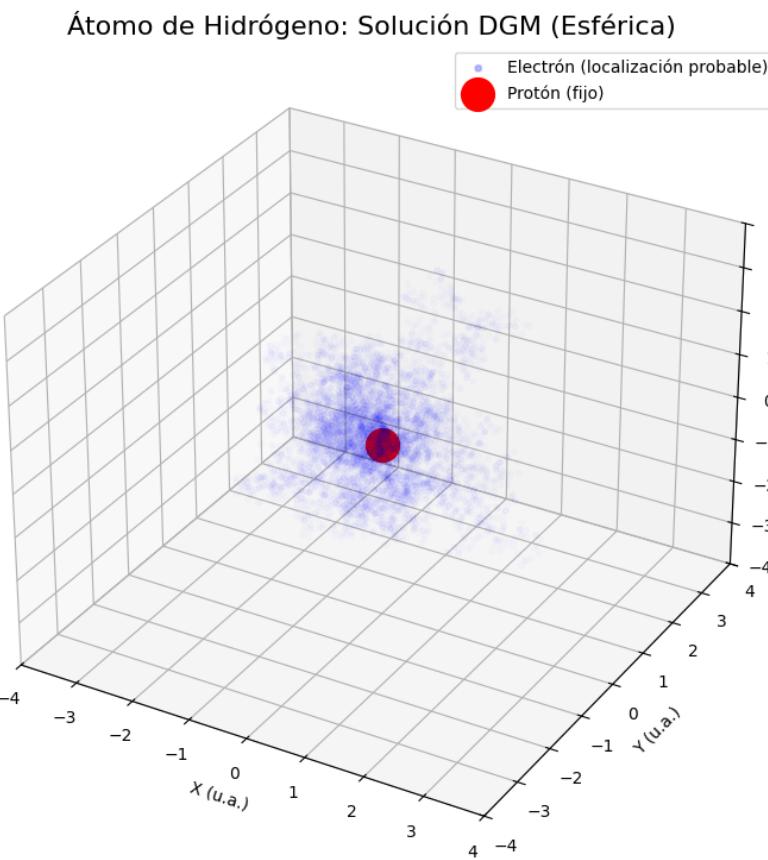


Figura 5.39: Densidad de probabilidad electrónica estimada mediante DGM (protón fijo).

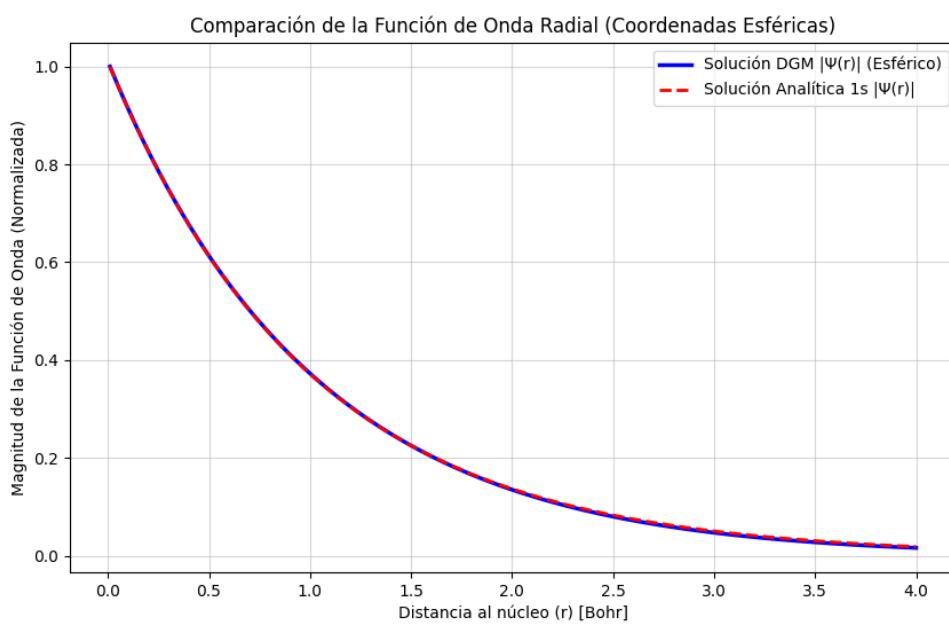


Figura 5.40: Perfil radial en coordenadas esféricas $|\psi(r)|^2$ obtenido por DGM vs. solución analítica $R_{1s}(r) = 2e^{-r}$.

presenta una caída más lenta y desviaciones en la amplitud.

La mejora evidente en la Figura 5.40 frente a la Figura 5.37 pone de manifiesto que la elección de un sistema de coordenadas alineado con la simetría del problema y la incorporación explícita de información física en la red neuronal (como el decaimiento exponencial) son factores determinantes para aumentar la precisión y estabilidad del método DGM en la resolución de la ecuación de Schrödinger para el átomo de hidrógeno.

Capítulo 6

Conclusiones y trabajos futuros

6.1. Conclusiones

A lo largo de este trabajo se ha comprobado que el método *Deep Galerkin Method* constituye una herramienta eficaz y versátil para la resolución de ecuaciones diferenciales parciales en diversos campos de la física y la ingeniería. En particular, su aplicabilidad fue evaluada en tres áreas representativas: la mecánica estructural (mediante los modelos de vigas de Euler–Bernoulli, Rayleigh y Timoshenko), la dinámica de fluidos (a través de la ecuación de Burgers) y la mecánica cuántica (por medio de la ecuación de Schrödinger). Los resultados obtenidos en los experimentos numéricos muestran que el método logra aproximaciones con alta precisión numérica y estabilidad computacional, incluso en dominios multidimensionales y en presencia de derivadas espaciales de orden elevado. Esta eficacia se mantuvo tanto en formulaciones acopladas como desacopladas, reflejándose en el adecuado comportamiento de los errores absolutos y relativos obtenidos. Asimismo, el método demostró una notable capacidad para reproducir las soluciones analíticas de los distintos modelos físicos abordados, incluyendo ecuaciones de vigas, de Burgers y de Schrödinger, evidenciando su versatilidad y robustez como esquema de resolución libre de mallas.

Un aspecto destacable es la capacidad del modelo para resolver las ecuaciones en derivadas parciales sin recurrir a discretizaciones explícitas en malla, lo que permite abordar problemas en espacios de alta dimensión sin incurrir en el fenómeno conocido como "la maldición de la dimensionalidad". Asimismo, la incorporación de estrategias como la separación de componentes de la solución (e.g., en la función de onda cuántica), el uso de arquitecturas profundas con compuertas, y técnicas de aprendizaje como el decaimiento

exponencial de la tasa de aprendizaje, han contribuido a mejorar el desempeño general del método.

A pesar de estos avances, se reconoce que existen limitaciones que abren la puerta a múltiples líneas de investigación futuras, tanto para mejorar la precisión como para extender el alcance del método a nuevas clases de problemas.

6.2. Trabajos futuros

Con base en la experiencia obtenida en esta tesis, se proponen las siguientes direcciones para trabajos futuros:

- **Extensión a geometrías más complejas:** Implementar el método DGM en dominios no rectangulares, con condiciones de frontera irregulares o dependientes del tiempo, especialmente en los modelos de viga de Euler–Bernoulli, Timoshenko y en placas delgadas (Kirchhoff–Love), con el fin de aproximarse mejor a casos reales en ingeniería estructural.
- **Optimización de hiperparámetros:** Incorporar algoritmos de selección automática de hiperparámetros (por ejemplo, *Bayesian optimization*, búsqueda aleatoria o técnicas evolutivas) para mejorar la eficiencia del entrenamiento y evitar configuraciones subóptimas.
- **Análisis teórico de convergencia:** Establecer bases teóricas sobre la convergencia del método DGM en diferentes tipos de ecuaciones y condiciones de frontera, con el objetivo de formalizar garantías de estabilidad y precisión que actualmente solo se verifican de forma empírica.
- **Aplicación a sistemas cuánticos de mayor complejidad:** Extender el método a moléculas como el ion de hidrógeno, la molécula de hidrógeno, o el agua; lo que implica trabajar con funciones de onda de alta dimensionalidad, y donde la malla tradicional se vuelve inviable. Estas aplicaciones permitirán aprovechar la escalabilidad inherente del DGM y explorar su ventaja frente a métodos numéricos clásicos.

Estas líneas de investigación permitirán afianzar el uso de modelos neuronales en la solución de EDP y consolidar el papel del aprendizaje profundo como una alternativa poderosa y flexible a los métodos numéricos tradicionales.

Apéndice A

Algoritmo de Metropolis–Hastings

El algoritmo de *Metropolis–Hastings* [44, 45, 46] constituye un procedimiento de muestreo estocástico enmarcado dentro de la familia de los métodos de *Monte Carlo* mediante *cadenas de Markov* (MCMC). Su propósito principal es generar muestras de una distribución de probabilidad objetivo $P(x)$, aún en aquellos casos donde dicha distribución no puede ser evaluada de manera directa o normalizada. Esta característica lo convierte en una herramienta especialmente valiosa en contextos donde la densidad objetivo solo se conoce hasta una constante de proporcionalidad, como ocurre habitualmente en inferencia bayesiana, física estadística y mecánica cuántica computacional.

El fundamento del método radica en la construcción de una *cadena de Markov ergódica* cuya distribución estacionaria coincida con la distribución objetivo $P(x)$. Una vez alcanzado el estado estacionario, las muestras generadas por la cadena pueden considerarse como provenientes de $P(x)$, lo cual permite realizar inferencias estadísticas o simulaciones a partir de una representación aproximada de la misma.

El algoritmo opera mediante un proceso iterativo que alterna entre la generación de una propuesta aleatoria y una etapa de *aceptación o rechazo* basada en un criterio probabilístico que garantiza el cumplimiento de la condición de equilibrio detallado. Esta estrategia asegura la convergencia asintótica de la cadena hacia la distribución deseada, incluso cuando $P(x)$ no es conocida en forma explícita.

El algoritmo de Metropolis–Hastings no solo constituye una herramienta esencial en estadística bayesiana y física estadística, sino que también desempeña un papel crucial en el ámbito de la **mecánica cuántica computacional**.

Una de las aplicaciones más notables se encuentra en el método de *Monte Carlo de función de onda variacional*, donde se busca estimar el valor esperado de un operador

cuántico, típicamente la energía, con respecto a una función de onda de prueba $\psi_T(x)$. Dado que esta función de onda puede ser compleja y no normalizable de forma explícita, el algoritmo de Metropolis–Hastings permite generar una secuencia de configuraciones $x \in \mathbb{R}^N$ distribuidas de acuerdo con la densidad $|\psi_T(x)|^2$, sin necesidad de conocer la constante de normalización:

La flexibilidad del algoritmo y su capacidad para muestrear distribuciones de alta dimensión con estructuras complejas lo han consolidado como una herramienta central en simulaciones cuánticas modernas, desde modelos de átomos y moléculas hasta sistemas de materia condensada o gases cuánticos confinados.

Algoritmo 3 Algoritmo de Metropolis-Hastings

Objetivo: Generar una secuencia de muestras $\{x^{(1)}, x^{(2)}, \dots, x^{(N)}\}$ de una distribución de probabilidad objetivo $P(x)$.

- 1: **Inicialización:** Escoger un punto de partida inicial $x^{(0)}$ y asignarlo como el estado actual: $x_{actual} \leftarrow x^{(0)}$.
- 2: **para** $i = 1$ **to** N **hacer**
- 3: Proponer un nuevo estado candidato x' a partir del estado actual x_{actual} usando una distribución de propuesta $Q(x'|x_{actual})$.
- 4: Calcular el **ratio de aceptación**, α :

$$\alpha(x'|x_{actual}) = \min \left(1, \frac{P(x')}{P(x_{actual})} \frac{Q(x_{actual}|x')}{Q(x'|x_{actual})} \right)$$

- 5: Generar un número aleatorio uniforme $u \sim U(0, 1)$.
 - 6: **si** $u < \alpha$ **entonces**
 - 7: Aceptar el nuevo estado: $x_{actual} \leftarrow x'$.
 - 8: **si no**
 - 9: Rechazar el nuevo estado y mantener el estado actual: $x_{actual} \leftarrow x_{actual}$.
 - 10: **fin si**
 - 11: Guardar el estado actual como la siguiente muestra en la cadena: $x^{(i)} \leftarrow x_{actual}$.
 - 12: **fin para**
 - 13: **Retornar:** La secuencia de muestras $\{x^{(i)}\}_{i=1}^N$.
-

Apéndice B

Teorema de aproximación universal

Uno de los fundamentos teóricos que respalda el uso de redes neuronales para aproximar soluciones de ecuaciones diferenciales es el *Teorema de Aproximación Universal*. En su forma más conocida, este teorema establece que una red neuronal *feedforward* con una sola capa oculta, bajo condiciones adecuadas sobre la función de activación, puede aproximar cualquier función continua definida en un conjunto compacto con una precisión arbitraria.

Teorema (Aproximación Universal). Sea $K \subset \mathbb{R}^n$ compacto y $f : K \rightarrow \mathbb{R}$ una función continua. Si $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ es una función de activación continua y no polinómica (por ejemplo, la sigmoide), entonces, para todo $\varepsilon > 0$, existe una red neuronal con una sola capa oculta —con parámetros apropiados— tal que su salida $N(x)$ satisface

$$\sup_{x \in K} |f(x) - N(x)| < \varepsilon.$$

Este resultado fue demostrado originalmente por George Cybenko [47] en el contexto de funciones sigmoides y más tarde generalizado por Hornik, Stinchcombe y White [48] para un conjunto más amplio de funciones de activación.

Pese a su importancia, el teorema presenta varias limitaciones prácticas:

- Es un resultado de existencia, no constructivo: garantiza que existe una red con los parámetros adecuados, pero no especifica cómo encontrarlos ni cuántas neuronas son necesarias en la práctica.
- No proporciona información sobre la velocidad de convergencia ni sobre la dificultad del entrenamiento (optimización, regularización o generalización).
- En espacios de entrada de alta dimensión, la cantidad de neuronas requeridas para alcanzar una precisión determinada puede crecer de forma exponencial, fenómeno

conocido como la *maldición de la dimensionalidad*. Este incremento refleja tanto la complejidad del dominio de la EDP como las limitaciones de las arquitecturas poco profundas para representar funciones multivariadas.

En el contexto del *Deep Galerkin Method* (DGM), el teorema justifica teóricamente la existencia de una red neuronal capaz de aproximar con precisión la solución (o el residuo) de una ecuación diferencial parcial. Sin embargo, no garantiza que el proceso de entrenamiento mediante optimización numérica logre converger hacia dicha solución.

Versiones más recientes del teorema extienden este principio al caso de redes profundas con funciones de activación como ReLU o GELU, e incluso al caso de operadores continuos (como en *DeepONet*) [49, 50]. Estas generalizaciones fortalecen el sustento teórico del uso de arquitecturas neuronales complejas en la aproximación de soluciones de EDPs y operadores diferenciales.

Bibliografía

- [1] Richard E. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- [2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [3] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.
- [4] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [5] Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.
- [6] S. P. Timoshenko and J. M. Gere. *Theory of Elastic Stability*. McGraw-Hill, 1955.
- [7] Stephen P. Timoshenko. *History of strength of materials*. McGraw-Hill, 1953.
- [8] Stephen P. Timoshenko. Lxvi. on the correction for shear of the differential equation for transverse vibrations of prismatic bars. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 41(245):744–746, 1921.
- [9] A. E. H. Love. *A Treatise on the Mathematical Theory of Elasticity*. Cambridge University Press, Cambridge, 1888.
- [10] S. P. Timoshenko and S. Woinowsky-Krieger. *Theory of Plates and Shells*. McGraw-Hill, New York, 2 edition, 1959.
- [11] J.M. Burgers. A mathematical model illustrating the theory of turbulence. *Advances in Applied Mechanics*, 1:171–199, 1948.

- [12] David J. Griffiths and Darrell F. Schroeter. *Introduction to Quantum Mechanics*. Cambridge University Press, 3rd edition, 2018.
- [13] James M. Gere and Stephen P. Timoshenko. *Mechanics of Materials*. Brooks/Cole, Cengage Learning, 5 edition, 2001.
- [14] Lord (John William Strutt) Rayleigh. *The Theory of Sound*. Macmillan, 1877. Vol. 1.
- [15] Leonard Meirovitch. *Analytical Methods in Vibrations*. Macmillan, 1967.
- [16] Stephen P. Timoshenko. *Strength of Materials, Part I: Elementary Theory and Problems*. Van Nostrand, 3 edition, 1955.
- [17] J. N. Reddy. *Theory and Analysis of Elastic Plates and Shells*. CRC Press, Boca Raton, 2 edition, 2007.
- [18] R. Szilard. *Theories and Applications of Plate Analysis*. John Wiley & Sons, New Jersey, 2004.
- [19] G. B. Whitham. *Linear and Nonlinear Waves*. Wiley, 2011.
- [20] J. Smoller. *Shock Waves and Reaction-Diffusion Equations*. Springer, 2012.
- [21] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 2000.
- [22] M. J. Lighthill and G. B. Whitham. On kinematic waves. ii. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society A*, 229(1178):317–345, 1955.
- [23] P. I. Richards. Shock waves on the highway. *Operations Research*, 4(1):42–51, 1956.
- [24] J. D. Cole. On a quasi-linear parabolic equation occurring in aerodynamics. *Quarterly of Applied Mathematics*, 9:225–236, 1951.
- [25] E. Hopf. The partial differential equation $u_t + uu_x = \mu u_{xx}$. *Communications on Pure and Applied Mathematics*, 3(3):201–230, 1950.
- [26] Konstantin Eduardovich Plokhotnikov. Solving the schrodinger equation on the basis of finite-difference and monte-carlo approaches. *Journal of Applied Mathematics and Physics*, 9(2):328–369, 2021.

- [27] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [28] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2 edition, 1999.
- [29] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.
- [30] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, 2013.
- [31] Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.
- [32] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, 2010.
- [33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2015.
- [34] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [35] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [36] Georgios Is Detorakis. Practical aspects on solving differential equations using deep learning: A primer. *arXiv preprint arXiv:2408.11266*, 2024.
- [37] Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

- [38] Cao Yanhua Luo Song. Deep learning for solving the time-dependent schrodinger equation. *Journal of Systems Science and Mathematical Sciences*, 45(5):1400–1412, 2025.
- [39] Taniya Kapoor, Hongrui Wang, Alfredo Núñez, and Rolf Dollevoet. Physics-informed neural networks for solving forward and inverse problems in complex beam systems. *IEEE Transactions on Neural Networks and Learning Systems*, 35(5):5981–5995, 2023.
- [40] Stefan Zeller, Maksim Kunitski, Jörg Voigtsberger, Anton Kalinin, Alexander Schottelius, Carl Schober, Markus Waitz, Hendrik Sann, Alexander Hartung, Tobias Bauer, et al. Imaging the he2 quantum halo state using a free electron laser. *Proceedings of the National Academy of Sciences*, 113(51):14651–14655, 2016.
- [41] Paola Gori-Giorgi, Tamás Gál, and Evert Jan Baerends. Asymptotic behaviour of the electron density and the kohn–sham potential in case of a kohn–sham homo nodal plane. *Molecular Physics*, 114(7-8):1086–1097, 2016.
- [42] M. Griesemer. Exponential decay and ionization thresholds in non-relativistic quantum electrodynamics. *Journal of Functional Analysis*, 210(2):321–340, 2004.
- [43] Deqing Jiang, Justin Sirignano, and Samuel N Cohen. Global convergence of deep galerkin and pinns methods for solving partial differential equations. *arXiv preprint arXiv:2305.06000*, 2023.
- [44] Anthony Alfred Barker. Monte carlo calculations of the radial distribution functions for a proton? electron plasma. *Australian Journal of Physics*, 18(2):119–134, 1965.
- [45] Bernd Albert Berg and Alexei Bazavor. *Markov chain Monte Carlo simulations and their statistical analysis: with web-based Fortran code*. World Scientific Publishing Company, 2004.
- [46] David DL Minh and Do Le Minh. Understanding the hastings algorithm. *Communications in Statistics-Simulation and Computation*, 44(2):332–349, 2015.
- [47] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

- [48] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
- [49] Takato Nishijima. Universal approximation theorem for neural networks, 2021. arXiv preprint.
- [50] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, 2019. arXiv preprint.