

UCUENCA

Universidad de Cuenca

Facultad de Ingeniería

Carrera de Ingeniería Civil

Resolución de la ecuación diferencial de una viga mediante la implementación de Physics-Informed Neural Networks

Trabajo de titulación previo a la obtención del título de Ingeniero Civil


Autores:

Pablo Javier Molineros Calle


Karen Viviana Sacoto Macías

Directores:

Esteban Patricio Samaniego Alvarado

ORCID:  0000-0002-8728-491X

Angel Oswaldo Vázquez Patiño

ORCID:  0000-0002-1679-3995

Cuenca, Ecuador

2023 -10- 02

Resumen

El presente trabajo de titulación se enfoca en la implementación y validación del método Physics-Informed Neural Networks (PINNs), utilizando un enfoque variacional en una aplicación común de la Mecánica de Sólidos, específicamente en el área estructural que aborda la deflexión en vigas bajo la teoría de Euler-Bernoulli. La formulación matemática empleada en este método se fundamenta en el enfoque variacional de la viga analizada, lo que permite obtener una solución aproximada mediante la implementación de una red neuronal.

Dado que la no linealidad del modelo puede influir en la resolución de este método, se realizaron modificaciones tanto en la formulación matemática como en los hiperparámetros de la implementación en la red neuronal para abordar esta complejidad. Aunque los métodos tradicionales poseen la rigurosidad y validez necesarias para resolver la mayoría de los problemas en ciencia e ingeniería, los nuevos métodos basados en redes neuronales, como el que se presenta en este trabajo de titulación, resultan prometedores debido al constante aumento en la capacidad computacional disponible.

La solución aproximada obtenida mediante el método estudiado muestra resultados consistentes en los tres problemas de vigas presentados, en comparación con la solución exacta. Esta investigación contribuye al crecimiento y desarrollo de enfoques novedosos para abordar problemas complejos en el campo de la mecánica de sólidos y demuestra el potencial de la integración de redes neuronales en la resolución de ecuaciones diferenciales en ingeniería.

Palabras clave: enfoque variacional, machine learning, mecánica de sólidos, Euler Bernoulli



El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Cuenca ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por la propiedad intelectual y los derechos de autor.

Repositorio Institucional: <https://dspace.ucuenca.edu.ec/>

Abstract

This thesis focuses on the implementation and validation of the Physics-Informed Neural Networks (PINNs) method, using a variational approach in a common application of Solid Mechanics, specifically in the structural area that deals with deflection in beams under the Euler-Bernoulli theory. The mathematical formulation used in this method is based on the variational approach of the analyzed beam, which allows obtaining an approximate solution through the implementation of a neural network.

Since the nonlinearity of the model may influence the resolution of this method, modifications were made to both the mathematical formulation and the hyperparameters of the neural network implementation to address this complexity. Although traditional methods have the rigor and validity necessary to solve most problems in science and engineering, new methods based on neural networks, such as the one presented in this degree work, are promising due to the constant increase in the available computational capacity.

The approximate solution obtained by the studied method shows consistent results in the three beam problems presented, as compared to the exact solution. This research contributes to the growth and development of novel approaches to tackle complex problems in the field of solid mechanics and demonstrates the potential of the integration of neural networks in the solution of differential equations in engineering.

Keywords: variational approach, machine learning, solid mechanics, Euler Bernoulli



The content of this work corresponds to the right of expression of the authors and does not compromise the institutional thinking of the University of Cuenca, nor does it release its responsibility before third parties. The authors assume responsibility for the intellectual property and copyrights.

Institutional Repository: <https://dspace.ucuenca.edu.ec/>

Índice de contenido

Resumen	2
Abstract	3
Índice de contenido.....	4
Índice de figuras	7
Índice de tablas	9
Dedicatoria	10
Dedicatoria	11
Agradecimientos.....	12
Introducción	13
Antecedentes	13
Problemática	14
Potenciales soluciones.....	16
Justificación	16
Objetivos	17
Objetivo principal.....	17
Objetivos secundarios	17
Resumen del contenido	17
Capítulo 2: Formulación matemática	19
Mecánica de sólidos.....	19
Aproximación de la ecuación de gobierno.....	20
Problema de valores de contorno	21
Problema variacional	30
Equivalencia de soluciones	30
Capítulo 3: Métodos de resolución de ecuaciones diferenciales.....	31
Métodos de resolución convencionales.....	31

Physics-Informed Neural Networks (PINNs).....	32
Origen y funcionamiento.....	32
Funcionamiento y elementos de la Physics-Informed Neural Networks.....	35
Función de pérdida y métodos de minimización	37
Capítulo 4: Implementación	39
Composición del sistema	39
Características geométricas y mecánicas de la viga	39
Problema 1: Viga simplemente apoyada	41
Problema 2: Viga empotrada – libre	42
Problema 3: Viga empotrada – simplemente apoyada	43
Modelado del sistema	44
Implementación de PINN	51
Hiperparámetros utilizados en la Physics-Informed Neural Networks.....	51
Arquitectura de la red.....	52
Función de activación	53
Algoritmos de optimización	53
Épocas de entrenamiento y tasa de aprendizaje	54
Inicialización de pesos	54
Puntos de colocación	54
Problemas de aplicación	55
Función de energía y función de pérdida balanceada	55
Generalidades de la Red Neuronal – Hiperparámetros implementados	56
Arquitectura de la red.....	56
Función de activación e inicialización de pesos	56
Algoritmo de optimización y épocas de entrenamiento	57

Puntos de colocación	58
Problema 1: Viga simplemente apoyada	58
Problema 2: Viga empotrada – libre	59
Problema 3: Viga empotrada – simplemente apoyada	60
Capítulo 5: Resultados y Discusiones.....	62
Problema 1: Viga simplemente apoyada	62
Problema 2: Viga empotrada – libre	65
Problema 3: Viga empotrada – simplemente apoyada	68
Sensibilidad de hiperparámetros.....	72
Arquitectura de la red	72
Capítulo 6: Conclusiones.....	74
Desarrollo de la formulación matemática	74
Desarrollo de la implementación	76
Comparación implementación de PINN con método analítico.....	77
Bibliografía.....	79
Anexos	83
Anexo A. Código fuente para replicación	83

Índice de figuras

Ilustración 1. Hipótesis de la viga bajo la teoría Euler Bernoulli [16].	21
Ilustración 2. Componentes de tensión positiva en un pequeño elemento cúbico Wang et al. [11].	25
Ilustración 3. Esfuerzos resultantes (sistema equivalente de fuerzas)	26
Ilustración 4. Equilibrio de un elemento de viga sometido a cargas axiales y transversales por Wang et al. [11].	27
Ilustración 5. Casos típicos de vigas [11].	29
Ilustración 6. Esquema de los elementos de una red neuronal informada por la física (PINN) por Lawal et al. [8].	36
Ilustración 7. Viga simplemente apoyada que reacciona ante fuerza puntual por Komperød M. [32].	40
Ilustración 8. Viga con condiciones de simple apoyo en sus extremos por Katsikis et al. [6].	41
Ilustración 9. Viga con condición empotrada en un extremo y libre en otro por Katsikis et al. [6].	42
Ilustración 10. Viga con condición empotrada en un extremo y simplemente apoyada en otro Katsikis et al.[6].	43
Ilustración 11. Deflexión del Problema 1 con función de pérdida desbalanceada.	47
Ilustración 12. Proceso empírico del balanceo de función de pérdida para el Problema 1. .	48
Ilustración 13. Diferentes fases de una molécula durante la minimización de su energía por Roy K et al. [33].	49
Ilustración 14. Calibración del término de disipación de energía.	50
Ilustración 15. Calibración del término de disipación.	51

Ilustración 16. Error en convergencia a la solución real por falta de condiciones de contorno.	61
Ilustración 17. Historial de pérdidas para el problema 1	63
Ilustración 18. Resultados de convergencia de la red para el problema 1	64
Ilustración 19. QQ Plot entre resultado analítico y resultado de la PINN del problema 1	64
Ilustración 20. Historial de pérdidas para el problema 2.	65
Ilustración 21. Variación de resultados del problema 2 con 1000 iteraciones máximas para L-BFGS y 500 iteraciones para Adam.	66
Ilustración 22. Historial de pérdidas para el problema 2	67
Ilustración 23. Resultados del problema 2 con 1500 iteraciones máximas para L-BFGS y 1000 iteraciones para Adam.....	67
Ilustración 24. QQ Plot entre resultado analítico y resultado de la PINN del problema 2.....	68
Ilustración 25. Historial de pérdidas para el problema 3	69
Ilustración 26. Resultados de convergencia de la red para el problema 3 con ajuste de loss = $2E/I$	70
Ilustración 27. Resultados de convergencia de la red para el problema 3 con ajuste de loss = $5E/I$	71
Ilustración 28. Q-Q Plot entre resultado analítico y resultado de la PINN del problema 3....	71

Índice de tablas

Tabla 1. Comparación entre resultados de convergencia correcta de la red con distinta densidad de capas.....	72
Tabla 2. Comparación entre resultados de convergencia correcta de distintas distribuciones de red.	73

Dedicatoria

Dedico este trabajo de tesis de grado a Dios, fuente de sabiduría y guía constante en mi camino académico. Con amor iluminó mi mente y corazón en cada paso de esta travesía, dándome la fortaleza y la perseverancia para superar los desafíos.

También dedico este logro a mi familia, mi madre, mi padre, mi hermana Rosita, mi hermano Stalin, y sobrinos Sammy y Sebastián, quienes han sido mi apoyo inquebrantable a lo largo de este viaje. Su aliento, confianza y sobre todo con su amor me han sostenido en los momentos difíciles y me han inspirado a seguir adelante.

Karen Viviana Sacoto Macías

Dedicatoria

Dedico este trabajo de titulación a mi amada familia, mi papá, hermanos y sobrino, quienes han sido mi fuente de inspiración y apoyo constante a lo largo de toda mi vida. En particular, quiero dedicar este logro a mi querida mamá, Nancy, quien desde que era pequeño ha sido mi ejemplo de determinación y superación. Su amor, dedicación y confianza en mí han sido la fuerza motriz detrás de mis esfuerzos académicos y personales. Este trabajo es un tributo a su inquebrantable apoyo y al papel fundamental que ha desempeñado en mi formación.

Gracias, mamá, por ser mi roca y guía a lo largo de este viaje.

A mis queridas mascotas, Piluca, Tobías, Matilda, Saeta y Lucy, por su constante compañía a lo largo de este viaje. Sus presencias llenas de cariño y alegría han sido un bálsamo reconfortante en los momentos de estudio y reflexión a lo largo de mi carrera.

Pablo Javier Molineros Calle

Agradecimientos

A nuestros tutores, el Ingeniero Esteban Samaniego y el Ingeniero Angel Vázquez, por su orientación experta y apoyo inquebrantable a lo largo de este trabajo de titulación. Su compromiso con nuestro crecimiento académico y profesional ha sido una fuente constante de inspiración, y estamos enormemente agradecidos por la invaluable contribución que han hecho a nuestra formación.

Queremos expresar nuestro profundo agradecimiento a nuestras familias, cuyo apoyo incondicional y amor constante han sido el motor de nuestro éxito académico. Sus sacrificios y aliento nos han dado la fuerza necesaria para alcanzar este logro. Estamos eternamente agradecidos por todo lo que han hecho por nosotros

A nuestras amistades forjadas en la universidad, en especial a Kevin, Alex, Christian, Dayanna, Juan y Sebastián, cuyo apoyo incondicional y las vivencias compartidas desde el inicio de nuestra carrera nos ayudaron a superar los desafíos universitarios y crecer como individuos.

A los distinguidos docentes de la Facultad de Ingeniería de la Universidad de Cuenca, cuyos profundos conocimientos y dedicación a la enseñanza nos brindaron las bases sólidas que nos abrieron el camino necesario para llevar a cabo este proyecto de investigación.

Finalmente, queremos expresar nuestro agradecimiento a todas las personas que de una u otra manera contribuyeron a este logro, su apoyo y aliento fueron fundamentales en nuestro camino hacia la culminación de esta etapa académica

Karen Viviana Sacoto Macías y Pablo Javier Molineros Calle

Introducción

Antecedentes

Los avances recientes en el campo de la ciencia de materiales y la ingeniería han generado la necesidad de crear nuevos materiales heterogéneos con propiedades físicas y mecánicas específicas. En consecuencia, hay una creciente demanda de herramientas más avanzadas con una mayor capacidad para capturar la respuesta física de estos materiales ante cargas externas [1]. En este contexto, se suele recurrir a métodos de aproximación numérica como los elementos finitos (FEM, por sus siglas en inglés) o diferencias finitas, entre otras técnicas comunes.

Para lograr una representación precisa de la respuesta física en diversos sistemas en el ámbito de la ciencia de materiales, resultan esenciales las ecuaciones diferenciales, ya sean ordinarias (EDOs) o en derivadas parciales (EDPs). En circunstancias muy particulares, es viable encontrar soluciones analíticas para estas ecuaciones [2]. Por lo general, se recurre a métodos numéricos para obtener aproximaciones de la solución, dada la complejidad que muchas veces requiere la descripción matemática de los fenómenos reales. En la actualidad, existen numerosos métodos disponibles para resolver tales ecuaciones. Sin embargo, cada uno de estos métodos presenta limitaciones numéricas o depende de la disponibilidad de datos.

Si bien métodos tradicionales como FEM han demostrado ser eficaces en una amplia gama de aplicaciones como en el análisis estructural y la mecánica de sólidos [3], presentan algunas limitaciones importantes. Uno de los aspectos clave es que requieren una discretización espacial del dominio, dividiendo el problema en elementos más pequeños mediante una malla o rejilla [3]. Esta discretización puede ser costosa computacionalmente y puede resultar en soluciones no deseadas.

Un enfoque contemporáneo para abordar ecuaciones diferenciales (ED) en el campo de la mecánica de sólidos es la utilización de aprendizaje automático, mediante Redes Neuronales Informadas por la Física (PINN, por sus siglas en inglés) [4]. Estas redes se entrenan empleando principios físicos y, en algunos casos, se pueden apoyar en datos provenientes de simulaciones o mediciones reales. Su objetivo es aprender a resolver ecuaciones diferenciales específicas. Este enfoque permite resolver de manera eficaz y precisa

problemas complejos en el ámbito de la mecánica de sólidos, lo cual reviste gran importancia en aplicaciones ingenieriles y de investigación.

Las PINNs se presentan como una solución para ecuaciones diferenciales al ofrecer la capacidad de ajustar la red neuronal sin necesidad de datos de entrenamiento. En lugar de ello, se entrena a partir de la minimización de una función que contiene información tanto de la ecuación diferencial como de las condiciones de contorno [5]. Esto se logra con una mínima manipulación en la ecuación original y utilizando condiciones de contorno como datos para el entrenamiento de la red mediante la retro propagación, un tipo de entrenamiento utilizado para resolver diversos problemas complejos en ciencia e ingeniería [6]. Este enfoque permite resolver de manera eficiente problemas de ecuaciones diferenciales en diferentes campos de aplicación.

Algunos autores como *Haghighat et al.* [7] plantean la implementación de PINNs para medios sólidos explorando la elasticidad lineal de los elementos y en *Katsikis et al.* [6] profundizan en el dominio de los elementos estructurales y plantean soluciones a las ecuaciones diferenciales de barras y vigas bajo distintas condiciones de empotramiento (condiciones de contorno) y comparan los resultados de predicción de la red neuronal con la solución analítica en los distintos casos de las vigas y barras, demostrando que el método es viable para aplicaciones similares a la planteada en este trabajo de titulación.

Desde el artículo de *Raissi et al* [4], se ha observado un crecimiento exponencial en el uso de Physics-Informed Neural Networks (PINNs) [8]. Aunque su potencial es prometedor, el campo de los PINNs aún se encuentra en una etapa temprana de investigación y desarrollo. Los desafíos actuales incluyen la selección de la arquitectura de la red neuronal, la representación adecuada de la información física y la mejora de la eficiencia computacional en problemas de gran escala [9]. Resolver estos desafíos permitirá aprovechar plenamente las capacidades de los PINNs.

Problemática

La práctica común de modelar fenómenos a través de enfoques físicos tradicionales se basa en la incorporación de parámetros que buscan representar la realidad, tomando en cuenta factores físicos como la resistencia al movimiento, la gravedad, entre otros [10]. Sin embargo, estos modelos presentan limitaciones en el contexto de las ecuaciones diferenciales, ya que

la representación de fenómenos específicos puede requerir consideraciones locales o globales más complejas en su formulación.

Las ecuaciones diferenciales, ya sean ordinarias o parciales, desempeñan un papel fundamental en la representación y comprensión de fenómenos del mundo real. Estas ecuaciones imponen la necesidad de desarrollar métodos de resolución con características que permitan obtener soluciones que no solo concuerden con observaciones empíricas, sino que también reflejen un sentido físico coherente con el comportamiento natural del fenómeno bajo estudio.

Los métodos de resolución pueden ser tanto numéricos como analíticos. Sin embargo, según el alcance del estudio del fenómeno, la aplicabilidad de los diversos métodos, sobre todo de los analíticos, se ve restringida cuando se requiere un análisis de problemas de manera directa e inversa, cuando la cantidad de datos disponibles que caracterizan el fenómeno es reducida, entre otros [6].

A pesar de los avances en el análisis estructural y la mecánica de sólidos, aún existen desafíos significativos que limitan la precisión y eficiencia de los métodos convencionales. Estos métodos se basan en la discretización espacial del dominio y, en ocasiones, se ven afectados por la complejidad geométrica de los sistemas, los cambios abruptos en las propiedades del material y la dependencia de la calidad de la malla utilizada. Además, el entrenamiento de redes neuronales tradicionales para la resolución de problemas científicos y de ingeniería requiere de grandes volúmenes de datos, los cuales no siempre están disponibles. Esto plantea la necesidad de desarrollar enfoques innovadores que superen estas limitaciones y brinden soluciones viables con la información disponible.

En este sentido, las PINNs se presentan como una alternativa prometedora que combina el poder del aprendizaje automático con la incorporación de restricciones físicas [4]. Las PINNs permiten integrar el conocimiento de las leyes físicas directamente en el proceso de entrenamiento de la red neuronal, lo que resulta en soluciones más precisas y consistentes, incluso cuando los datos disponibles son limitados. A diferencia de los métodos convencionales, las PINNs no requieren una discretización espacial explícita, lo que los hace más flexibles para abordar sistemas con geometrías complejas y cambios abruptos en las propiedades del material.

Potenciales soluciones

Bajo este concepto se plantea la siguiente pregunta de investigación: ¿Qué características tiene el uso de redes neuronales informadas por la física (PINN) en la solución numérica de la ecuación diferencial de una viga, en comparación con los métodos numéricos tradicionales? Cuestionamiento que inicia la investigación debido al gran número de propuestas que existen para solucionar ecuaciones diferenciales.

Uno de los problemas que busca solucionar el presente trabajo es la necesidad de gran cantidad de datos de entrada necesarios para el entrenamiento de una red neuronal convencional, que, si bien es un proceso válido, puede suponer errores en el condicionamiento del sistema en casos donde los datos experimentales son relacionados a un fenómeno que obedece las leyes de la física [10]. De esta manera surge la posibilidad de reducir (o eliminar por completo) la cantidad de datos de entrenamiento cuando se trate de un fenómeno en el dominio de una ecuación física, por lo que las PINNs aparecen como una herramienta útil para este fin.

Justificación

La solución de ecuaciones diferenciales que describen el comportamiento de diversos fenómenos físicos brinda una mayor comprensión y caracterización de estos. Para lograrlo, existen diversos métodos que nos permiten encontrar soluciones a estas ecuaciones, ya sea de manera analítica o numérica.

Este trabajo de titulación utiliza Physics-Informed Neural Networks (PINNs) para resolver la forma energética equivalente al problema de valores de contorno del comportamiento elástico de una viga. Se propone explorar las posibilidades de este método novedoso con miras a que en el futuro se viabilice superar algunas de las limitaciones de los métodos convencionales utilizados en el análisis estructural y la mecánica de sólidos. Este método se basa en el uso de redes neuronales para resolver problemas de valores de contorno, aprovechando la capacidad de estas redes para aprender patrones complejos en los datos y la posibilidad de introducir la caracterización de los fenómenos físicos.

En general, PINNs son una herramienta valiosa en el análisis estructural y simulación numérica, y están ganando cada vez más popularidad en la comunidad científica debido a su capacidad para resolver problemas con condiciones de contorno. Sin embargo, aún se necesita investigar más sobre su rendimiento y precisión en comparación con métodos

tradicionales como FEM, para determinar en qué situaciones específicas es más adecuado utilizar cada uno de ellos. De esta manera, si bien los PINNs son una herramienta poco estudiada, es de interés su desarrollo debido a su potencialidad. En el presente trabajo de titulación, se implementará una Physics-Informed Neural Network para un problema variacional de energía en vigas para validar su método resolutivo en comparación con métodos ya comprobados, en específico con soluciones analíticas.

Objetivos

Objetivo principal

- Implementar una Physics-Informed Neural Network para la resolución de la ecuación diferencial de flexión en una viga.

Objetivos secundarios

- Elaborar una aproximación al modelo matemático de la ecuación diferencial de la viga
- Elaborar un modelo numérico tomado como base de la aplicación de PINNs para la resolución de ecuaciones diferenciales que caracterizan el comportamiento elástico de vigas.
- Implementar una solución a la ecuación diferencial de una viga con condiciones de contorno específicas bajo un planteamiento variacional.
- Validar los resultados obtenidos a partir de los PINNs mediante métodos comprobados.

Resumen del contenido

Capítulo 2 Formulación matemática

En este capítulo se analiza la formulación matemática de vigas bajo la teoría de Euler-Bernoulli. Se revisa la aproximación de la ecuación de gobierno desde la forma fuerte que hace referencia al problema de valores de contorno y desde su forma equivalente que es el planteamiento variacional.

Capítulo 3 Métodos de resolución de ecuaciones diferenciales

En este capítulo, se realiza una revisión de los métodos numéricos más destacados utilizados para la resolución de ecuaciones diferenciales. Se aborda el enfoque del método de elementos finitos, ampliamente empleado en la solución de problemas en los ámbitos científicos e ingenieriles, debido a su reconocida validez y rigurosidad. Sin embargo, a pesar de sus virtudes, este método presenta limitaciones particulares que pueden ser superadas mediante la exploración y aplicación de nuevas metodologías. Por el objetivo del presente trabajo de titulación se examina especialmente el potencial de las Redes Neuronales Informadas por la Física como alternativa, subrayando su capacidad para abordar estas limitaciones. Durante este análisis, se presta especial atención a las características esenciales de estas redes neuronales, con un enfoque particular en la función de pérdida, que desempeña un rol central en el funcionamiento integral de la red.

Capítulo 4 Implementación

El capítulo de implementación del presente trabajo de titulación se centra en la aplicación práctica de los conceptos de redes neuronales informadas por la física (PINN, por sus siglas en inglés), los cuales son traducidos a la práctica mediante la adaptación del algoritmo desarrollado por Federico Magnani. Este capítulo se estructura en una serie de subcapítulos, cada uno de los cuales aborda un aspecto esencial del proceso:

Composición del Sistema: Se destaca la importancia de la resolución en términos de un enfoque variacional para abordar los diferentes casos de problemas de vigas.

Características Geométricas y Mecánicas: Se profundiza en las propiedades geométricas y mecánicas que influyen en la respuesta de las vigas ante los esfuerzos. Además, se detallan las condiciones de apoyo que son esenciales para la implementación exitosa del algoritmo PINN. Los problemas específicos, como las vigas simplemente apoyadas, empotradas-libres y empotradas-simplemente apoyadas, son analizados en términos de restricciones a desplazamiento, giro y curvatura. Estas restricciones son incorporadas para obtener ecuaciones exactas de deflexión mediante métodos de resolución matemática.

Modelado del Sistema: Se aborda la construcción de la función de pérdida para las vigas, incluyendo las consideraciones específicas sobre los escalares que ponderan los términos de penalización. Se explora el término de disipación añadido a la función de pérdida y se proporciona una explicación detallada de su inclusión.

Implementación de PINN: Se examinan con detalle los hiperparámetros asociados a la red neuronal, considerando tanto las directrices físicas como las recomendaciones de literatura relevantes para garantizar una convergencia efectiva de la red hacia respuestas precisas.

Aplicación de Problemas: Basándose en las recomendaciones para los hiperparámetros, se presentan los puntos cruciales abordados durante la implementación. Se pone especial énfasis en la función de pérdida, identificando un caso particular donde se requieren mayor cantidad de restricciones.

Capítulo 2: Formulación matemática

Mecánica de sólidos

La mecánica de sólidos es una rama de la física que se ocupa del estudio de la deformación y el movimiento de los cuerpos sólidos bajo la aplicación de cargas externas y de los fenómenos internos inducidos por estas [11]. Las leyes básicas de movimiento que describen el equilibrio entre las fuerzas externas y las fuerzas internas causadas, y las teorías que caracterizan el comportamiento mecánico de los materiales son principios de la teoría básica de física generalmente aplicados en el estudio de cuerpos sólidos. El comportamiento de los sólidos ante cargas externas tales como fuerzas, desplazamientos, aceleraciones, así como ante los efectos producidos como la fuerza de inercia en los cuerpos, cambios térmicos, etc., constituye el objetivo de estudio de la mecánica de sólidos. Por tanto, la mecánica de sólidos estudia el comportamiento de cuerpos sólidos mediante principios físicos básicos.

La mecánica de sólidos permite el estudio de la elasticidad en cuerpos sólidos deformables en base a las deformaciones y esfuerzos generados ante la acción de cargas externas. La elasticidad es una propiedad que permite a los sólidos recuperar su forma original después de retirar la carga aplicada, es decir, cuando un sólido se deforma, adquiere una mayor energía potencial elástica y aumenta su energía interna, sin que se produzcan transformaciones termodinámicas irreversibles ni presente cambios permanentes. Entonces, el comportamiento del cuerpo sólido elástico deformable responde a las postulaciones de la teoría elástica [12].

En el campo de la ingeniería civil, se lleva a cabo el diseño, la construcción y el mantenimiento de estructuras. Para lograrlo, es necesario aplicar los principios proporcionados por la mecánica de sólidos deformables. Las vigas son elementos de las estructuras que son

analizados principalmente por su papel en resistencia y estabilidad en las estructuras ante variedad de cargas. En el presente trabajo de titulación se analiza el comportamiento de estas bajo la teoría elástica lineal.

Aproximación de la ecuación de gobierno

Dentro del campo estructural, las ecuaciones diferenciales desempeñan un papel fundamental al describir el comportamiento elástico de los cuerpos deformables. En el caso específico del comportamiento elástico lineal de una viga bajo la teoría de Euler-Bernoulli, se representa mediante la ecuación de gobierno, que es una ecuación diferencial ordinaria de cuarto orden. Por tanto, la ecuación de gobierno a analizar es una ecuación diferencial ordinaria.

La ecuación de gobierno de un sistema puede obtenerse a partir de dos métodos, bajo la condición de equilibrio y mediante el método energético. El primer método considera el equilibrio de fuerzas y momentos en un volumen de control del sistema, mientras que, el segundo método contempla la energía potencial del sistema como funcional y en función a las variaciones de este se establece las ecuaciones Euler-Lagrange correspondientes [13].

El problema de valores de contorno (BVP, por sus siglas en inglés) constituye una formulación fuerte para describir problemas en el ámbito ingenieril. Su propósito está en determinar la solución de una ecuación diferencial de gobierno, ya sea ordinaria o parcial, sujeta a condiciones de contorno. Estas condiciones de contorno pueden ser de tipo Dirichlet, que especifican el valor de la solución en puntos específicos de la región de interés, o de tipo Neumann, que establecen las derivadas de la solución en dichos puntos. El análisis de los BVP permite abordar de manera precisa y sistemática los problemas de ingeniería [14].

La aplicación de principios variacionales de la mecánica de sólidos y mecánica estructural permite plantear las ecuaciones que describen el comportamiento de elementos estructurales [15]. En este tipo de planteamiento, la solución se obtiene al minimizar el funcional correspondiente al sistema. Al emplear esta formulación, se puede abordar el análisis y diseño de estructuras, permitiendo obtener soluciones precisas para problemas en ingeniería.

La solución tanto del problema variacional como del problema de valores de contorno son matemáticamente equivalentes. A continuación, se lleva a cabo el planteamiento del problema de valores de contorno, el problema variacional y se demuestra la equivalencia de las soluciones.

Problema de valores de contorno

Una viga es un elemento estructural esbelto que puede estar sujeto a compresión, tracción y flexión, diseñado de tal forma que resista en su mayoría cargas transversales. Como respuesta a dichas sollicitaciones, la viga se deforma por flexión, y es analizada mediante la teoría elástica denominada “teoría de vigas de Euler-Bernoulli”. A continuación, se desarrolla el planteamiento del problema de valores de contorno, es decir, el planteamiento de la ecuación diferencial junto con las condiciones de contorno apropiadas.

La teoría de vigas de Euler-Bernoulli se formula en base al sistema de coordenadas Cartesiano (Ilustración 1), en el cual el eje “x” coincide con el eje longitudinal de la viga, el eje “z” se considera en la dirección de la altura de la viga, y el eje “y” se toma a lo ancho de la viga.

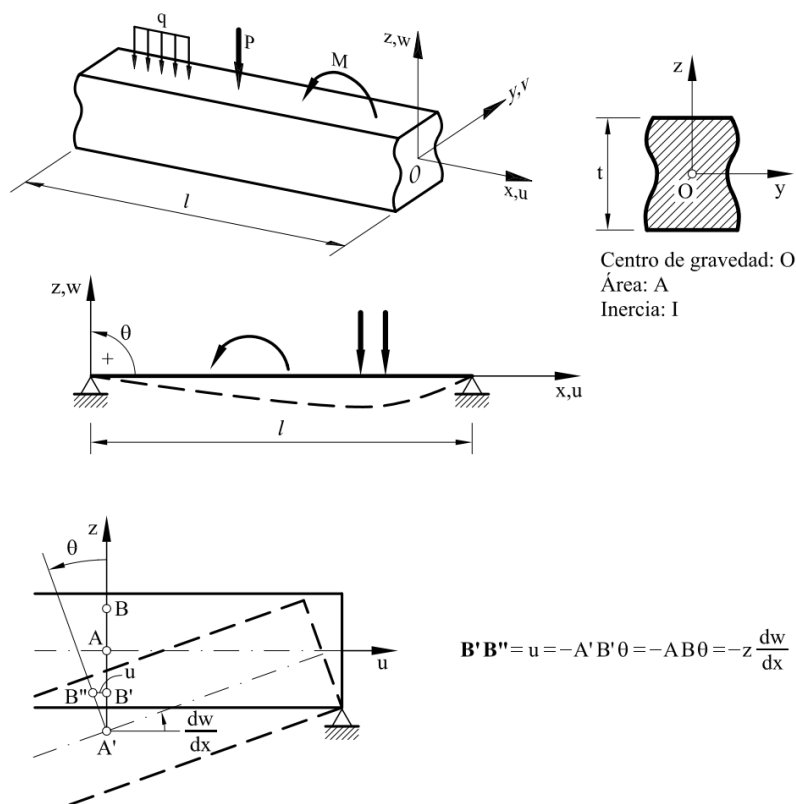


Ilustración 1. Hipótesis de la viga bajo la teoría Euler Bernoulli [16].

Como se citó en Wang et al. [11] esta teoría se basa en las siguientes tres hipótesis cinemáticas:

1. La sección transversal de la viga se considera como una superficie rígida que solo gira, puesto que no cambia significativamente la forma ni la geometría de la sección bajo cargas aplicadas.
2. Después de la deformación se considera que la sección transversal de la viga permanece plana y ortogonal al eje deformado de la viga.
3. La deformación angular del eje neutro de la viga es pequeña.

A partir de las hipótesis antes expuestas, se establecen las expresiones de desplazamiento “u”, “v”, y “w” (1) de un punto ubicado arbitrariamente (x, y, z) en dirección de los ejes “x”, “y”, y “z” respectivamente. Además, “u₀” y “w₀” representan los desplazamientos en el punto (x,0,0).

$$u(x, y, z) = u_0(x) - z \sin \theta \quad (1)$$

$$v(x, y, z) = 0$$

$$w(x, y, z) = w_0(x) + z \cos \theta - z$$

En base a la hipótesis la cual considera que existen pequeñas deformaciones, se tiene que:

$$\sin \theta \approx \theta, \quad \cos \theta \approx 1 \quad (2)$$

Así también, se expresa la rotación (θ) como:

$$\theta = \frac{dw}{dx} \quad (3)$$

Entonces, considerando las expresiones (2) y (3), las formulaciones de desplazamiento (1) pueden reescribirse de la siguiente manera:

$$u(x, y, z) = u_0(x) - z \cdot \frac{dw}{dx} \quad (4)$$

$$v(x, y, z) = 0$$

$$w(x, y, z) = w_0(x)$$

La ecuación que caracteriza la teoría de vigas de Euler-Bernoulli resulta de la combinación de la expresión cinemática (3), las expresiones de desplazamiento (4), y las ecuaciones de gobierno para las respuestas lineales de los sólidos elásticos. Esta ecuación juntamente con las condiciones de contorno comprende el problema de valores de contorno conocida como la forma fuerte.

Bajo la teoría de elasticidad lineal, se hacen varias suposiciones fundamentales, que incluyen la consideración de deformaciones pequeñas, la existencia de un estado neutro y un proceso de deformación isotérmica y adiabático. En este contexto, un problema elástico lineal se describe mediante un conjunto de ecuaciones para determinar los desplazamientos, deformaciones y las tensiones. Dentro de esas ecuaciones tenemos a la ecuación de Cauchy, ecuación constitutiva, ecuación geométrica junto con condiciones de contorno y condiciones iniciales. La solución de estos se puede llevar a cabo desde el enfoque de los desplazamientos y desde el enfoque de tensiones. En el presente trabajo de titulación, la formulación del problema de la viga de Euler-Bernoulli, se lleva a cabo desde el enfoque de desplazamientos [12].

A continuación, de acuerdo con *Wang et al.* [11] se presentan las ecuaciones de gobierno en una viga con comportamiento elástico:

1. Relación deformación-desplazamiento

La deformación (ε) es la variación del desplazamiento y/o ángulo por unidad de longitud. De forma matricial, la deformación es:

$$\varepsilon = \mathbf{L} \cdot \mathbf{u} \quad (5)$$

Donde:

L = operador diferencial matricial.

\mathbf{u} = vector de desplazamientos.

Para la viga definida en la Ilustración 1, tenemos como deformación:

$$\varepsilon_{xx} = \frac{\partial u}{\partial x} = \frac{du_0}{dx} - z \cdot \frac{d^2 w}{dx^2} \quad (6)$$

$$\varepsilon_{yy} = \frac{\partial v}{\partial x} = 0$$

$$\varepsilon_{zz} = \frac{\partial w(x)}{\partial x} = 0$$

$$\varepsilon_{xy} = \frac{1}{2} \left(\frac{du}{dy} + \frac{dv}{dx} \right) = 0$$

$$\varepsilon_{yz} = \frac{1}{2} \left(\frac{dv}{dz} + \frac{dw}{dw} \right) = 0$$

$$\varepsilon_{zx} = \frac{1}{2} \left(\frac{dw}{dx} + \frac{du}{dz} \right) = 0$$

En base a las hipótesis analizadas anteriormente, en la expresión (6) la única componente del tensor de deformación que no es igual a cero es la que se encuentra en la dirección del eje longitudinal de la viga. En esta, el primer término representa la deformación resultante debido a la extensión uniforme de toda la sección transversal, mientras que, el segundo término considera la contribución de flexión e introduce la definición de la curvatura del eje de la viga (7).

$$\kappa = \frac{d^2 w}{dx^2} \quad (7)$$

2. Relación constitutiva

Las relaciones constitutivas describen el comportamiento mecánico de los materiales en un cuerpo continuo. En general, para materiales elásticos isótropos la relación constitutiva se conoce como ley de Hooke generalizada (8), que establece la relación entre esfuerzo y la

deformación en un sólido deformado. Los esfuerzos representan la interacción interior en medios continuos causado por fuerzas externas distribuidas en todo el volumen o en la frontera, o concentradas en varios lugares.

Considerando un elemento cúbico muy pequeño del cuerpo (Ilustración 2), se analizan las componentes normales (σ) como cortantes (τ) del esfuerzo actuante. Las componentes actuantes en un punto son seis en total, tres componentes normales ($\sigma_x, \sigma_y, \sigma_z$) y tres componentes cortantes ($\tau_{xy}, \tau_{yz}, \tau_{zx}$) considerando equilibrio.

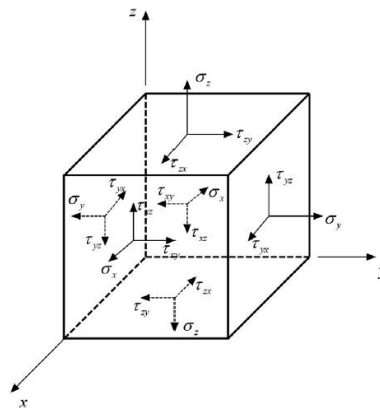


FIGURE 2.3 Positive stress components on a small cubic element.

Ilustración 2. Componentes de tensión positiva en un pequeño elemento cúbico Wang et al. [11].

$$\sigma = \mathbf{D} \cdot \epsilon \tag{8}$$

Donde:

\mathbf{D} = matriz de coeficientes elásticos y/o de rigidez referidos a las coordenadas del material.

ϵ = componentes de deformación.

Sin embargo, para la relación constitutiva unidireccional para una viga de material lineal isotrópico la relación entre esfuerzo y la deformación está dada mediante el módulo elástico de Young. El esfuerzo considerado corresponde al actuante a lo largo del eje de la viga (9):

$$\sigma_x = E \cdot \epsilon_{xx} \tag{9}$$

Donde:

E = módulo de elasticidad de Young.

ϵ_{xx} = deformación en dirección del eje longitudinal de la viga.

El módulo de elasticidad de Young puede variar a lo largo del eje longitudinal de la viga para denotar el cambio de los materiales.

3. Esfuerzos resultantes

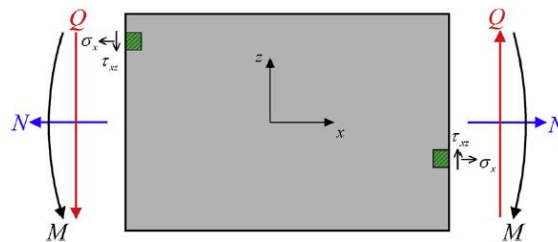


FIGURE 2.12 Positive sign conventions for axial force, moment, and shear.

Ilustración 3. Esfuerzos resultantes (sistema equivalente de fuerzas) en un diferencial de viga por Wang et al. [11].

Los esfuerzos resultantes son analizados como un sistema equivalente de fuerzas (Ilustración 3). Estas fuerzas representan los esfuerzos internos actuantes en la sección transversal de la viga. Entonces, para la presente viga (Ilustración 1), la fuerza axial (N), momento flector (M) y fuerza cortante (V), tenemos (10):

$$N = \int_A \sigma_x dA \tag{10}$$

$$M = \int_A z \sigma_x dA$$

$$V = \int_A \tau_{xz} dA$$

Considerando la expresión de esfuerzo (9), reescribimos las expresiones (10):

$$N = EA \frac{du_0}{dx} \tag{11}$$

$$M = E \frac{du_0}{dx} \int_A z dA - E \frac{d^2w_0}{d^2x} \int_A z^2 dA = EI \frac{d^2w_0}{d^2x}$$

$$V = \int_A \tau_{xz} dA = 0$$

En (11) se obtiene que las fuerzas de corte son nulas en la sección transversal, lo que corrobora que el efecto de las fuerzas de corte en la deformación de la viga es mínimo.

4. Ecuaciones de equilibrio

Las ecuaciones de equilibrio describen la relación entre esfuerzos y fuerzas externas. En la Ilustración 4, se considera un segmento de viga sometida a fuerzas cortantes ($q(x)$) y fuerzas en dirección del eje longitudinal ($f(x)$).

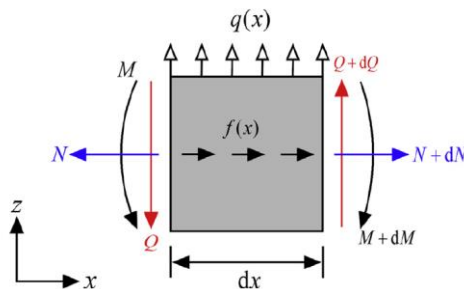


FIGURE 2.13 Equilibrium of a beam element subjected to axial and transverse loads.

Ilustración 4. Equilibrio de un elemento de viga sometido a cargas axiales y transversales por Wang et al. [11].

Como resultado, el equilibrio de fuerzas en el eje “x” y el eje “z” es:

$$\sum F_x = 0: \quad \frac{d}{dx} N(x) + f(x) = 0 \tag{12}$$

$$\sum F_z = 0: \quad \frac{d}{dx} Q(x) + q(x) = 0$$

Sustituyendo las ecuaciones constitutivas (11) en las ecuaciones de equilibrio de fuerza (12), se obtiene:

$$\frac{d}{dx} \left(EA \frac{du_0}{dx} \right) + f(x) = 0 \quad (13)$$

$$\frac{d^2}{dx^2} \left(EI \frac{d^2 w_0}{dx^2} \right) = q(x)$$

Se expande la segunda expresión dada en la ecuación (13), se obtiene:

$$\frac{d^2 EI_y}{dx^2} \frac{d^2 w_0}{dx^2} + 2 \frac{d^2 EI_y}{dx} \frac{d^3 w_0}{dx^3} + EI_y \frac{d^4 w_0}{dx^4} = q(x) \quad (14)$$

Considerando que el material (E) y que el momento de inercia (I_y) son constantes, la ecuación diferencial de gobierno de la teoría de vigas de Euler-Bernoulli obtenida a partir de (14) es:

$$EI_y \frac{d^4 w}{dx^4} = q(x) \quad 0 < x < L \quad (15)$$

En la expresión (15), el símbolo w_0 se sustituye por w puesto que es la solución buscada cuatro veces diferenciable donde el dominio está limitado por la longitud de la viga.

Para completar la formulación fuerte del problema, es decir el problema de valores de contorno (BVP), se definen las condiciones de contorno. Por lo que, la solución de un problema elástico lineal es única siempre que se consideren las condiciones de contorno apropiadas de tal forma que se eviten movimientos indeseados. Las condiciones de contorno limitan el espacio de soluciones para la ecuación (15) [11].

La ecuación diferencial (15) es de tipo lineal de cuarto orden, por lo que para ser resuelta requiere cuatro condiciones de contorno, dos condiciones en cada extremo. Las condiciones pueden ser de tipo esenciales (Dirichlet) o naturales (Neumann). Las condiciones esenciales corresponden al desplazamiento transversal (w) y giro (θ), mientras que las naturales, corresponden a la fuerza cortante (V) y al momento flector (M), es decir, corresponden a las condiciones cinemáticas y a las condiciones estáticas respectivamente.

De manera que, las condiciones de contorno especificadas pueden ser: todas de tipo esencial, todas de tipo natural, o algunas condiciones esenciales y otras naturales. En el último caso, una forma de definirla se basa en escoger un elemento de cada par conjugado (w_0, V) y $(\frac{dw_0}{dx}, M)$ como condición de contorno.

Las condiciones de contorno asociadas a una viga de Euler-Bernoulli, dependerán del estado de las restricciones de la viga. Dentro de los casos típicos se tiene vigas con extremos empotrados, apoyados o libres. A continuación, se detallan las condiciones de contorno correspondientes a los casos típicos mencionados (Ilustración 5).

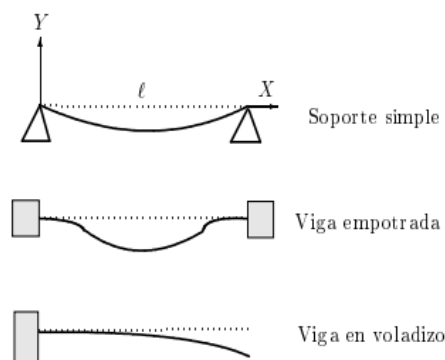


Ilustración 5. Casos típicos de vigas [11].

- Viga simplemente apoyada

En los extremos los desplazamientos transversales (w) y los momentos flectores (M) son conocidos y son nulos. Mientras que, las fuerzas cortantes (V) y giros (θ) son desconocidos.

- Viga empotrada

En los extremos los desplazamientos transversales (w) y los giros (θ) son nulos, y las fuerzas cortantes (V) y los momentos flectores (M) son desconocidos.

- Viga libre

En el extremo en $L=0$, el desplazamiento transversal (w) y el giro (θ) son nulos, mientras que, momento flector (M) y la fuerza cortante (V) se especifican. En el extremo en $L=L$, el desplazamiento transversal (w) y el giro (θ) son desconocidos, mientras que, momento flector (M) y la fuerza cortante (V) son nulos.

Problema variacional

Los métodos variacionales mediante sus correspondientes ecuaciones satisfacen los requerimientos estructurales que generalmente son representados mediante ecuaciones diferenciales. Los requerimientos son tales como, condiciones de equilibrio, condiciones de compatibilidad de deformaciones, y condiciones de contorno. El objetivo de los problemas variacionales consiste en determinar la función dependiente que permite que un determinado funcional tenga un valor estable ya sea un máximo o mínimo. El funcional es equivalente a la ecuación diferencial que describe el comportamiento mecánico del sistema [17]. Por tanto, la formulación variacional junto con las condiciones de contorno esenciales brinda una solución equivalente a la obtenida a partir del problema de valores de contorno (BVP).

En mecánica de sólidos y mecánica estructural, un funcional comprende la energía total de un sistema [15]. En el presente caso, la energía de la viga bajo carga uniformemente distribuida analizada bajo la teoría de Euler-Bernoulli (Ilustración 1). La energía potencial total (16) está comprendida por la energía interna del sistema (U) y el trabajo realizado por las fuerzas externas (V).

$$\Pi_{viga} = U + V \quad (16)$$

$$\Pi_{viga} = \int_0^L \frac{EI_y}{2} \left(\frac{d^2w}{dx^2} \right)^2 dx - \int_0^L wq dx$$

La expresión (15) y (16) tiene como datos al módulo de elasticidad (E), al momento de inercia (I_y) y a la carga transversal distribuida (q), mientras que, la deflexión transversal ($w = w(x)$) es la variable dependiente por obtener. Esta última debe ser tal que minimice el funcional para lo cual requiere ser lo suficientemente diferenciable y satisfacer las condiciones de contorno de Dirichlet.

Equivalencia de soluciones

Según *J. N. Ready* [15], la equivalencia de la solución de las expresiones (15) y (16) bajo las respectivas condiciones de contorno se verifica, ya que a partir del planteamiento variacional (16) se puede obtener la ecuación diferencial base, y viceversa. En el primer caso, para obtener el problema de valores de contorno, es decir la expresión diferencial y las condiciones de contorno, se requiere que el funcional (16) sea el mínimo considerando las condiciones de

contorno geométricas. Para lo que se requiere hallar la primera variación de este que corresponde al principio de mínima energía potencial.

Lo mencionado anteriormente respalda el principio utilizado por *Samaniego et al.* [14], el cual asume que la solución tanto del problema variacional como del problema de valores de contorno son iguales, es decir, la energía total del sistema es tal que existe una solución única del problema variacional que es igual que la solución de ecuación de Euler-Lagrange definida como la expresión en la forma fuerte.

Capítulo 3: Métodos de resolución de ecuaciones diferenciales

Métodos de resolución convencionales

Métodos analíticos y métodos numéricos convencionales

Las ecuaciones diferenciales, ya sean ordinarias (EDOs) o en derivadas parciales (EDPs), desempeñan un papel fundamental en la modelización matemática de numerosos fenómenos físicos [18]. En situaciones excepcionales, es posible encontrar soluciones analíticas para estas ecuaciones. Sin embargo, en la mayoría de los casos, se recurre a métodos numéricos para obtener aproximaciones de las soluciones debido a la dificultad e incluso a la imposibilidad de obtener soluciones a estas ecuaciones en la ciencia e ingeniería [19]. En la actualidad, existen una amplia gama de métodos disponibles para resolver estas ecuaciones.

Un aspecto fundamental de los métodos analíticos es su enfoque en la manipulación algebraica y el uso de ecuaciones matemáticas para obtener soluciones exactas. Si bien estos métodos son aplicables en una gran variedad de ecuaciones diferenciales, hay una gran cantidad de ecuaciones en las cuales las soluciones no se pueden obtener mediante métodos analíticos [20].

Los métodos analíticos son capaces de obtener soluciones exactas a partir de la ecuación de gobierno del sistema, que describe el comportamiento físico y establece la relación entre las variables. Estos métodos utilizan técnicas como la separación de variables para descomponer la ecuación en ecuaciones más simples y encontrar soluciones exactas. Además, se apoyan en las condiciones de contorno, restricciones impuestas en los límites del dominio, para determinar la solución exacta del sistema y comprender los fenómenos físicos subyacentes.

No obstante, es importante tener en cuenta que la aplicabilidad de los métodos analíticos puede verse limitada en casos de geometrías complejas o ecuaciones diferenciales no lineales. En situaciones donde no se cumplen las condiciones de simetría y linealidad, o cuando el sistema presenta geometrías más complejas, puede resultar difícil o incluso imposible obtener soluciones exactas mediante los métodos de resolución analíticos [19].

Por otro lado, algunos métodos numéricos se basan en la discretización del dominio y la aproximación de las soluciones mediante cálculos numéricos. Estos métodos, como el método de elementos finitos y el método de diferencias finitas son los métodos más estandarizados en el medio, estos métodos subdividen el dominio en elementos más pequeños y aproximan la solución en cada discretización [3]. Los métodos numéricos son las resoluciones más versátiles y pueden adaptarse a diferentes tipos de problemas y geometrías, incluso cuando no se cumplen las condiciones de simetría y linealidad. Sin embargo, las soluciones obtenidas son aproximadas y pueden requerir una mayor capacidad computacional en comparación con los métodos analíticos cuando se requiere una respuesta con una exactitud similar entre ambos métodos.

Physics-Informed Neural Networks (PINNs)

Origen y funcionamiento

Las redes neuronales

Una red neuronal artificial (ANN, por sus siglas en inglés) es un sistema que emula el funcionamiento de una red neuronal biológica, donde se intercambia información entre las unidades de procesamiento llamadas neuronas. En el caso de una red neuronal artificial, se adapta el sistema de manera que se convierte en un potente procesador distribuido en paralelo, con una capacidad inherente para el almacenamiento experimental del conocimiento y disponibilidad para su uso. Este sistema se asemeja al funcionamiento del cerebro humano en dos aspectos fundamentales: el aprendizaje como medio para adquirir "conocimiento" y el uso de pesos sinápticos, que determinan la fuerza de conexión entre las neuronas, para almacenar la información obtenida [21].

Las redes neuronales son una herramienta poderosa para resolver una amplia gama de problemas numéricos. En un proceso de entrenamiento convencional de una ANN, cada neurona recibe entradas, realiza cálculos utilizando funciones de activación y genera una

salida que se propaga a través de la red. Durante el entrenamiento, las redes neuronales ajustan los pesos de las conexiones entre las neuronas utilizando datos de entrada y salidas esperadas. Este ajuste de pesos les permite aprender patrones y realizar predicciones. No obstante, es importante tener en cuenta que el proceso de entrenamiento, en el cual la red aprende a identificar patrones en los datos recibidos, tiende a volverse más largo y computacionalmente intensivo debido a las operaciones involucradas y la necesidad de contar con una gran cantidad de datos de entrenamiento [22].

La exactitud de los resultados de una red neuronal se ve influenciada por dos factores clave: la cantidad de información utilizada en el entrenamiento y la precisión de las mediciones con relación al fenómeno real. Una mayor cantidad de datos disponibles durante el entrenamiento brinda a la red una mayor capacidad para obtener resultados esperados. Además, si las mediciones utilizadas en el entrenamiento se acercan al comportamiento esperado del fenómeno, la cantidad de datos necesarios para el entrenamiento puede reducirse significativamente [10].

No obstante, para lograr un entrenamiento adecuado de una red neuronal, se considera ideal un enfoque supervisado utilizando información perfecta en todo el dominio de interés. Sin embargo, en la práctica, es poco común tener acceso a datos perfectos en dicho dominio. En lugar de eso, nos enfrentamos a datos ruidosos, fragmentados e incompletos. Si se emplea un enfoque de entrenamiento supervisado tradicional, es probable obtener modelos indeseables [10].

Physics-Informed Neural Networks

En la mayoría de los casos prácticos de las redes neuronales convencionales, se puede afirmar que la red desarrollada no toma en cuenta modelos matemáticos subyacentes que expresen leyes físicas. Consecuentemente, se obtiene un algoritmo que desconoce la matemática, ya que el entrenamiento de la red no requiere ningún conocimiento previo sobre dicho modelo matemático [23].

Al utilizar ecuaciones diferenciales como restricciones en la arquitectura de la red neuronal, se puede mejorar la capacidad de la red para predecir fenómenos físicos de manera más precisa [4]. Al incorporar conocimiento físico en las redes neuronales, se aprovecha la poderosa capacidad de estas redes para aprender patrones complejos y obtener resultados más fiables y apegados a los fenómenos físicos en aplicaciones, especialmente en campos como la ingeniería civil: la mecánica de fluidos y la mecánica de materiales [24]. Resoluciones

exitosas a problemas de mecánica de fluidos y a ecuaciones diferenciales de alto orden fueron resueltas en [25] y [26].

En el contexto de la resolución de ecuaciones diferenciales y la mecánica de sólidos, han surgido las Redes Neuronales Informadas por la Física (PINNs por sus siglas en inglés). Las PINNs son una variante de las redes neuronales que combinan la capacidad de aprendizaje de las redes con la información física del problema. A diferencia de los métodos numéricos tradicionales que requieren una discretización espacial explícita, los PINNs tienen la capacidad de resolver ecuaciones diferenciales sin necesidad de una malla o rejilla [9]. En su lugar, los PINNs utilizan la información física del problema, como las ecuaciones de gobierno y las condiciones de contorno, para guiar el aprendizaje de la red neuronal y obtener soluciones aproximadas del fenómeno.

Una distinción fundamental entre un Physics-Informed Neural Network (PINN) y una red neuronal convencional reside en su enfoque de entrenamiento. Aunque el machine learning ha surgido como una opción prometedora, el entrenamiento de redes neuronales profundas demanda conjuntos de datos extensos, los cuales no siempre están disponibles en contextos científicos. Por el contrario, las PINNs para su entrenamiento utilizan información adicional obtenida al incorporar las leyes físicas pertinentes al problema en cuestión [5]. Esta estrategia permite aprovechar el conocimiento físico existente para guiar el aprendizaje de la red neuronal, superando así la limitación de datos masivos requeridos por los enfoques convencionales de redes neuronales.

El funcionamiento en general de las PINNs se basa en que la red neuronal construya un aproximador de la solución de la ecuación diferencial planteada utilizando diferenciación automática en los operadores no lineales que describen la ecuación diferencial. Además, la red neuronal debe aprender el sistema de solución ajustando los pesos de manera que se minimice una función de pérdida objetivo, compuesta en la mayoría de los casos por los términos de la ecuación diferencial y las condiciones de contorno impuestas. [27].

Las PINNs han demostrado ser una herramienta prometedora para resolver tanto la forma fuerte como la forma débil de las ecuaciones diferenciales. En la forma fuerte, las PINNs permiten estimar directamente las soluciones de las ecuaciones diferenciales considerando el problema de valores de contorno. También las PINNs utilizan el enfoque variacional para aproximar las soluciones mediante la minimización de un funcional como función de pérdida.

Estas metodologías son comparadas en el artículo de Li W et al. [28] como dos tipos de función de pérdida, PDE-based y energy based respectivamente.

Las PINNs poseen la singularidad de abordar tanto problemas directos como inversos. En los problemas directos, estas redes predicen las relaciones entre variables conocidas, como las deflexiones en vigas. Por otro lado, en los problemas inversos, emplean datos observados para estimar condiciones desconocidas (como Inferir propiedades de materiales en materiales funcionales), aprovechando su capacidad para incorporar información física durante el entrenamiento [5]. Esta versatilidad amplía su aplicabilidad en una variedad de campos científicos y de ingeniería.

Funcionamiento y elementos de la Physics-Informed Neural Networks

Una PINN utiliza los datos de entrada, que representan el dominio espacial discretizado, y técnicas de diferenciación automática para generar aproximaciones de los términos presentes en las ecuaciones diferenciales. Además, se emplean condiciones iniciales y de contorno del modelo investigado. En las PINNs si bien se utiliza un enfoque alternativo que permite obtener aproximaciones directas e inversas, se emplean elementos de una red neuronal tradicional como: funciones de activación, nodos, capas ocultas, entre otros elementos de las redes neuronales [9].

En la Ilustración 6 [8] se observa la estructura principal de la PINN incluyendo sus elementos principales, donde para la arquitectura de la red neuronal, las variables de entrada x , y , y t se utilizan para calcular las variables de salida u y v . Estas variables se utilizan para medir las desviaciones de las leyes físicas durante el entrenamiento, asegurando que la red generalice bien para entradas desconocidas. La pérdida física permite que las variables de salida se entrenen para incluir sus derivadas y considerar regiones locales alrededor de los valores de entrada [8].

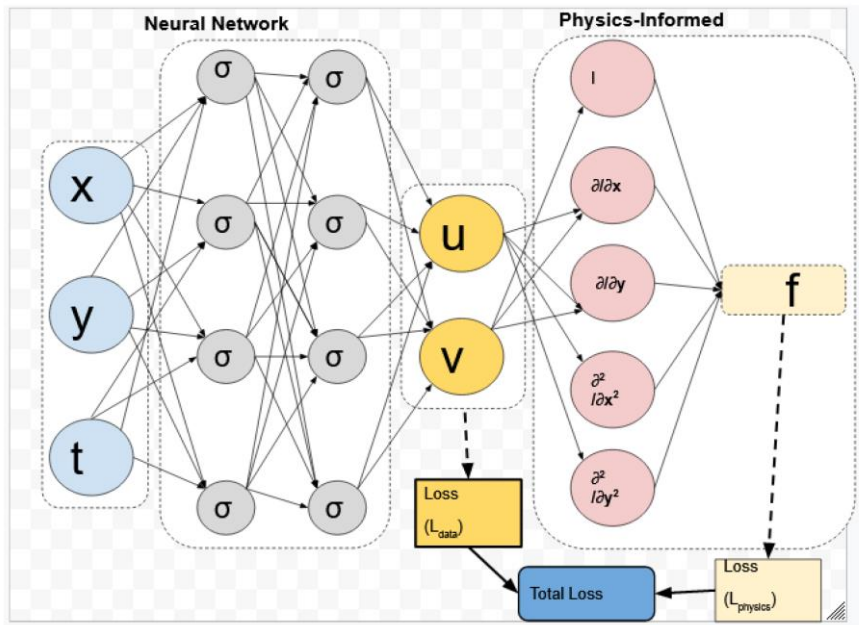


Ilustración 6. Esquema de los elementos de una red neuronal informada por la física (PINN) por Lawal et al. [8].

Los datos iniciales para el entrenamiento de la red, también conocidos como puntos de colocación, juegan un papel crucial en las PINNs. A pesar de que el algoritmo permite utilizar una pequeña cantidad de datos iniciales para entrenar el sistema, los resultados obtenidos llegan a ser válidos gracias al modelo matemático que incorpora las ecuaciones diferenciales, las condiciones iniciales y de contorno del fenómeno.

Los puntos de colocación pueden seguir diferentes distribuciones, como uniforme (espaciados uniformemente en el dominio de discretización), pseudoaleatoria (creados mediante un generador de números pseudoaleatorios) y Sobol (obtenidos a partir de la secuencia de baja discrepancia de Sobol) que es similar a los métodos de cuasi-Monte Carlo [29]. Según Markidis [29], se ha observado que la distribución de los puntos de colocación no tiene un efecto significativo en el error de entrenamiento cuando se utilizan conjuntos de datos extensos. Li et al. en [13] para funciones de pérdida basadas en métodos energéticos utiliza el muestreo aleatorio en distribuciones uniformes en cada época de entrenamiento, lo que significa que el conjunto de datos (puntos de colocación) varía durante todo el entrenamiento, para obtener resultados más acertados y consistentes.

Una resolución óptima de ecuaciones diferenciales utilizando PINNs requiere la selección adecuada de los hiperparámetros asociados a la red neuronal. Estos hiperparámetros

incluyen el optimizador de la función de pérdida, la arquitectura de la red, la función de activación y la inicialización de los pesos, como el método de Glorot o Xavier normal, entre otros. Cada uno de estos hiperparámetros desempeña un papel importante en el proceso de entrenamiento de la red y puede afectar significativamente el rendimiento y la precisión de las soluciones obtenidas [27].

A pesar de la aparente simplicidad conceptual de las PINNs, el problema de aprendizaje resultante es un desafío tanto desde el punto de vista teórico como práctico. La selección adecuada de los hiperparámetros mencionados anteriormente se vuelve crucial para obtener resultados confiables y precisos. La investigación de *Wang et al.* [30] destaca la complejidad inherente en el proceso de aprendizaje de las PINNs y subraya la importancia de abordar de manera rigurosa tanto los aspectos teóricos como los desafíos prácticos asociados a su implementación.

Función de pérdida y métodos de minimización

La incorporación de una red residual que incluye las ecuaciones físicas de gobierno es una de las innovaciones presentadas por las PINNs [9]. En el enfoque tradicional de los PINNs (PDE-Based) la función de pérdida tiene como objetivo minimizar las diferencias entre el resultado de la red neuronal y la información generada o medida, retornando los parámetros de mejor aproximación a la solución [23]. Esta función desempeña un papel fundamental en el proceso de entrenamiento de los PINNs, ya que guía la actualización de los pesos de la red neuronal para lograr soluciones que se ajusten a las restricciones físicas impuestas por las ecuaciones de gobierno y las condiciones de contorno [8].

La función de pérdida se construye con el objetivo de que la red neuronal satisfaga las ecuaciones diferenciales del problema. Para lograrlo, se utiliza una formulación de la función de pérdida que penalice la violación de las ecuaciones de gobierno en los puntos de colocación. Además, la función de pérdida también debe tener en cuenta las condiciones de contorno del problema. Esto implica incluir términos en la función de pérdida que penalicen las desviaciones de la red neuronal respecto a las condiciones de contorno especificadas [27]. Estos términos aseguran que la red aprenda a respetar las condiciones físicas impuestas en los límites del dominio.

Según *Raissi et al.* [4] en el enfoque tradicional de los PINNs la función de pérdida es la siguiente:

$$MSE = MSE_u + MSE_f \quad (17)$$

Donde:

$$MSE_u = \frac{1}{N_u} \sum_1^{N_u} |u(t_u^i, x_u^i) - u^i|^2 \quad (18)$$

$$MSE_f = \frac{1}{N_f} \sum_1^{N_f} |f(t_f^i, x_f^i)|^2$$

De esta manera, en (17), $\{t_u^i, x_u^i, u^i\}_{i=1}^{N_u}$ corresponden a los datos iniciales y de contorno respectivamente para $u(t, x)$ y $\{t_f^i, x_f^i\}_{i=1}^{N_f}$ detalla los puntos de colocación en $f(t, x)$. La función de pérdida MSE_u corresponde a los datos iniciales y de contorno, mientras que MSE_f aplica la ecuación diferencial del sistema para los puntos de colocación.

La selección de la función de pérdida, como el error cuadrático medio (MSE), el error absoluto medio (MAE) u otras alternativas, se considera como un hiperparámetro adicional que requiere una cuidadosa elección. En la práctica, es común utilizar un enfoque de prueba y error para determinar la función de pérdida más adecuada, siendo el MSE la opción más popular en el contexto de los PINNs [27].

Durante el entrenamiento de las PINNs, el objetivo principal consiste en minimizar la función de pérdida mediante la utilización de diversas técnicas de optimización. Entre estas técnicas se encuentran los algoritmos de descenso de gradiente estocástico (SGD por sus siglas en inglés), así como variantes más avanzadas como Adam y otros métodos de optimización. Estos algoritmos demuestran su capacidad para proporcionar soluciones precisas en problemas de optimización que involucran no linealidades y no convexidades [31].

Cada método tiene sus propias ventajas y desventajas, y es importante elegir el método adecuado para el problema específico en cuestión. Por ejemplo, el método de gradiente descendente es fácil de implementar, pero puede ser propenso a quedarse atrapado en puntos de silla locales, mientras que el método de L-BFGS tiene una mayor precisión, pero requiere más capacidad computacional.

Capítulo 4: Implementación

Composición del sistema

La composición del sistema se basa en la idea dada por *Samaniego et al.* [14] que introduce el conocimiento físico en el proceso de entrenamiento de la red neuronal. Esto se lleva a cabo al adoptar un planteamiento variacional para resolver las ecuaciones diferenciales junto con la aplicación de redes neuronales. Este enfoque no requiere derivar la forma débil del problema, ni depende de la calidad y cantidad de los datos de entrenamiento. Entonces, el sistema está definido desde el principio y condiciones de contorno respectivas.

En el presente trabajo de titulación, el conocimiento físico introducido es el comportamiento de la viga bajo la teoría de Euler-Bernoulli. Pues, como se mencionó anteriormente la solución del problema de valores de contorno es matemáticamente equivalente a la solución que resulta de la formulación variacional ligada a las condiciones de contorno respectivas. Entonces, la solución que resulta de la formulación variacional es tal que tal que minimice la energía potencial total de la viga. Esta idea es ligada con la forma bajo la cual se da el entrenamiento de las redes neurales. En este proceso se requiere minimizar la función de pérdida, es por ello por lo que la ecuación de la energía potencial total del sistema resulta una función de pérdida precisa en dicho contexto.

Características geométricas y mecánicas de la viga

La relación entre los esfuerzos y las fuerzas que actúan sobre una viga y su respuesta estructural es esencialmente transformada a través de distintos parámetros que dependen de las características geométricas y mecánicas inherentes a la viga. Las fuerzas ejercen momentos sobre la viga, los cuales, a su vez, influyen en las deflexiones experimentadas por la misma. Este proceso de transformación, desde esfuerzos y fuerzas iniciales hasta las deflexiones resultantes, está fuertemente condicionado por la longitud, la inercia y el módulo de elasticidad de la viga (Ilustración 7). Para obtener una representación precisa de las deflexiones en la viga, es crucial tener en cuenta esta secuencia de transformación y considerar meticulosamente las propiedades geométricas y mecánicas que gobiernan su comportamiento estructural.

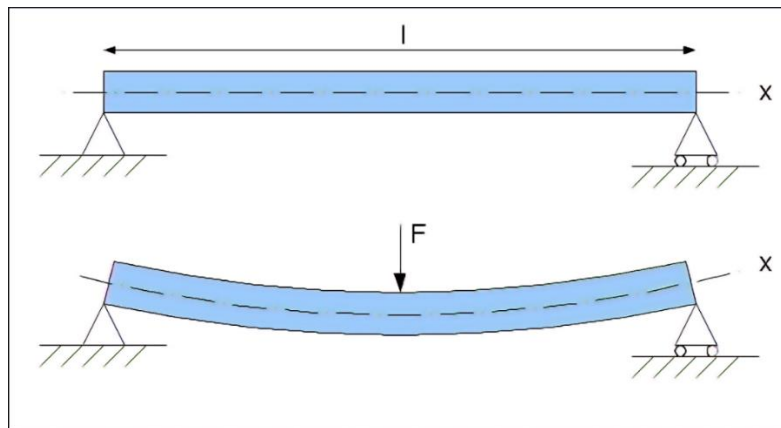


Ilustración 7. Viga simplemente apoyada que reacciona ante fuerza puntual por Komperød M. [32].

La viga bajo estudio presenta propiedades mecánicas clave que influyen en su comportamiento estructural. Uno de los parámetros fundamentales es el módulo de elasticidad (E), que refleja la rigidez del material y su capacidad para resistir deformaciones bajo cargas aplicadas. Además, la inercia (I) de la sección transversal de la viga es determinante para su resistencia a la flexión y torsión.

Los parámetros de elasticidad e Inercia corresponden a la elasticidad de una viga de hormigón y la inercia a una viga tipo I tomados del trabajo de *Katsikis et. al.* [6] donde se realiza un análisis bajo las mismas restricciones en las vigas planteadas en posteriores secciones.

De esta manera:

$$E = 200 \text{ GPa}$$

$$I = 0.000038929334 \text{ m}^4$$

Además de las propiedades mecánicas, los parámetros geométricos son cruciales para comprender la respuesta de la viga. La longitud total de la viga (L) y las dimensiones de su sección transversal, como el ancho y la altura, determinan su geometría.

Para el análisis, la longitud (L) de la viga corresponde a:

$$L = 2.7 \text{ m}$$

La distribución de cargas aplicadas a lo largo de la viga también desempeña un papel esencial en su análisis. En particular, la presencia de cargas distribuidas, como fuerzas uniformes a lo

largo de la longitud, o cargas concentradas en puntos específicos, influye directamente en los momentos flexionantes y las deflexiones experimentadas por la viga.

La carga distribuida (q) es de 60 KN/m.

Condiciones de Apoyo y Cargas Externas

Las condiciones de apoyo juegan un papel fundamental en la respuesta mecánica de la viga. Las condiciones de apoyo pueden ser del tipo simplemente apoyado, empotrado, articulado, entre otras. Estas condiciones definen cómo la viga está restringida en sus extremos y afectan la distribución de momentos y fuerzas a lo largo de su longitud. Además, las cargas externas, como fuerzas o momentos aplicados en puntos específicos de la viga, también influyen en su comportamiento estructural. La interacción entre las condiciones de apoyo y las cargas externas determina las deformaciones y las tensiones en la viga, lo que es esencial para su análisis y diseño adecuado.

Problema 1: Viga simplemente apoyada

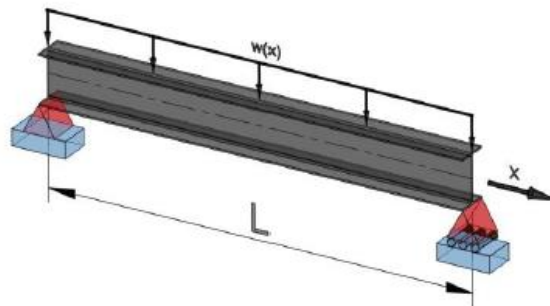


Ilustración 8. Viga con condiciones de simple apoyo en sus extremos por Katsikis et al. [6].

Basándonos en la forma fuerte de la ecuación diferencial de la viga desarrollada en la referencia (15) se procederá a mostrar la solución exacta de una viga simplemente apoyada.

En el caso de una viga simplemente apoyada (Ilustración 8 [6]), los desplazamientos en los puntos de apoyo son iguales a cero, lo que implica que $w(0) = 0, w(L) = 0$. Otra restricción asociada al estado de la viga es el momento, donde no hay restricciones de momento en los puntos de apoyo, por lo que $M(0) = 0, M(L) = 0$.

De manera que las condiciones de contorno para el problema 1 resultan:

$$w(0) = 0, \quad w(L) = 0, \quad EI \frac{d^2 w(0)}{dx^2} = 0, \quad EI \frac{d^2 w(L)}{dx^2} = 0 \quad (19)$$

En virtud de la forma fuerte de la ecuación de la viga (15), la solución exacta al problema 1 obtenida mediante métodos analíticos es la siguiente:

$$w(x) = \frac{q(x)}{24 EI} (-x^4 + 2Lx^3 - L^3x) \quad (20)$$

Problema 2: Viga empotrada – libre

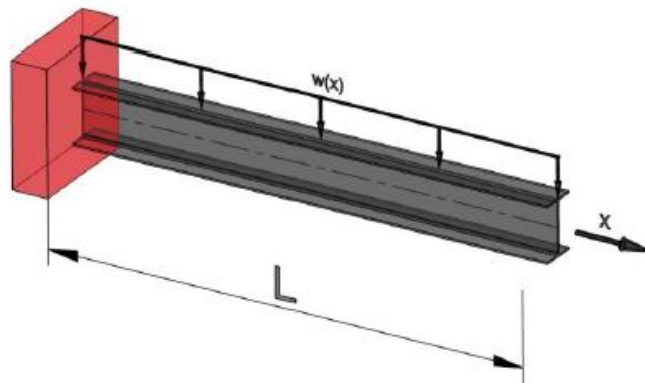


Ilustración 9. Viga con condición empotrada en un extremo y libre en otro por Katsikis et al. [6].

De igual manera para la viga empotrada en un extremo y libre en otro (Ilustración 9 [6]), basándonos en la forma fuerte de la ecuación diferencial de la viga desarrollada en (15), a continuación, se expone la solución exacta.

En este caso de restricción, para la viga en la cara del apoyo empotrado ($x = 0$), el desplazamiento es cero y la pendiente es igual a cero, de esta manera:

$$w(0) = 0, \quad \frac{dw(0)}{dx} = 0 \quad (21)$$

Para el extremo libre ($x = L$), el momento es igual a cero ($M(L) = 0$) al igual que el cortante ($V(L) = 0$) de esta manera las condiciones de empotramiento resultan:

$$EI \frac{d^2 w(L)}{dx^2} = 0, \quad EI \frac{d^3 w(L)}{dx^3} = 0 \quad (22)$$

En virtud de la forma fuerte de la ecuación de la viga (15), para las condiciones de empotramiento (21) y (22) la solución exacta al problema 2 obtenida mediante métodos analíticos es la siguiente:

$$w(x) = \frac{q(x)}{24 EI} (-x^4 + 4Lx^3 - 6L^2x^2) \quad (23)$$

Problema 3: Viga empotrada – simplemente apoyada

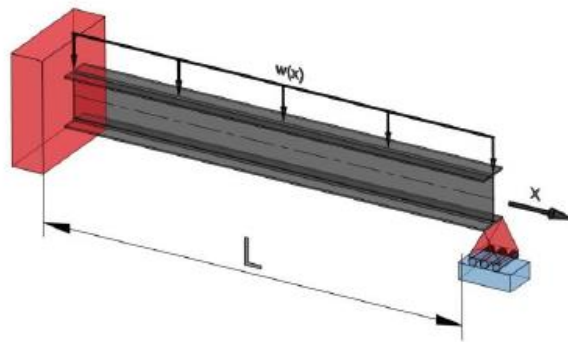


Ilustración 10. Viga con condición empotrada en un extremo y simplemente apoyada en otro Katsikis et al.[6].

En el último caso de estudio (Ilustración 10 [6]) para la viga empotrada en un extremo y simplemente apoyada en otro, en base a la forma fuerte de la ecuación diferencial de la viga desarrollada en (15). La solución exacta se desarrolla:

En problema, en la cara del apoyo empotrado ($x = 0$), tanto el desplazamiento como la pendiente son igual a cero, de esta manera:

$$w(0) = 0, \quad \frac{dw(0)}{dx} = 0 \quad (24)$$

En el extremo simplemente apoyado ($x = L$), tanto el momento como el desplazamiento son igual a cero, por lo que $M(L) = 0$ y $y(L) = 0$ de esta manera las condiciones de empotramiento resultan:

$$EI \frac{d^2 w(L)}{dx^2} = 0, \quad w(L) = 0 \quad (25)$$

Resolviendo para la forma fuerte de la ecuación de la viga (15), para las condiciones de empotramiento (24) y (25), la solución exacta al tercer problema obtenida mediante métodos analíticos es la siguiente:

$$w(x) = \frac{q(x)}{24 EI} \left(-x^4 + \frac{5}{2} Lx^3 - \frac{3}{2} L^2 x^2 \right) \quad (26)$$

Modelado del sistema

El proceso de aprendizaje dado en una red neuronal está guiado por las funciones de pérdida. Durante este proceso se lleva a cabo la optimización de pesos y sesgos al minimizar las dichas funciones. Estas funciones son escogidas en función del modelo analizado. En el presente trabajo de titulación se incluye el conocimiento de la física dentro del modelo en la función de pérdida considerando la representación de la ecuación diferencial por su formulación variacional para la ecuación de la energía en una viga.

La inclusión del conocimiento de la física en la función de pérdida ha sido implementada por otros autores dentro del marco de la mecánica de sólidos. Por ejemplo, *Li W et al.* [13] con el objetivo de conocer el comportamiento de placas bajo la teoría clásica presenta dos tipos de funciones de pérdida en las cuales la física se incluye en el proceso de aprendizaje. El primero tipo es denominado "PDE-based" en el cual se considera la forma fuerte del problema. Esta idea inicialmente fue propuesta por *Raissi et al.* [4], y también fue usada por *Katsikis D et al.* [6] en aplicaciones en barras y vigas. El segundo tipo es denominado "Energy-based", donde la función de pérdida está formada por la energía potencial de la placa junto con un término que representa las condiciones de contorno cinemáticas.

La función de pérdida está representada a través de la formulación variacional, donde el funcional en cuestión corresponde a la energía potencial total de la viga (16). Sin embargo, esta expresión no considera las condiciones de contorno cinemáticas, las cuales son introducidas como un término adicional a la formulación de la energía potencial total,

denominado “término de penalización” (T). Estas condiciones varían según las consideraciones de cada viga analizada, lo que conduce a la definición general de la función de pérdida (27).

$$\mathcal{L}_{viga} = \Pi_{viga} = U + V + T \quad (27)$$

La energía potencial total de la viga junto con el término de penalización (27) cumple con el principio de la mínima energía potencial, pues al hallar la variación de esta, el término se anula, de manera que el sistema se mantiene en equilibrio.

El balance de la función de pérdida consiste en escalar individualmente los términos que corresponden a la formulación y a los términos que consideran las condiciones de contorno cinemáticas. Esto se realiza para equiparar la importancia de las condiciones de contorno en contraste con los de la formulación debido a la diferencia de unidades, y así evitar que el modelo considere otra expresión diferente que cumpla con los requerimientos de la formulación base. La función de pérdida balanceada es la siguiente:

$$\mathcal{L}_{viga} = \lambda_0 \cdot (U + V) + \lambda_k \cdot T \quad (28)$$

Donde:

λ_0 = Escalar correspondiente a la formulación base.

λ_k = Escalar correspondiente a las condiciones de contorno cinemáticas.

El término de penalización depende de las condiciones de apoyo del problema considerado. De manera general, dentro del dominio cinemático de la viga tiene la siguiente forma:

$$T = \int_{\Gamma_d} |w(x) - \widehat{w}^i| + \int_{\Gamma_d} |w'(x) - \widehat{w}'| + \int_{\Gamma_d} |w''(x) - \widehat{w}''| \quad (29)$$

Donde, \widehat{w} , \widehat{w}' , \widehat{w}'' representan deflexión, giro y curvatura respectivamente y corresponden a las condiciones de contorno propias de cada problema, mientras que w , w' , w'' representan deflexión, giro y curvatura respectivamente y son resultado de las predicciones.

La aplicación de la función de pérdida para el [Problema 1: Viga simplemente apoyada](#). Primero, se define las condiciones de contorno que formarán parte del término de penalización. Se consideran dos condiciones de contorno (30), la deflexión nula al inicio y al final de la viga.

$$w(0) = 0, \quad w(L) = 0 \quad (30)$$

Se considera la aproximación de la función de pérdida (27) que corresponde a la energía potencial total de la viga simplemente apoyada junto a las condiciones de contorno añadidas.

$$\mathcal{L}_{viga} = \frac{1}{Q_0} \sum_{i=1}^{Q_0} \left| \frac{EI_y}{2} \left(\frac{d^2 w^i}{dx^2} \right)^2 - w^i q \right|^2 + \frac{1}{Q_T} \sum_{i=1}^{Q_T} |w(x^i) - \hat{w}^i|^2 \quad (31)$$

En (31), Q_0 representa el número de puntos de colocación dentro del dominio y Q_T número de puntos en el contorno cinemático. El superíndice i indica el i -ésima punto de entrenamiento, \hat{w}^i es la deflexión correspondiente a la condición de contorno, mientras que w^i son las deflexiones predichas.

Como resultado de la implementación del [Problema 1: Viga simplemente apoyada](#), considerando la función de pérdida no balanceada (31), se obtiene la deflexión (Ilustración 11) que cumple con los requerimientos de la formulación base, sin embargo, no cumple con las condiciones de contorno cinemáticas impuestas.

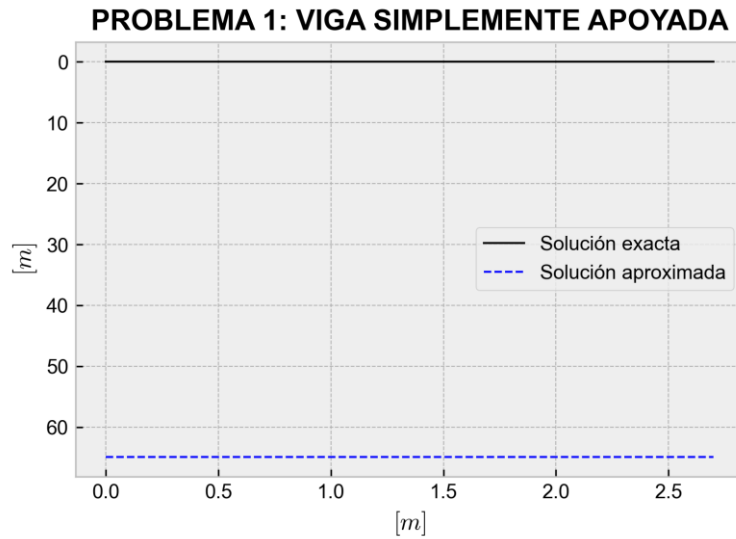


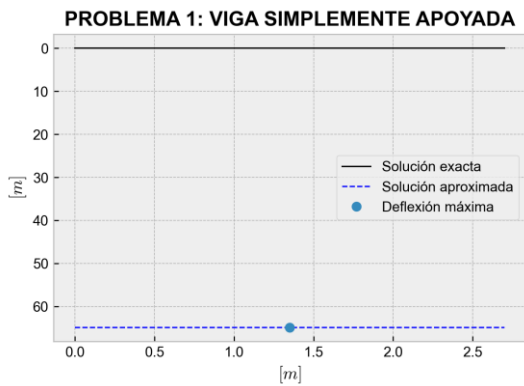
Ilustración 11. Deflexión del Problema 1 con función de pérdida desbalanceada.

El balanceo de la ecuación de pérdida (32) responde a un proceso empírico. En el presente problema, así como en los otros problemas de aplicación se consideran aplicables los mismos valores escalares. Estos favorecen al balanceo y a la posterior convergencia del modelo.

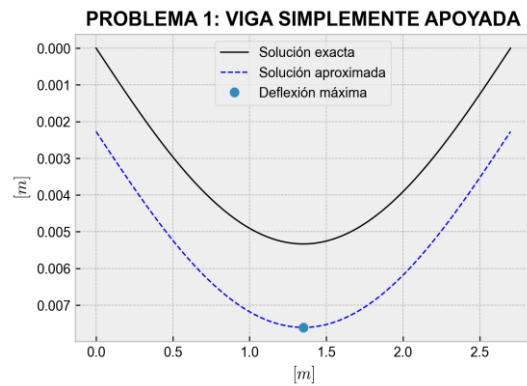
$$\mathcal{L}_{viga} = \lambda_0 \cdot \frac{1}{Q_0} \sum_{i=1}^{Q_0} \left| \frac{EI_y}{2} \left(\frac{d^2 w^i}{dx^2} \right)^2 - w^i q \right|^2 + \lambda_k \cdot \frac{1}{Q_T} \sum_{i=1}^{Q_T} |w(x^i) - \hat{w}^i|^2 \quad (32)$$

$$\lambda_0 = 1 \quad \lambda_k = 2 E/I$$

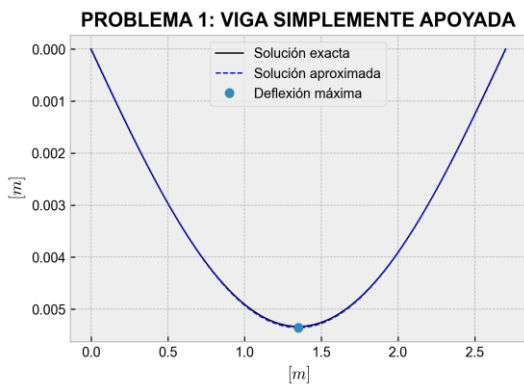
En la Ilustración 12 se observa que al aumentar la magnitud del valor escalar λ_k correspondiente las condiciones de contorno, el resultado de la deflexión converge a la solución. Podemos observar que a partir del valor de $\lambda_k = 2 E/I$ se obtiene un ajuste correcto. Además de mantener las mismas condiciones de inicialización de la red, mismas iteraciones e igual semilla de TensorFlow para reducir aleatoriedad.



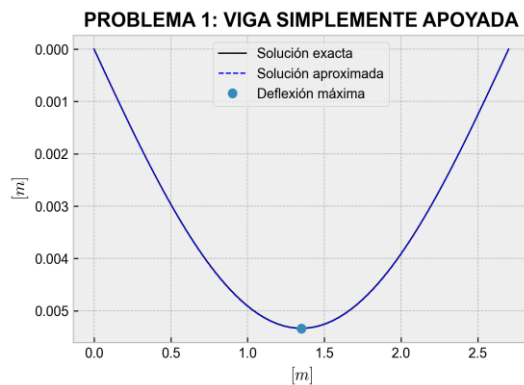
$$\lambda_0 = 1 \quad \lambda_k = 1$$



$$\lambda_0 = 1 \quad \lambda_k = 0.01 E/I$$



$$\lambda_0 = 1 \quad \lambda_k = 2 E/I$$



$$\lambda_0 = 1 \quad \lambda_k = 4 E/I$$

Ilustración 12. Proceso empírico del balanceo de función de pérdida para el Problema 1.

Para obtener los resultados expuestos en Ilustración 12 además del balanceo de la función de pérdida se requirió modificar dicha función. Esta modificación se realizó debido a que al implementar la expresión (32) la deflexión obtenida en toda la viga tendió a cero, lo cual puede ser atribuido a que es una respuesta correspondiente a un mínimo local. La [Implementación de PINN](#), explicada en el siguiente apartado, considera dentro del proceso de optimización una forma de evitar la influencia de los mínimos locales. Es por ello por lo que adicional a lo considerado en la implementación, se aplica lo mencionado por Roy K et al. [33], que las

barreras de energía (Ilustración 13) pueden ser superadas al aumentar energía de disipación, que en nuestro caso corresponde a la energía interna del sistema (U).

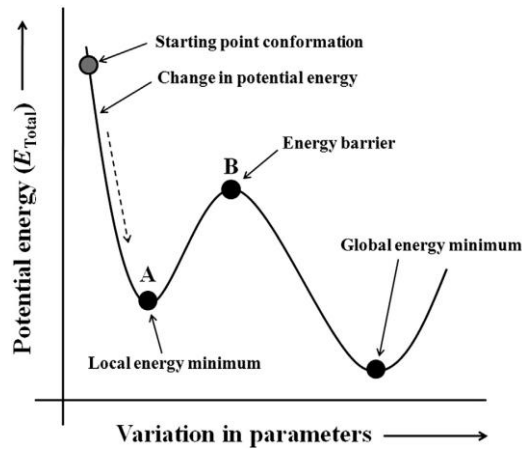


Ilustración 13. Diferentes fases de una molécula durante la minimización de su energía por Roy K et al. [33]

Entonces, en el presente caso a la función de pérdida (32) se adiciona un término extra de disipación denominado U_{extra} . Por tanto, la función de pérdida toma la siguiente forma (33):

$$\mathcal{L}_{viga} = \lambda_0 \frac{1}{Q_0} \sum_{i=1}^{Q_0} \left| \frac{EI_y}{2} \left(\frac{d^2 w^i}{dx^2} \right)^2 + U_{extra} - w^i q \right|^2 + \lambda_k \frac{1}{Q_T} \sum_{i=1}^{Q_T} |w(x^i) - w^i|^2 \quad (33)$$

El proceso de definir U_{extra} , es decir, el proceso de añadir energía de disipación también se lo realiza de manera empírica para cada problema considerado. En los tres problemas expuestos, se requiere el mismo término extra de disipación. En la Ilustración 14, se

ejemplifica el proceso de calibración del término extra de disipación desde 0 hasta $0.5 EI$ para el [Problema 2: Viga empotrada – libre](#).

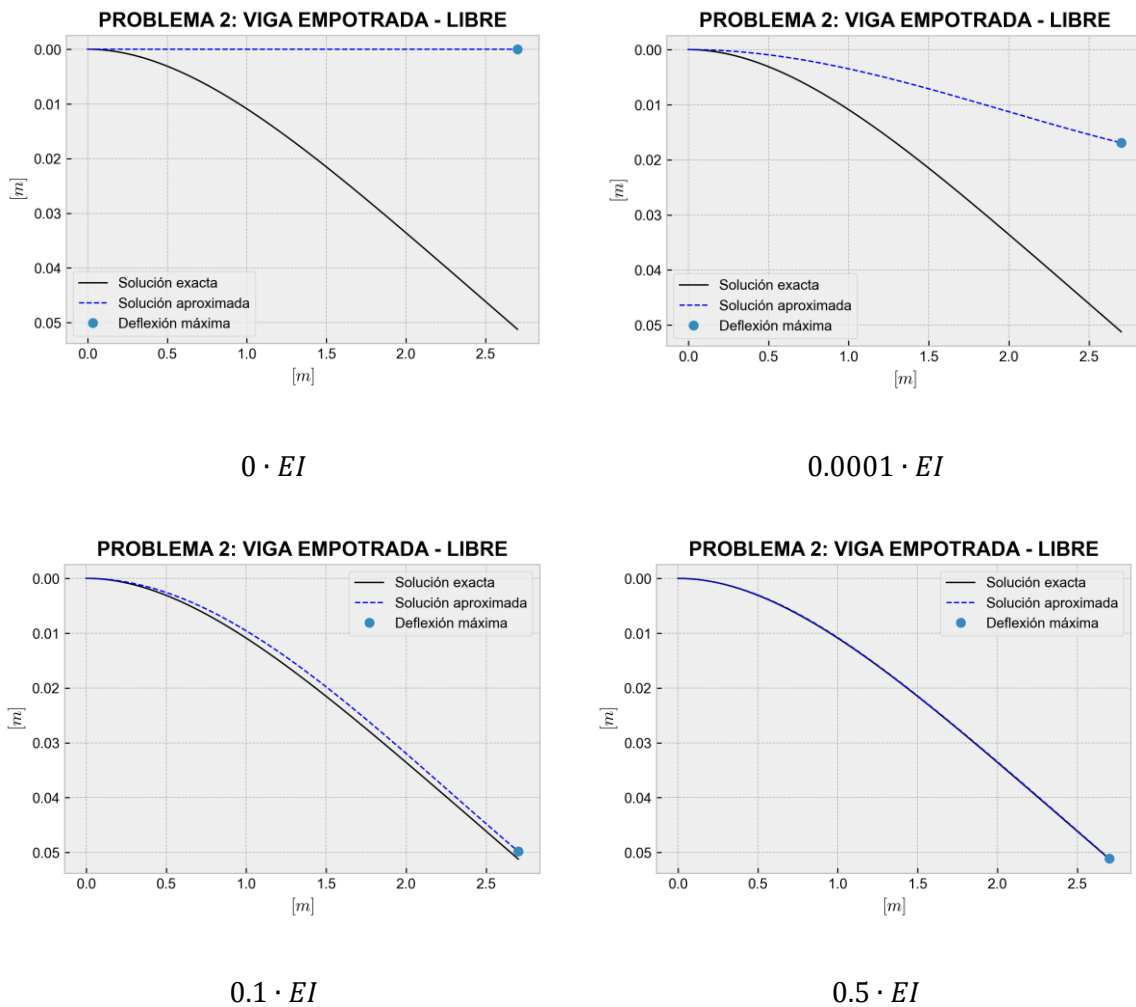


Ilustración 14. Calibración del término de disipación de energía.

En la Ilustración 14 y en la Ilustración 15, se observa que para valores entre $0.5 \cdot EI$ y $1 \cdot EI$, el término de disipación permite la convergencia del modelo. En la Ilustración 15 se observa que, al aumentar 10 veces dicho término de disipación, el ajuste de la función de pérdida se ve influenciado. Esto evidencia la relación entre los valores de los escalares correspondientes al balanceo de la función de pérdida y el valor del término extra de disipación, puesto que, para obtener un ajuste preciso se requiere modificar los escalares mediante el balanceo.

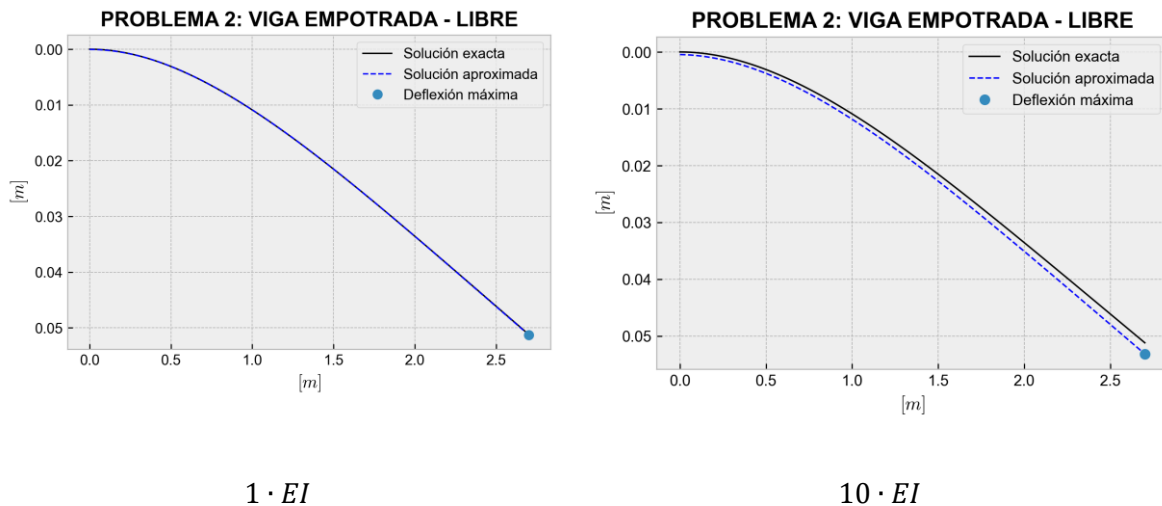


Ilustración 15. Calibración del término de disipación.

Implementación de PINN

Para llevar a cabo la implementación de la Physics-Informed Neural Network (PINN) en este estudio, se utilizó como base el código de redes neuronales de [Federico Magnani](#) y bajo la modificación del cotutor de la presente tesis Ing. Ángel Vázquez PhD., donde se aplicó el lenguaje de programación Python 3 junto con la biblioteca de código abierto Tensor Flow 2. El código es una interpretación actualizada de la primera implementación realizada en "*Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*" por Raissi, Perdikaris y Karniadakis [4].

Esta nueva interpretación bajo la librería actualizada proporcionó una plataforma robusta y eficiente para la implementación de redes neuronales artificiales y la realización de cálculos numéricos necesarios para resolver la ecuación diferencial de la energía en una viga.

Hiperparámetros utilizados en la Physics-Informed Neural Networks

Como se habló en el apartado [Funcionamiento y elementos de las PINNs](#), los hiperparámetros son variables que definen la arquitectura y el comportamiento de la red, y su elección adecuada es crucial para obtener resultados precisos y estables. Entre los hiperparámetros considerados se encuentran el número de capas ocultas, la cantidad de neuronas por capa, la tasa de aprendizaje, la función de activación, y el tamaño y distribución de los conjuntos de entrenamiento, validación y prueba. A continuación, se mostrarán los

parámetros elegidos en la implementación de la red neuronal informada por la física para la resolución de la ecuación variacional de la energía en vigas. En [34] Bischof analiza varias recomendaciones de selección de hiperparámetros en las redes neuronales informadas por la física.

Arquitectura de la red

La arquitectura de la red neuronal utilizada en la PINN se diseñó con capas ocultas que contenían neuronas completamente conectadas. Existen distintas formas de conformar las capas de la red neuronal, dependiendo de la cantidad de capas se puede clasificar a la red como superficiales o "shallow network" y profundas o "deep network". En el primer caso, la cualidad es que la red contiene hasta dos capas ocultas, y en el segundo caso las redes pueden contener más de dos capas [9].

Durante el proceso de implementación, se consideró tanto el uso de "deep networks" como el de "shallow networks". Si bien las "deep networks" ofrecen una mayor capacidad de representación y complejidad, y en artículos sobre distintas implementaciones de PINNs como [4] [13] se utilizan redes profundas por la complejidad de las ecuaciones utilizadas. Sin embargo, para la presente tesis que involucra una expresión integral de variables dependientes, es decir la formulación variacional del problema de una ecuación diferencial que, al ser relativamente sencilla, el uso de una red superficial resultó más adecuado.

La elección de las "shallow networks" se debió a la simplicidad de la ecuación y al objetivo de evitar el sobreajuste de la red. *Bischof et al.* [35] sugiere que redes superficiales y amplias tienden a obtener mejores resultados. Al limitar el número de capas ocultas y parámetros, las "shallow networks" son menos propensas a problemas de overfitting (sobreajuste) debido a su simplicidad, sacrificando su capacidad para aprender representaciones de datos complejas, lo que puede afectar el rendimiento en problemas más desafiantes. Sin embargo, las redes superficiales son lo suficientemente capaces de producir todo tipo de combinaciones no lineales en funciones continuas [9].

La implementación de "redes superficiales" en este contexto ha demostrado ser efectiva para obtener resultados satisfactorios y mejorar la comprensión del comportamiento de las soluciones de las ecuaciones diferenciales abordadas. No obstante, se reconoce que, en casos de problemas más complejos o con conjuntos de datos más extensos, la adopción de "redes profundas" podría ser necesaria para capturar representaciones más elaboradas y precisas. Además, en relación con la densidad de neuronas en las capas, a pesar de que en

la literatura se han obtenido resultados satisfactorios mediante el uso de redes con amplitud, es importante destacar que la sección "[Arquitectura de la red](#)" en los resultados analiza específicamente las densidades de capas y los tipos de red adecuadas para el problema variacional en cuestión.

Función de activación

Las funciones de activación son elementos clave en las redes neuronales, ya que introducen no linealidad en el modelo y permiten la captura de relaciones complejas entre las variables. En esta tesis, la elección de la función de activación se ha considerado en base a distintas implementaciones como el artículo inicial de PINNs de Raissi et al. [4]. La mayoría de los autores que trabajan con PINNs utilizan la tangente hiperbólica para garantizar la convergencia y el rendimiento óptimo de la red [9]. La tangente hiperbólica (\tanh) es utilizada debido a su naturaleza acotada y su simetría alrededor del origen, lo que permite que las salidas de la red estén acotadas en un rango entre -1 y 1. Esto es especialmente beneficioso en el contexto de ecuaciones diferenciales, ya que evita salidas no deseadas que pudiesen llevar a divergencias o inestabilidades numéricas.

Algoritmos de optimización

Como se habló anteriormente en la sección de [Función de pérdida y métodos de minimización](#), los métodos de descenso de gradiente estocástico son los más utilizados en procesos de minimización ya que proporcionan soluciones precisas en una variedad de problemas de optimización.

En los métodos de descenso de gradiente estocástico, es común utilizar optimizadores populares como Adam y Broyden-Fletcher-Goldfarb-Shanno (BFGS) (o su modificación L-BFGS). Estos optimizadores son elegidos por sus propiedades y capacidades para abordar problemas de optimización no lineales. En el trabajo de comparación de métodos de optimización realizado por *Markidis* [29], se utiliza una estrategia que combina ambos optimizadores. Inicialmente, se utiliza el optimizador Adam para superar los mínimos locales en el espacio de la función diferenciable. Luego, se emplea el optimizador BFGS, que utiliza la matriz Hessiana para calcular la dirección de optimización y obtener resultados más precisos. Esta secuencia de optimizadores se utiliza para aprovechar las fortalezas de cada uno y lograr un mejor rendimiento en la optimización del problema en cuestión. Esta metodología es utilizada también en *He et al.*, como fue citado en *Cuomo et al.* [9] para una mejor y más rápida convergencia.

En esta tesis se ha adoptado una metodología de dos optimizadores sucesivos con el objetivo de sortear la convergencia de la red a mínimos locales y buscar que la red alcance mínimos globales. Esta estrategia se emplea para reducir la incertidumbre relacionada con las causas de errores de convergencia en la red neuronal. En particular, se utiliza inicialmente el optimizador Adam y, posteriormente, se aplica el optimizador L-BFGS, una variante del BFGS que aproxima el cálculo de la matriz hessiana para ahorrar memoria computacional, en lugar de estimarla completamente. Esta combinación de optimizadores busca obtener una convergencia más estable y precisa durante el entrenamiento de la red.

Épocas de entrenamiento y tasa de aprendizaje

En el contexto de las redes neuronales, una "época" hace referencia a todas las iteraciones del optimizador necesarias para recorrer todo el conjunto de datos durante el entrenamiento. Según *Markidis* [29], para obtener resultados precisos en PINNs son necesarias épocas en el orden de las mil unidades. Es decir, un entrenamiento prolongado y detallado es fundamental para obtener resultados óptimos en este tipo de problemas.

En el caso del optimizador Adam, *Bischof* [34] recomienda empezar probando en una tasa de aprendizaje desde 0.1 y reducir el valor para observar cómo evoluciona la función de pérdida, en la presente tesis se utiliza el valor de 0.001 que es el valor predeterminado de la librería. Para el optimizador L-BFGS no existe una tasa de aprendizaje ya que es un algoritmo adaptativo.

Inicialización de pesos

La inicialización de pesos es crucial para la convergencia de la red neuronal. Kumar en [36] muestra la estrecha relación entre las funciones de activación y la inicialización de pesos, recalcando que activadores RELU fallan en trabajar con Xavier normal. Al usar tangente hiperbólica como función de activación la presente PINN, el hiperparámetro inicializador de pesos se ha elegido el Glorot normal (Xavier normal) debido a su capacidad para proporcionar una inicialización adecuada que favorece la estabilidad del entrenamiento.

Puntos de colocación

El método energético (Energy-Based) en el que se basa la presente tesis muestra una dependencia significativamente mayor del tamaño de la muestra para el entrenamiento de la red en comparación con las implementaciones basadas en ecuaciones diferenciales (PDE

Based). Esto se debe a que en el método energético hace uso de todos los puntos de colocación durante el entrenamiento [13]. No obstante, debido a la presencia de una ecuación diferencial de segundo orden, el algoritmo no experimenta dificultades de convergencia ante cambios en la ubicación de los puntos de la red neuronal. Aunque no hay recomendaciones específicas sobre la cantidad de puntos necesarios para lograr la convergencia, Markidis [29] enfatizan que un aumento en el número de puntos de colocación para el entrenamiento de la red conlleva a un incremento en el valor del error cuadrático medio (MSE, por sus siglas en inglés).

Problemas de aplicación

En el siguiente segmento, se presentarán las modificaciones clave del código para su implementación en las tres aplicaciones de vigas propuestas. Estas adaptaciones son esenciales para adecuar la metodología de Physics-Informed Neural Networks a los diferentes escenarios de análisis de vigas y obtener resultados adecuados. Exploraremos las transformaciones realizadas en el código original para abordar las variantes de vigas simplemente apoyadas, empotradas-libres y empotradas-simplemente apoyadas. Cada ajuste se centrará en parámetros críticos, alineándolos con las condiciones de contorno, las características geométricas y las propiedades mecánicas de la viga. Este análisis conciso de las modificaciones del código brindará una visión concreta de cómo se ha logrado adaptar la metodología PINN a problemas reales de ingeniería estructural.

Función de energía y función de pérdida balanceada

De la sección [Modelado del sistema](#) a partir de la calibración en la Ilustración 14 del término extra de energía requerido en la función para una correcta convergencia llamado término de disipación de energía. De esta manera en el código se expresa de la siguiente manera la función de energía:

```
f_w = (self.E*self.I/2) * tf.square(w_xx) - self.q * w +  
0.5*(self.E*self.I)
```

La función de pérdida balanceada expresada en la ecuación (33) en código se representa:

```
loss = 2*self.E/self.I*y0 + yS + 2*self.E/self.I*yB
```

La cual representa a la función de pérdida con dos condiciones de contorno representados por y_0 y y_B . El término y_S representa el resultado de los puntos de colocación al pasar por la red neuronal.

Generalidades de la Red Neuronal – Hiperparámetros implementados

Arquitectura de la red

Siguiendo las recomendaciones previamente detalladas para la implementación de la Red Neuronal en la sección de implementación [Arquitectura de la red](#), se propone una distribución de red superficial para el presente trabajo de titulación. Esta arquitectura consta de una única capa de entrada al igual que en la salida, donde esas capas tienen una sola neurona. Esta elección se fundamenta en que el problema en cuestión presenta una única variable independiente a lo largo de la longitud de la viga (eje x).

Para las capas ocultas, se configuran dos capas con la característica de ser no amplias como se propone en [35] (capas ocultas con 256 neuronas) consistiendo cada una en 20 neuronas completamente interconectadas entre sí.

Así se establece la siguiente configuración en el código:

```
layers = [1,20,20,1]
```

En la sección de [Resultados: Arquitectura de la red](#), se lleva a cabo un análisis de la sensibilidad con respecto a la variación del parámetro crucial en las redes neuronales informadas por la física: la arquitectura de la red. Dicha variación abarca tanto redes superficiales como profundas, además de explorar cambios en la amplitud de las capas, es decir, en el número de neuronas que las conforman.

Función de activación e inicialización de pesos

Dentro de la implementación se aplican las condiciones analizadas en las secciones [Función de activación](#) e [Inicialización de pesos](#) de la siguiente manera:

```
self.add(tf.keras.layers.InputLayer(input_shape=(layers[0],)))

self.add(tf.keras.layers.Lambda(
    lambda X: 2.0*(X-self.lb)/(self.ub-self.lb)-1.0))
for width in layers[1:-1]:

    self.add(tf.keras.layers.Dense(
```



```

width, activation=tf.nn.tanh,
kernel_initializer="glorot_normal"))

self.add(tf.keras.layers.Dense(
    layers[-1], activation=None,
    kernel_initializer="glorot_normal"))

```

Donde en un principio se añade la capa de entrada con una normalización de los datos de entrada, se continua con las capas ocultas con una función de activación de tangente hiperbólica y se finaliza con la capa de salida.

Algoritmo de optimización y épocas de entrenamiento

Dentro del algoritmo se incluyen las optimizaciones planteadas para sortear distintos problemas de minimización (como se explicó en la sección [Algoritmos de optimización](#)). De esta manera, se define el número de iteraciones para cada optimizador, si el valor es de cero el optimizador no hará ninguna iteración. De esta manera en el código se define:

```

adam_iterations = 500 # Numero de iteraciones de Adam
lbfgs_max_iterations = 1000 # Máximas iteraciones para L-BFGS

```

Para realizar 500 iteraciones en Adam y definir 1000 iteraciones máximas en L-BFGS

En el algoritmo para el entrenamiento de la red neuronal son necesarios ambos parámetros de iteración, definidos como:

```
Adam_hist,LBFGS_hist = model.train(adam_iterations, lbfgs_max_iterations)
```

Para la definición de las épocas de entrenamiento y distintos parámetros en la implementación se utilizan la librería de TensorFlow para Adam y Scipy para L-BFGS, donde para Adam se define la tasa de aprendizaje y en L-BFGS se definen las máximas iteraciones en función del parámetro definido con anterioridad.

Definición de parámetros de Adam:

```
optimizer=tf.keras.optimizers.Adam(learning_rate=0.001)
```

Definición de parámetros de L-BFGS:

```

results = scipy.optimize.minimize(self.loss_and_flat_grad,
                                  self.model.get_weights(),
                                  method='L-BFGS-B',
                                  jac=True,
                                  callback=self.callback,

```

```
options = {'maxiter': maxiter,
          'maxfun': 50000,
          'maxcor': 50,
          'maxls': 50,
          'ftol' : 1.0 * np.finfo(float).eps})
```

Puntos de colocación

En la sección [Funcionamiento y elementos de la PINN](#) se habló sobre las distintas distribuciones posibles para los puntos de colocación, de esta manera para el algoritmo implementado se utiliza una distribución con un espaciamiento equitativo en función del número de puntos requeridos.

Se define en el código:

```
num_train_samples = 181 # número de puntos de colocación

x_1 = np.linspace(0, L, num_train_samples).reshape((-1,1)) # distribución
de puntos de colocación
```

Problema 1: Viga simplemente apoyada

Para la viga simplemente apoyada, bajo las condiciones de contorno conocidas proporcionadas en [Problema 1: Viga simplemente apoyada](#) (19), se establecen requisitos conocidos tanto para el desplazamiento como para el momento. En el proceso de implementación, el algoritmo utiliza una ecuación diferencial de segundo orden, lo que implica que solo se necesitan dos condiciones de contorno para lograr su resolución. En el contexto del primer problema abordado en este trabajo de titulación, se emplearán únicamente las dos condiciones de desplazamiento nulo para el inicio y final de la sección ($w(0) = 0$, $w(L) = 0$):

En el código implementado tenemos:

```
x_0 = np.zeros((1, 1)) # Posición de contorno  $w(x_0) = 0$ 
x_1 = np.linspace(0, L, num_train_samples).reshape((-1,1)) # Puntos de
colocación
x_2 = L*np.ones((1, 1)) # Posición de contorno  $w(x_2) = 0$ 
```

```
w_0 = 0*np.ones((1, 1)) # Condición de contorno  $w_x(0) = w_0$ 
w_2 = 0*np.ones((1, 1)) # Condición de contorno  $w_x(L) = w_2$ 
```

Que corresponden a los valores y posiciones que toman las condiciones de contorno para el problema 1.

Dentro de las condiciones de contorno y la conformación de la función de pérdida que establecemos en la PINN para la viga simplemente apoyada:

```
X0 = tf.stack([x0], axis=1) #Condición  $u(0) = 0$ 
w_pred = self.model(X0) #Paso de condición X0 por la red
y0 = tf.reduce_mean(tf.square(w0 - w_pred)) #Penalty term de condición X0
f_w = self.net_f_wv() #Termino de energía para loss
yS = tf.reduce_mean(tf.square(f_u)) #Paso de colocacion points por la red
X2 = tf.stack([x2], axis=1) #Condición  $u(L) = 0$ 
w_pred = self.model(X2) #Paso de condición X2 por la red
yB = tf.reduce_mean(tf.square(w2-w_pred)) #Penalty term de condición X2
```

Donde la sumatoria y balanceo de los términos yS , $y0$ y yB conforma la función de pérdida.

Problema 2: Viga empotrada – libre

En el caso de una viga empotrada en un extremo y libre en el otro, se establecen, bajo las condiciones de contorno conocidas, definidas en (21) y (22), los requisitos para el desplazamiento y giro al inicio de la sección, y fuerza cortante y momento al final de la sección. En el proceso de implementación, el algoritmo se basa en una ecuación diferencial de segundo orden, lo que implica que solo se requieren dos condiciones de contorno para lograr su resolución. En el marco del segundo problema abordado en este trabajo de titulación, se utilizarán únicamente dos condiciones en la cara del empotramiento, donde el desplazamiento y el giro son nulos ($w(0) = 0, \frac{dw(0)}{dx} = 0$):

En el código implementado tenemos:

```
x_0 = np.zeros((1, 1)) # Posición de contorno  $w(x_0)$ 
x_1 = np.linspace(0, L, num_train_samples).reshape((-1,1)) # Puntos de
colocación
x_2 = np.zeros((1, 1)) # Posición de contorno  $w(x_2)$ 

w_0 = 0*np.ones((1, 1)) # Condición de contorno  $w(0) = w_0$ 
w_2 = 0*np.ones((1, 1)) # Condición de contorno  $w_x(0) = w_2$ 
```

Que corresponden a los valores y posiciones que toman las condiciones de contorno para el problema 2.

Dentro de las condiciones de contorno y la conformación de la función de pérdida que establecemos en la PINN para la viga empotrada - libre:

```
X0 = tf.stack([x0], axis=1) #Condición  $w(0) = 0$ 
w_pred = self.model(X0) #Paso de condición X0 por la red
y0 = tf.reduce_mean(tf.square(w0 - w_pred)) #Penalty term de condición X0
f_w = self.net_f_wv() #Termino de energía para loss
yS = tf.reduce_mean(tf.square(f_w)) #Paso de colocation points por la red
X2 = tf.stack([x2], axis=1) #Condición  $w_x(0) = 0$ 
w_pred, w_x_pred, *_ = self.net_uv(X2) #Paso de condición X2 por la red
yB = tf.reduce_mean(tf.square(w_x_pred-w2)) #Penalty term de condición X2
```

Donde la sumatoria y balanceo de los términos yS , $y0$ y yB conforma la función de pérdida.

Problema 3: Viga empotrada – simplemente apoyada

En el caso de una viga empotrada en un extremo y simplemente apoyada en el otro, se establecen, bajo las condiciones de contorno conocidas, definidas en (24) y (25), los requisitos para el desplazamiento, giro y el momento tanto al inicio como al final de la sección.

Este caso de implementación del problema 3, si bien el algoritmo se basa en una ecuación diferencial de segundo orden, que implicaría que solo se requieren dos condiciones de contorno para lograr su resolución. Al aplicar dos condiciones de contorno, una para la cara del empotramiento con giro cero y en el extremo simplemente apoyado con desplazamiento cero ($\frac{dw(0)}{dx} = 0$, $w(L) = 0$), obtenemos la siguiente Ilustración 16:

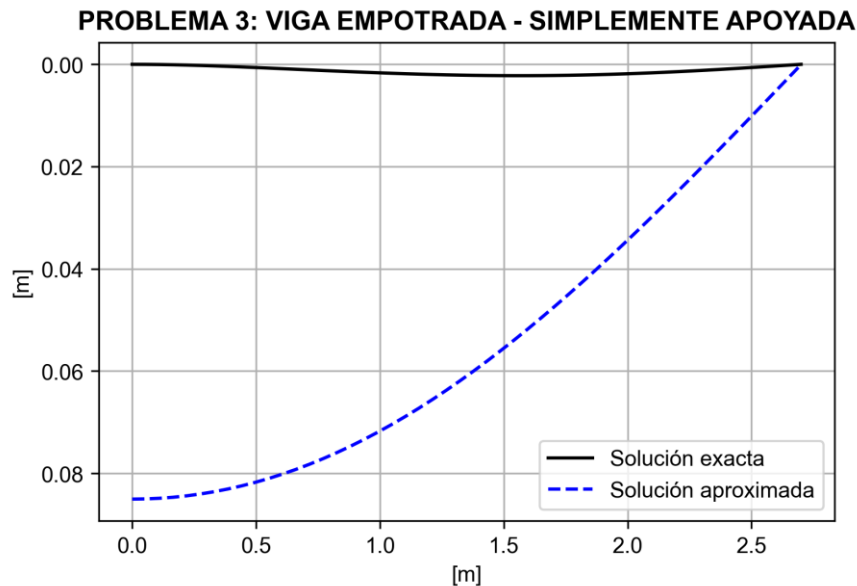


Ilustración 16. Error en convergencia a la solución real por falta de condiciones de contorno.

Aquí es evidente que en el extremo izquierdo se inicia con una curvatura igual a cero. Sin embargo, el desplazamiento, al carecer de restricciones, queda a discreción de la red para cumplir con las condiciones de contorno y minimizar el error. A pesar de que la red neuronal converge, se obtiene una respuesta incorrecta en comparación con la solución exacta. Por lo tanto, es imperativo agregar una tercera condición de contorno al problema.

La condición extra de contorno de acuerdo con la definición en (24) y (25) que se incluirá será de desplazamiento cero en la zona de la cara del apoyo ($w(0) = 0$), de esta manera en el código de Python 3:

```
x_0 = np.zeros((1, 1)) # Posición de contorno w(x_0)
x_1 = np.linspace(0, L, num_train_samples).reshape((-1,1)) # puntos de
colocación
x_2 = np.zeros((1, 1)) # Posición de contorno w(x_2)
x_3 = L*np.ones((1, 1)) # Posición de contorno w(x_3)

w_0 = 0*np.ones((1, 1)) # Condición de contorno w(0) = w_0
w_2 = 0*np.ones((1, 1)) # Condición de contorno w_x(0) = w_2
w_3 = 0*np.ones((1, 1)) # Condición de contorno w(L) = w_3
```

Dentro de las condiciones de contorno y la conformación de la función de pérdida que establecemos en la PINN para la viga empotrada – simplemente apoyada:

```
X0 = tf.stack([x0], axis=1) #Condición w(0) = 0
```

```

w_pred = self.model(X0) #Paso de condición X0 por la red
y0 = tf.reduce_mean(tf.square(w0 - w_pred)) #Penalty term de condición X0
f_w = self.net_f_wv() #Termino de energía para loss
yS = tf.reduce_mean(tf.square(f_w)) #Paso de colocation points por la red
X2 = tf.stack([x2], axis=1) #Condición  $w_x(L) = 0$ 
w_pred, w_x_pred, *_ = self.net_wv(X2) #Paso de condición X2 por la red
yB = tf.reduce_mean(tf.square(w_x_pred-w2)) #Penalty term de condición X2
X3 = tf.stack([x3], axis=1) #Condición  $u(L) = 0$ 
w_pred, *_ = self.net_wv(X3) #Paso de condición X3 por la red
yC = tf.reduce_mean(tf.square(w_pred-w3)) #Penalty term de condición X3

```

Donde la sumatoria y balanceo de los términos y_S , y_0 , y_B y y_C conforma la función de perdida, y y_C es el termino extra de penalización referido a la nueva condición de contorno.

Capítulo 5: Resultados y Discusiones

Se realizaron experimentos numéricos para estudiar la deformación de una viga bajo diferentes condiciones de carga y restricciones de apoyo. Se compararon las deformaciones obtenidas mediante la PINN con las soluciones analíticas conocidas, y se calcularon métricas de error para evaluar la precisión del modelo. Además, se realizaron análisis de sensibilidad para estudiar la influencia de diferentes parámetros del modelo en los resultados obtenidos.

Para los resultados presentados en esta sección se empleó un procesador Intel(R) Core(TM) i9-12900H para el procesamiento, mientras que la tarjeta gráfica GeForce RTX 3070 Ti aceleró los cálculos mediante el procesamiento de las redes neuronales. La implementación se realizó en Python 3.10.12, y para el aprendizaje profundo se utilizó TensorFlow en su versión 2.11.1. Esta configuración tecnológica permitió alcanzar resultados sólidos en la resolución de ecuaciones diferenciales variacionales con la metodología PINN.

Problema 1: Viga simplemente apoyada

Para el problema de valores de contorno en una viga simplemente apoyada ([Problema 1: Viga simplemente apoyada](#)), las condiciones de contorno implementadas en la PINN correspondieron al desplazamiento nulo en los extremos, permitiendo que los grados de libertad restantes (curvatura y momento) quedaran sin restricciones para la red neuronal.

Al emplear los hiperparámetros definidos previamente ([Hiperparámetros utilizados en la Physics-Informed Neural Networks](#)), que incluyen una arquitectura superficial de red con 2 capas ocultas y 20 neuronas cada una, función de activación tangente hiperbólica, y la inicialización de pesos con Xavier normal, junto con las épocas de entrenamiento específicas

para ambos algoritmos de optimización (500 iteraciones para Adam y 1000 iteraciones máximas para L-BFGS), se observa en la Ilustración 17 que el algoritmo converge sin necesidad de utilizar todas las iteraciones del segundo optimizador.



Ilustración 17. Historial de pérdidas para el problema 1

La convergencia de los resultados de la Red Neuronal Informada por la Física (PINN) se observan en la Ilustración 18 , donde se establece la relación entre la solución analítica y los resultados obtenidos por la red neuronal informada por la física. Esta relación se verifica a través de tres métricas: el coeficiente de determinación (R^2), el error cuadrático medio (MSE) en cada punto de colocación entre los resultados, y la deflexión máxima en la viga, en este caso ubicada en $L/2$.

En la Ilustración 18, se aprecia que el valor del error cuadrático medio (MSE) es de $1.2e-10$, una expresión que puede ser considerada prácticamente nula. En la posición $L/2$, la deflexión exacta es de $5.33e-3$ m, mientras que la deflexión aproximada mediante la PINN es de $5.35e-3$ m, lo que equivale a un error del 0.41% entre ambos valores.

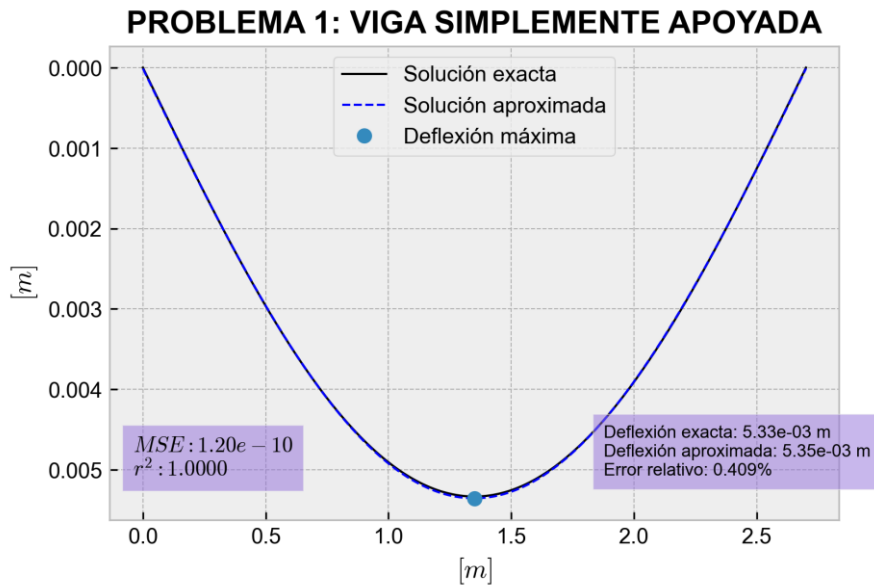


Ilustración 18. Resultados de convergencia de la red para el problema 1

Además, al comparar los resultados analíticos con los obtenidos a través del método numérico PINN, se utiliza un QQ Plot, tal como se muestra en la Ilustración 19. En esta figura se observa que los puntos forman una línea recta a 45 grados entre ambos conjuntos de resultados.

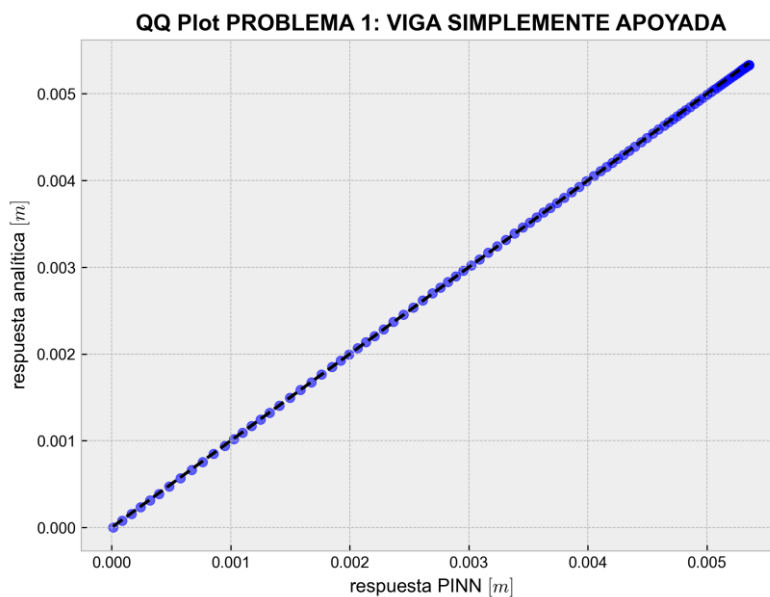


Ilustración 19. QQ Plot entre resultado analítico y resultado de la PINN del problema 1

Problema 2: Viga empotrada – libre

Para el [Problema 2: Viga empotrada – libre](#) las condiciones de contorno utilizadas en la PINN correspondieron tanto al desplazamiento como el giro nulo en el extremo empotrado. Esto permitió que los grados de libertad restantes en el otro extremo de la viga fueran capturados por la red neuronal.

Al aplicar los [Hiperparámetros utilizados en la Physics-Informed Neural Networks](#), al igual que en sección del problema anterior (arquitectura, inicialización de pesos, etc.), en este problema se estableció una configuración de épocas de entrenamiento específica para ambos algoritmos de optimización (500 iteraciones para Adam y 1000 iteraciones máximas para L-BFGS). Observamos en la Ilustración 20 que el algoritmo converge utilizando todas las iteraciones del optimizador L-BFGS.

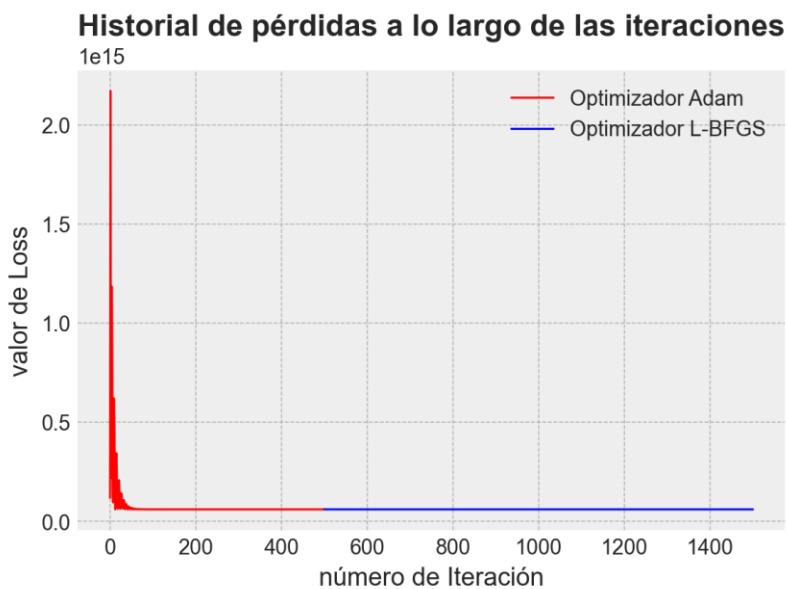


Ilustración 20. Historial de pérdidas para el problema 2.

En este escenario, al ejecutar el algoritmo durante diez iteraciones, se obtuvieron los resultados de la Ilustración 21.a en seis ocasiones, mientras que en cuatro ocasiones se obtuvieron los resultados de la Ilustración 21.b. Donde el error cuadrático medio aumenta de $7.32e-8$ a $7.12e-7$, es perceptible a simple vista que en la Ilustración 21.b la curva aproximada que representa la deflexión no se ajusta completamente a la curva de la solución exacta en toda la discretización de puntos.

La deflexión máxima en este problema se produce en $x=L$, punto que se obtiene al encontrar donde curvatura sea máxima o mínima. Para la Ilustración 21.a el error de la deflexión entre la solución exacta y la solución aproximada mediante PINN es de 1% y se puede observar que es mayor al error de la Ilustración 21.b de 0.68%, efecto contrario a los resultados de MSE para cada figura, de manera que es notoria la necesidad de un mejor ajuste de la red neuronal.

En el presente problema, la máxima deflexión ocurre en $x=L$, identificando el punto donde la curvatura es máxima o mínima mediante derivación de la ecuación de deflexión (ecuación (23)). En la Ilustración 21.a, el error en la deflexión entre la solución exacta y la aproximada por la PINN es del 1%, siendo mayor que el error en la Ilustración 21.b, que es del 0.68%. Este contraste en los errores se presenta en sentido opuesto a los resultados del MSE para cada figura. Esto subraya la necesidad evidente de lograr un mejor ajuste de la red neuronal en este caso.

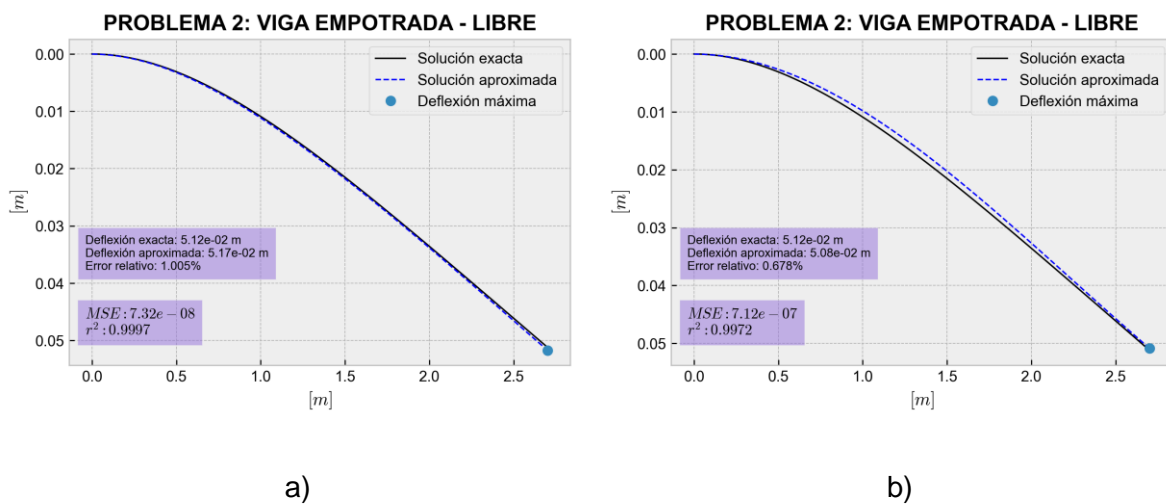


Ilustración 21. Variación de resultados del problema 2 con 1000 iteraciones máximas para L-BFGS y 500 iteraciones para Adam.

En otro escenario, al aumentar las iteraciones de los optimizadores a 1000 en el caso de Adam y 1500 las máximas iteraciones de L-BFGS para diez repeticiones del algoritmo se observa en la Ilustración 22 que de igual manera que en el escenario planteado anteriormente, el algoritmo converge al pasar 2500 iteraciones.



Ilustración 22. Historial de pérdidas para el problema 2

Para el incremento de iteraciones en los optimizadores, en la Ilustración 23 se puede observar una mejoría en términos de convergencia ya que en las diez repeticiones del algoritmo no fue perceptible a simple vista una desviación en la trayectoria de la deflexión aproximada por la red neuronal. De igual manera el error cuadrático medio entre las dos soluciones disminuyó a $8.95e-9$ al igual que el error en la deflexión en $x = L$ pasando a 0.19%

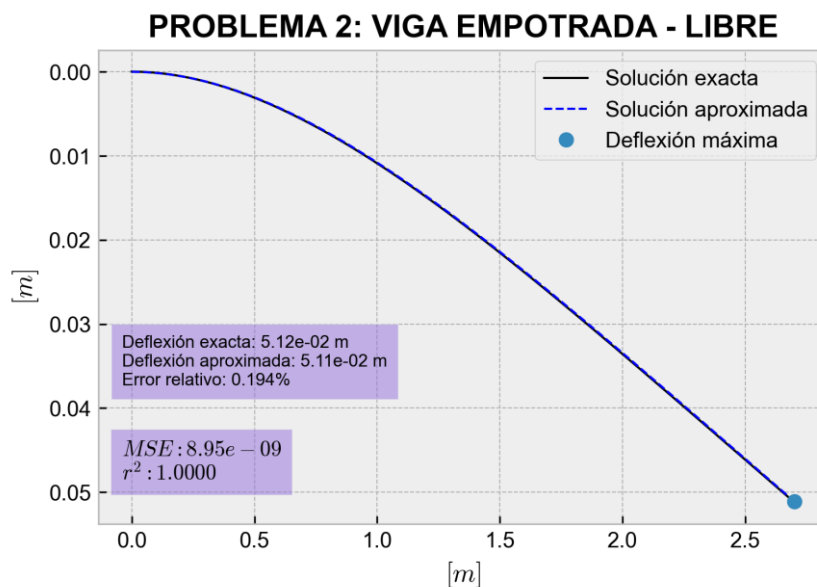


Ilustración 23. Resultados del problema 2 con 1500 iteraciones máximas para L-BFGS y 1000 iteraciones para Adam.

Los resultados presentados en la Ilustración 23 demuestran la capacidad de mejorar la convergencia hacia un resultado más preciso, en línea con las leyes físicas incorporadas en el modelo de la red neuronal, al ajustar ciertos hiperparámetros. En este caso específico, se logró este efecto al modificar el número de iteraciones de los optimizadores.

Adicionalmente, en la comparación de la deflexión y el MSE de los resultados analíticos y aquellos obtenidos mediante el método numérico PINN, se emplea un gráfico QQ Plot, que se presenta en la Ilustración 24. En esta representación gráfica, se puede apreciar cómo los puntos enfrentando las distribuciones analíticas y aproximadas se disponen en una línea diagonal de 45 grados, estableciendo una relación directa entre los dos conjuntos de resultados.

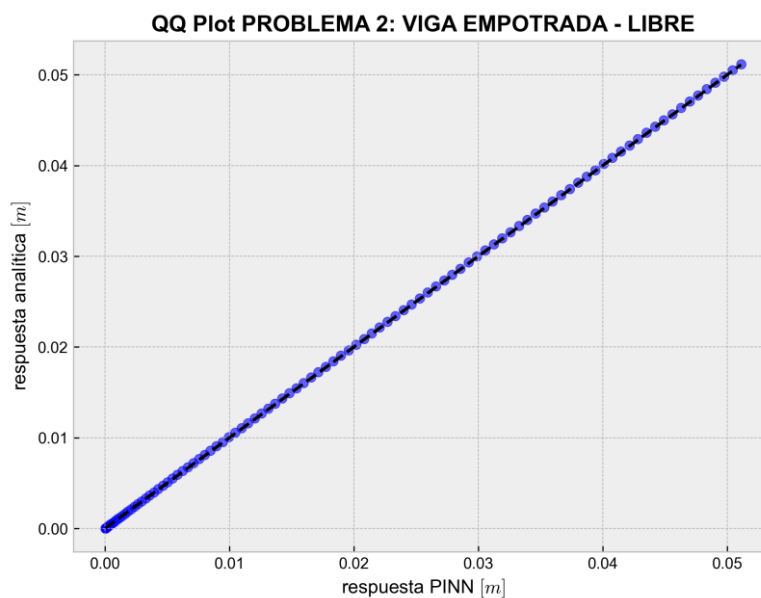


Ilustración 24. QQ Plot entre resultado analítico y resultado de la PINN del problema 2

Problema 3: Viga empotrada – simplemente apoyada

Para el [Problema 3: Viga empotrada – simplemente apoyada](#) las condiciones de contorno utilizadas en la PINN correspondieron al desplazamiento nulo en ambos extremos de la viga, así como curvatura nula en la cara del apoyo (como se especifica en la sección de problemas de aplicación en el caso del [Problema 3](#)).

Al igual que en los problemas anteriores se aplican los [Hiperparámetros utilizados en la Physics-Informed Neural Networks](#), ya establecidos. Observamos en la Ilustración 25 que el algoritmo converge sin la necesidad de utilizar todas las iteraciones planteadas (500 para Adam y 1000 para L-BFGS).

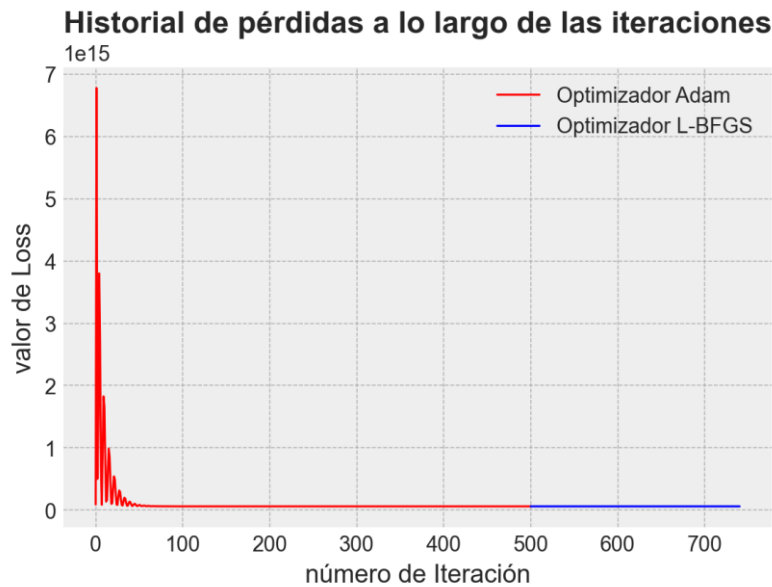


Ilustración 25. Historial de pérdidas para el problema 3

A partir de la convergencia exitosa de la red neuronal se obtienen las graficas de la Ilustración 26, donde es notoria la diferencia entre los resultados de la distribución exacta y la calculada por el metodo numerico PINN, este comportamiento se discutio en el apartado [Modelado del sistema](#) y se demostro graficamente en la Ilustración 12. De manera que se propone un aumento en los valores escalares para el ajuste de la función de perdida en los terminos de penalización (Actualmente 2 E/l).

PROBLEMA 3: VIGA EMPOTRADA - SIMPLEMENTE APOYADA

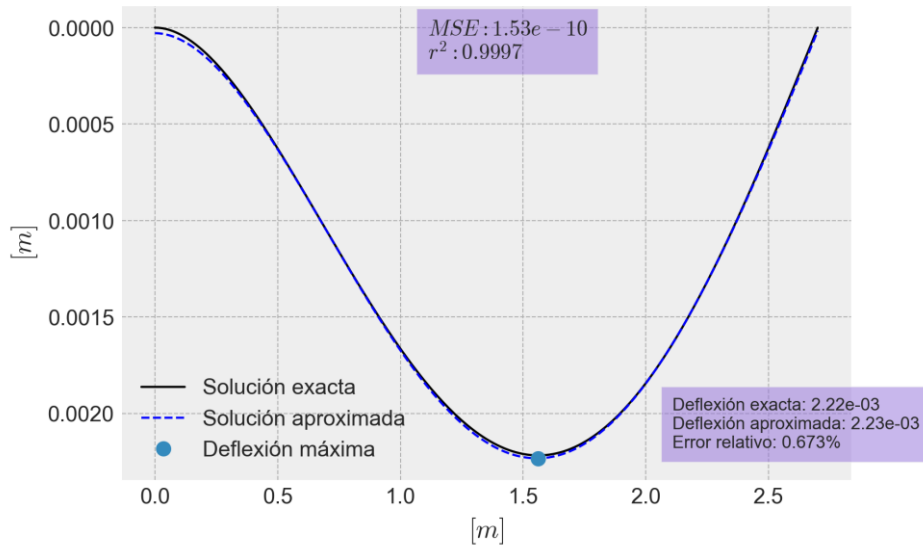


Ilustración 26. Resultados de convergencia de la red para el problema 3 con ajuste de loss = 2E/I

Al ajustar los valores escalares que acompañan a los términos de penalización a 5E/I, los resultados mostrados en la Ilustración 27 revelan una mejora comparada con la Ilustración 26. En específico, se observa una mejoría en el valor del MSE, disminuyendo de 1.53e-10 a 1.08e-10. Además, el error relativo en la deflexión máxima se reduce de 0.673% a 0.186% (donde la deflexión máxima se presenta en $x = -L * \frac{\sqrt{33}-15}{16} \approx 0.578 * L$) destacando la importancia de optimizar estos parámetros para una respuesta más precisa del modelo. Se reconoce la necesidad de futuras investigaciones para abordar posibles limitaciones y mejorar aún más el rendimiento del sistema modelado.

PROBLEMA 3: VIGA EMPOTRADA - SIMPLEMENTE APOYADA

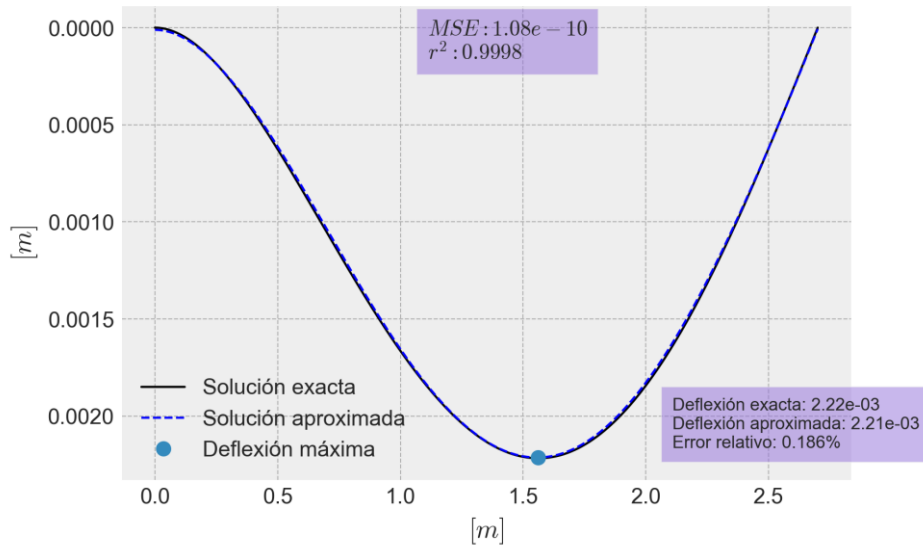


Ilustración 27. Resultados de convergencia de la red para el problema 3 con ajuste de $loss = 5E/l$

Al igual que en las discusiones de los anteriores problemas, se emplea un gráfico QQ Plot (Ilustración 28) para comparar los resultados analíticos con los obtenidos mediante el método numérico PINN. Este gráfico muestra cómo las distribuciones de ambos conjuntos se relacionan en una línea diagonal de 45 grados.

QQ Plot PROBLEMA 3: VIGA EMPOTRADA - SIMPLEMENTE APOYADA

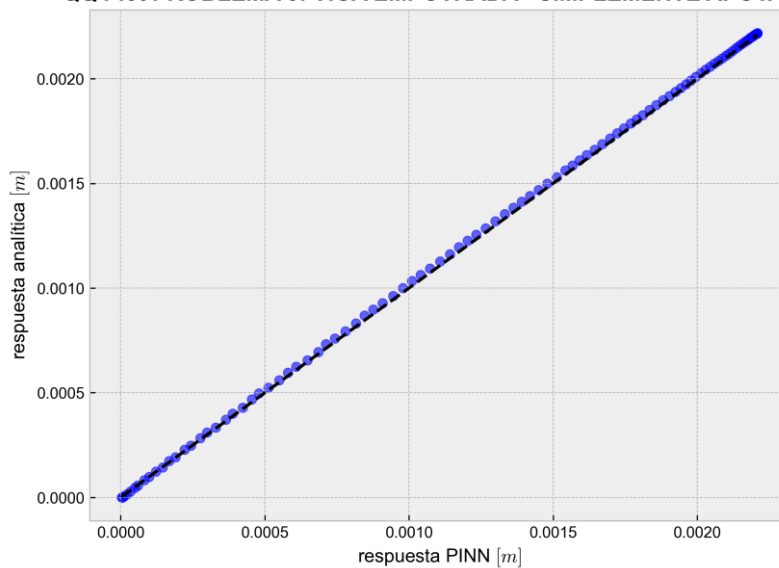


Ilustración 28. Q-Q Plot entre resultado analítico y resultado de la PINN del problema 3

Sensibilidad de hiperparámetros

Arquitectura de la red

A continuación, se presenta el resultado de la variación en la densidad de la red, pasando de una capa ancha con 256 neuronas a capas con 20 neuronas cada una, mientras se mantiene la configuración superficial de la red. Estos resultados se basan en las condiciones establecidas en el problema 1 previamente planteado.

<p>PROBLEMA 1: VIGA SIMPLEMENTE APOYADA</p> <p>— Solución exacta - - - Solución aproximada ● Deflexión máxima</p> <p>MSE: $1.37e-10$ r^2: 0.9999</p> <p>Deflexión exacta: $5.33e-03$ m Deflexión aproximada: $5.35e-03$ m Error relativo: 0.406%</p> <p>a) Resultado de la red con capas anchas</p>	<p>PROBLEMA 1: VIGA SIMPLEMENTE APOYADA</p> <p>— Solución exacta - - - Solución aproximada ● Deflexión máxima</p> <p>MSE: $1.11e-10$ r^2: 1.0000</p> <p>Deflexión exacta: $5.33e-03$ m Deflexión aproximada: $5.34e-03$ m Error relativo: 0.228%</p> <p>b) Resultado de la red con capas estrechas</p>
<p>layers = [1, 256, 256, 1]</p>	<p>layers = [1, 20, 20, 1]</p>
<p>Tiempo de entrenamiento promedio en 5 iteraciones del código.</p>	
<p>40.83 s</p>	<p>15.72 s</p>

Tabla 1. Comparación entre resultados de convergencia correcta de la red con distinta densidad de capas.

La Tabla 1 compara las convergencias de la red neuronal informada por la física bajo diferentes densidades de capas en una configuración de red superficial. Aunque los resultados de la red neuronal son similares en términos del coeficiente de determinación y el error cuadrático medio a lo largo de los puntos de colocación, resulta crucial comparar los tiempos de entrenamiento en la red neuronal.

En el caso de la Tabla 1.a, donde el tiempo promedio es de 40.83 segundos, se observa que el tiempo más corto de convergencia de la red es de 15.76 segundos, mientras que el más largo es de 77.58 segundos. En la Tabla 1.b, el tiempo de convergencia más breve es de 4.45

segundos, mientras que el más largo es de 26.39 segundos. Aunque estos tiempos son relativos a las especificaciones de la computadora, es notable la mejora en el tiempo al optimizar el hiperparámetro de la densidad de capas en las redes neuronales, sin comprometer la precisión de la respuesta de la PINN.

Bajo el caso del problema 3, se llevó a cabo una variación en el número de capas, alternando entre una configuración de red profunda y una superficial, manteniendo en ambos casos el mismo número de neuronas por capa (256). El número de iteraciones para el optimizador Adam se mantuvieron en 500 y las iteraciones máximas para L-BFGS se establecieron en 1000.

<p>PROBLEMA 3: VIGA EMPOTRADA - SIMPLEMENTE APOYADA</p> <p>a) Resultado para red superficial</p>	<p>PROBLEMA 3: VIGA EMPOTRADA - SIMPLEMENTE APOYADA</p> <p>b) Resultado para red profunda</p>
<p>layers = [1, 256, 256, 1]</p>	<p>layers = [1, 256, 256, 256, 1]</p>
<p>Tiempo de entrenamiento promedio en 5 iteraciones del código.</p>	
<p>30.71 s</p>	<p>148.56 s</p>
<p>Error entre solución exacta y aproximada</p>	
<p>0.124%</p>	<p>0.127%</p>

Tabla 2. Comparación entre resultados de convergencia correcta de distintas distribuciones de red.

De manera similar a cuando se modifica la densidad de las capas en la red, agregar capas a la configuración de la red y alternar entre arreglos superficiales y profundos afecta el tiempo de entrenamiento y convergencia de la red.

En la Tabla 2.a, se observa que el tiempo promedio para 5 iteraciones del programa es aproximadamente de 30.71 segundos con una disposición superficial de la red, utilizando dos capas de 256 neuronas cada una. Por otro lado, en la Tabla 2.b, con una disposición profunda, el tiempo de convergencia de la red se incrementa significativamente a 148.56 segundos. Sin embargo, esta prolongación del tiempo de convergencia no conlleva una mejora en el ajuste en términos del coeficiente de determinación y el error entre la deflexión máxima calculada analíticamente y la obtenida mediante PINNs.

Este contraste en los tiempos de optimización es notable. Aunque se trata de una ecuación diferencial de segundo orden aparentemente sencilla, la optimización de este parámetro puede ser crucial en investigaciones futuras con ecuaciones que presenten mayor complejidad para su resolución computacional.

Capítulo 6: Conclusiones

Desarrollo de la formulación matemática

El objetivo del trabajo de titulación es comprobar la validez de incluir conocimiento físico desde un planteamiento variacional en una red neuronal aplicándolo en un caso puntual de la mecánica de sólidos, específicamente en vigas bajo la teoría de Euler-Bernoulli. El procedimiento considerado fue desarrollado por *Samaniego et al.* [14]. Este considera que bajo la teoría del planteamiento variacional se requiere la minimización de un funcional para obtener una solución aproximada, mientras que, el proceso de entrenamiento de una red neuronal se da al minimizar la función de pérdida. Estos conceptos se entrelazan y permiten obtener la solución de un fenómeno físico generalmente representado mediante una ecuación diferencial y sus respectivas condiciones iniciales y de contorno.

El planteamiento de la función de pérdida bajo un planteamiento variacional considera como funcional a la energía potencial total de la viga que cumple con el principio de mínima energía potencial total. Se adiciona un término de penalización (T) que contiene las condiciones de contorno cinemáticas propias de cada problema analizado. La función de pérdida requiere ser balanceada empíricamente (28) para que la diferencia de unidades entre los términos sea equiparada, de forma que, las condiciones de contorno adquieran la importancia necesaria limitando así el espacio de soluciones. Sin embargo, se requirió modificar una vez más dicha función pues el resultado del problema analizado era nulo.

La siguiente modificación se asume que se debe a la complejidad del espacio en el cual se obtiene la solución aproximada, es por ello, que en primera estancia se consideró lo mencionado por *Roy K et al.* [33] donde es posible la existencia de barreras energéticas que limiten la obtención del resultado, es decir, se llega a un mínimo local en el proceso de optimización y no a un mínimo global que representa la solución aproximada del sistema, y que una posible solución a dicho problema es adicionar energía de disipación. En el presente caso corresponde a la energía interna del sistema (U), permitiendo de tal forma que durante el proceso de aprendizaje la minimización no converja estrictamente a cero sino toma el valor de la perturbación elevado al cuadrado.

En resumen, dentro de la formulación matemática se considera el término de penalización, el valor del término extra de disipación, y el balanceo de la función de pérdida como factores que influyen directamente en la convergencia del modelo y que su adición y calibración es necesaria para la obtención de resultados precisos acordes a las restricciones impuestas por la física introducida en la red neuronal que correspondan al problema de viga estudiado.

La configuración del término de penalización está ligada a las condiciones de contorno, que en el presente caso se esperaría la necesidad de dos restricciones pues responden a la formulación variacional de segundo orden. Tanto en el primer como en el segundo problema se cumple dicho principio, pero en el tercer problema requiere de tres restricciones de acuerdo con las condiciones de apoyo y de carga. Entonces el número de restricciones consideradas en el término de penalización no está estrictamente determinado por el orden de la ecuación analizada sino la configuración del sistema analizado.

Tanto el proceso del balanceo de la función de pérdida como la adición del valor extra de disipación se realizaron de manera empírica en virtud de la convergencia a la solución. La convergencia se da cuando el escalar correspondiente a las condiciones de contorno tome valores a partir $\lambda_k = 2 \frac{E}{I}$ y cuando el valor extra de disipación sea mayor a $0.5 EI$ para el caso del Problema 1 y problema 2, con una variación para el escalar a $\lambda_k = 5 \frac{E}{I}$ en el problema 3. Existe una relación de correspondencia entre dichos valores, puesto que al variarlos de manera exagerada no es posible la convergencia del modelo a la solución esperada.

Dentro del balanceo de la función de pérdida el valor del escalar correspondiente a las condiciones de contorno permite la convergencia del modelo. Mientras mayor sea la complejidad del sistema analizado se requiere un mayor énfasis en las restricciones. Este es

el caso del tercer problema que requiere un escalar de 5 E/I en comparación con los primeros dos problemas que llegan a una convergencia a partir de un escalar a partir de 2 E/I .

Desarrollo de la implementación

La implementación del sistema bajo la inclusión del conocimiento físico dentro del proceso de aprendizaje de una red neuronal requiere de la calibración de los hiperparámetros en virtud de la efectividad de la convergencia del problema analizado. De manera general, entre los factores de mayor influencia tenemos: la arquitectura de la red, la función de activación, los algoritmos de optimización, las épocas de entrenamiento y tasa de aprendizaje, la inicialización de pesos, y los puntos de colocación.

De forma general, en la [Arquitectura de la red](#) se analizó la influencia del tipo de red neuronal ya sea superficial o profunda, así como de la densidad de las capas ocultas. En cuanto al tipo de red neuronal, se opta por una red superficial de dos capas ocultas debido a la simplicidad de la ecuación diferencial analizada para lo cual se prioriza evitar el sobreajuste de la red. Esto se ejemplifica mediante las condiciones del problema 3, y se obtiene que el tiempo de convergencia para una red neuronal superficial es menor. En cuanto a la densidad de las capas ocultas se opta por una red no densa con 20 neuronas completamente conectadas. Esto se muestra para las condiciones del problema 1, y resulta que el tiempo de entrenamiento es menor con resultados similares. Tanto para la selección de tipo de red neuronal como densidad de las capas ocultas no existe un cambio significativo en el ajuste en términos del coeficiente de determinación y en el error con relación a la solución exacta, solamente existe variación en el tiempo de convergencia, que en ecuaciones con relaciones más complejas entre variables puede suponer una mejora en la prueba y error de los ajustes de hiperparámetros.

La *función de activación* utilizada es la tangente hiperbólica. Esta introdujo la no linealidad en el modelo y por ende en el espacio de aproximación, de tal forma que, permitió a la red neuronal aprender y representar las características de los casos de las vigas analizados. A pesar de que la función objetivo es lineal su uso favoreció en el proceso de convergencia principalmente por su adaptabilidad con los algoritmos de optimización basados en gradientes.

Puesto que el problema variacional se aproxima en un espacio de dimensión finita, se convierte en un problema de optimización para lo cual se hace uso del método de descenso

de gradiente estocástico junto con *algoritmos de optimización* sucesivos, el optimizador Adam y el optimizador L-BFGS respectivamente. El primer optimizador se incluye con el objetivo de superar los mínimos locales, mientras que, el segundo optimizador permite calcular la dirección de optimización. En conjunto favorecen la obtención de la solución correspondiente al mínimo global, es decir, la solución aproximada esperada. Ligado al funcionamiento de los optimizadores, se requiere definir el número de *épocas de entrenamiento* requeridas que pueden variar según el caso de viga analizado. Con la variación de estos hiperparámetros se obtuvo un correcto resultado para los tres casos de viga planteados.

La velocidad de convergencia y la eficiencia del modelo está influenciada por la *inicialización de pesos*, razón por la cual se utiliza una inicialización Glorot Normal que es compatible con funciones de activación no lineales, y además permite que sean hallados los gradientes durante el proceso de optimización al mantener en equilibrio de la varianza. Además de la inicialización de pesos, el número de *puntos de colocación* juega un papel importante en la eficiencia del método. Esto se debe a que el método energético lleva el proceso de entrenamiento con todos los puntos de colocación reconociendo que un aumento del número de puntos de colocación conlleva a un incremento en el valor del error cuadrático medio, es por ello por lo que se escogió de manera empírica que sea un número de 181 con el único objetivo de comparación entre la deflexión aproximada y la deflexión analítica.

En síntesis, durante la implementación de un sistema que fusiona el conocimiento físico en el aprendizaje de redes neuronales, se vuelve primordial adaptar los hiperparámetros con precisión, según los requisitos emergentes o específicos del problema analizado. Este ajuste cobra vital importancia para alcanzar una convergencia eficiente en el análisis del problema, dado que este enfoque carece de una secuencia de pasos predeterminada.

Comparación implementación de PINN con método analítico

El comportamiento de las vigas bajo la teoría de Euler-Bernoulli en su formulación fuerte se presenta como una ecuación diferencial lineal ordinaria de cuarto orden. Para su resolución se cuenta con el método analítico que requiere de cuatro condiciones de contorno para obtener la solución exacta. En el apartado de composición del sistema se expone la solución exacta cada problema planteado ([Problema 1: Viga simplemente apoyada](#), [Problema 2: Viga empotrada – libre](#), [Problema 3: Viga empotrada – simplemente apoyada](#)).

Problema 1: Viga simplemente apoyada

En el primer problema las métricas usadas nos indican que la configuración del modelo tanto en el enfoque matemático como en el enfoque de la implementación contribuyen a una convergencia casi perfecta. El modelo llega a su convergencia antes que el segundo optimizador utilizado L-BFGS cumpla las iteraciones máximas planteadas. Esto también se evidencia en la invariación de los resultados en corridas de código sucesivas. Esta situación se contribuye a la simplicidad del problema, pues el espacio de soluciones es limitado bajo las restricciones de desplazamiento en los dos extremos de la viga.

Problema 2: Viga empotrada – libre

En el segundo problema bajo la configuración general explicada anteriormente se observó variaciones en la convergencia durante las corridas del código. Es decir, al utilizar mismo número de iteraciones la red no convergió completamente, presentó errores que al aumentar el número de iteraciones de los optimizadores desaparecieron de tal forma que durante las corridas del código también desapareció la variabilidad entre soluciones y el modelo convergió. Por tanto, las métricas que comparan la solución obtenida con la solución exacta mejoran de un error relativo del orden del 0.6 – 1% al 0.19%. La necesidad de incrementar el número de iteraciones está ligada a la complejidad de las restricciones, puesto que en el presente problema se requiere restricciones en giro y desplazamiento en un solo extremo quedando a discreción de la red neuronal el desplazamiento del extremo libre.

Problema 3: Viga empotrada – simplemente apoyada

En el tercer problema bajo la configuración general explicada anteriormente se observó que en los extremos de la viga la solución aproximada presentaba variaciones con la solución exacta. Esto se solucionó al utilizar escalares más fuertes en el proceso del balanceo de la función de pérdida. Con tal modificación el modelo convergió a la solución incluso sin utilizar todas las iteraciones del optimizador. Aumentar el valor del escalar correspondiente a las condiciones de contorno en la función de pérdida con el objetivo de mejorar la convergencia se contribuye a las restricciones que debe cumplir el modelo para representar las condiciones de apoyo en los extremos de la viga.

Bibliografía

- [1] M. Bazmara, M. Silani, M. Mianroodi, and M. sheibanian, “Physics-informed neural networks for nonlinear bending of 3D functionally graded beam,” *Structures*, vol. 49, pp. 152–162, Mar. 2023, doi: 10.1016/j.istruc.2023.01.115.
- [2] A. Díaz Díaz, R. Castañeda Balderas, A. F. Domínguez Alvarado, and C. I. Martínez Morfín, “Soluciones de Ecuaciones Diferenciales por Elemento Finito (SEDEF),” *Ingeniería Investigación y Tecnología*, vol. 21, no. 1, pp. 1–11, Jan. 2020, doi: 10.22201/fi.25940732e.2020.21n1.002.
- [3] V. jagota, A. Sethi, and D.-K. Kumar, “Finite Element Method: An Overview,” *Walailak J Sci Technol*, vol. 10, pp. 1–8, Jun. 2013, doi: 10.2004/wjst.v10i1.499.
- [4] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” vol. 378, pp. 686–707, 2018, doi: 10.1016/j.jcp.2018.10.045.
- [5] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, May 2021, doi: 10.1038/s42254-021-00314-5.
- [6] D. Katsikis, A. D. Muradova, and G. E. Stavroulakis, “A Gentle Introduction to Physics-Informed Neural Networks, with Applications in Static Rod and Beam Problems,” vol. 9, pp. 103–128, 2022, doi: 10.15377/2409-5761.2022.09.8.
- [7] E. Haghghat, M. Raissi, A. Moure, H. Gomez, and R. Juanes, “A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics,” *Comput Methods Appl Mech Eng*, vol. 379, p. 113741, Jun. 2021, doi: 10.1016/j.cma.2021.113741.
- [8] Z. K. Lawal, H. Yassin, D. T. C. Lai, and A. Che Idris, “Physics-Informed Neural Network (PINN) Evolution and Beyond: A Systematic Literature Review and Bibliometric Analysis,” *Big Data and Cognitive Computing*, vol. 6, no. 4, p. 140, Nov. 2022, doi: 10.3390/bdcc6040140.

- [9] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific Machine Learning Through Physics-Informed Neural Networks: Where we are and What's Next," *J Sci Comput*, vol. 92, no. 3, p. 88, Sep. 2022, doi: 10.1007/s10915-022-01939-z.
- [10] Henderson I., "Physics Informed Neural Networks (PINNs): An Intuitive Guide," Towards Data Science. Accessed: Jun. 13, 2023. [Online]. Available: <https://towardsdatascience.com/physics-informed-neural-networks-pinns-an-intuitive-guide-fff138069563>
- [11] H. Wang and Q.-H. Qin, "Mechanics of solids and structures," in *Methods of Fundamental Solutions in Solid Mechanics*, Elsevier, 2019, pp. 53–90. doi: 10.1016/B978-0-12-818283-3.00002-6.
- [12] X. O. Olivella and C. Agelet de Saracibar, *Mecánica de medios continuos para ingenieros*, vol. 92. 2002.
- [13] W. Li, M. Z. Bazant, and J. Zhu, "A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches," *Comput Methods Appl Mech Eng*, vol. 383, p. 113933, Sep. 2021, doi: 10.1016/j.cma.2021.113933.
- [14] E. Samaniego *et al.*, "An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications," *Comput Methods Appl Mech Eng*, vol. 362, p. 112790, Apr. 2020, doi: 10.1016/j.cma.2019.112790.
- [15] Ready J. N., *Energy Principles and Variational Method*. Texas, 2002.
- [16] E. Oñate, "Flexión de Vigas," in *Cálculo de Estructuras por el Método de Elementos Finitos*, 1992.
- [17] E. Suhir, *Structural Analysis in Microelectronic and Fiber-Optic Systems*. Dordrecht: Springer Netherlands, 1991. doi: 10.1007/978-94-011-6535-8.
- [18] Diprima R. and Boyce W., *Ecuaciones diferenciales*. Mexico: Limusa, 2009.

- [19] O. R. Bosede, "On Some Numerical Methods for Solving Initial Value Problems in Ordinary Differential Equations," *IOSR Journal of Mathematics*, vol. 1, no. 3, pp. 25–31, 2012, doi: 10.9790/5728-0132531.
- [20] B. Denis, "An Overview of Numerical and Analytical Methods for solving Ordinary Differential Equations," Dec. 2020.
- [21] E. Aldabas-Rubira, "Introducción al reconocimiento de patrones mediante redes neuronales," 2002, Accessed: Jun. 11, 2023. [Online]. Available: <http://www.jcee.upc.es/JCEE2002/Aldabas.pdf>
- [22] L. Cruz Varona, "Paralelización del entrenamiento de redes neuronales en sistemas heterogéneos," Jul. 2021, Accessed: Jun. 11, 2023. [Online]. Available: <http://hdl.handle.net/10902/22828>
- [23] E. Kharazmi, Z. Zhang, and G. E. M. Karniadakis, "hp-VPINNs: Variational physics-informed neural networks with domain decomposition," *Comput Methods Appl Mech Eng*, vol. 374, p. 113547, Feb. 2021, doi: 10.1016/j.cma.2020.113547.
- [24] S. Wang, X. Yu, and P. Perdikaris, "When and why PINNs fail to train: A neural tangent kernel perspective," *J Comput Phys*, vol. 449, p. 110768, Jan. 2022, doi: 10.1016/j.jcp.2021.110768.
- [25] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," *Science (1979)*, vol. 367, no. 6481, pp. 1026–1030, Feb. 2020, doi: 10.1126/science.aaw4741.
- [26] J. Han, A. Jentzen, and W. E., "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505–8510, Aug. 2018, doi: 10.1073/pnas.1718942115.
- [27] A. F. Psaros, K. Kawaguchi, and G. E. Karniadakis, "Meta-learning PINN loss functions," *J Comput Phys*, vol. 458, p. 111121, Jun. 2022, doi: 10.1016/j.jcp.2022.111121.

- [28] W. Li, M. Z. Bazant, and J. Zhu, "A physics-guided neural network framework for elastic plates: Comparison of governing equations-based and energy-based approaches," *Comput Methods Appl Mech Eng*, vol. 383, p. 113933, Sep. 2021, doi: 10.1016/j.cma.2021.113933.
- [29] S. Markidis, "The Old and the New: Can Physics-Informed Deep-Learning Replace Traditional Linear Solvers?," *Front Big Data*, vol. 4, Nov. 2021, doi: 10.3389/fdata.2021.669097.
- [30] S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient pathologies in physics-informed neural networks," Jan. 2020.
- [31] N. Sukumar and A. Srivastava, "Exact imposition of boundary conditions with distance functions in physics-informed deep neural networks," *Comput Methods Appl Mech Eng*, vol. 389, p. 114333, Feb. 2022, doi: 10.1016/j.cma.2021.114333.
- [32] M. Komperød, "Deriving the Beam Equation using the Minimum Total Potential Energy Principle and Solving the Equation Numerically," Nov. 2018, pp. 65–72. doi: 10.3384/ecp1815365.
- [33] K. Roy, S. Kar, and R. N. Das, *Understanding the Basics of QSAR for Applications in Pharmaceutical Sciences and Risk Assessment*. Elsevier, 2015. doi: 10.1016/C2014-0-00286-9.
- [34] Bischof R., "10 Useful Hints and Tricks for Improving Physics-Informed Neural Networks (PINNs)," Towards Data Science.
- [35] R. Bischof and M. Kraus, "Mixture-of-Experts-Ensemble Meta-Learning for Physics-Informed Neural Networks," Aug. 2022.
- [36] S. K. Kumar, "On weight initialization in deep neural networks," Apr. 2017.

Anexos

Anexo A. Código fuente para replicación

Los códigos necesarios para la replicación de los resultados del presente trabajo de titulación se encuentran en el siguiente enlace:

[PabloMolineros/PINN_VARIACIONAL_TESIS \(github.com\)](https://github.com/PabloMolineros/PINN_VARIACIONAL_TESIS)

<https://1drv.ms/f/s!AhtZY3Dab86xhOFegaQBqG86PidQcA?e=iEw9Y9>

