



PHYSICS-INFORMED NEURAL NETWORKS (PINN)

Oscilador Armónico Subamortiguado 1D



24 DE JUNIO DE 2025

JON GIL ALVARO

1.-¿Qué son las PINNs?

Las Physics-Informed Neural Networks (PINNs) son una clase de redes neuronales diseñadas no solo para ajustarse a datos observados, sino también para incorporar información física del sistema que se está modelando, usualmente en forma de ecuaciones diferenciales.

En lugar de aprender una función únicamente a partir de un conjunto de datos de entrenamiento (como hace una red neuronal tradicional, o RNA), una PINN también se entrena para satisfacer una ecuación física que gobierna el sistema. Esto se consigue añadiendo un término de penalización al error total del modelo, que mide cuánto incumple la red la ecuación física.

2.-Casos de uso

A continuación, se van a presentar casos en los que las PINNs son útiles:

- Se dispone de pocos datos medidos, pero se conoce bien la ecuación física que describe el fenómeno.
- Se desea obtener soluciones numéricas aproximadas a ecuaciones diferenciales, incluso en dominios donde las soluciones analíticas no están disponibles.
- Se quiere estudiar problemas inversos, es decir, deducir parámetros del sistema a partir de observaciones parciales.

3.-Desarrollo del Problema: Oscilador Armónico Subamortiguado 1D

3.1.- Objetivo del Estudio

Este trabajo tiene como objetivo comparar el rendimiento de dos enfoques de redes neuronales: una red neuronal artificial clásica (RNA) y una Physics-Informed Neural Network (PINN), aplicadas al problema del oscilador armónico amortiguado en una dimensión. La comparación se centra especialmente en cómo ambas redes ajustan la solución del sistema cuando se dispone de una cantidad muy limitada de datos de entrenamiento.

3.2.- Formulación física

En este proyecto se ha planteado el problema físico de un oscilador armónico amortiguado en 1D en régimen subamortiguado (es decir, con un coeficiente de amortiguamiento menor que la frecuencia natural del sistema).

La ecuación diferencial ordinaria (ODE) que describe este sistema es:

$$m \frac{d^2 x}{dt^2} + \mu \frac{dx}{dt} + kx = 0$$

Cuyas condiciones iniciales son:

$$x(0) = 1$$

$$\frac{dx}{dt} = 0$$

El régimen será subamortiguado debido a que:

$$\delta < w_0$$

Siendo el valor de los mismos:

$$\delta = \frac{\mu}{2m}$$

$$w_0 = \sqrt{k/m}$$

3.3.- Planteamiento para resolver el problema

1. Se define la solución exacta basada en la forma conocida para el caso subamortiguado.

$$x(t) = e^{-\delta t} (2A \cdot \cos(\phi + wt))$$

Siendo la frecuencia angular del sistema amortiguado:

$$w = \sqrt{w_0^2 - \delta^2}$$

La fase inicial:

$$\phi = \arctan(-\delta/w)$$

La amplitud ajustada:

$$A = \frac{1}{2 \cos(\phi)}$$

2. Se seleccionan 10 puntos de esta solución.
3. Se entrena ambos modelos:
 - Una **RNA** clásica solo con esos 10 puntos.
 - Una **PINN** que, además de esos puntos, incorpora la PDE como parte de la función de pérdida.

3.4.- Ventajas de las PINNs frente a una RNA tradicional

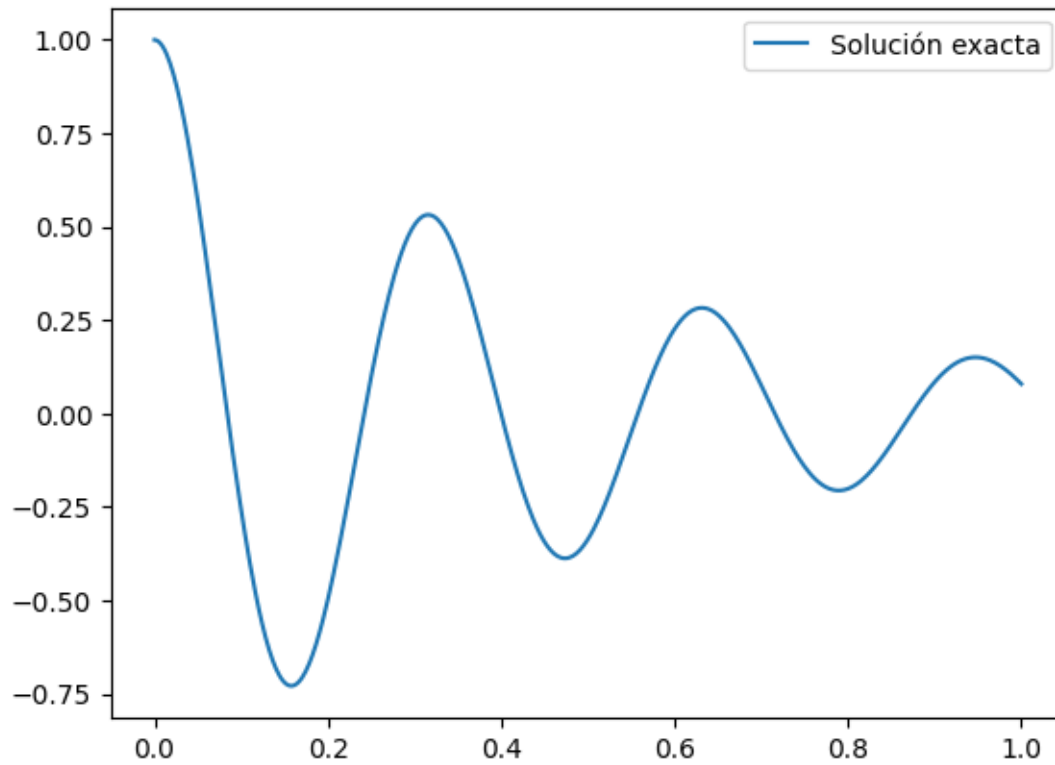
En la siguiente Tabla 1, se muestra un pequeño “trade-off” entre las redes neuronales artificiales tradicionales y las Physics-Informed Neural Networks.

Característica	RNA tradicional	PINN
Necesidad de datos	Alta	Baja (gracias a la física)
Conocimiento físico	No considerado	Incorporado en el entrenamiento
Generalización fuera del dominio	Limitada	Mejor generalización

Tabla 1. Trade-off

4.-Implementación de la Solución Analítica

Se implementa la función “oscillator()” en PyTorch para modelar esta solución. Esta función recibe como entrada los parámetros físicos δ y w_0 , junto con un vector de tiempos, y devuelve el desplazamiento $x(t)$ correspondiente. Se incluye una verificación con assert para asegurar que se cumple la condición de subamortiguamiento $\delta < w_0$.



Gráfica 1. Solución analítica

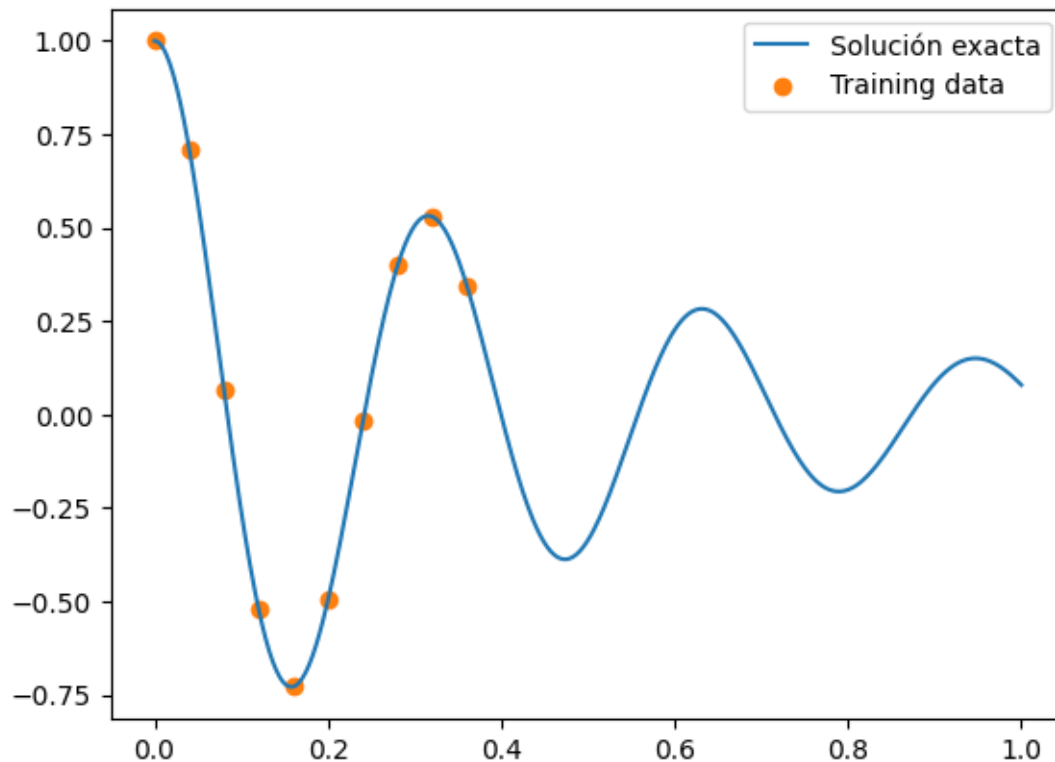
5.-Generación de los Datos de Entrenamiento

Para este estudio se trabaja con los siguientes parámetros:

- Coeficiente de amortiguamiento: $\delta = 2$
- Frecuencia natural: $w_0 = 20$

El dominio temporal para la resolución de la solución analítica se define como el intervalo $[0, 1]$ dividido en 500 puntos equiespaciados. Sobre la solución analítica generada, se van a extraer únicamente 10 puntos que serán utilizados para entrenar tanto la RNA como la PINN.

En la Gráfica 2, se puede observar tanto la solución exacta con los 10 puntos seleccionados para el entrenamiento. Esta visualización inicial es clave para observar la capacidad de extrapolación de los modelos.



Gráfica 2. Puntos utilizados durante el entrenamiento

6.-Entrenamiento de una Red Neuronal Artificial (RNA) clásica

Una vez seleccionados los 10 puntos de entrenamiento extraídos de la solución analítica, se entrena una red neuronal artificial (RNA) convencional para aproximar el comportamiento del sistema. Esta RNA no incorpora ninguna información física del problema: únicamente ajusta sus parámetros en base a los datos disponibles, siguiendo el enfoque tradicional de aprendizaje supervisado.

6.1.- Arquitectura de la RNA

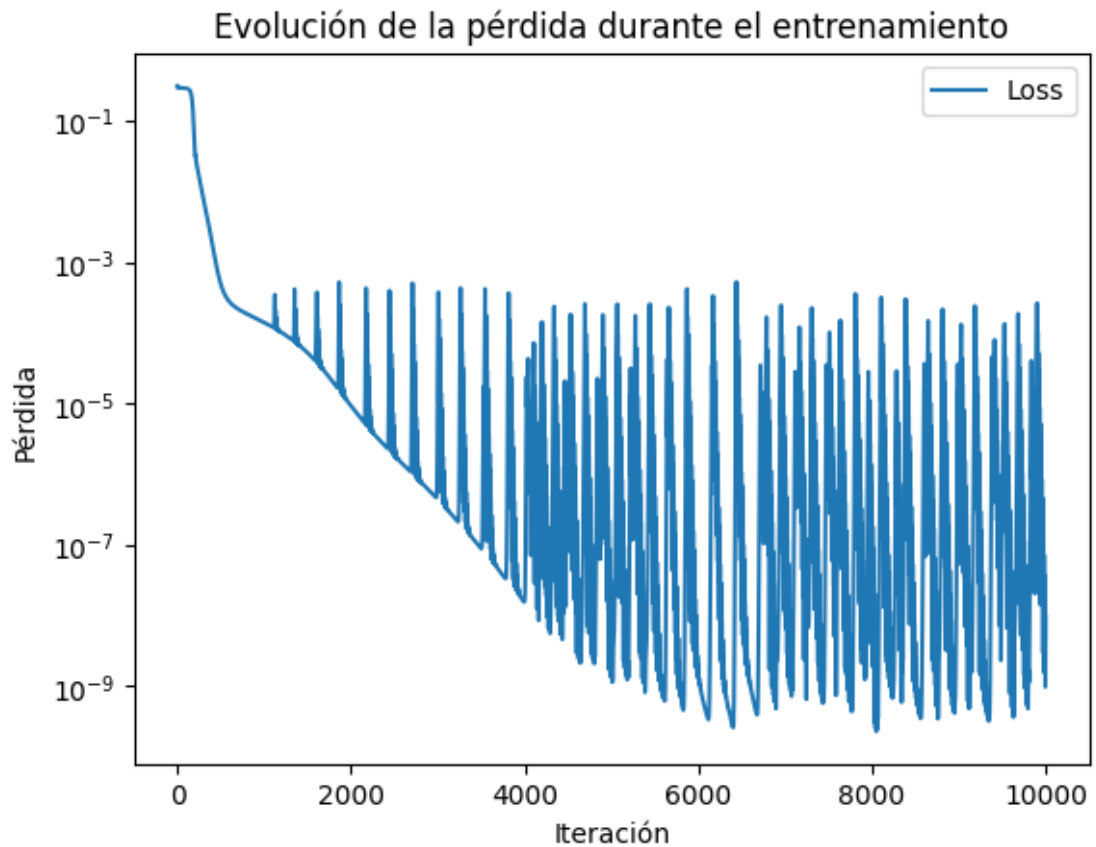
La red implementada cuenta con:

- 1 neurona de entrada (correspondiente al tiempo t)
- 3 capas ocultas con 32 neuronas cada una
- 1 neurona de salida (posición $x(t)$)
- Función de activación Tanh en cada capa intermedia.

El entrenamiento se lleva a cabo utilizando el optimizador Adam con una tasa de aprendizaje de 10^{-3} , y una función de pérdida MSE (Mean Squared Error), que mide la discrepancia entre la predicción de la red y los datos observados.

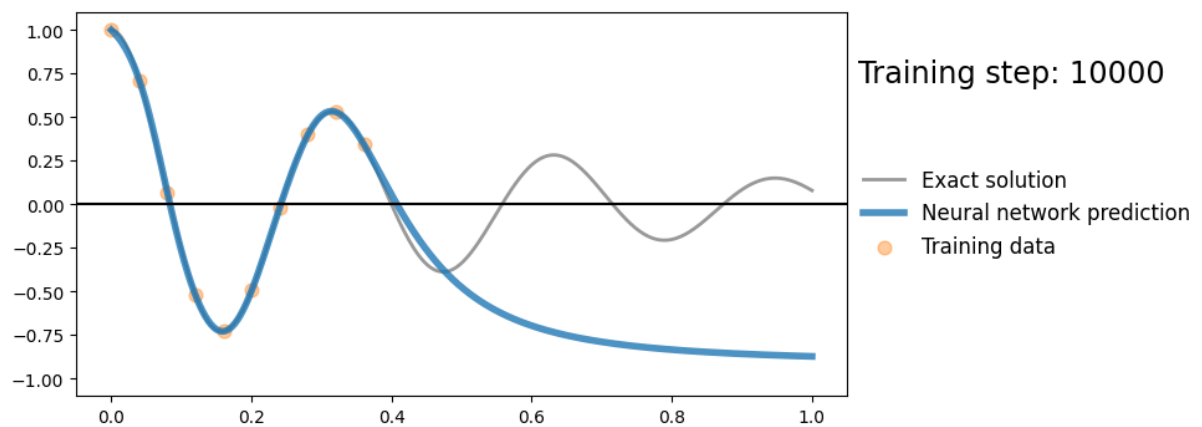
6.2.- Observaciones durante el entrenamiento

- A pesar de que la pérdida converge rápidamente a valores cercanos a cero, la RNA no logra aproximar la solución exacta del sistema.



Gráfica 3. Evolución de la pérdida de la RNA

- La red simplemente memoriza los 10 puntos proporcionados, sin capturar la dinámica oscilatoria que define el sistema.
- Este comportamiento se observa en la siguiente figura, donde la predicción de la RNA se aleja notablemente de la solución real fuera de los puntos de entrenamiento:



Gráfica 4. Resolución de la RNA

6.3.- Interpretación de resultados

Este resultado es coherente con la naturaleza de las RNAs tradicionales:

- Se ajustan eficazmente a los datos observados, pero carecen de capacidad para generalizar si la cantidad de datos es limitada o si estos no representan completamente la dinámica del sistema.
- En contextos industriales o científicos, donde obtener datos puede ser costoso, limitado o ruidoso, este tipo de enfoque puede ser inadecuado.

En la siguiente sección se introducirá el entrenamiento de una PINN, que permite superar estas limitaciones al incorporar la física del sistema como parte del proceso de aprendizaje.

7.-Entrenamiento de una PINN (Physics-Informed Neural Network)

Después de comprobar que una red neuronal artificial clásica no es capaz de generalizar correctamente a partir de pocos datos, se implementa un enfoque más robusto y físicamente consistente: una PINN.

7.1.- Principio de funcionamiento de una PINN

A diferencia de las RNA tradicionales, en las PINNs la función de pérdida no sólo contempla el error en los datos de entrenamiento, sino también el **residuo de la ecuación diferencial** y el residuo de las condiciones de contorno impuestas. De este modo, se penalizan aquellas predicciones que no respetan la física subyacente, incluso en regiones del dominio donde no hay datos disponibles.

7.2.- Condiciones y arquitectura del modelo

- Se entrena la misma arquitectura utilizada previamente para la RNA (entrada → 3 capas ocultas de 32 neuronas → salida),
- Se utiliza la activación Tanh y el optimizador Adam, pero con una tasa de aprendizaje reducida a 10^{-4} para mejorar la estabilidad al incorporar gradientes de derivadas,
- Se evalúan tres tipos de pérdida:

1. Pérdida en los datos de entrenamiento: \mathcal{L}_1

- Se calcula el error cuadrático medio entre la salida del modelo y los 10 puntos extraídos de la solución exacta:

$$\mathcal{L}_1 = MSE(x_{pred}, x_{real})$$

2. Pérdida por condición de contorno: \mathcal{L}_2

- Para el oscilador amortiguado, se impone que la velocidad inicial $\dot{x}(0)$ sea cero, es decir:

$$\left. \frac{dx}{dt} \right|_{t=0} = 0$$

3. Pérdida física (residuo de la ecuación diferencial): \mathcal{L}_3

- Se calcula el residuo de la EDO en 30 puntos uniformemente distribuidos en el dominio $[0,1]$
- Estas derivadas se obtienen automáticamente mediante autodiferenciación con PyTorch.

4. Función de pérdida total:

$$\mathcal{L}_{PINN} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3$$

7.3.- *Cálculo automático de derivadas: el motor de la física en las PINNs*

Una de las características más poderosas de PyTorch, y absolutamente esencial en el funcionamiento de las PINNs, es su capacidad de realizar autodiferenciación a través del módulo autograd.

Gracias a esta funcionalidad, es posible calcular de forma automática y precisa derivadas de cualquier orden con respecto a las entradas del modelo, incluso si esas derivadas están incrustadas en operaciones altamente no lineales y en estructuras de red complejas.

Esto es particularmente importante porque, en el contexto de las PINNs, las derivadas son necesarias para evaluar el residuo de la ecuación diferencial, es decir, la diferencia entre lo que predice la red y lo que dicta la ley física.

Este simple comando activa una poderosa cadena de cálculo simbólico automático que reemplaza, en esencia, al procedimiento tradicional de derivación simbólica o diferenciación numérica. A diferencia de los métodos clásicos, este enfoque:

- Es exacto, en el sentido de que no sufre errores por redondeo o por paso de malla (como en diferencias finitas).
- Es eficiente, pues solo se calculan las derivadas necesarias en el orden exacto requerido.
- Permite construir derivadas de derivadas (segundas, terceras, etc.) simplemente aplicando `autograd.grad()` varias veces, gracias al parámetro `create_graph=True`, que mantiene el grafo abierto para derivadas posteriores.

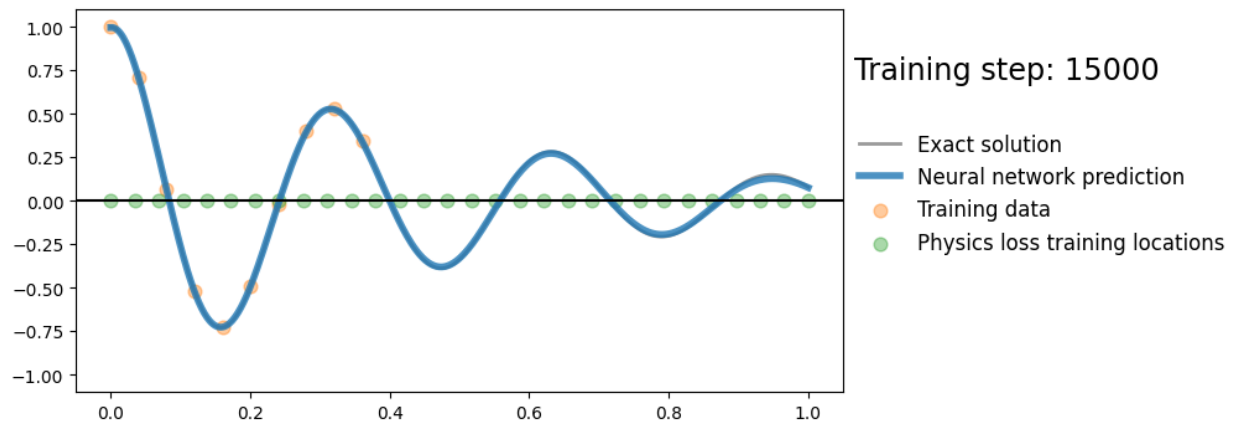
7.4.- *Visualización de los resultados*

Cada 500 iteraciones se genera una gráfica que incluye:

- La solución exacta del sistema (línea gris)
- La predicción de la PINN (línea azul)
- Los datos de entrenamiento (puntos naranjas)

- Los **30 puntos físicos** donde se evalúa el residuo (puntos verdes sobre el eje $y=0$)

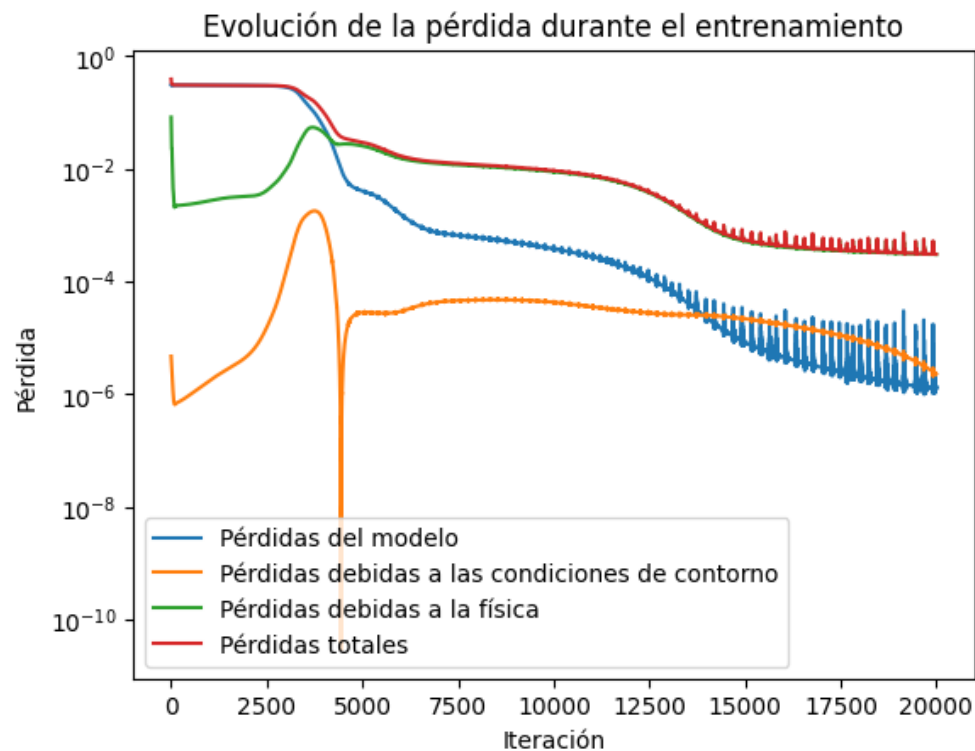
Esta última característica es fundamental: aunque estos puntos no contienen valores conocidos de $x(t)$, sirven para asegurar que el modelo predice soluciones que respetan la física.



Gráfica 5. Resolución de la PINN

7.5.- Evolución de las pérdidas

En la siguiente Gráfica 6, se muestra cómo evolucionan las distintas componentes de la pérdida a lo largo del entrenamiento:



Gráfica 6. Evolución de las pérdidas de la PINN

Se observa que:

- Todas las componentes $(\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3)$ convergen a valores cercanos a **cero** en escala logarítmica.
- Esto indica que el modelo respeta tanto los datos disponibles como las condiciones de contorno y, crucialmente, **la ecuación diferencial**.

7.6.- *Interpretación*

Este experimento demuestra la capacidad superior de las PINNs para generalizar incluso con pocos datos. La red no solo ajusta los puntos observados, sino que reconstruye la solución completa del sistema físico con alta fidelidad. Esto es especialmente importante en contextos donde obtener datos completos es costoso, peligroso o incluso imposible.

La inclusión de los puntos verdes (evaluación del residuo) evidencia cómo el modelo se ve guiado a través del dominio, obligándolo a mantener el comportamiento físico correcto en todas partes, no solo en los puntos de entrenamiento.

8.-Conclusiones

1. Limitaciones de las redes neuronales tradicionales (ANN)

- A pesar de que la red neuronal artificial logra minimizar la pérdida (loss) de entrenamiento hasta valores cercanos a cero, su capacidad de generalización es muy limitada.
- El modelo ajusta con rapidez los valores de los puntos de entrenamiento, pero no respeta la estructura subyacente del sistema físico, resultando en una predicción errónea o incoherente fuera de esos puntos.
- Esto demuestra un claro caso de sobreajuste (overfitting): el modelo memoriza los datos, pero no aprende el comportamiento general.
- En problemas reales, donde rara vez se dispone de datos completos o exhaustivos, este enfoque no es adecuado ni fiable.

2. Ventajas de las PINNs (Physics-Informed Neural Networks)

- La PINN incorpora el conocimiento de la física directamente en la función de pérdida, de modo que el aprendizaje no se basa solo en ajustar datos, sino en satisfacer la ecuación diferencial que gobierna el fenómeno.
- Gracias a la autodiferenciación de PyTorch, se pueden calcular automáticamente las derivadas necesarias de la salida de la red, sin recurrir a esquemas numéricos tradicionales como diferencias finitas.
- El resultado es un modelo que no solo se ajusta a los datos conocidos, sino que también predice de manera coherente en el resto del dominio.

3. Análisis de las pérdidas

- Se ha observado una convergencia clara y sostenida de las tres pérdidas que componen la PINN:
 - La pérdida de datos del modelo (loss_1) disminuye a medida que la red aprende los puntos conocidos.
 - La pérdida por condiciones de contorno (loss_2) se reduce progresivamente, asegurando que se cumplan las condiciones impuestas en los bordes del dominio.
 - La pérdida física (loss_3), asociada al residuo de la ecuación diferencial, converge hacia cero, garantizando que la solución respeta la dinámica física del sistema.
- La suma total de estas pérdidas (loss_PINN) disminuye a lo largo del entrenamiento, lo que indica un aprendizaje efectivo tanto desde el punto de vista de los datos como desde el punto de vista físico.

4. Importancia de los puntos de evaluación física

- Los 30 puntos verdes distribuidos uniformemente sobre el dominio representan las ubicaciones donde se evalúa el residuo físico de la ecuación diferencial.

- Aunque no contienen información de entrenamiento en el sentido clásico (no hay "valores objetivo"), estos puntos son fundamentales para guiar el aprendizaje físico de la red.
- Su efecto es equivalente a "inyectar conocimiento del sistema físico", reforzando la solución aprendida para que sea compatible con la dinámica real del problema.
- Este enfoque permite entrenar modelos robustos incluso cuando se dispone de pocos datos de observación.

5. Conclusión general

- Las PINNs representan un enfoque híbrido y poderoso que combina lo mejor de dos mundos: la capacidad de aproximación universal de las redes neuronales y la solidez de las leyes físicas.
- En entornos industriales o científicos donde los datos son escasos, costosos o ruidosos, pero la física es bien conocida, las PINNs se perfilan como una alternativa muy prometedora frente a los métodos clásicos puramente basados en datos.
- Este experimento demuestra empíricamente que una red informada físicamente generaliza mejor y predice con más coherencia que una red puramente basada en ajuste de datos.