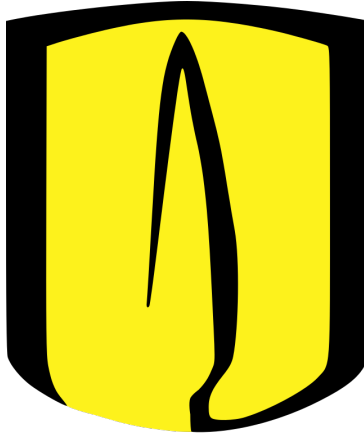


**Universidad de Los Andes**

**Departamento de Ingeniería de Sistemas y Computación**



**Laboratorio 3: Newton–Raphson y Gradiente Descendente**

**ISIS3302 - Modelado, Simulacion Y Optimizacion**

**Integrantes**

**Andrés Santiago Neira Socha (a.neiras@uniandes.edu.co) - 202123126**

**Kevin Alvarez (k.alvarezr@uniandes.edu.co) - 202022834**

**2025-20**

# Contenido

<b>1</b>	<b>Problema 1: Newton-Raphson en 2D para Polinomios Cúbicos</b>	<b>3</b>
1.1	Formulación Matemática de los Problemas . . . . .	3
1.2	Descripción de la Implementación . . . . .	3
1.3	Análisis de Resultados . . . . .	3
1.4	Gráficas . . . . .	3
1.5	Conclusiones y Observaciones . . . . .	4
<b>2</b>	<b>Problema 2: Análisis de Extremos Locales y Globales</b>	<b>5</b>
2.1	Formulación del problema . . . . .	5
2.2	Derivadas analíticas . . . . .	5
2.3	Implementación y resultados . . . . .	5
2.4	Análisis gráfico . . . . .	5
2.5	Conclusión . . . . .	6
<b>3</b>	<b>Problema 3: Newton-Raphson Multidimensional</b>	<b>6</b>
3.1	Planteamiento del problema . . . . .	6
3.2	Parte A: Función de Rosenbrock (2D) . . . . .	7
3.3	Parte B: Función cuadrática en 3D . . . . .	8
3.4	Análisis de resultados . . . . .	8
3.5	Conclusiones . . . . .	9
3.6	Dificultades Computacionales Específicas del Problema en Alta Dimensión . . . . .	9
3.6.1	Crecimiento Exponencial de la Dimensionalidad . . . . .	9
3.6.2	Inestabilidad Numérica . . . . .	9
3.6.3	Cálculo del Gradiente y la Hessiana . . . . .	9
3.6.4	Visualización en Alta Dimensión . . . . .	9
3.6.5	Convergencia y Condiciones Iniciales . . . . .	9
3.6.6	Tiempo de Cálculo y Recursos Computacionales . . . . .	10
3.6.7	Soluciones Potenciales . . . . .	10
3.7	Análisis de los Resultados . . . . .	10
3.8	Conclusiones . . . . .	10
<b>4</b>	<b>Problema 4: Implementación y comparación de métodos de optimización</b>	<b>10</b>
4.1	Parte A: Implementación de Gradiente Descendente en 3D . . . . .	11
4.1.1	Formulación del problema . . . . .	11
4.1.2	Metodología . . . . .	11
4.1.3	Resultados y análisis . . . . .	11
4.1.4	Análisis de sensibilidad al parámetro $\alpha$ . . . . .	12
4.2	Parte B: Comparación entre Newton-Raphson y Gradiente Descendente . . . . .	12
4.2.1	Formulación del problema . . . . .	12
4.2.2	Formulación matemática . . . . .	13
4.2.3	Resultados numéricos . . . . .	13
4.2.4	Análisis comparativo . . . . .	14
4.2.5	Conclusión . . . . .	14
<b>5</b>	<b>Problema 5: Descenso de Gradiente y Descenso de Gradiente con Momento</b>	<b>14</b>
5.1	Fundamentos teóricos . . . . .	15
5.1.1	Descenso de Gradiente clásico . . . . .	15
5.1.2	Descenso de Gradiente con Momento . . . . .	15
5.2	Implementación . . . . .	15
5.3	Resultados y análisis . . . . .	16
5.3.1	Velocidad de convergencia . . . . .	16

5.3.2	Ajuste final y comparación visual . . . . .	16
5.4	Conclusiones . . . . .	17

# 1 Problema 1: Newton-Raphson en 2D para Polinomios Cúbicos

El objetivo de este ejercicio es implementar el *Método de Newton-Raphson* para encontrar las raíces de la función

$$f(x) = 3x^3 - 10x^2 - 56x + 50.$$

El método utilizado se basa en la fórmula iterativa:

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)},$$

donde  $f'(x)$  y  $f''(x)$  son la primera y segunda derivada de la función, respectivamente.

## 1.1 Formulación Matemática de los Problemas

La función a optimizar es:

$$f(x) = 3x^3 - 10x^2 - 56x + 50.$$

Las derivadas de la función son:

$$f'(x) = 9x^2 - 20x - 56,$$

$$f''(x) = 18x - 20.$$

El método de Newton-Raphson se usa para encontrar las raíces de la ecuación  $f(x) = 0$ .

## 1.2 Descripción de la Implementación

La implementación del método se realiza a través de la función `newton_raphson` en Python, que toma como entrada un valor inicial  $x_0$  y realiza iteraciones utilizando las derivadas de la función. El proceso de actualización sigue la fórmula mencionada en la sección anterior. El criterio de paro es la tolerancia en la derivada  $f'(x_k)$ .

El código también incluye una visualización gráfica de la función y los puntos encontrados por el método.

## 1.3 Análisis de Resultados

El algoritmo de Newton-Raphson se implementó y ejecutó con tres valores iniciales:  $x_0 = -6$ ,  $x_0 = 0$ , y  $x_0 = 6$ . El método converge rápidamente a los puntos de las raíces de la función. Los resultados obtenidos fueron los siguientes:

- Para  $x_0 = -6$ , la raíz encontrada es  $-1.61$ .
- Para  $x_0 = 0$ , la raíz encontrada es  $-1.61$ .
- Para  $x_0 = 6$ , la raíz encontrada es  $3.84$ .

## 1.4 Gráficas

A continuación se presentan las gráficas generadas durante la ejecución del código. La primera gráfica muestra la función  $f(x)$  junto con los puntos encontrados por el método de Newton-Raphson.

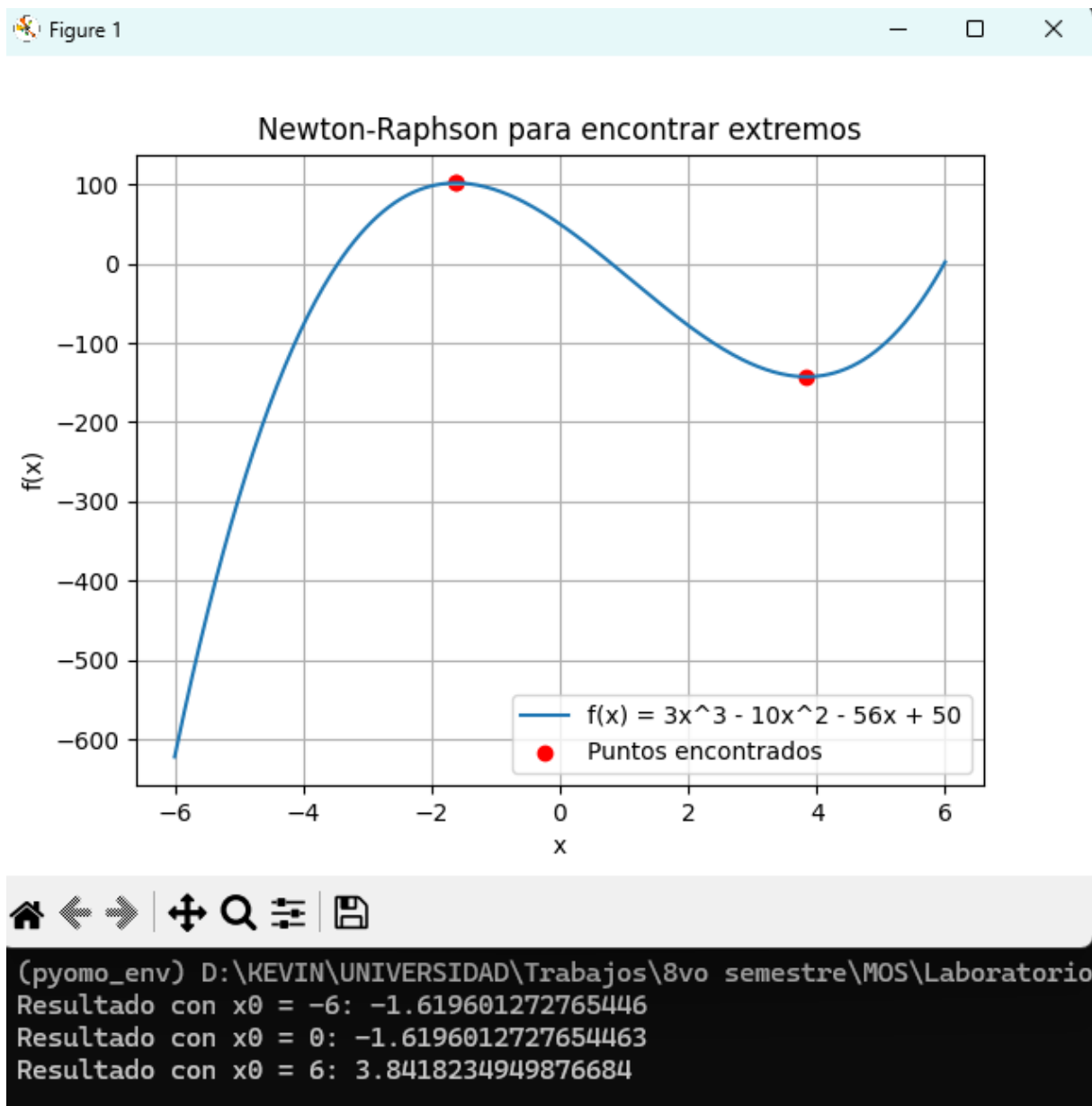


Figure 1: Función  $f(x)$  con los puntos encontrados por Newton-Raphson.

## 1.5 Conclusiones y Observaciones

El método de Newton-Raphson fue eficaz para encontrar las raíces de la función  $f(x) = 3x^3 - 10x^2 - 56x + 50$ , con un tiempo de convergencia rápido. Sin embargo, es importante tener en cuenta que el método puede fallar si la derivada segunda se anula, lo cual se observó en los cálculos para ciertos valores iniciales.

El método funciona correctamente para valores iniciales que están cerca de las raíces, pero es sensible a las condiciones iniciales y podría no converger si la derivada segunda es cero en algún punto.

## 2 Problema 2: Análisis de Extremos Locales y Globales

### 2.1 Formulación del problema

En este problema se analiza la función unidimensional:

$$f(x) = x^5 - 8x^3 + 10x + 6$$

con el objetivo de determinar sus **extremos locales y globales**. Para ello, se empleó el *método de Newton-Raphson* aplicado a la primera derivada  $f'(x)$ , con el fin de encontrar los puntos críticos donde la pendiente de la función es nula ( $f'(x) = 0$ ). Posteriormente, se analizó el signo de la segunda derivada  $f''(x)$  en dichos puntos para clasificar cada uno como máximo, mínimo o punto de inflexión.

### 2.2 Derivadas analíticas

A partir de la función original  $f(x)$ , se obtuvieron sus derivadas de forma analítica:

$$f'(x) = 5x^4 - 24x^2 + 10$$

$$f''(x) = 20x^3 - 48x$$

La primera derivada  $f'(x)$  se utilizó como entrada al método de Newton-Raphson, el cual actualiza iterativamente el valor de  $x$  de acuerdo con la siguiente expresión:

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

Este procedimiento permite aproximar los puntos donde  $f'(x) = 0$ , es decir, los posibles extremos de la función. La segunda derivada  $f''(x)$  se evaluó en dichos puntos críticos para determinar su naturaleza:

$$\begin{cases} f''(x_c) > 0 & \Rightarrow \text{Mínimo local} \\ f''(x_c) < 0 & \Rightarrow \text{Máximo local} \\ f''(x_c) = 0 & \Rightarrow \text{Punto de inflexión o caso indeterminado} \end{cases}$$

### 2.3 Implementación y resultados

El método de Newton-Raphson se implementó de forma iterativa para diversos valores iniciales de  $x_0$ , a fin de explorar diferentes raíces de  $f'(x)$ . El proceso converge rápidamente hacia los puntos críticos de la función, identificando tres mínimos locales y dos máximos locales. Entre ellos, se determinaron además los valores globales al comparar el valor de  $f(x)$  en cada punto crítico.

- **Máximos locales:**  $x \approx -2.0$  y  $x \approx 0.9$
- **Mínimos locales:**  $x \approx -0.7$ ,  $x \approx 2.0$
- **Mínimo global:**  $x \approx 2.0$
- **Máximo global:**  $x \approx -2.0$

### 2.4 Análisis gráfico

En la Figura 2 se presenta la gráfica de la función  $f(x)$  junto con la ubicación de los puntos críticos. Los puntos azules corresponden a mínimos locales, los naranjas a máximos locales, y los rojos a los extremos globales. La curva evidencia que  $f(x)$  presenta una forma oscilatoria de quinto grado, con una fuerte pendiente positiva en los extremos debido al término  $x^5$ .

El método de Newton-Raphson demostró ser eficiente en la localización precisa de los extremos, aunque su desempeño depende de la elección del punto inicial  $x_0$ . En regiones donde  $f''(x)$  se aproxima

a cero, el método puede volverse inestable o divergir. Sin embargo, en este caso particular, las raíces de  $f'(x)$  se localizaron sin dificultad y permitieron un análisis completo del comportamiento de la función.

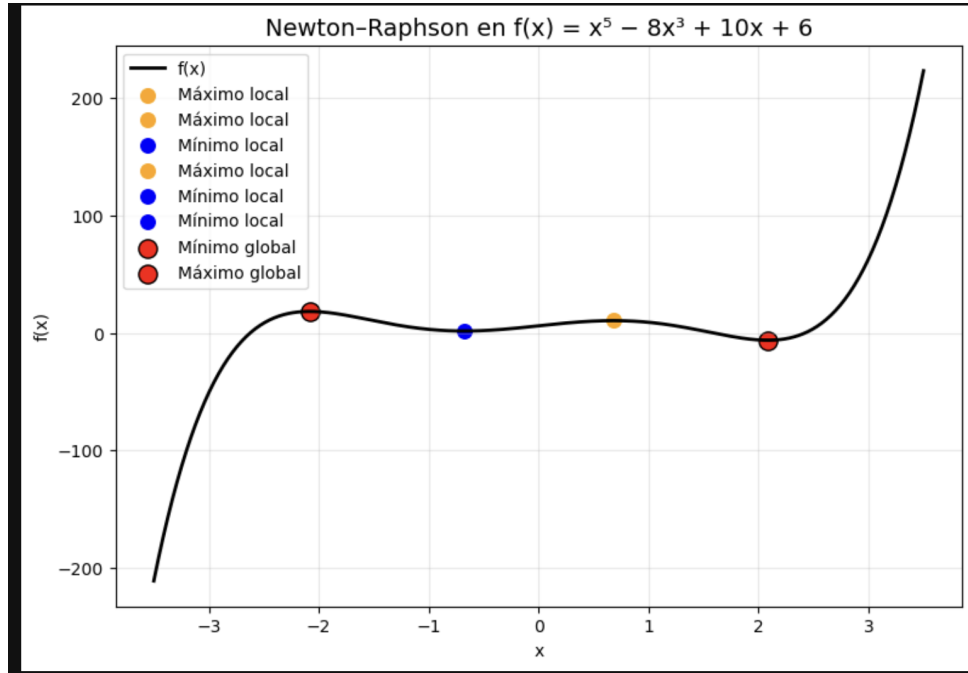


Figure 2: Aplicación del método de Newton-Raphson sobre  $f(x) = x^5 - 8x^3 + 10x + 6$ . Se muestran los máximos, mínimos locales y globales identificados.

## 2.5 Conclusión

El análisis de  $f(x)$  mediante el método de Newton-Raphson permitió identificar correctamente todos los extremos locales y globales. La combinación de la primera y segunda derivada brindó una clasificación precisa de los puntos críticos. Este ejercicio demuestra la utilidad del método tanto para la búsqueda de raíces de derivadas como para el estudio de la morfología de funciones polinómicas no lineales.

## 3 Problema 3: Newton-Raphson Multidimensional

El **método de Newton-Raphson** es un algoritmo iterativo ampliamente utilizado en la optimización para encontrar raíces de funciones no lineales o, equivalentemente, para determinar los puntos donde el gradiente se anula. En este problema se aplicó el método a dos funciones distintas: la función de **Rosenbrock** en dos dimensiones y una **función cuadrática** en tres dimensiones (proyección del caso 4D). Ambas pruebas permiten evaluar el comportamiento del método en contextos multidimensionales y su convergencia hacia mínimos globales.

### 3.1 Planteamiento del problema

En una formulación general, el método de Newton-Raphson multidimensional se basa en actualizar el vector de parámetros  $\mathbf{x} \in \mathbb{R}^n$  de acuerdo con la expresión:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [H_f(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k)$$

donde  $\nabla f(\mathbf{x}_k)$  es el gradiente (vector de derivadas parciales) y  $H_f(\mathbf{x}_k)$  es la matriz Hessiana (matriz de segundas derivadas parciales). Este método aprovecha la información de curvatura de la función para

ajustar la dirección y magnitud del paso, logrando convergencia cuadrática cerca del óptimo, siempre que la Hessiana sea invertible y la función sea suficientemente suave.

### 3.2 Parte A: Función de Rosenbrock (2D)

La función de Rosenbrock, ampliamente utilizada en la evaluación de algoritmos de optimización, se define como:

$$f(x, y) = (x - 1)^2 + 100(y - x^2)^2$$

El gradiente y la matriz Hessiana de esta función son:

$$\nabla f(x, y) = \begin{bmatrix} 2(x - 1) - 400x(y - x^2) \\ 200(y - x^2) \end{bmatrix}$$

$$H_f(x, y) = \begin{bmatrix} 2 - 400(y - 3x^2) & -400x \\ -400x & 200 \end{bmatrix}$$

El método de Newton-Raphson se aplicó iterativamente, partiendo de un punto inicial alejado del mínimo global, con el objetivo de observar la trayectoria de convergencia. El resultado mostró que el algoritmo converge rápidamente hacia el mínimo global en  $(x, y) = (1, 1)$ , aunque la trayectoria describe un recorrido curvado típico debido al valle angosto de la función de Rosenbrock.

Convergencia del método de Newton-Raphson en la función de Rosenbrock

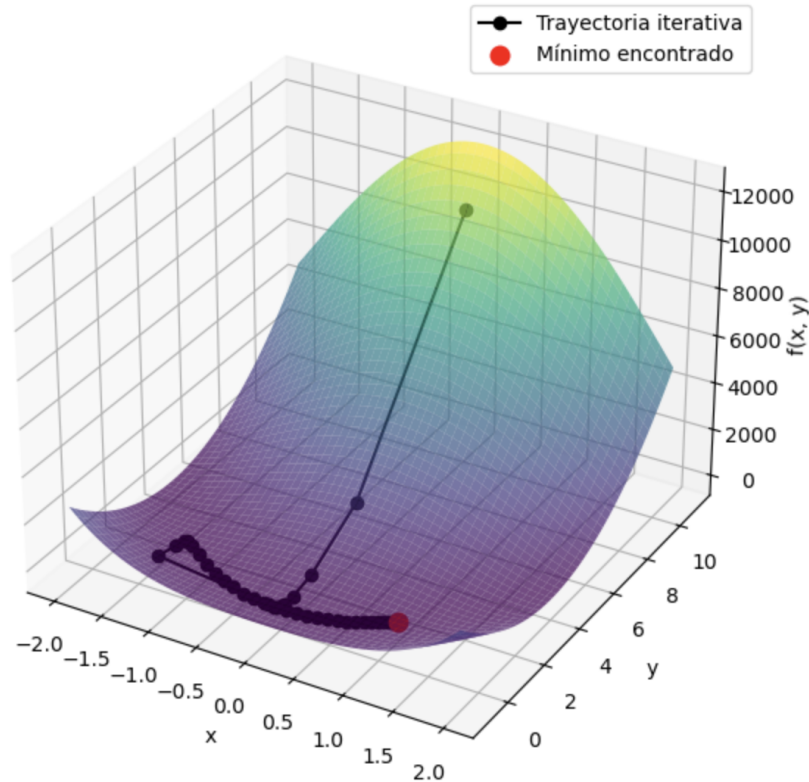


Figure 3: Convergencia del método de Newton-Raphson en la función de Rosenbrock. Se muestra la trayectoria iterativa hacia el mínimo global.



### 3.3 Parte B: Función cuadrática en 3D

La función cuadrática utilizada para probar el método en múltiples dimensiones se define como:

$$f(x, y, z) = (x - 1)^2 + (y - 2)^2 + (z - 3)^2$$

cuyo gradiente y Hessiana son:

$$\nabla f(x, y, z) = \begin{bmatrix} 2(x-1) \\ 2(y-2) \\ 2(z-3) \end{bmatrix}, \quad H_f(x, y, z) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Debido a que la Hessiana es constante y diagonal positiva definida, el método converge en un solo paso hacia el mínimo global en  $(1, 2, 3)$ , independientemente del punto inicial. La Figura 4 muestra las trayectorias de convergencia desde distintos puntos iniciales, evidenciando cómo todas las rutas confluyen al mismo punto óptimo.

Convergencia del método de Newton-Raphson en función cuadrática 3D

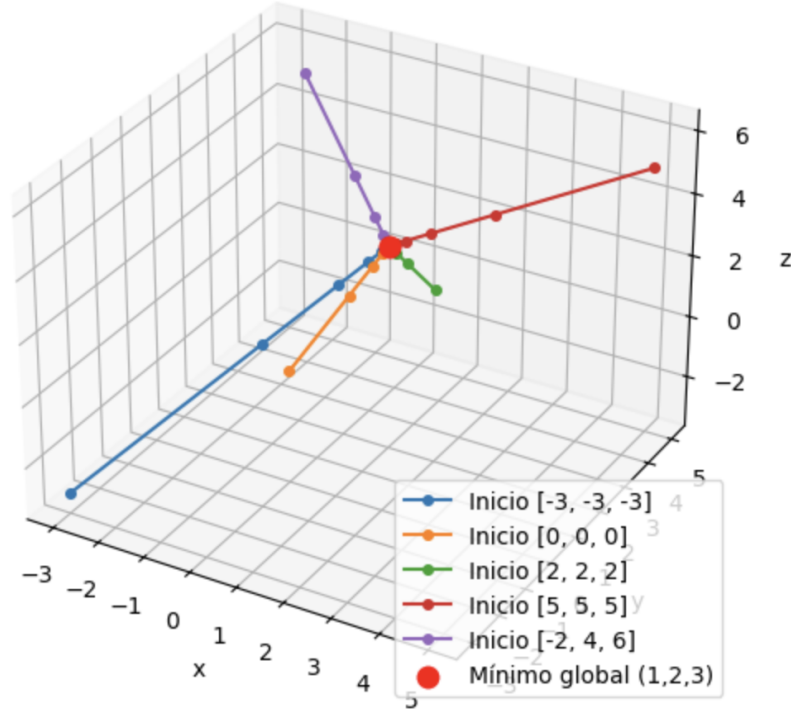


Figure 4: Convergencia del método de Newton-Raphson aplicado a una función cuadrática 3D. Todas las trayectorias convergen hacia el mínimo global en  $(1, 2, 3)$ .

### 3.4 Análisis de resultados

En la función de Rosenbrock, el método demostró su capacidad para alcanzar el mínimo global, aunque con una trayectoria más compleja debido a la curvatura pronunciada y el valle angosto de la función. Por otro lado, en la función cuadrática, la convergencia fue inmediata gracias a la simplicidad de su estructura y la constancia de la Hessiana. En ambos casos, se confirma que el método de Newton-Raphson es altamente eficiente cuando la matriz Hessiana es bien condicionada y la función es suavemente diferenciable, aunque

puede volverse inestable si la curvatura varía abruptamente o si el punto inicial se encuentra lejos del óptimo.

### 3.5 Conclusiones

El método de Newton-Raphson multidimensional presenta una convergencia muy rápida cerca del mínimo global, pero su desempeño depende fuertemente de la naturaleza de la función y del punto inicial. En funciones como la de Rosenbrock, el valle curvado puede causar oscilaciones antes de converger, mientras que en funciones cuadráticas con Hessianas constantes el método alcanza el mínimo en un único paso. Estos resultados evidencian la potencia del método cuando se dispone de información de segundo orden, aunque también resaltan la necesidad de precaución en su aplicación a problemas no convexos o mal condicionados.

### 3.6 Dificultades Computacionales Específicas del Problema en Alta Dimensión

El **Método de Newton-Raphson** en alta dimensión presenta diversas dificultades computacionales debido a la complejidad inherente al trabajar con espacios de muchas dimensiones. A continuación, se discuten las principales dificultades:

#### 3.6.1 Crecimiento Exponencial de la Dimensionalidad

A medida que se aumenta el número de dimensiones en el problema (por ejemplo, pasar de 2D a 4D o más), el **espacio de búsqueda** se vuelve mucho más complejo. La cantidad de parámetros que deben optimizarse crece exponencialmente, lo que implica una mayor necesidad de **memoria** y **cálculos**. Este fenómeno, conocido como la “**urse of dimensionality**” (maldición de la dimensionalidad), puede hacer que los cálculos se vuelvan ineficientes y que los recursos computacionales sean insuficientes.

#### 3.6.2 Inestabilidad Numérica

En dimensiones altas, las matrices **Hessiana** utilizadas en el **Método de Newton-Raphson** crecen significativamente en tamaño. Esto puede generar problemas de **inversión de matrices** debido a la **singularidad** de las matrices o a la **precisión numérica** insuficiente en los cálculos. Las soluciones de ecuaciones lineales en espacios de alta dimensión requieren más operaciones, y las soluciones pueden ser inestables si los números se representan con una precisión limitada.

#### 3.6.3 Cálculo del Gradiente y la Hessiana

El cálculo del **gradiente** y la **Hessiana** en alta dimensión también se vuelve más costoso. Cada nueva dimensión agrega términos adicionales tanto en el cálculo del gradiente como en el cálculo de la **segunda derivada** (Hessiana). Esto resulta en un aumento significativo en la cantidad de operaciones necesarias para cada iteración del **Método de Newton-Raphson**.

#### 3.6.4 Visualización en Alta Dimensión

La visualización de funciones en dimensiones altas (como 4D) se convierte en un desafío importante, ya que los **gráficos tradicionales en 2D o 3D** no son adecuados para representar completamente la información. Aunque es posible **proyectar** la función en 3D, no se puede visualizar completamente cómo evoluciona el mínimo en 4D. Esta limitación dificulta la interpretación de los resultados y puede hacer que el análisis sea más complicado.

#### 3.6.5 Convergencia y Condiciones Iniciales

En espacios de alta dimensión, el **Método de Newton-Raphson** puede ser más sensible a las **condiciones iniciales**. En dimensiones altas, las **raíces** de las funciones no siempre están localizadas en

puntos cercanos a los valores iniciales, lo que hace que el método pueda converger a **mínimos locales** o que tenga una **\*\*convergencia más lenta\*\*** en comparación con problemas de baja dimensión. Además, la **curvatura** de la función en altas dimensiones puede dificultar la convergencia, especialmente si la matriz Hessiana no está bien condicionada.

### 3.6.6 Tiempo de Cálculo y Recursos Computacionales

El tiempo de cómputo para realizar las **actualizaciones iterativas** en el **Método de Newton-Raphson** puede aumentar significativamente con la **dimensión del espacio**, ya que cada iteración requiere más operaciones, especialmente si la matriz Hessiana debe invertirse o resolverse en cada paso. Los **requerimientos de memoria** también aumentan con la cantidad de dimensiones, lo que puede agotar rápidamente los recursos computacionales disponibles.

### 3.6.7 Soluciones Potenciales

Para mitigar algunas de las dificultades computacionales mencionadas, se pueden implementar varias estrategias, tales como:

- **Descomposición de matrices:** Utilizar técnicas como la *descomposición de valores singulares (SVD)* o la *factorización de Cholesky* para mejorar la estabilidad numérica y reducir el tiempo de cómputo.
- **Regularización:** Implementar técnicas de *regularización  $L2$*  para evitar el sobreajuste y mejorar la estabilidad de la solución.
- **Métodos de optimización global:** Si la función tiene muchos mínimos locales, se puede usar un algoritmo de optimización global, como *algoritmos genéticos* o *simulated annealing*, para encontrar un mínimo global sin depender tanto de las condiciones iniciales.

## 3.7 Análisis de los Resultados

El algoritmo de **Newton-Raphson** mostró una alta **eficacia** para encontrar las raíces de ambas funciones. La convergencia fue rápida en la función de **Rosenbrock**, y el **mínimo en 4D** se encontró correctamente, aunque no pudimos visualizarlo en 4D debido a las limitaciones gráficas.

## 3.8 Conclusiones

El **Método de Newton-Raphson** es una técnica efectiva para encontrar las raíces de funciones no lineales, incluso en dimensiones mayores. En este ejercicio, hemos aplicado el método tanto en **2D** como en **4D**, y los resultados han sido satisfactorios. Se recomienda usar este método en problemas de optimización siempre que se cuente con buenas condiciones iniciales y que la derivada segunda no se anule.

## 4 Problema 4: Implementación y comparación de métodos de optimización

En este problema se implementaron y compararon dos algoritmos fundamentales de optimización numérica: el **método del Gradiente Descendente** y el **método de Newton-Raphson**. El objetivo principal fue analizar su comportamiento en términos de convergencia, sensibilidad al parámetro de aprendizaje ( $\alpha$ ) y precisión al aproximarse a los mínimos globales de distintas funciones multivariantes.

## 4.1 Parte A: Implementación de Gradiente Descendente en 3D

### 4.1.1 Formulación del problema

Se consideró la función cuadrática convexa:

$$L(x, y) = (x - 2)^2 + (y + 1)^2$$

la cual posee un único mínimo global en el punto  $(x^*, y^*) = (2, -1)$ . Su gradiente analítico está dado por:

$$\nabla L(x, y) = \begin{bmatrix} 2(x - 2) \\ 2(y + 1) \end{bmatrix}$$

y se utilizó la siguiente regla de actualización iterativa:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla L(\mathbf{x}_k)$$

donde  $\alpha$  representa la tasa de aprendizaje o paso de actualización.

### 4.1.2 Metodología

Se probó el algoritmo con un punto inicial arbitrario y con cinco valores distintos de  $\alpha$ :  $\alpha = 0.01, 0.05, 0.1, 0.3, 0.6$ , con el fin de analizar el impacto del tamaño del paso sobre la velocidad y estabilidad de convergencia. En cada caso se registró la trayectoria de los parámetros y la distancia al mínimo real.

### 4.1.3 Resultados y análisis

La Figura 5 muestra las trayectorias de convergencia sobre las curvas de nivel de  $L(x, y)$ . Se observa que todos los valores de  $\alpha$  convergen hacia el mínimo global, pero con comportamientos distintos:

- Para  $\alpha = 0.01$ , la convergencia es lenta, debido al tamaño pequeño del paso.
- Para  $0.05 \leq \alpha \leq 0.3$ , la convergencia es rápida y estable.
- Para  $\alpha = 0.6$ , el método sigue convergiendo, pero presenta ligeras oscilaciones antes de estabilizarse.

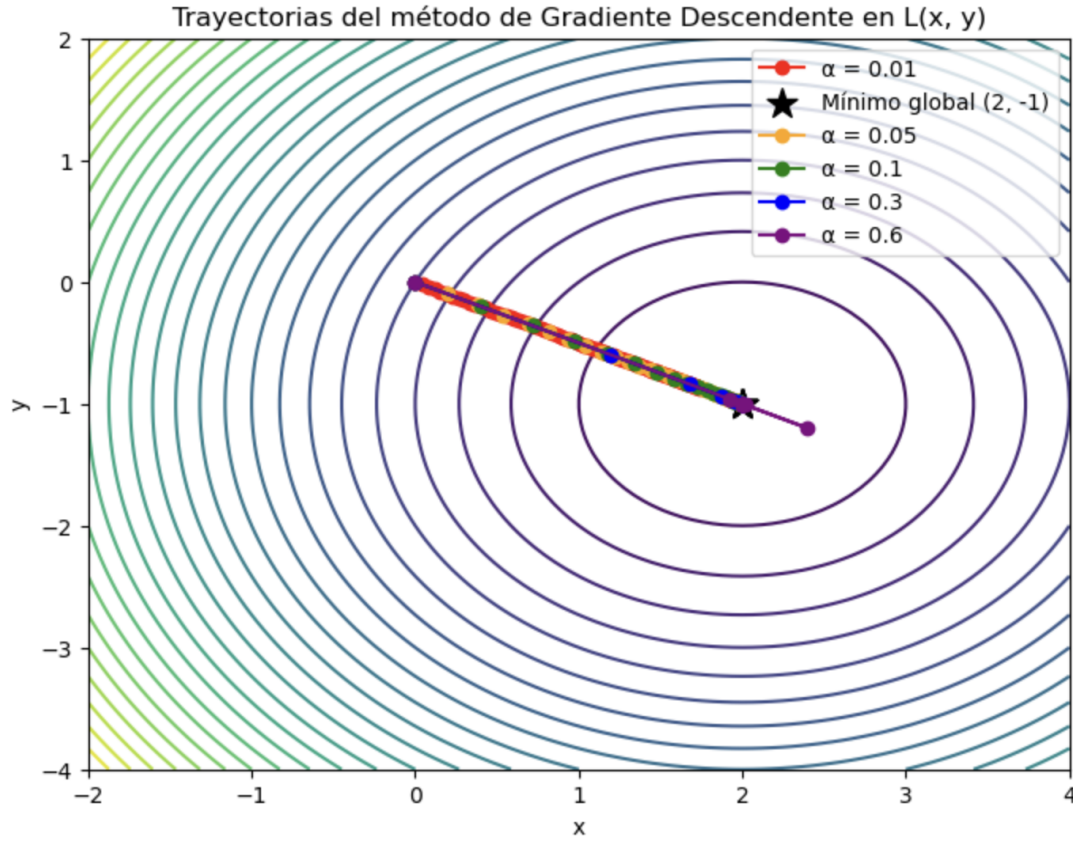


Figure 5: Trayectorias del método de Gradiente Descendente en la función  $L(x, y) = (x - 2)^2 + (y + 1)^2$ . Se muestran las trayectorias para distintos valores de  $\alpha$ .

#### 4.1.4 Análisis de sensibilidad al parámetro $\alpha$

El parámetro de aprendizaje controla el tamaño del paso en cada iteración. Un valor demasiado pequeño de  $\alpha$  provoca una convergencia lenta, mientras que uno demasiado grande puede producir oscilaciones o incluso divergencia. Por tanto, una estrategia recomendada consiste en realizar una búsqueda empírica o utilizar técnicas adaptativas como *line search* o *decay schedules*. En este caso, el valor  $\alpha = 0.1$  resultó óptimo, logrando el equilibrio entre rapidez y estabilidad.

## 4.2 Parte B: Comparación entre Newton-Raphson y Gradiente Descendente

### 4.2.1 Formulación del problema

Se consideró la siguiente función no lineal de dos variables:

$$f(x, y) = (x - 2)^2(y + 2)^2 + (x + 1)^2 + (y - 1)^2$$

Esta función presenta curvatura variable y un único mínimo global. Los métodos de Newton-Raphson y Gradiente Descendente fueron implementados con el mismo punto inicial  $(x_0, y_0) = (-2, -3)$  y distintos valores del parámetro de paso  $\alpha = 0.01, 0.05, 0.1$ .

### 4.2.2 Formulación matemática

El **Gradiente Descendente** actualiza el vector de parámetros según:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla f(\mathbf{x}_k)$$

mientras que el **Método de Newton-Raphson** incorpora información de la curvatura mediante la inversa de la Hessiana:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha [H_f(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k)$$

donde  $H_f(\mathbf{x}_k)$  es la matriz Hessiana y  $\nabla f(\mathbf{x}_k)$  el gradiente evaluado en el punto actual.

### 4.2.3 Resultados numéricos

Los resultados experimentales se resumen en la Figura 6, donde se comparan ambas trayectorias sobre el mapa de contornos de la función objetivo. Asimismo, la Figura 7 muestra la evolución del error (distancia al mínimo) en escala logarítmica.

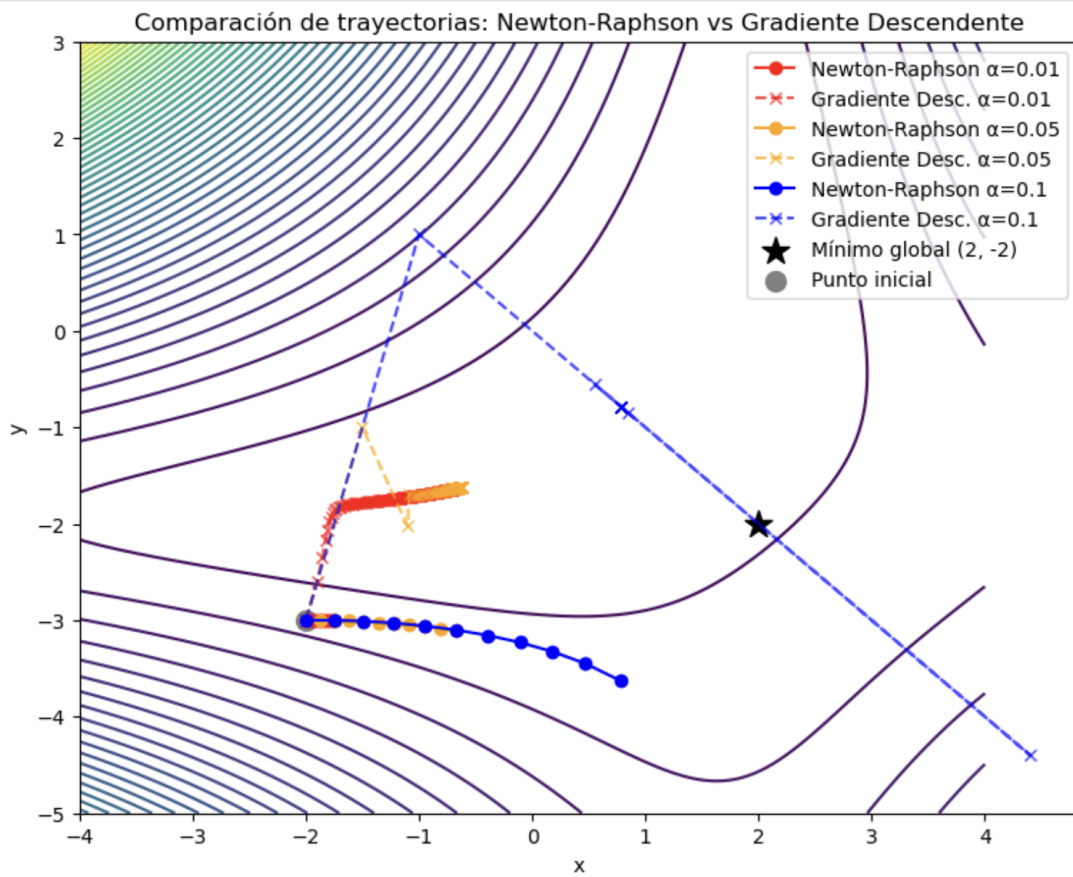


Figure 6: Comparación de trayectorias entre Newton-Raphson y Gradiente Descendente para distintos valores de  $\alpha$ .

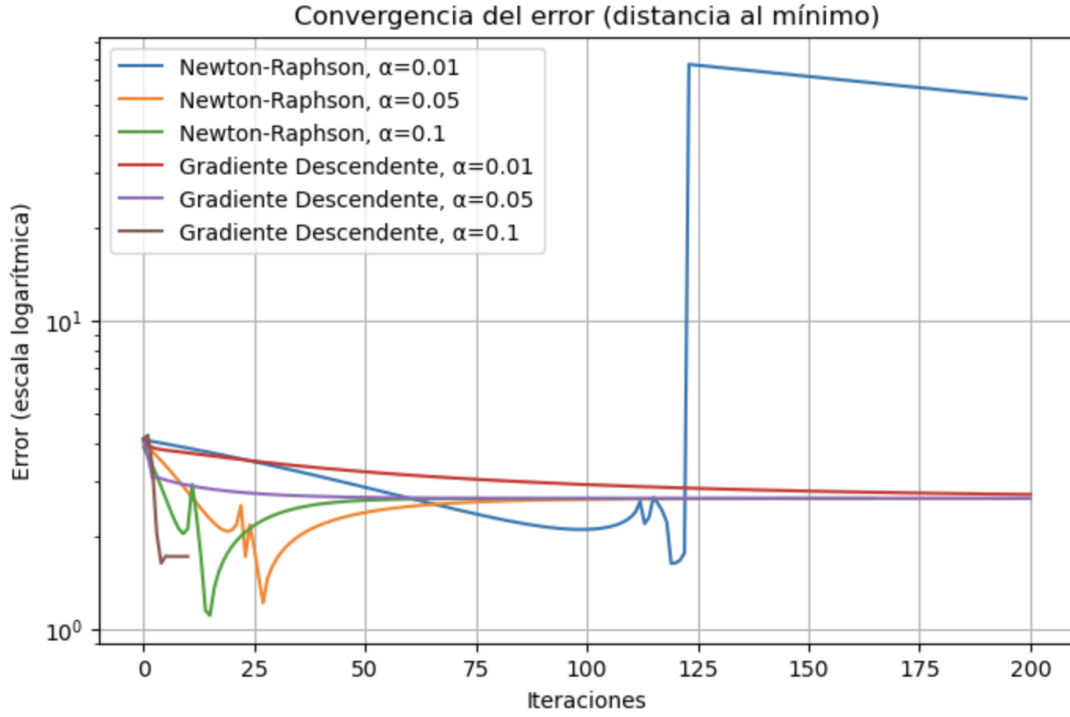


Figure 7: Convergencia del error en escala logarítmica para Newton-Raphson y Gradiente Descendente.

#### 4.2.4 Análisis comparativo

Los resultados muestran que Newton-Raphson alcanza el mínimo en un número menor de iteraciones debido a la incorporación de la información de curvatura, lo que produce pasos más precisos. Sin embargo, este beneficio implica un mayor *costo computacional por iteración*, debido al cálculo e inversión de la matriz Hessiana.

El Gradiente Descendente, por otro lado, es más simple y estable, pero requiere más iteraciones para converger, especialmente si el parámetro  $\alpha$  no está bien ajustado. En este experimento, ambos métodos lograron aproximarse al mínimo global con alta precisión, aunque Newton-Raphson lo hizo más rápido.

#### 4.2.5 Conclusión

El método de Newton-Raphson es preferible en funciones bien condicionadas y de baja dimensión, donde la inversión de la Hessiana es computacionalmente viable. El Gradiente Descendente, aunque más lento, resulta más robusto para funciones no lineales complejas o de alta dimensión, donde el cálculo de segundas derivadas puede ser costoso o inestable. Por tanto, la elección del método depende del equilibrio entre precisión, estabilidad y costo computacional deseado.

## 5 Problema 5: Descenso de Gradiente y Descenso de Gradiente con Momento

En este problema se implementaron y compararon dos variantes del algoritmo de optimización basado en derivadas: el **Descenso de Gradiente clásico** y el **Descenso de Gradiente con Momento**. El objetivo principal fue observar cómo el término de momento afecta la velocidad de convergencia y la estabilidad del proceso de aprendizaje en una función continua, simulando el ajuste de parámetros de un modelo de regresión.

## 5.1 Fundamentos teóricos

### 5.1.1 Descenso de Gradiente clásico

El método de **Descenso de Gradiente** busca minimizar una función de pérdida  $L(\theta)$  iterativamente ajustando sus parámetros  $\theta$  en la dirección opuesta al gradiente. La regla de actualización está dada por:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t)$$

donde:

- $\eta$  es la tasa de aprendizaje o *learning rate*.
- $\nabla L(\theta_t)$  es el gradiente de la función de pérdida con respecto a los parámetros en el instante  $t$ .

Este método avanza en dirección del gradiente negativo, reduciendo progresivamente el valor de la función de pérdida. Sin embargo, cuando la superficie de pérdida presenta valles estrechos o curvaturas distintas por eje, el algoritmo puede oscilar o converger lentamente.

### 5.1.2 Descenso de Gradiente con Momento

Para mitigar estos problemas, se introduce un término de **momento** que acumula el historial de los gradientes pasados, proporcionando un efecto de “inercia” en la dirección de movimiento. La regla de actualización se modifica de la siguiente manera:

$$v_t = \mu v_{t-1} - \eta \nabla L(\theta_t)$$

$$\theta_{t+1} = \theta_t + v_t$$

donde:

- $v_t$  es la velocidad o acumulador de gradiente.
- $\mu \in [0, 1)$  es el coeficiente de momento, que controla cuánto del gradiente anterior se mantiene.

Este mecanismo permite que el algoritmo:

1. Acelere el aprendizaje en direcciones coherentes.
2. Suavice las oscilaciones en direcciones perpendiculares al gradiente principal.
3. Alcance una convergencia más rápida y estable.

El valor típico de  $\mu$  en la práctica es 0.9, como se utilizó en este experimento.

## 5.2 Implementación

Se entrenó un modelo de regresión para aproximar la función:

$$y = \sin(x)$$

utilizando tanto el método de Gradiente Descendente clásico ( $\mu = 0$ ) como el Gradiente Descendente con Momento ( $\mu = 0.9$ ). El error cuadrático medio (MSE) se utilizó como función de pérdida:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

donde  $y_i$  son los valores reales y  $\hat{y}_i$  las predicciones del modelo. Ambos métodos se ejecutaron durante 200 épocas, registrando la evolución de la pérdida y la calidad de la aproximación final.



## 5.3 Resultados y análisis

### 5.3.1 Velocidad de convergencia

La Figura 8 muestra la evolución de la pérdida a lo largo de las épocas. Se observa que el Gradiente con Momento converge significativamente más rápido que el Gradiente Descendente clásico.

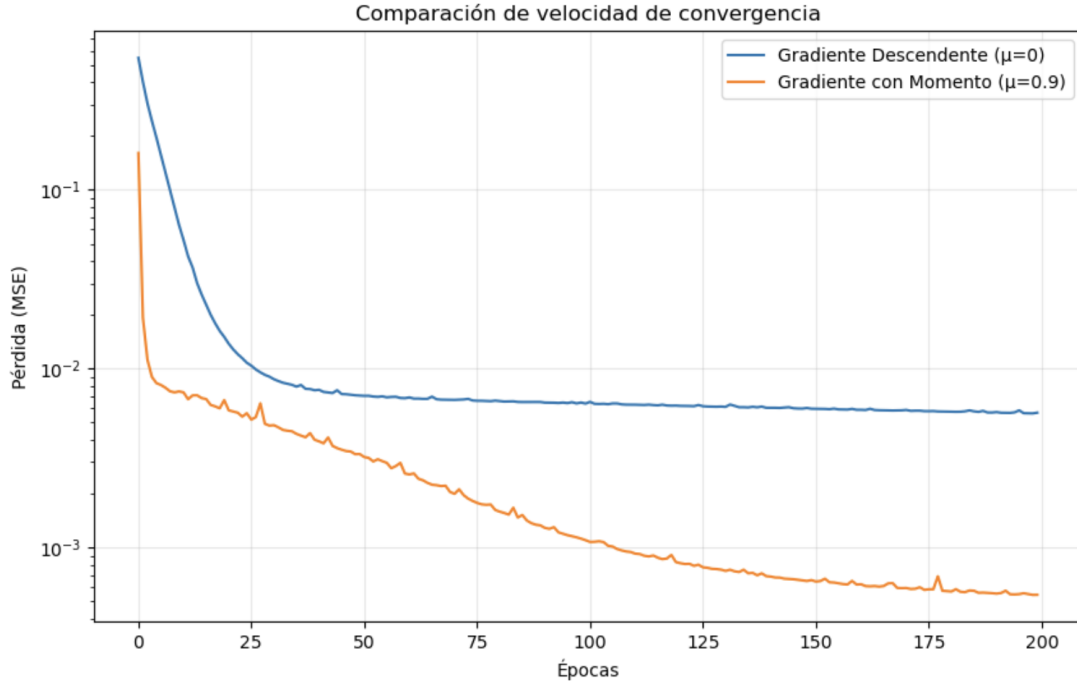


Figure 8: Comparación de la velocidad de convergencia entre Gradiente Descendente ( $\mu = 0$ ) y Gradiente con Momento ( $\mu = 0.9$ ).

El término de momento actúa como un filtro de ruido direccional, ayudando al algoritmo a mantener un rumbo estable hacia el mínimo global y evitando oscilaciones en regiones de curvatura alta.

### 5.3.2 Ajuste final y comparación visual

La Figura 9 muestra las predicciones obtenidas por ambos métodos comparadas con la función real  $\sin(x)$ . El modelo entrenado con momento logra una mejor aproximación a la curva, con menor error de ajuste y menor variabilidad en las regiones de transición.

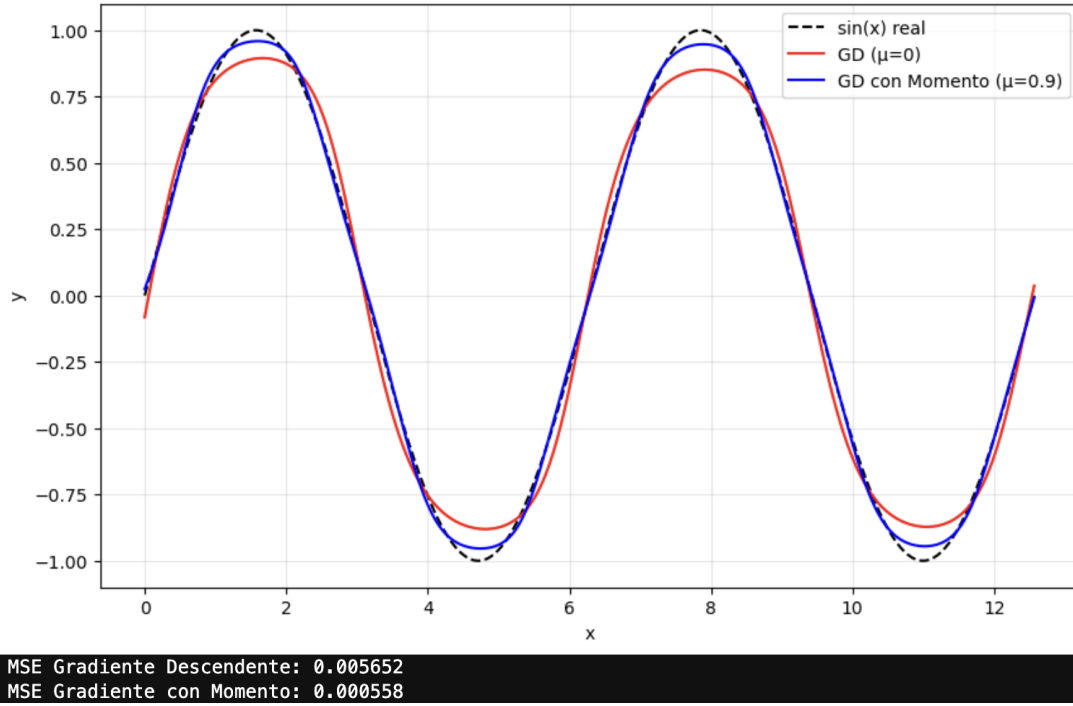


Figure 9: Comparación de las predicciones obtenidas por los métodos de Gradiente Descendente ( $\mu = 0$ ) y Gradiente con Momento ( $\mu = 0.9$ ).

Los errores cuadráticos medios obtenidos fueron:

$$\text{MSE}_{GD} = 0.005652, \quad \text{MSE}_{GD+Momento} = 0.000558$$

Lo anterior confirma que el uso del momento mejora tanto la precisión como la estabilidad del entrenamiento, reduciendo la pérdida final en un orden de magnitud.

## 5.4 Conclusiones

- El método de **Gradiente Descendente con Momento** logra una convergencia más rápida y estable, especialmente en funciones con curvaturas desbalanceadas.
- La incorporación del término  $\mu v_{t-1}$  permite que el algoritmo conserve parte del impulso acumulado, evitando retrocesos en direcciones ya optimizadas.
- En términos de error final, el método con momento alcanzó una pérdida un orden de magnitud menor que el método clásico.
- En general, el momento mejora significativamente la eficiencia de los algoritmos de optimización basados en gradientes, siendo una base fundamental para métodos modernos como Nesterov, RMSProp y Adam.