Tarea Evaluativa de Optimización

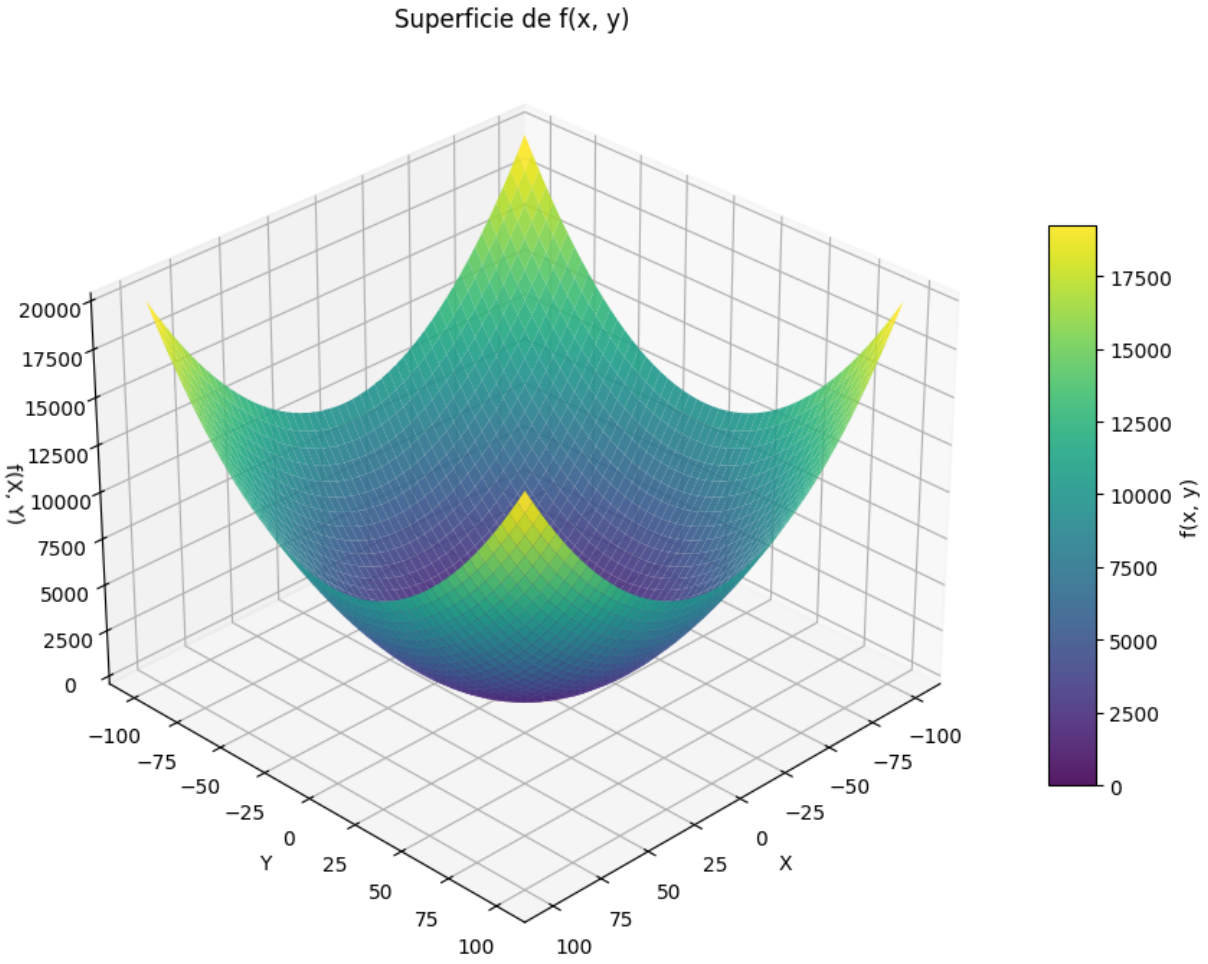
Carlos Mazorra Matos c-311

12 de noviembre de 2025

Función, dominio y primeras observaciones

- **Función:**

$$f : \mathbb{R}^2 \to \mathbb{R}, \qquad f(x, y) = x^2 + y^2 - 10\,\cos^2(x) - 10\,\cos\left(\frac{x^2}{30}\right).$$



- **Dominio:** $\operatorname{dom} f = \mathbb{R}^2$.

- **Continuidad:** La función $f(x, y)$ es una suma algebraica de funciones polinómicas (continuas en todo el dominio $\mathbb{R}^2$) y cosenos (continuos en $\mathbb{R}$) cuyos argumentos son también funciones polinómicas. Por tanto la función es continua en todo el dominio.

- **Cota inferior:** usando $\cos^2(x) \leq 1$ y $\cos(t) \leq 1$ para todo t, se tiene

$$f(x,y) \geq x^2 + y^2 - 10 - 10 = x^2 + y^2 - 20 \geq -20.$$

En particular, f es *coerciva* (tiende a $+\infty$ cuando $\|(x,y)\| \to \infty$), por lo que admite al menos un mínimo global.

Hallando el gradiente:

$$\begin{aligned}\frac{\partial f}{\partial x}(x,y) &= 2x - 10\,\frac{\mathrm{d}}{\mathrm{d}x}(\cos^2 x) - 10\,\frac{\mathrm{d}}{\mathrm{d}x}\Big(\cos\big(\tfrac{x^2}{30}\big)\Big) \\ &= 2x + 10\,\sin(2x) + \frac{2x}{3}\,\sin\left(\frac{x^2}{30}\right), \\ \frac{\partial f}{\partial y}(x,y) &= 2y.\end{aligned}$$

$$\nabla f(x,y) = \begin{bmatrix} 2x + 10\,\sin(2x) + \frac{2x}{3}\,\sin\big(\frac{x^2}{30}\big) \\ 2y \end{bmatrix}.$$

y la Hessiana:

$$\frac{\partial^2 f}{\partial x^2}(x,y) = 2 + 20\,\cos(2x) + \frac{2}{3}\,\sin\left(\frac{x^2}{30}\right) + \frac{2x^2}{45}\,\cos\left(\frac{x^2}{30}\right),$$

$$\frac{\partial^2 f}{\partial x\,\partial y}(x,y) = 0, \qquad \frac{\partial^2 f}{\partial y^2}(x,y) = 2.$$

$$\nabla^2 f(x,y) = \begin{bmatrix} 2 + 20\,\cos(2x) + \frac{2}{3}\,\sin\big(\frac{x^2}{30}\big) + \frac{2x^2}{45}\,\cos\big(\frac{x^2}{30}\big) & 0 \\ 0 & 2 \end{bmatrix}.$$

llegamos a que el término $20\cos(2x)$ oscila en $[-20, 20]$; existen valores de x (p.ej. $x \approx \frac{\pi}{2}$) donde la entrada $(1,1)$ del Hessiano se hace claramente menor que $2 - 20 < 0$ (los términos restantes pueden ser pequeños), por lo que $\nabla^2 f$ no es semidefinida positiva en todo $\mathbb{R}^2$. Luego, f *no* es convexa globalmente.

Puntos estacionarios

Los puntos estacionarios satisfacen $\nabla f(x,y) = 0$.

- De $\frac{\partial f}{\partial y} = 2y = 0$ se obtiene necesariamente $y^* = 0$.
- Para x, debemos resolver la ecuación

$$g(x) = 2x + 10\,\sin(2x) + \frac{2x}{3}\,\sin\left(\frac{x^2}{30}\right) = 0.$$

Es inmediato que $x = 0$ es solución (pues $\sin(0) = 0$).

Por lo tanto, $(0,0)$ es un punto estacionario.

Evaluando la Hessiana en el punto $(0,0)$ obtenemos:

$$\nabla^2 f(0,0) = \begin{bmatrix} 2 + 20\,\cos(0) + \frac{2}{3}\,\sin(0) + \frac{2(0)^2}{45}\,\cos(0) & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 2 + 20(1) + 0 + 0 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 22 & 0 \\ 0 & 2 \end{bmatrix}.$$

Los valores propios de la matriz Hessiana $\nabla^2 f(0,0)$ son los elementos de la diagonal, $\lambda_1 = 22$ y $\lambda_2 = 2$. Como ambos valores propios son **positivos** ($\lambda_1 > 0$ y $\lambda_2 > 0$), el Hessiano es **definido positivo** en $(0,0)$, y por tanto, el punto $(0,0)$ es un **mínimo local estricto**.

Y si usamos las cotas del inicio,

$$f(x,y) \geq x^2 + y^2 - 20 \geq -20.$$

En $(0,0)$ se obtiene

$$f(0,0) = 0 + 0 - 10\cos^2(0) - 10\cos(0) = -10 - 10 = -20,$$

con lo cual $(0,0)$ *alcanza* la cota inferior. Como los máximos valores que pueden tomar $\cos^2(x)$ y $\cos(\frac{x^2}{30})$ es 1 y $x^2 + y^2 \geq 0$ entonces $(0,0)$ es el **único mínimo global** y

$$\min_{(x,y)\in\mathbb{R}^2} f(x,y) = f(0,0) = -20.$$

Descripción General del método de Máximo descenso

El **Método de Máximo Descenso**, también conocido como **Gradiente Descendente** (*Gradient Descent*), es un algoritmo iterativo de optimización utilizado para encontrar mínimos locales de funciones diferenciables de varias variables $f : \mathbb{R}^n \to \mathbb{R}$.

La premisa fundamental del método es que el **gradiente** de una función en un punto dado ($\nabla f(\mathbf{x}_k)$) apunta en la dirección de *máximo ascenso*. Por lo tanto, moverse en la dirección del **gradiente negativo** ($-\nabla f(\mathbf{x}_k)$) garantiza el *descenso* más rápido posible en el valor de la función, al menos localmente.

Partiendo de un punto inicial $\mathbf{x}_0$, se genera una secuencia de puntos $\mathbf{x}_1, \mathbf{x}_2, \ldots$ que se espera converja a un mínimo local de f.

En cada iteración k, el siguiente punto $\mathbf{x}_{k+1}$ se calcula mediante la siguiente fórmula de actualización:

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

Donde:

- $\mathbf{x}_k \in \mathbb{R}^n$: Es el punto actual en la iteración k.
- $\nabla f(\mathbf{x}_k)$: Es el **gradiente** de f evaluado en $\mathbf{x}_k$. Indica la dirección del máximo ascenso.
- $-\nabla f(\mathbf{x}_k)$: Es la **dirección de descenso**.
- $\alpha_k > 0$: Es el **tamaño del paso**..

El algoritmo se desarrolla en los siguientes pasos iterativos:

1. **Inicialización:** Escoger un punto inicial $\mathbf{x}_0$ y establecer un criterio de parada (tolerancia ϵ).
2. **Cálculo del Gradiente:** En la iteración k, calcular el gradiente $\nabla f(\mathbf{x}_k)$.
3. **Criterio de Parada:** Si $\nabla f(\mathbf{x}_k)$ es menor que ϵ, terminar el proceso.
4. **Determinación del Paso:** Elegir el tamaño del paso $\alpha_k > 0$ (fijo, o mediante un método como la búsqueda de paso óptimo).
5. **Actualización:** Calcular el nuevo punto $\mathbf{x}_{k+1}$ usando la fórmula de actualización.
6. **Repetir:** Incrementar $k \leftarrow k + 1$ y volver al paso 2.

Este algoritmo tiene la ventaja de ser conceptualmente sencillo y fácil de implementar; solo requiere el cálculo de las derivadas parciales (el gradiente), no necesitando la costosa matriz Hessiana ($\nabla^2 f$) y funciona eficientemente en problemas con un número muy grande de variables (alta dimensionalidad). No obstante, cuenta con ciertas desventajas a tener en cuenta. La

convergencia puede ser muy lenta cuando la función tiene contornos alargados o presenta un condicionamiento pobre (es decir, cuando los valores propios de la Hessiana son muy diferentes). Además, es altamente sensible a la elección del tamaño del paso (α) donde un α muy grande puede causar divergencia (sobrepasar el mínimo) y un α muy pequeño conduce a una convergencia excesivamente lenta. Por último, el método está garantizado para converger solo a un mínimo local y no necesariamente al mínimo global.

Descripción General del método de Newton

El **Método de Newton** es un potente algoritmo iterativo de optimización utilizado para encontrar puntos estacionarios (mínimos, máximos o puntos de silla) de una función dos veces diferenciable $f : \mathbb{R}^n \to \mathbb{R}$. A diferencia del Gradiente Descendente, el método utiliza información de la **curvatura** de la función a través de la matriz Hessiana, lo que le confiere una velocidad de convergencia superior cerca del óptimo.

La función $f(\mathbf{x})$ se aproxima localmente en la iteración k mediante una expansión de Taylor de segundo orden alrededor de $\mathbf{x}_k$:

$$f(\mathbf{x}) \approx f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T(\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^T \mathbf{H}_f(\mathbf{x}_k)(\mathbf{x} - \mathbf{x}_k)$$

El siguiente punto $\mathbf{x}_{k+1}$ se determina encontrando el punto crítico (gradiente cero) de esta aproximación cuadrática.

En cada iteración k, el siguiente punto $\mathbf{x}_{k+1}$ se calcula como:

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

Donde:

- $\nabla f(\mathbf{x}_k)$: Es el **vector gradiente** de f en $\mathbf{x}_k$.
- $\mathbf{H}_f(\mathbf{x}_k)$: Es la **matriz Hessiana** de f en $\mathbf{x}_k$, definida por la matriz de segundas derivadas parciales:

$$\mathbf{H}_f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

El término $\mathbf{p}_k = \mathbf{H}_f(\mathbf{x}_k)^{-1} \nabla f(\mathbf{x}_k)$ es la **dirección de Newton**, que ajusta la dirección y el tamaño del paso basándose en la curvatura local.

La forma estándar de implementar la iteración es resolviendo un sistema de ecuaciones lineales en lugar de calcular explícitamente la inversa de la Hessiana.

1. **Inicialización:** Escoger un punto inicial $\mathbf{x}_0$ y un criterio de parada ϵ.
2. **Cálculo:** En la iteración k, calcular el gradiente $\nabla f(\mathbf{x}_k)$ y la Hessiana $\mathbf{H}_f(\mathbf{x}_k)$.
3. **Criterio de Parada:** Si $\nabla f(\mathbf{x}_k) < \epsilon$, detener el algoritmo.
4. **Resolución del Sistema:** Resolver el sistema de ecuaciones lineales para obtener la dirección $\mathbf{p}_k$:
$$\mathbf{H}_f(\mathbf{x}_k)\mathbf{p}_k = \nabla f(\mathbf{x}_k)$$
5. **Actualización:** Calcular el nuevo punto:
$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{p}_k$$
6. **Repetir:** Incrementar $k \leftarrow k + 1$ y volver al paso 2.

El método de Newton cuenta con las ventajas de que posee una convergencia cuadrática cerca de un mínimo local estricto, lo que significa que el número de cifras significativas correctas en el resultado se duplica en cada paso llegando a converger en muy pocas iteraciones y la dirección de Newton incluye automáticamente el tamaño del paso que se requiere para alcanzar el mínimo de la aproximación cuadrática. Sin embargo, este requiere calcular, almacenar y, a menudo, invertir o resolver un sistema lineal con la matriz Hessiana. Para funciones con n variables, esto implica n^2 cálculos de segundas derivadas por iteración. También, si la matriz Hessiana $\mathbf{H}_f(\mathbf{x}_k)$ no es definida positiva (es singular o indefinida), el método puede fallar, divergir, o conducir a un máximo o punto de silla y es sensible a la elección del punto inicial, especialmente en regiones no convexas de la función, donde la convergencia puede ser errática.

Razones para usar los métodos propuestos para la optimización de $f(x, y)$

El método de Máximo Descenso es conceptualmente sencillo y requiere cálculos relativamente básicos, solo requiere el cálculo del gradiente $\nabla f(x, y)$, funciona bien para explorar regiones amplias del dominio y es robusto para encontrar mínimos locales. Sin embargo, la convergencia puede ser muy lenta si los términos oscilatorios del gradiente hacen que la superficie cambie bruscamente y necesita un ajuste cuidadoso del tamaño de paso.

Mientras tanto, el método de Newton utiliza la matriz Hessiana $\mathbf{H}_f(x, y)$, que incorpora la curvatura de la función, puede converger muy rápido cerca de un mínimo local y la dirección y magnitud del paso se ajustan automáticamente, aprovechando la información de la curvatura. Pero también hay que tener en cuenta que debido a los términos oscilatorios $\cos(2x)$, la entrada $(1, 1)$ de la Hessiana puede tomar valores negativos. Esto puede hacer que $\mathbf{H}_f$ sea indefinida, lo que podría desviar el método o hacerlo converger a un máximo o un punto silla.

Análisis de Resultados de Algoritmos de Optimización

La implementación del método de Newton demostró una convergencia cuadrática extremadamente rápida, alcanzando el mínimo global en una sola iteración ($\mathbf{x}_0 \to \mathbf{x}_1$) en todos los casos donde el punto inicial satisfacía $x_0 = 0$. Este comportamiento se ilustra claramente en el experimento con $\mathbf{x}_0 = [0, 90]$ (Figura 2).

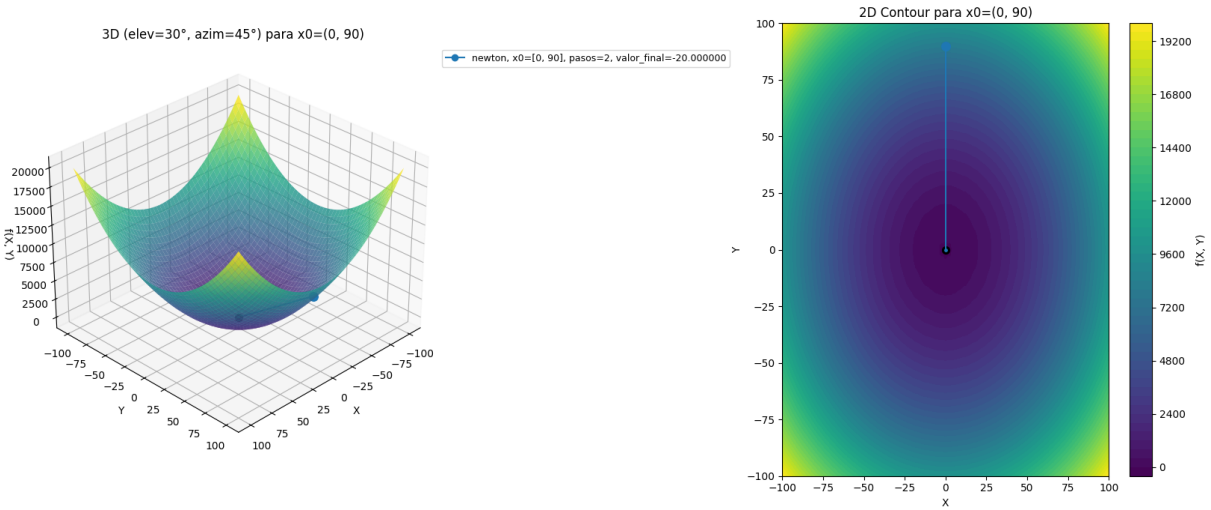


Figura 1: Gráfico de resultados del algoritmo de Newton en $\mathbf{x_0} = [\mathbf{0}, \mathbf{90}]$

La sensibilidad del método se hizo evidente cuando se seleccionó un punto inicial con $x_0 \neq 0$.

En estos casos, el algoritmo falló consistentemente en converger al mínimo global $f = -20$, quedando atrapado en mínimos locales o puntos estacionarios con valores significativamente mayores. El experimento con $\mathbf{x}_0 = [50, 50]$ (Figura 3) subraya esta inestabilidad en regiones no convexas de la función, destacando la vulnerabilidad de Newton a una selección deficiente del punto de inicio.

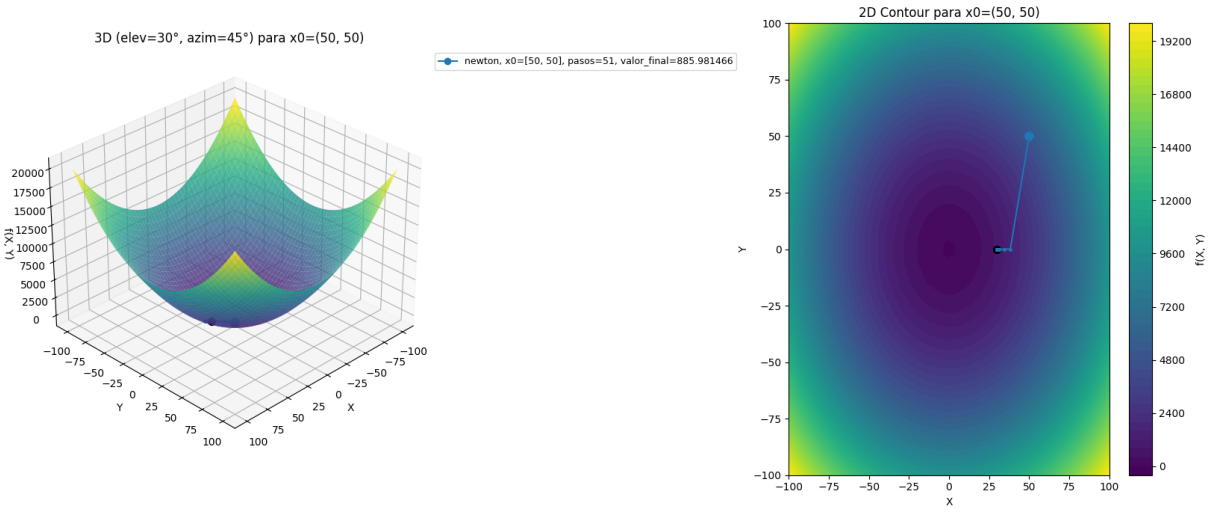


Figura 2: Gráfico de resultados del algoritmo de Newton en $\mathbf{x_0} = [\mathbf{50}, \mathbf{50}]$

Si, en vez de Newton, decidimos usar el método de Máximo Descenso podemos notar que el exito de este depende mucho de el tamaño del paso que tomemos y la cantidad de iteraciones que le dejemos hacer. Si tomamos lr = 0.1 logra converger a -20 (o un valor muy cercano como -19.999...) desde todos los puntos de inicio probados siendo el más rápido para este algoritmo y encontrando el mínimo en menos de 100 pasos (p.ej.Figura 4).

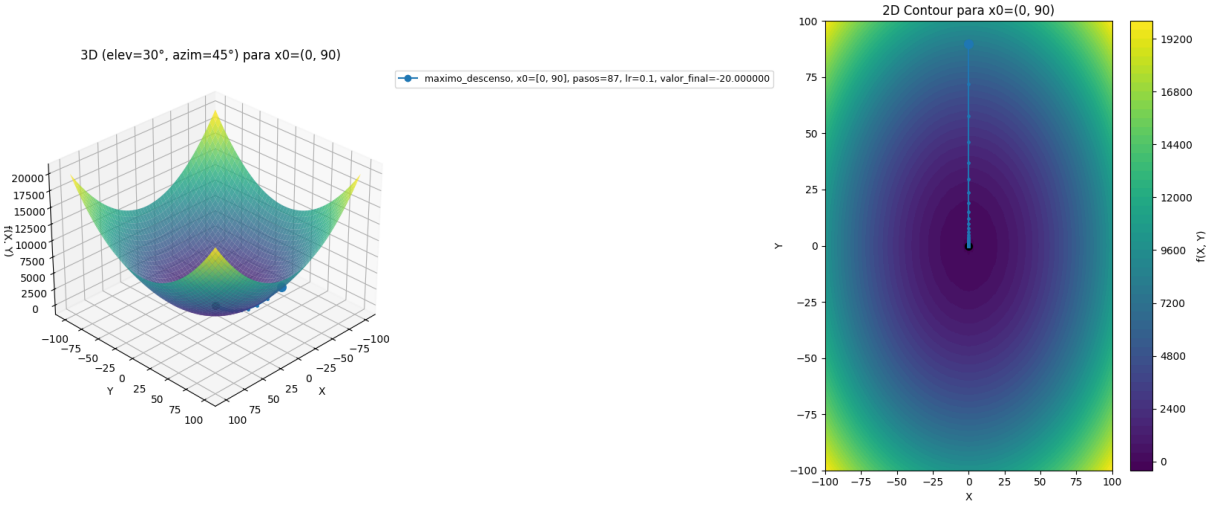


Figura 3: Gráfico de resultados del algoritmo de Máximo Descenso con lr=0.1

Probando para lr=0.01 se llega al éxito, pero de forma lenta comparado con el anterior. Este lr también logró converger al valor de -20 desde los diferentes puntos de inicio, pero necesitó cerca el máximo de 1000 iteraciones para hacerlo (Figura 5). Cuando se limitó a 50 o 200 iteraciones, este lr no fue suficiente y el algoritmo se detuvo prematuramente.

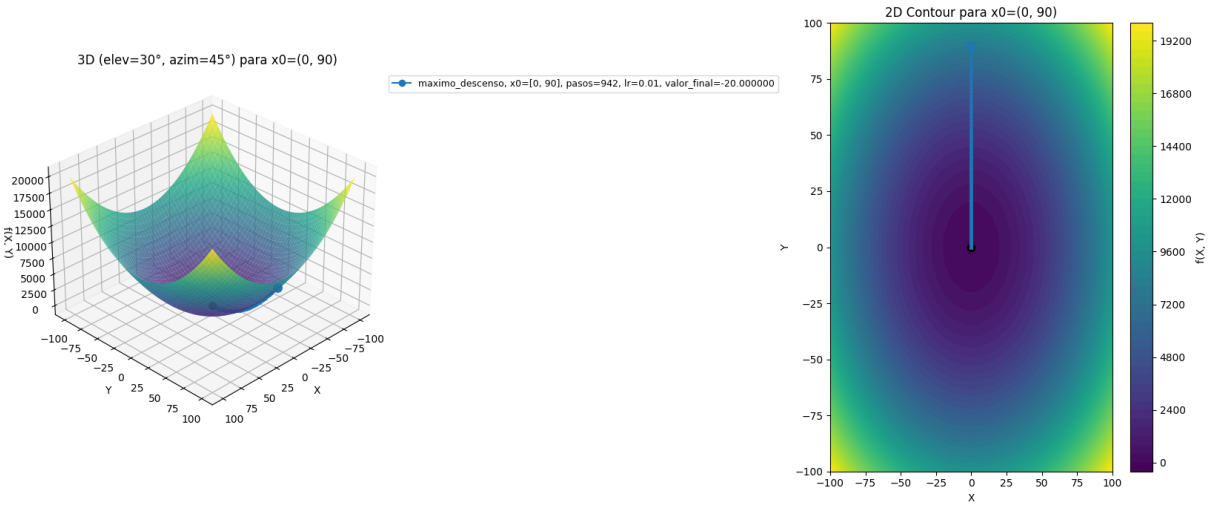


Figura 4: Gráfico de resultados del algoritmo de Máximo Descenso con lr=0.01

Finalmente, con lr = 0.001 se falla. Este tamaño del paso es demasiado pequeño porvocando que el algoritmo avance muy lentamente y no logre converger en ninguno de los experimentos, ni siquiera con 1000 iteraciones (Figura 6) evidenciando la importancia crítica de ajustar el lr.

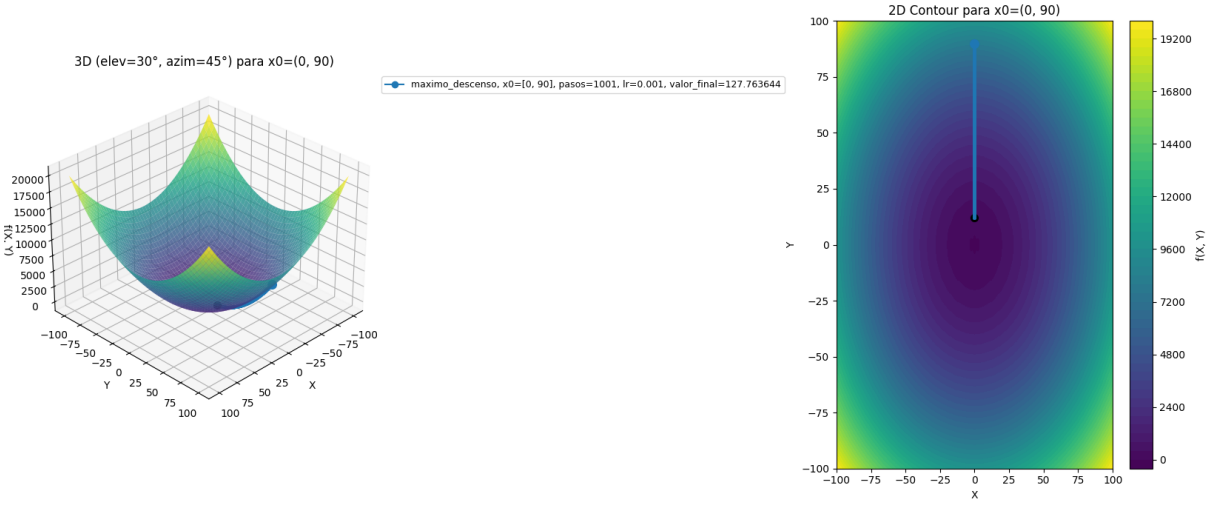


Figura 5: Gráfico de resultados del algoritmo de Máximo Descenso con lr=0.001

Comparativa de Algoritmos y conclusiones

- El **Método de Newton** es rápido pero inestable. Cuando funciona (en este caso, cuando x_0 está en el eje Y), es increíblemente eficiente (1 paso). Sin embargo, es muy sensible al punto de inicio y converge a mínimos locales incorrectos si x_0 no está en la cuenca de atracción adecuada. La función tiene un mínimo global claro en [0, 0] (valor -20), pero también posee múltiples mínimos locales o puntos estacionarios en el eje X que atrapan al algoritmo de Newton si se inicia cerca de ellos.
- El **Máximo Descenso** es lento pero seguro. Demostró ser mucho más robusto, encontrando el mínimo correcto (-20) desde todos los puntos de inicio probados. Su desventaja es que es mucho más lento que Newton y su éxito demostró ser altamente dependiente de la correcta elección del tamaño del paso (α, o learning rate, lr) y del número máximo de iteraciones permitidas.