

# Modelación y Simulación 2025

## Lab 03

14.agosto.2025

1. Implementar los siguientes métodos de descenso gradiente (naïve = tamaño de paso  $\alpha$  constante):

- descenso gradiente naïve con dirección de descenso aleatoria
- descenso máximo naïve
- descenso gradiente de Newton, con Hessiano exacto
- un método de gradiente conjugado (Fletcher-Reeves, Hestenes-Stiefel, Polak-Ribière)
- el método BFGS.

En cada uno de los métodos, su función debe recibir los siguientes argumentos:

- la función objetivo  $f$ ,
- el gradiente de la función objetivo  $df$ ,
- el hessiano  $ddf$  (cuando sea necesario),
- un punto inicial  $\mathbf{x}_0 \in \mathbb{R}^n$ ,
- el tamaño de paso  $\alpha > 0$ ,
- el número máximo de iteraciones  $maxIter$ ,
- la tolerancia  $\varepsilon$ , así como un criterio de paro.

Como resultado, sus algoritmos deben devolver: la mejor solución encontrada *best* (la última de las aproximaciones calculadas); la secuencia de iteraciones  $\mathbf{x}_k$ ; la secuencia de valores  $f(\mathbf{x}_k)$ ; la secuencia de errores en cada paso (según el error de su criterio de paro).

Además, es deseable indicar el número de iteraciones efectuadas por el algoritmo, y si se obtuvo o no convergencia del método.

2. Testar sus algoritmos del Ejercicio 1 con las siguientes funciones:

a) La función  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ , dada por

$$f(x, y) = x^4 + y^4 - 4xy + \frac{1}{2}y + 1.$$

Punto inicial:  $\mathbf{x}_0 = (-3, 1, -3, 1)^T$ , Óptimo:  $\mathbf{x}^* = (-1.01463, -1.04453)^T$ ,  $f(\mathbf{x}^*) = -1.51132$ .

b) La función de Rosembrock 2-dimensional  $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ , dada por

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

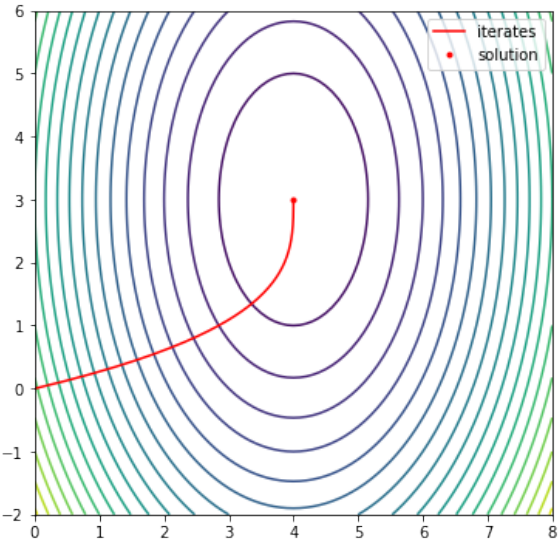
Punto inicial:  $\mathbf{x}_0 = (-1.2, 1)^T$ , Óptimo:  $\mathbf{x}^* = (1, 1)^T$ ,  $f(\mathbf{x}^*) = 0$ .

c) La función de Rosembrock 7-dimensional  $f : \mathbb{R}^7 \rightarrow \mathbb{R}$ , dada por

$$f(\mathbf{x}) = \sum_{i=1}^6 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2.$$

Punto inicial:  $\mathbf{x}_0 = (-1.2, 1, 1, 1, 1, -1.2, 1)^T$ , Óptimo:  $\mathbf{x}^* = (1, 1, \dots, 1)^T$ ,  $f(\mathbf{x}^*) = 0$ .

Para las funciones 2D, muestre visualizaciones de la secuencia de aproximaciones  $\{\mathbf{x}_k\}$  convergiendo al mínimo local de su función.



En cada uno de los casos, hallar un tamaño de paso  $\alpha$  que garantice la convergencia de los métodos, y elabore una tabla comparativa de los resultados, error, número de iteraciones requeridas por cada método. Por ejemplo:

Algoritmo de optimización	Convergencia (Sí/No)	Número de Iteraciones	Solución	Error
Descenso gradiente Descenso gradiente aleatorio Descenso máximo Descenso de Newton Fletcher-Reeves BFGS				

Elabore gráficas que muestren el error de aproximación, en función del número de iteración, y muestre la comparación de la evolución de la convergencia en sus cinco métodos. A partir de estas gráficas, discuta cuál de los métodos es más efectivo, en cada caso. Para ello, debe tomar en cuenta:

- la solución aproximada obtenida
- el error de aproximación
- la norma del gradiente en la solución.

3. Construya una función “suma de gaussianas” 2-dimensional, en la forma

$$f(\mathbf{x}) = - \sum_{i=1}^k \exp \left( - \frac{1}{2\sigma} ||\mathbf{x} - \mathbf{x}_i||_2^2 \right),$$

donde  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$  son puntos en el rectángulo  $[0, 8] \times [0, 8]$  elegidos de forma aleatoria (distribución uniforme). Use  $k = 8$  (8 gaussianas), Aquí,  $\sigma > 0$  es un parámetro de escala definido por el usuario.

**Nota:** Una vez se construyen los puntos aleatorios  $\mathbf{x}_k$  deberá congelarlos para trabajar siempre con la misma función  $f$ .

Aplique varias veces alguno de los métodos implementados a la función  $f$ , con inicializaciones  $\mathbf{x}_0$  distintas, de forma que se puedan obtener los diferentes mínimos locales de la función.

Elabore una tabla con todos los mínimos encontrados, y muestre visualizaciones de diferentes secuencias de aproximaciones  $\{\mathbf{x}_i\}$  convergiendo a cada uno de los mínimos locales de su, al igual que en el ejercicio anterior.

4. Considere el siguiente conjunto de datos que se incluye en el archivo `datos_lab3.csv`. Estos datos corresponden a una serie de tiempo.

Se quiere realizar un modelo de regresión de la forma

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 \sin(7x) + \beta_4 \sin(13x),$$

que explique la relación entre las variables  $x$  y  $y$ .

Para ello, vamos a formular un problema de optimización en la variable vectorial  $\beta = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4) \in \mathbb{R}^5$ . Hallar el modelo de regresión corresponde a hallar el vector  $\beta$  que minimiza la función de error regularizada

$$\mathbf{E}_\lambda(\beta) = \sum_{i=1}^n (f(\mathbf{x}_i) - y_i)^2 + \lambda \sum_{i=1}^{n-1} (f(\mathbf{x}_{i+1}) - f(\mathbf{x}_i))^2.$$

Implementar en Python un algoritmo de optimización para resolver el problema de regresión en los siguientes 3 casos:

- a) cuando  $\lambda = 0$ , regresión lineal sin regularización.
- b) cuando  $\lambda = 100$ , regresión lineal con regularización de Tychonoff.
- c) cuando  $\lambda = 500$ .

Compare las tres soluciones obtenidas en la misma gráfica, junto con los datos originales. Discuta las soluciones obtenidas y sus resultados. Explique cuál es el papel de la constante  $\lambda$ .