Optimización

Método del Descenso por Gradiente usando Backtracking y Bisección

Tarea 4

23 de febrero de 2021

Erika Rivadeneira Pérez
erika.rivadeneira@cimat.mx
Matemáticas Aplicadas - CIMAT

1. Resumen

En el presente reporte se expone el método del descenso por gradiente para encontrar mínimos en una función dada. Se utilizan los métodos de backtracking y de bisección para encontrar el tamaño de paso adecuado. Se probará el método del descenso en las funciones de Rosembrock, Wood y una función suavizadora. Finalmente, se comparan los resultados obtenidos.

2. Introducción

El método de descenso por gradiente es uno de los algoritmos de optimización más populares en aprendizaje automático, particularmente por su uso extensivo en el campo de las redes neuronales.

El encontrar mínimos puede ser complicado en algunas funciones y existen algunas opciones para encontrarlo, tales como

- Analítica, que consiste en calcular la derivada cerrada de una función y encontrar los puntos donde la derivada es igual a cero.
- Métodos numéricos, localizarse en un punto de la función y tratar de descender al punto mínimo usando información de la primera derivada (steepest gradient descent). También podemos usar información de la segunda derivada (Newtown's gradient descent).
- Usar métodos aproximativos.

En este reporte se considera el método numérico del descenso por gradiente (steepest descent) para encontrar mínimos, localizándonos en un punto de la función y descendiendo al mínimo de la misma.

3. Metodología

El algoritmo de descenso por gradiente hace uso de una estrategia por búsqueda lineal, donde elige una dirección d_k y busca a lo largo de esta dirección desde la iteración actual x_k para una nueva iteración con un valor de función más bajo.
Dado $\varepsilon > 0$ buscamos $\|g_k\| < \varepsilon$. Para esto definimos una sucesión de puntos $\{x_k\}$ dada por

$$x_{k+1} = x_k - \alpha_k d_k$$

La sucesión anterior converge a un mínimo local bajo ciertas condiciones, como la primer condición de Wolfe, como se vio en clases. El valor de la dirección de búsqueda d_k considerado es $-g(x_k)$.

Para encontrar el valor de la longitud de paso α_k se utilizan backtracking y el método de bisección. El objetivo de estos dos métodos es el generar una sucesión $\{\alpha_k\}$ que nos garantice la convergencia de manera eficiente. El método de backtracking busca de manera iterativa un α_k tal que

$$f\left(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k\right) \leq f\left(\boldsymbol{x}_k\right) + c_1 \alpha \nabla f_k^T \boldsymbol{d}_k$$

para alguna constante $c_1 \in (0,1)$, verificando así la primera condición débil de Wolfe.
Por otro lado, el método de bisección genera una sucesión $\{\alpha_k\}$ por medio de las condiciones de Wolfe débiles, las cuales son

$$f\left(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k\right) \leq f\left(\boldsymbol{x}_k\right) + c_1 \alpha \nabla f_k^T \boldsymbol{d}_k$$
$$\nabla f\left(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k\right)^T \boldsymbol{d}_k \geq c_2 \nabla f_k^T \boldsymbol{d}_k$$

con $0 < c_1 < c_2 < 1$. Con valores típicos de $c_1 = 10^{-4}$ y $c_2 = 0{,}9$.[1]

4. Resultados

Se implementó el algoritmo de descenso por gradiente utilizando los métodos de búsqueda de línea de backtracking y bisección en las funciones de Rosembrock, Wood y la función suavizadora considerando puntos de inicio x_0. Además, se graficó (x_k, f_k) y $(x_k, ||g_k||)$ (ver figuras 1,2 y 3). Los resultados se muestran a continuación:

4.1. Función de Rosembrock

Para la función de Rosembrock se consideraron los casos de $n = 2$ y $n = 100$, con

$$\begin{aligned}
f(\boldsymbol{x}) =& \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(1 - x_i\right)^2\right] \\
\nabla f(x) =& \left(-400 x_1\left(x_2 - x_1^2\right) - 2\left(1 - x_1\right), \cdots,\right. \\
& \left. 200\left(x_k - x_{k-1}^2\right) - 400 x_k\left(x_{k+1} - x_k^2\right) - 2\left(1 - x_k\right), \cdots, 200\left(x_n - x_{n-1}^2\right)\right) \\
\boldsymbol{x}_0 =& [0,1,3,21,2,4,5,2,3,4]^T \quad \text{(vector inicial aleatorio)} \\
\boldsymbol{x}_0 =& [-1{,}2,1,1,\cdots,1,-1{,}2,1]^T \\
\boldsymbol{x}^* =& [1,1,\cdots,1,1]^T \\
f\left(\boldsymbol{x}^*\right) =& 0.
\end{aligned}$$

Obteniendo:

- Para $n = 10$ y $x_0 = [0,1,3,21,2,4,5,2,3,4]^T$:
  - **Por backtracking:**
    - Solución:

$$x^* = [1{,}00000002, 1{,}00000005, 1{,}0000001, 1{,}0000002, 1{,}0000004, 1{,}00000081, 1{,}00000162, 1{,}00000325, 1{,}00000652, 1{,}00001307]^T$$

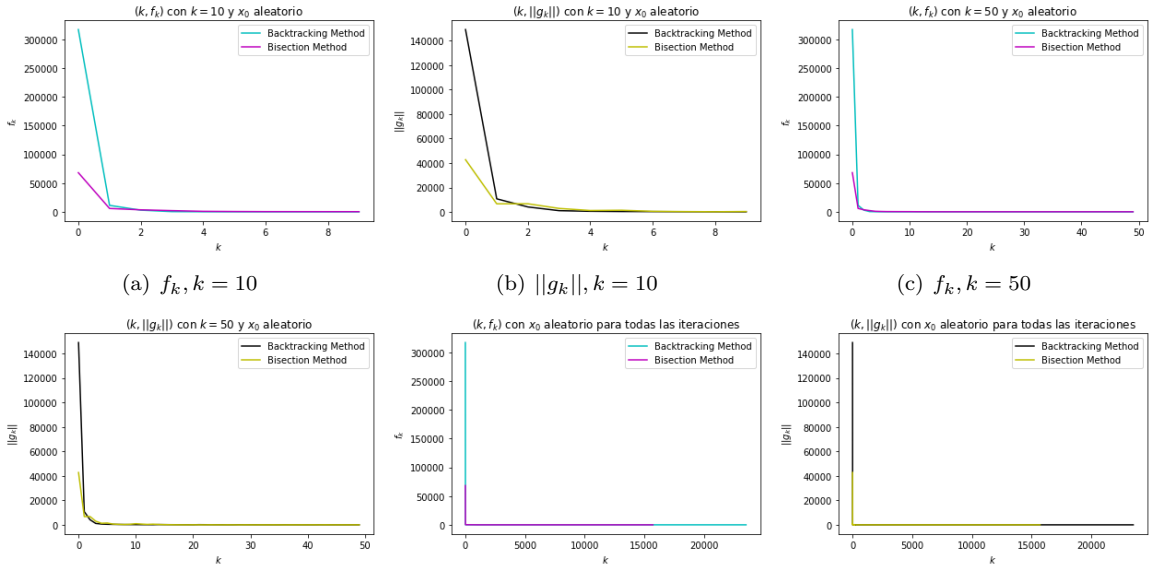
    - $\nabla f(x^*) = 9{,}902162e-06$
    - Iteraciones: 23519
    - Tiempo de ejecución: 24,77seg.
  - **Por bisección:**

- Solución:

$$x^* = [0{,}99999997, 0{,}99999994, 0{,}99999989, 0{,}99999977, 0{,}99999954,$$
$$0{,}99999909, 0{,}99999817, 0{,}99999633, 0{,}99999264, 0{,}99998525]$$

- $\nabla f(x^*) = 9{,}97425e-06$
- Iteraciones: 15715
- Tiempo de ejecución: 14,25seg.

Las gráficas de (x_k, f_k) y de $(x_k, ||g_k||)$ correspondientes a esta función para el vector inicial aleatorio se encuentran en la figura (1).



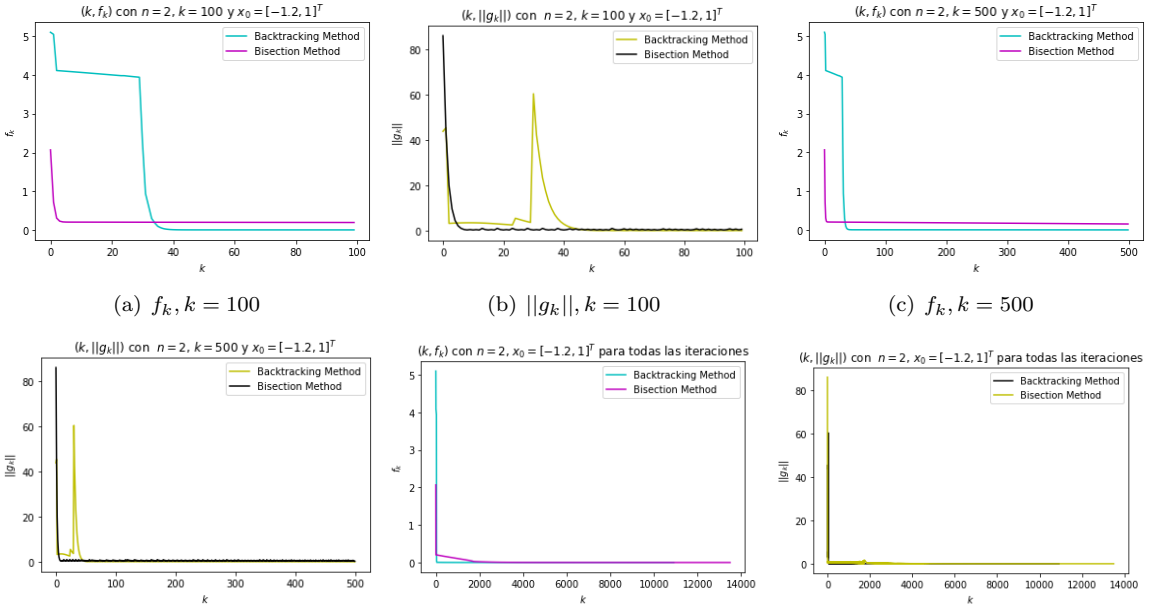
(a) $f_k, k = 10$ (b) $||g_k||, k = 10$ (c) $f_k, k = 50$

(d) $||g_k||, k = 50$ (e) f_k para todas las iteraciones (f) $||g_k||$ para todas las iteraciones

Figura 1: Convergencia en la función de Rosembrock con vector inicial aleatorio

- Para $n = 2$ y $x_0 = [-1{,}2, 1]^T$:
  - **Por backtracking:**
    - Solución: $x^* = [0{,}99999218, 0{,}99998432]^T$
    - $\nabla f(x^*) = 9{,}9971e-06$
    - Iteraciones: 10916
    - Tiempo de ejecución: 2,42seg.
  - **Por bisección:**
    - Solución: $x^* = [1{,}00000878, 1{,}00001762]^T$
    - $\nabla f(x^*) = 9{,}971e-06$
    - Iteraciones: 13498
    - Tiempo de ejecución: 2.47seg.

  Las gráficas de (x_k, f_k) y de $(x_k, ||g_k||)$ correspondientes a esta función para $n = 2$ y $x_0 = [-1{,}2, 1]$ se encuentran en la figura (2).



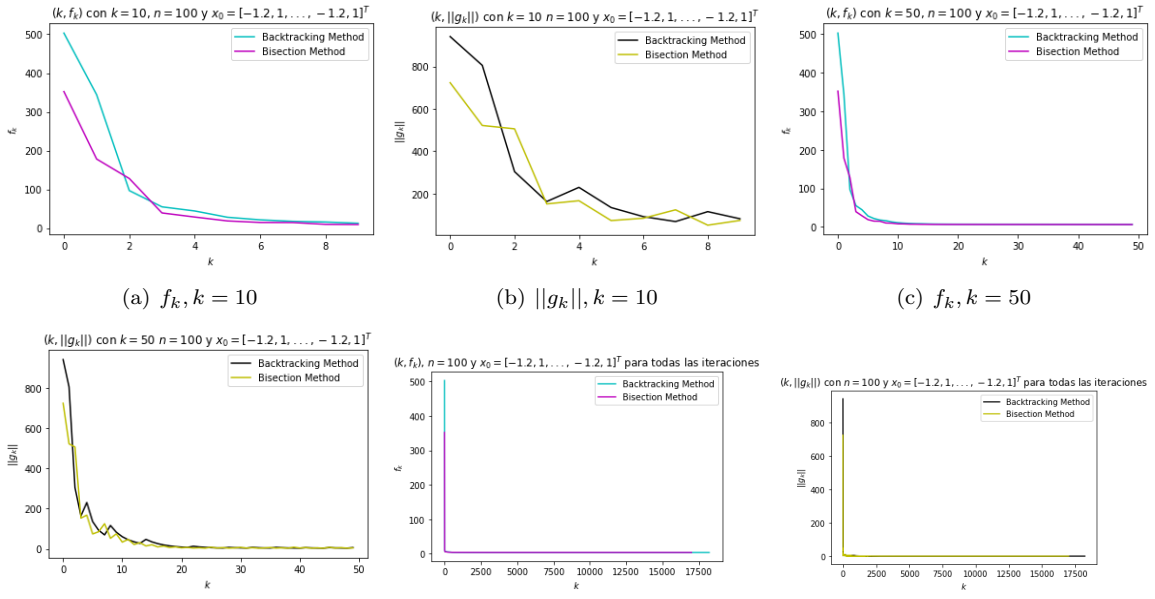
(a) $f_k, k = 100$ (b) $||g_k||, k = 100$ (c) $f_k, k = 500$

(d) $||g_k||, k = 500$ (e) f_k para todas las iteraciones (f) $||g_k||$ para todas las iteraciones

Figura 2: Convergencia en la función de Rosembrock para $n = 2$ y vector inicial $x_0 = [-1{,}2, 1]$

- Para $n = 100$ y $x_0 = [-1{,}2, 1, 1, ..., 1, -1{,}2, 1]^T$
  - **Por backtracking:**
    - Solución: $x^* = [-0{,}9932861, 0{,}99665107, \ldots, 0{,}999993530{,}99998703]$
    - $\nabla f(x^*) = 9{,}9825e - 06$
    - Iteraciones: 18213
    - Tiempo de ejecución: 182.62seg.
  - **Por bisección:**
    - Solución: $x^* = [-0{,}9932861, 0{,}99665107, \ldots, 0{,}99999299, 0{,}99998595]$
    - $\nabla f(x^*) = 9{,}9936e - 06$
    - Iteraciones: 16996
    - Tiempo de ejecución: 131.61seg.

Las gráficas de (x_k, f_k) y de $(x_k, ||g_k||)$ correspondientes a esta función para $n = 100$ y $x_0 = [-1{,}2, 1, 1, ..., 1, -1{,}2, 1]$ se encuentran en la figura (3).



(a) $f_k, k = 10$ (b) $||g_k||, k = 10$ (c) $f_k, k = 50$

(d) $||g_k||, k = 50$ (e) f_k para todas las iteraciones (f) $||g_k||$ para todas las iteraciones

Figura 3: Convergencia en la función de Rosembrock para $n = 100$ y vector inicial $x_0 = [-1{,}2, 1, ..., -1{,}2, 1]$

4.2. Función de Wood

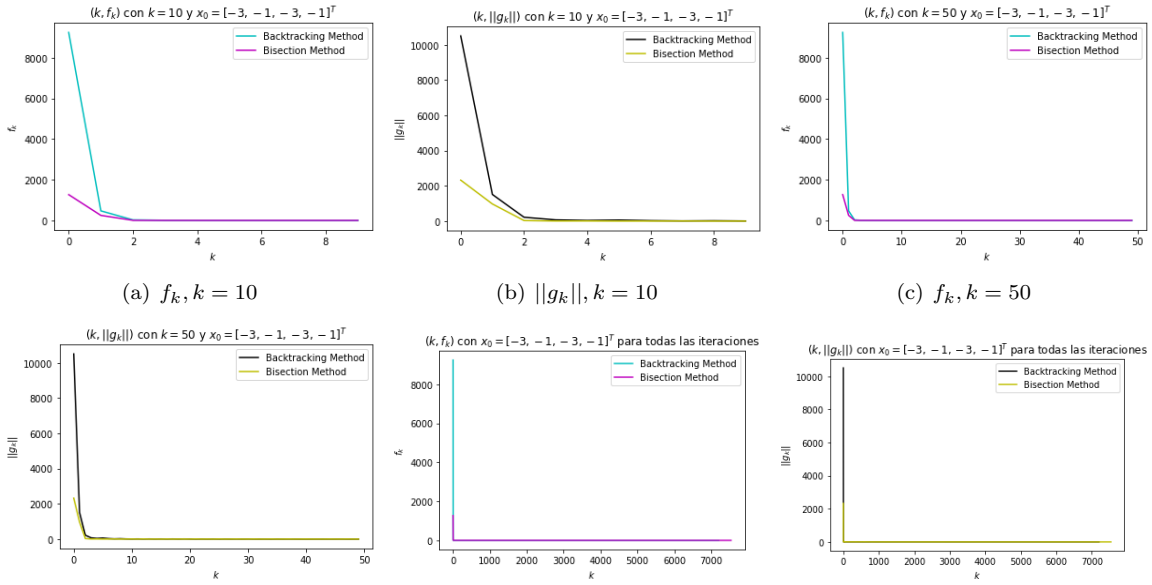
Se consideró:

$$\begin{aligned} f(\boldsymbol{x}) =& 100\left(x_1^2 - x_2\right)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90\left(x_3^2 - x_4\right)^2 \\ &+ 10{,}1\left[(x_2 - 1)^2 + (x_4 - 1)^2\right] + 19{,}8\,(x_2 - 1)\,(x_4 - 1) \\ \boldsymbol{x}_0 =& [-3, -1, -3, -1]^T \\ \boldsymbol{x}^* =& [1, 1, 1, 1]^T \\ f\left(\boldsymbol{x}^*\right) =& 0. \end{aligned}$$

Obteniendo:

- **Por backtracking:**
  - Solución: $x^* = [1{,}00000307, 1{,}00000617, 0{,}99999692, 0{,}99999383]^T$
  - $\nabla f(x^*) = 9{,}9911e - 06$
  - Iteraciones: 7205
  - Tiempo de ejecución: 2.69seg.
- **Por bisección:**
  - Solución: $x^* = [1{,}00000343, 1{,}00000689, 0{,}99999656, 0{,}99999312]^T$
  - $\nabla f(x^*) = 9{,}9541e - 06$
  - Iteraciones: 7541
  - Tiempo de ejecución: 2.26seg.

Las gráficas de (x_k, f_k) y de $(x_k, ||g_k||)$ correspondientes a esta función se encuentran en la figura (4).



(a) $f_k, k = 10$ (b) $||g_k||, k = 10$ (c) $f_k, k = 50$

(d) $||g_k||, k = 50$ (e) f_k para todas las iteraciones (f) $||g_k||$ para todas las iteraciones

Figura 4: Convergencia en la función de Wood para el vector inicial $x_0 = [-3, -1, -3, -1]$

4.3. Función de Suavizamiento

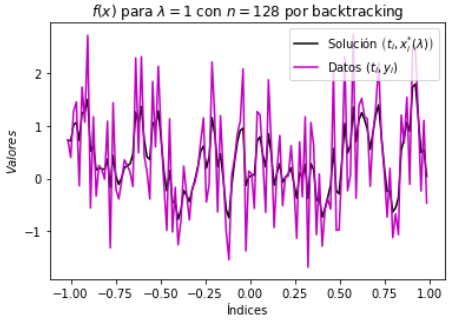
Por el método del descenso por gradiente se obtuvo el mínimo de $f(\boldsymbol{x})$ para $\eta \sim \mathcal{N}(0, \sigma)$ y $\lambda, \sigma > 0$ y se graficó (t_i, y_i) y $(t_i, x_i^*(\lambda))$ (ver figura 5).

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} (x_i - y_i)^2 + \lambda \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2$$

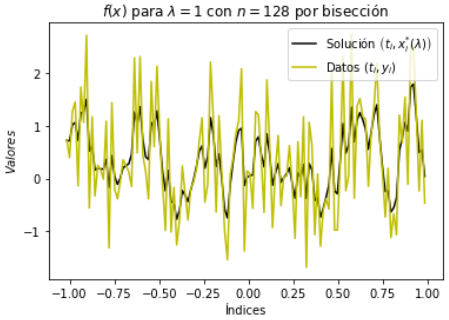
$$y_i = t_i^2 + \eta, t_i = \frac{2}{n-1}(i-1) - 1, i = 1, 2, \cdots, n.$$

Se consideraron los casos cuanddo $\lambda \in \{1, 10, 1000\}$ con $n = 128$. Obteniendo:

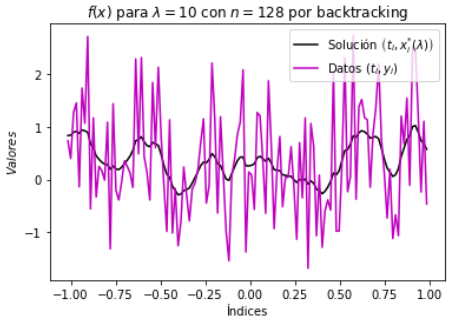
- Para $\lambda = 1$:
  - Por backtracking: $\nabla f(x^*) = 4{,}29893e - 06$ con 25 iteraciones.
  - Por bisección: $\nabla f(x^*) = 9{,}2868e - 06$ con 67 iteraciones.
- Para $\lambda = 10$:
  - Por backtracking: $\nabla f(x^*) = 9{,}624e - 06$ con 235 iteraciones.
  - Por bisección: $\nabla f(x^*) = 8{,}7395e - 06$ con 251 iteraciones.
- Para $\lambda = 100$:
  - Por backtracking: $\nabla f(x^*) = 8{,}9513e - 06$ con 2578 iteraciones.
  - Por bisección: $\nabla f(x^*) = 8{,}7395e - 06$ con 251 iteraciones.



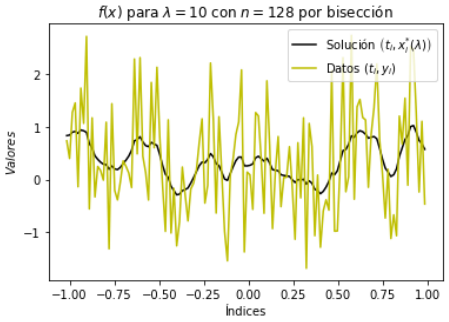
(a) $f(x)$ para $\lambda = 1$ con $n = 128$ por backtracking



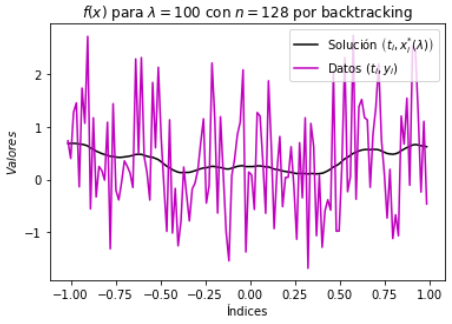
(b) $f(x)$ para $\lambda = 1$ con $n = 128$ por bisección



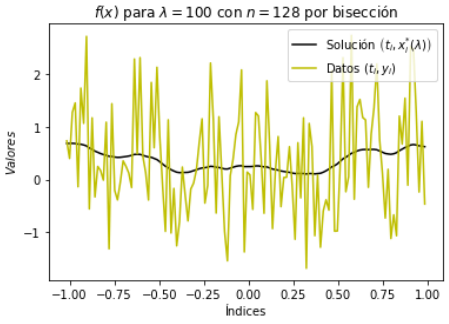
(c) $f(x)$ para $\lambda = 10$ con $n = 128$ por backtracking



(d) $f(x)$ para $\lambda = 10$ con $n = 128$ por bisección



(e) $f(x)$ para $\lambda = 100$ con $n = 128$ por backtracking



(f) $f(x)$ para $\lambda = 100$ con $n = 128$ por bisección

Figura 5: Convergencia en la función de Wood para el vector inicial $x_0 = [-3, -1, -3, -1]$

5. Conclusiones

A pesar de que se solicitaba que el algoritmo convergiera al mínimo dado, el método convergió a mínimos locales. En particular se obtuvo el vector solución $x^* = [-0{,}9932861, 0{,}99665107, ..., 0{,}999993530, 0{,}99998703]$ en la función de Rosembrock para $n = 100$. Dado que los métodos utilizados no disciernen entre mínimos locales y globales, se puede considerar que el resultado obtenido fue satisfactorio.

A pesar de que el método de bisección hace uso de más condiciones de Wolf no se observó que sea estrictamente mejor que el método de bisección pues varia dependiendo de cada caso, esto se concluye por el tiempo de ejecución y las iteraciones realizadas.

Finalmente, se pudo observar que mientras mayor sea el valor de λ en la función de suavizamiento se obtiene una mejor aproximación a los valores reales reduciendo el ruido de la función.

Referencias

[1] J. Nocedal and S. J. Wright. Numerical Optimization. Springer Series in Operation Research, 2000.