Métodos numéricos
Zaira Carolina Martinez Vargas

Tarea 4:
Método de Newton y gradiente descendente

2023-02-25

**Resumen**

El objetivo de este trabajo fue explorar los métodos de gradiente descendente con paso fijo y el método de Newton para encontrar mínimos en distintas funciones.

# 1. Introducción

Los algoritmos de optimización sin restricciones, comienzan con un vector inicial $x_0$, y buscan de forma iterativa el mínimo de una función. Para decidir la dirección de la función, se utiliza información de esta, por ejemplo, su gradiente o Hessiano. Existen distintos algoritmos para encontrar los mínimos, sin embargo, en este trabajo nos enfocaremos en el método de gradiente descendente con paso fijo, y con el método de Newton.

# 2. Métodos

## Método de gradiente descendente

Supón que se busca el mínimo de una función f(x), x $\in \mathbb{R}^n$ y f: $\mathbb{R}^n \rightarrow \mathbb{R}$ [1]. La idea de este método, es tomar la dirección negativa del gradiente. No obstante, los resultados de este método, varían dependiendo del tamaño de paso escogido.

**Algorithm 1** Gradiente descendente

**Entrada:** Vector inicial $x_0$, gradiente de la función en este punto $g0$, tolerancia TOL, máximo de iteraciones max_iter y tamaño de paso $\alpha$.
**Salida:** Mínimo de la función
**for** k = 0 hasta max_iter **do**
    x = x - $\alpha$*g0
    g0 = $\nabla$ f(x)
    **if** $||\nabla f(x)|| \leq$ TOL **then**
        converge y regresa x
regresa x

## Método de Newton para búsqueda en línea

Este método utiliza el Hessiano para acelerar la convergencia, sin embargo, el cálculo del Hessiano no es muy eficiente si se intenta hacer con la idea de diferencias finitas. Así, itera de la forma

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

se debe de verificar que la matriz Hessiana no sea una matriz singular, y que sea invertible.

**Algorithm 2** Método de Newton

**Entrada:** Vector inicial $x_0$, gradiente de la función en este punto $g0$, tolerancia TOL, máximo de iteraciones max_iter y tamaño de paso $\alpha$.
**Salida:** Mínimo de la función
Obtener el valor de $d = (\nabla^2 f(x_0))^{-1}(\nabla f(x_0))$
**for** k = 0 hasta max_iter **do**
    x = x + d
    d = $(\nabla^2 f(x))^{-1}(\nabla f(x))$
    **if** $||\nabla f(x)|| \leq$ TOL **then**
        converge y regresa x
regresa x

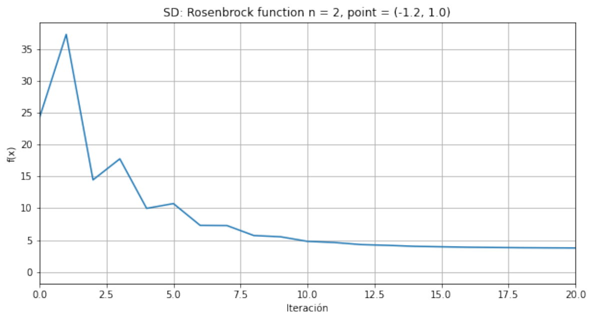
# 3. Resultados (gráficas, tablas, estadísticas de ejecución, etc)

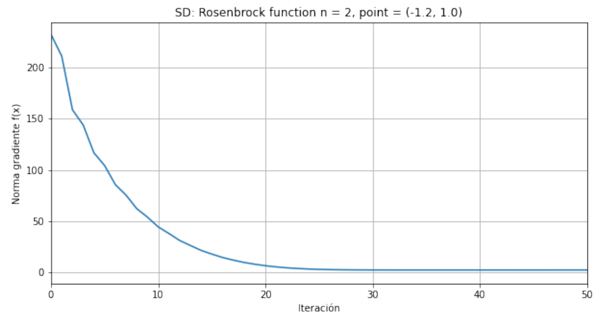
Estos algoritmos se probaron con tres funciones: Rosenbrock, Wood y una función extra. Se graficó la función $f$ evaluada en los puntos x de cada iteración y la norma del gradiente, ya que esta se utilizó como criterio de parada de los métodos.

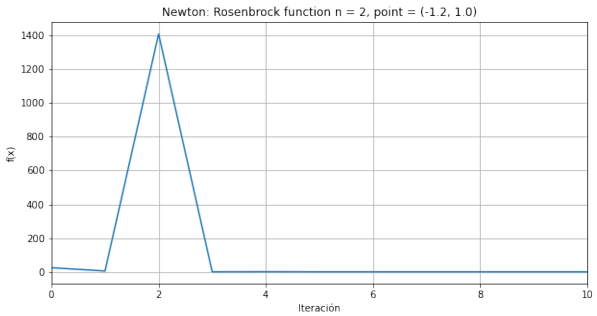
## Función Rosenbrock

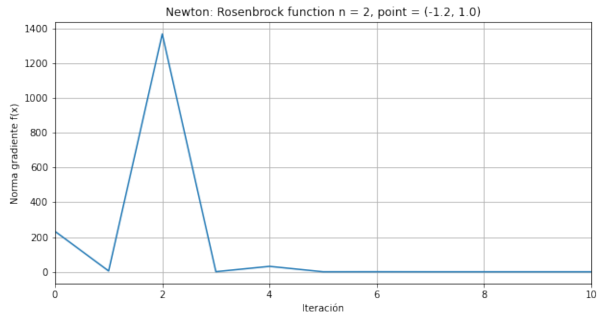
$$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right],$$

primero, la función se probó para n = 2, y un vector inicial de (-1.2, 1.0), para el método de gradiente descendente se utilizó la abreviación SD.

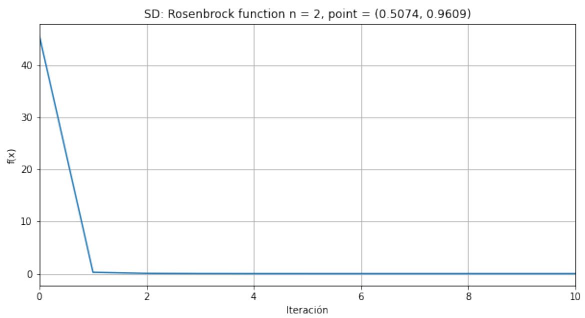


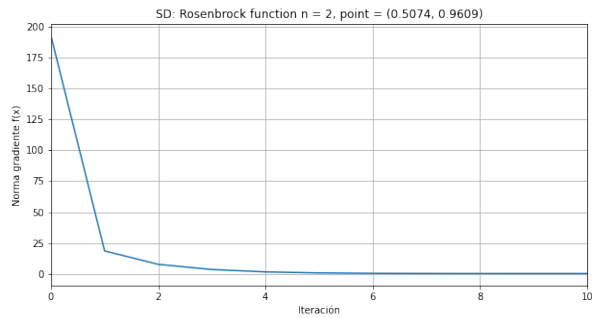



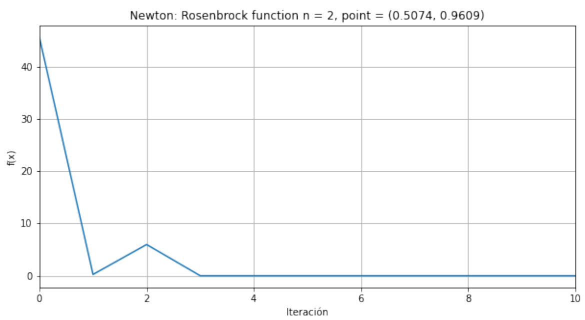



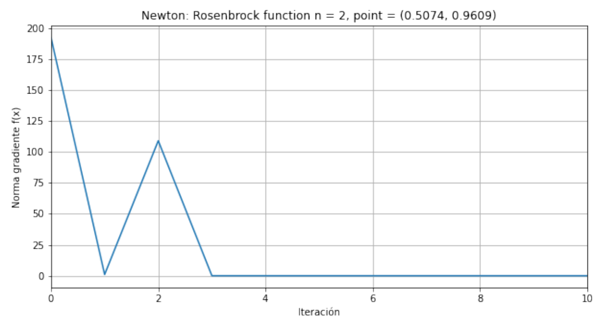




Después, se probó para un vector aleatorio con ambos métodos:

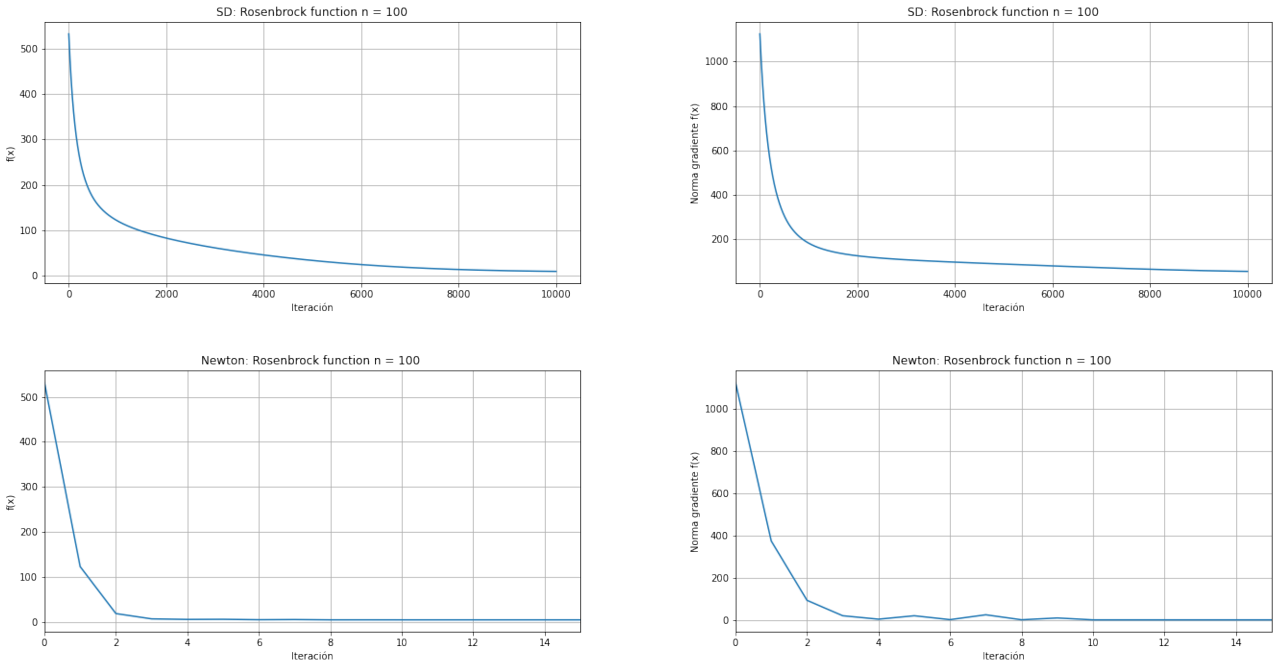


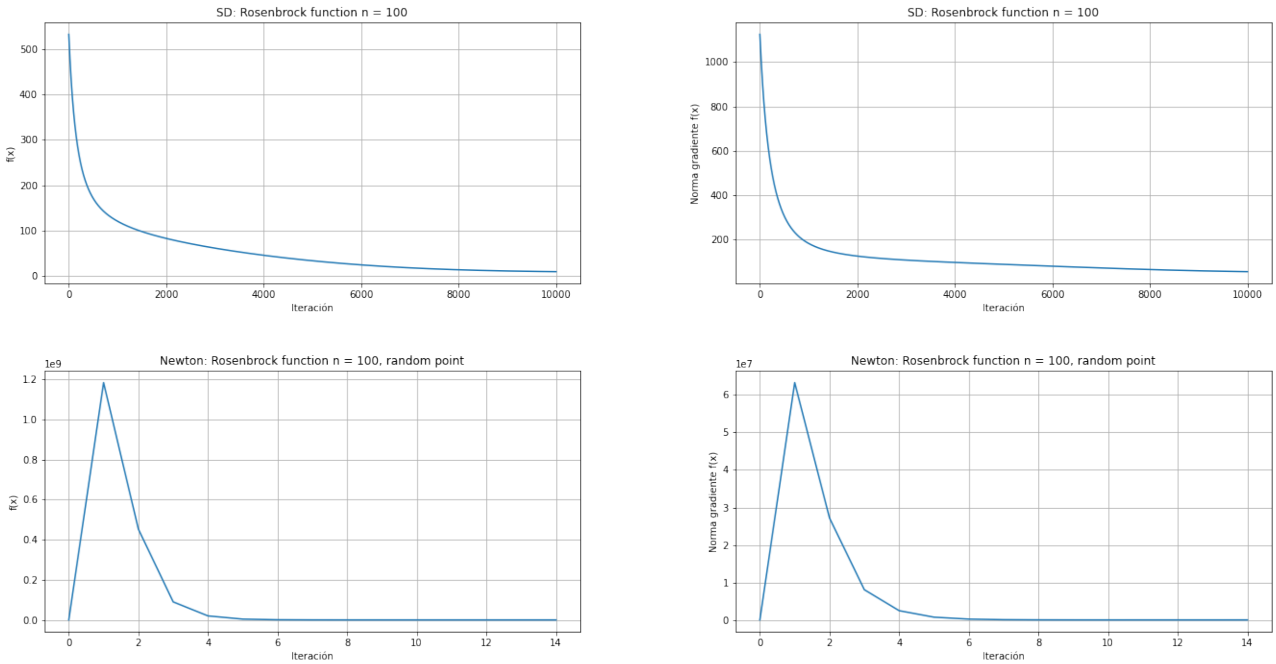




Este procedimiento se repitió para n = 100,



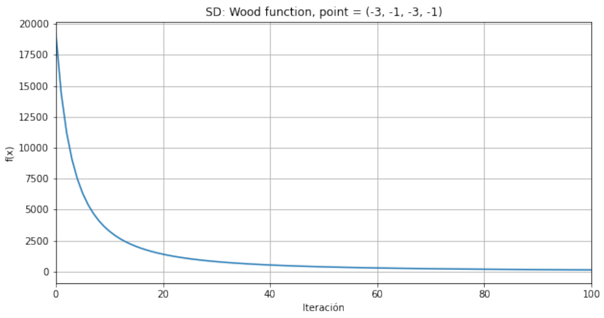

de la misma forma, se generó un vector aleatorio con n = 100, y se obtuvo

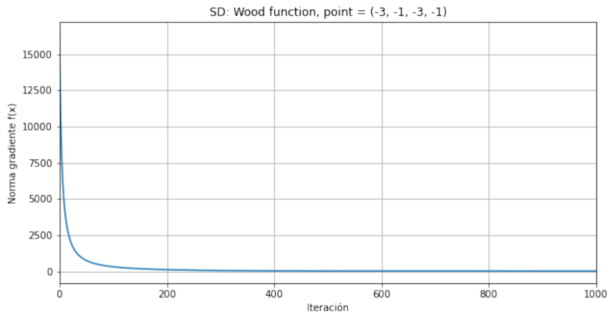


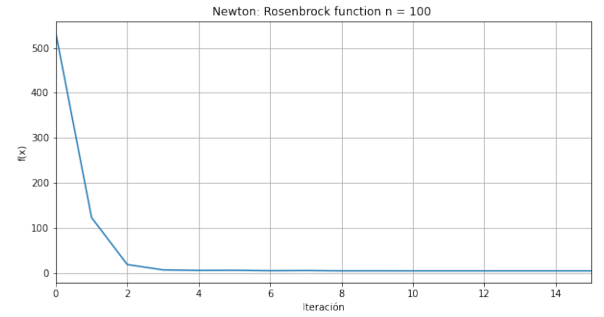

## Función Wood

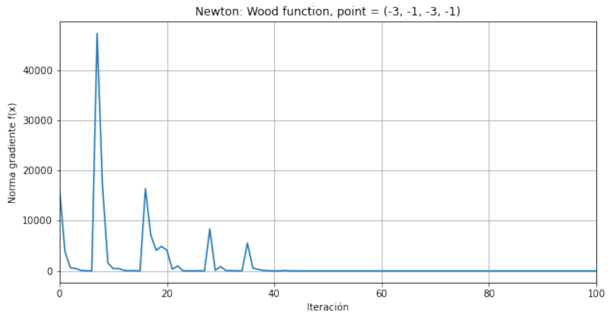
$$f(\mathbf{x}) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10{,}1\left[(x_2 - 1)^2 + (x_4 - 1)^2\right] + 19{,}8(x_2 - 1)(x_4 - 1)$$

Se utilizó el vector $x_0 = [-3, -1, -3, -1]$, como inicializador, y se obtuvieron:

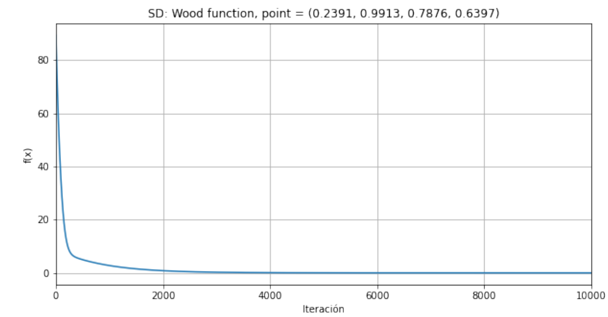


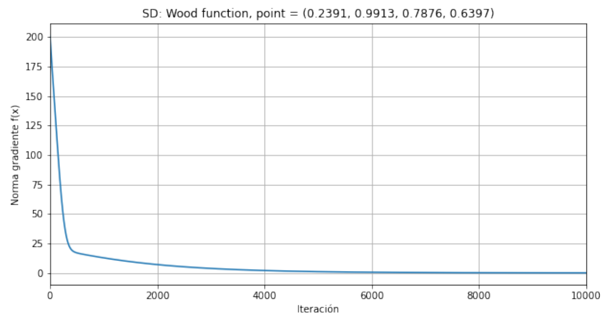



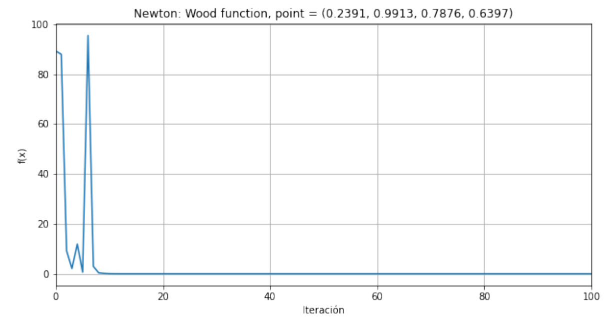



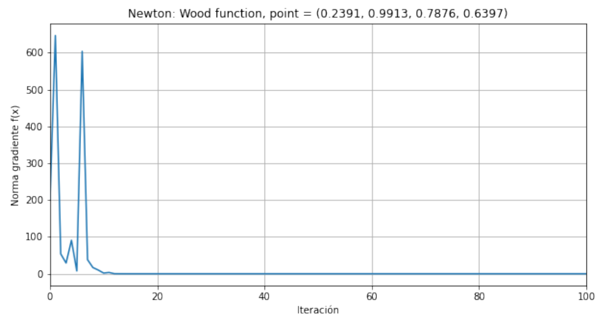




Además, con un vector aleatorio de inicializador:



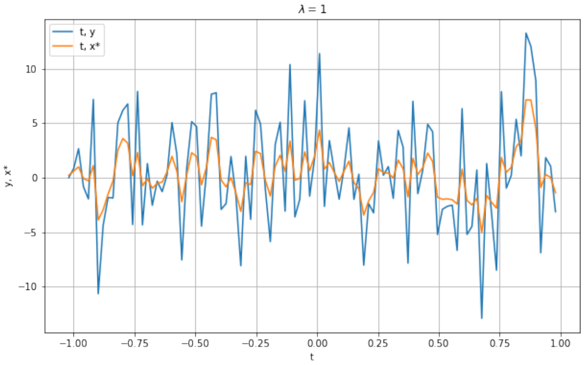



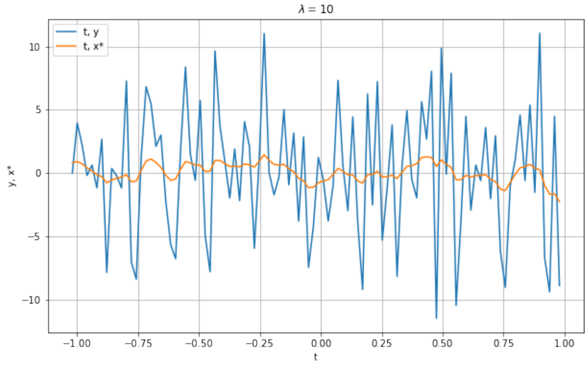



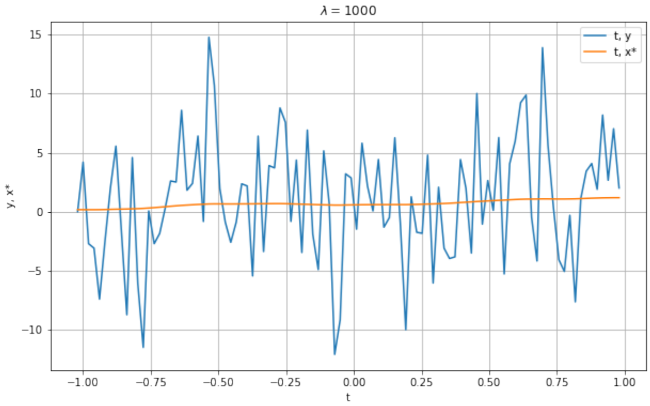

## Función extra

$$
\begin{aligned}
&f(\mathbf{x}) = \sum_{i=1}^{n}(x_i - y_i)^2 + \lambda \sum_{i=1}^{n-1}(x_{i+1} - x_i)^2, \\
&y_i = \left[\frac{2}{n-1}(i-1) - 1\right]^2 + \eta, \eta \sim N(0,\sigma), \sigma > 0, \\
&\lambda = 1, 10, 1000, n = 128
\end{aligned}
$$

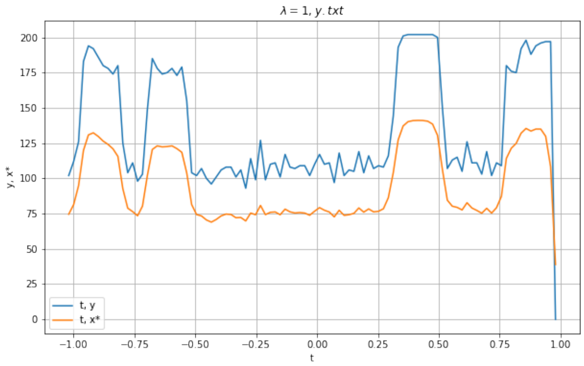
Para esta función, no se conoce el mínimo, y se busca graficar el valor de $(t_i, y_i)$ y $(t_i, x_i^*(\lambda))$. Las tres primeras figuras de esta sección son para un $y$ que contiene un elemento aleatorio de una distribución normal, y un valor de $\sigma = 5$, y un vector inicial de ceros.

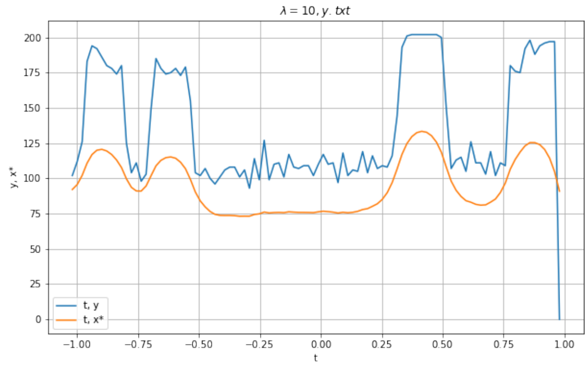


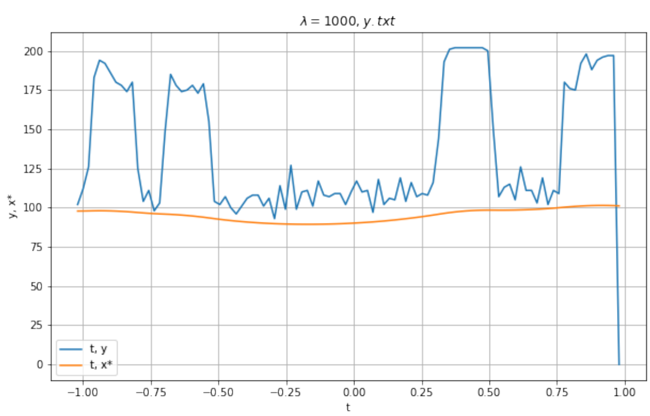




En este caso, se utilizó el archivo proporcionado y para cada $\lambda$ se observa un diferente comportamiento.



# 4. Conclusiones

Para mejorar los resultados o la convergencia con las funciones de Rosenbrock y Wood, cambiar el tamaño de paso era significativo. En la mayoría de las ejecuciones se utilizó $\alpha = 0.002$, por prueba y error. Además, en las gráficas se observa el comportamiento inicial con el método de Newton, una subida y bajada muy pronunciada para después acercarse más al mínimo. También, en la función de Rosenbrock, se observó una convergencia más rápida con el método de Newton. En la última función, se observa como se comporta la búsqueda del mínimo, como un desplazamiento y reducción con respecto a parámetros de la función. Y al final, un desplazamiento muy marcado.

# Referencias

[1] Stephen Wright, Jorge Nocedal, et al. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.