Tarea 3-Optimización 1

1st Daniel Vallejo Aldana
Departamento de Matemáticas
Universidad de Guanajuato
daniel.vallejo@cimat.mx

***Resumen*—El propósito del presente trabajo es implementar el método de Newton para búsqueda en linea para encontrar el argumento que minimiza la función planteada en el problema, así mismo se busca comparar este método con el método de gradiente descendente, especialmente en el número de iteraciones que tiene que realizar cada algoritmo para encontrar el minimizador de la función.**

***Palabras clave*—Método de Newton, gradiente descendente.**

I. Introducción

Sea $f : \mathbb{R}^n \to \mathbb{R}$ y consideremos la iteración de Newton, la cual está dada de la siguiente forma

$$\mathbf{d}_k = -\mathbf{H}(\mathbf{x}_k)\mathbf{g}_k$$

Donde $(\mathbf{x}_k)$ representa la matriz Hessiana de la función f. Como la la matriz hessiana de la función f no es necesariamente positiva definida, podemos ver que la dirección $\mathbf{d}_k$ puede no ser de descenso. Sabemos también que para una vecindad alrededor del punto solución $\mathbf{x}^*$ tal que $H(\mathbf{x}^*)$ es definido positivo, el hessiano $H(\mathbf{x})$ también va a ser positivo definido por lo que el método de Newton va a estar bien definido en esta región y va a converger de forma cuadrática, provisto de que las longitudes de paso α_k son eventualmente siempre 1. [1]

De lo anterior podemos ver que una desventaja del método de Newton para búsqueda en linea es que si el punto inicial está lejos del valor de la solución, éste método puede no converger ya que $H(\mathbf{x}_k)$ puede no ser positiva definida o ser una matriz singular, donde una matriz singular es aquella no invertible.

II. Método/Algoritmo

II-A. *Método de Newton para búsqueda en linea*

Para el método de Newton se utilizó el algoritmo basado en la dirección de descenso de Newton, donde cada vez que se calculaba el Hessiano en el punto $\mathbf{x}_k$ se verifica que la matriz sea invertible y definida positiva para garantizar que la dirección de Newton es en efecto una dirección de descenso.

Input:x_0
Output: x^*
$k = 0$
while $\|\mathbf{g}_k\| \geq \tau_{grad}$
    Verifica si la matriz H($\mathbf{x}_k$) es definida positiva y no singular
    $\mathbf{d}_k = -H(\mathbf{x}_k)\mathbf{g}_k$
    $\alpha_k = arg_{min} f(\mathbf{x}_k + \alpha_k \mathbf{d}_k)$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$
    $k = k + 1$
end while

II-B. *Método de gradiente descendente*

Para el método de gradiente descendente se utiizó el siguiente algoritmo descrito también en la tarea 2.

Input: x_0
Output: x^*
$k = 0$, $g_0 = \nabla f(x_0)$
while $\|g_0\| \neq 0$ **do**
    $\alpha_k = argmin_{\alpha>0} f(x_k + \alpha_k g_k)$
    $x_{k+1} = x_k + \alpha_k g_k$
    $g_{k+1} = \nabla f(x_{k+1})$
    k=k+1
end while

Para este caso $\alpha_k = \frac{\mathbf{g}_k^t \mathbf{g}_k}{\mathbf{g}_k^t \mathbf{H}(\mathbf{x}_k \mathbf{g}_k}$

III. Análisis de resultados

Se generaron las tablas contenidas en el apéndice A, así como las gráficas del apéndice B en donde se muestran los resultados del método de gradiente descendente y los del método de Newton.

IV. Conclusiones

Con base en los resultados obtenidos al realizar el presente trabajo se puede concluir que al ser la función f de tipo cuadrático el método de Newton converge en una sola iteración mientras que el método de gradiente descendente requiere de un mayor número de iteraciones para poder llegar al óptimo, así mismo se concluye que en este caso la elección de punto inicial no afecta en la convergencia de ambos algoritmos. Finalmente se pude concluir que a medida que el valor de λ aumenta la gráfica que describe el óptimo vs el conjunto de índices se va suavizando.

APÉNDICES

Apéndice A
Tablas donde se muestran los resultados

Caso $\lambda = 1$:

Método de Newton	
Iteraciones	$\|\nabla f(\mathbf{x}_k\|$
1	1.6403537336495988e-12

La tabla anterior muestra el número de iteraciones y el valor del gradiente en el punto $\mathbf{x}_k$ usando el método de Newton.

Método gradiente descendente	
Iteraciones	$\|\nabla f(\mathbf{x}_k\|$
1	715.6594747376078
2	209.44953074370804
33	0.00013687566905231128
34	9.99538439815332e-05

La tabla anterior muestra el número de iteraciones y el valor del gradiente en el punto $\mathbf{x}_k$ usando el método de gradiente descendente.

Caso $\lambda = 100$:

Método de Newton	
Iteraciones	$\|\nabla f(\mathbf{x}_k\|$
1	1.293787306947764e-11

La tabla anterior muestra el número de iteraciones y el valor del gradiente en el punto $\mathbf{x}_k$ usando el método de Newton.

Método gradiente descendente	
Iteraciones	$\|\nabla f(\mathbf{x}_k\|$
1	6313.686854193508
2	2418.413226002537
3412	0.00011652024866197986
3413	9.997366399364417e-05

La tabla anterior muestra el número de iteraciones y el valor del gradiente en el punto $\mathbf{x}_k$ usando el método de gradiente descendente.

Caso $\lambda = 1000$:

Método de Newton	
Iteraciones	$\|\nabla f(\mathbf{x}_k\|$
1	1.6363315500143799e-09

La tabla anterior muestra el número de iteraciones y el valor del gradiente en el punto $\mathbf{x}_k$ usando el método de Newton.

Método gradiente descendente	
Iteraciones	$\|\nabla f(\mathbf{x}_k\|$
1	17938.348189021533
2	14975.87213487309
6998	182.46030982351215
6999	157.1641049205263

La tabla anterior muestra el número de iteraciones y el valor del gradiente en el punto $\mathbf{x}_k$ usando el método de gradiente descendente. En este caso el algoritmo salió por número de iteraciones y no por tolerancia del gradiente.

Apéndice A
Gráficas donde se muestran los resultados

Caso: $\lambda = 1{,}0$: A continuación se presentan las gráficas obtenidas para el caso $\lambda = 1$

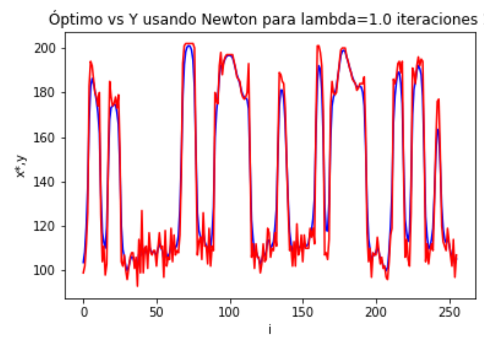

Figura 1. Gráfica obtenida utilizando método de Newton

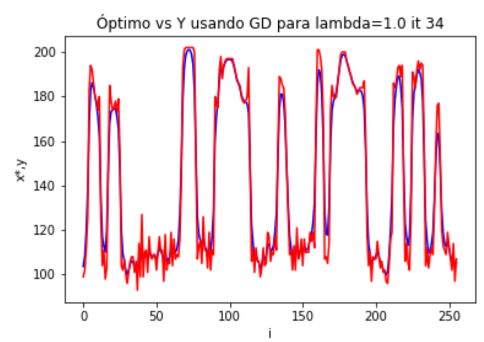

Figura 2. Gráfica obtenida utilizando método de gradiente descendente

Caso: $\lambda = 100{,}0$: A continuación se presentan las gráficas obtenidas para el caso $\lambda = 100$

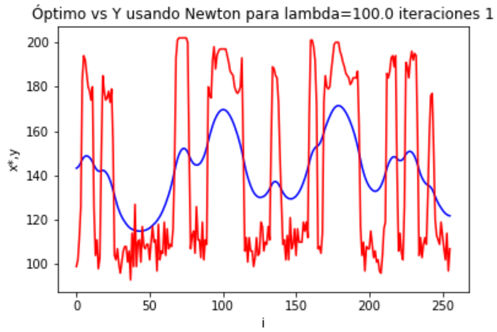

Figura 3. Gráfica obtenida utilizando método de Newton

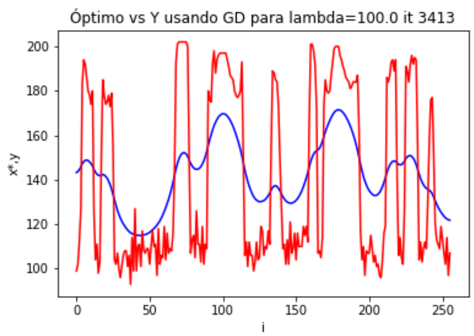

Figura 4. Gráfica obtenida utilizando método de gradiente descendente

Caso: $\lambda = 1000{,}0$*:* A continuación se presentan las gráficas obtenidas para el caso $\lambda = 1000$

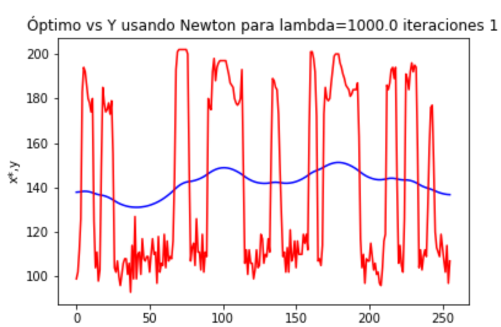

Figura 5. Gráfica obtenida utilizando método de Newton

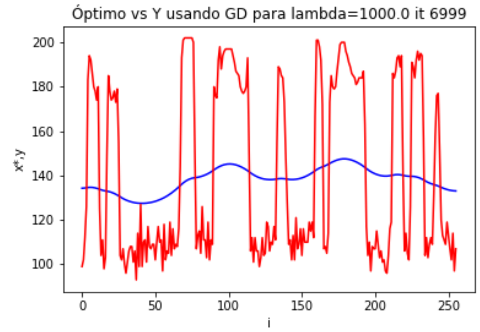

Figura 6. Gráfica obtenida utilizando método de gradiente descendente

Apéndice C
Problema 1

Problema 1. Consider the function $f(x_1, x_2) = (x_1 + x_2)^2$ at the point $x^t = [1, 0]$. We consider the search direction $p^t = [-1, 1]$. Show that p is a descent direction and find all minimizers of the function.

Solución Por definición, una dirección de descenso es un vector $\mathbf{d} \in \mathbb{R}^n$ tal que $f(\mathbf{x} + \tau\mathbf{d}) < f(\mathbf{x})$ para $\tau \in (0, T)$. Procederemos de acuerdo a la definición utilizando las condiciones dadas en el problema.
Consideremos ahora

$$\begin{aligned} f([1,0] + \tau[-1,1]) &= f(1-\tau, \tau) \\ &= (1 - \tau + \tau^2)^2 \\ &= \tau^4 - 2\tau^3 + 3\tau^2 - 2\tau + 1 \end{aligned}$$

Luego $f(\mathbf{x}^t) = 1$, por lo que tenemos que encontrar un τ tal que satisfaga la siguiente condición

$$\begin{aligned} \tau^4 - 2\tau^3 + 3\tau^2 - 2\tau + 1 < 1 \\ 0 < \tau < 1 \end{aligned}$$

Al resolver la primera desigualdad

Por lo que para $\tau \in (0, 1)$ se cumple que $\mathbf{p}^t = [-1, 1]$ es una dirección de descenso que es lo que se buscaba probar ■ Encontraremos ahora todos los minimizadores de la función anterior para esto consideremos $\nabla f(\mathbf{x})$ definido de la siguiente forma

$$\nabla f(\mathbf{x}) = \left[2(x_1 + x_2^2), 4x_2(x_1 + x_2^2)\right]^t$$

Los puntos críticos los obtenemos al resolver el siguiente sistema de ecuaciones

$$2(x_1 + x_2^2) = 0$$
$$4x_2(x_1 + x_2^2) = 0$$

Suponiendo que la función f es real valuada, los puntos críticos son los puntos $[x_1, +-\sqrt{-x_1}]$ para $x_1 \le 0$, luego al evaluar f en dichos puntos $\mathbf{x}$ se tiene que $f(\mathbf{x}) = 0$, como $f(\mathbf{x}) \ge 0$ para todo $\mathbf{x} \in \mathbb{R}^2$ entonces los puntos $[x_1, +-\sqrt{-x_1}]$ para $x_1 \le 0$ son los minimzadores de f.

Apéndice D
Problema 2

Problema 2. Find all values of the parameter a such that $[1, 0]^t$ is the minimizer or the maximizer of the function

$$\begin{aligned} f(x_1, x_2) &= a^3 x_1 e^{x_2} + 2a^2 log(x_1 + x_2) \\ &- (a+2)x_1 + 8ax_2 + 16x_1x_2 \end{aligned}$$

Solución Vamos a calcular primero todas las derivadas parciales de la función f. Se tiene entonces lo siguiente

$$\frac{\partial f(x_1,x_2)}{\partial x_1} = a^3 e^{x_2} + \frac{2a^2}{x_1 + x_2} - (a+2) + 16x_2$$
$$\frac{\partial f(x_1,x_2)}{\partial x_2} = a^3 x_1 e^{x_2} + \frac{2a^2}{x_1 + x_2} + 8a + 16x_1$$

Como se busca que el punto $[1, 0]$ sea un minimizador entonces $[1, 0]$ debe de ser punto crítico de f por lo que debe de cumplir las siguientes ecuaciones

$$a^3 + 2a^2 - a - 2 = 0$$
$$a^3 + 2a^2 + 8a + 16 = 0$$

Por lo que para $a = -2$ se tiene que $[1, 0]$ es punto crítico de f. Por lo tanto para $a = -2$ se tiene que el punto $[1, 0]$ es minimizador de la función f.

Apéndice E
Problema 3

Problema 3. Let $f : \mathbb{R}^n \to \mathbb{R}$ be given by $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^t\mathbf{Q}\mathbf{x} - \mathbf{b}^t\mathbf{x}$ with $\mathbf{b} \in \mathbb{R}^n$ and $\mathbf{Q} \in \mathbb{R}^{n\times n}$ a real symmetric and positive definite matrix. Consider the algorithm

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \beta\alpha_k\mathbf{g}_k$$

where $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$ and $\alpha_k = \frac{\mathbf{g}_k^t\mathbf{g}_k}{\mathbf{g}_k^t\mathbf{Q}\mathbf{g}_k}$. Show that $\{\mathbf{x}_k\}$ converges to $\mathbf{x}^* = \mathbf{Q}^{-1}\mathbf{b}$ for any initial point $\mathbf{x}^0$ if and only if $0 < \beta < 2$.

Demostración $\Rightarrow$ Supongamos primero que la secuencia $\{\mathbf{x}_k\}$ converge a $\mathbf{Q}^{-1}\mathbf{b}$ para cualquier punto inicial $\mathbf{x}^0$. Como $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^t\mathbf{Q}\mathbf{x} - \mathbf{b}^t\mathbf{x}$ entonces se cumple que $\mathbf{g}_k = \mathbf{Q}\mathbf{x}_k - \mathbf{b}$. Consideremos γ_k definido de la siguiente forma

$$\gamma_k = \beta\alpha_k \frac{\mathbf{g}_k^t\mathbf{g}_k}{\mathbf{g}_k^t\mathbf{Q}^{-1}\mathbf{g}_k}\left(2\frac{\mathbf{g}_k^t\mathbf{g}_k}{\mathbf{g}_k^t\mathbf{Q}\mathbf{g}_k} - \beta\alpha_k\right)$$

Como la sucesión $\{\mathbf{x}_k\}$ converge entonces γ_k debe de cumplir que $0 < \gamma_k \le 1$. Luego, por hipótesis del problema se tiene que $\alpha_k = \frac{\mathbf{g}_k^t\mathbf{g}_k}{\mathbf{g}_k^t\mathbf{Q}\mathbf{g}_k}$ por lo cual

$$\gamma_k = \beta(2-\beta)\frac{(\mathbf{g}_k^t\mathbf{g}_k)^2}{\mathbf{g}_k^t\mathbf{Q}\mathbf{g}_k\mathbf{g}_k^t\mathbf{Q}^{-1}\mathbf{g}_k}$$

Sea a el eigenvalor mas pequeño de Q y A el eigenvalor más grande de Q, entonces del Lemma 2.4 se tiene lo siguiente.

$$0 < \frac{a}{A} \le \frac{(\mathbf{g}_k^t\mathbf{g}_k)^2}{\mathbf{g}_k^t\mathbf{Q}\mathbf{g}_k\mathbf{g}_k^t\mathbf{Q}^{-1}\mathbf{g}_k} \le \frac{A}{a} \le 1$$

Luego como $0 < \gamma_k \le 1$ entonces se debe de cumplir que $0 < \beta(2-\beta) \le 1$ lo cual se cumple solamente si $0 < \beta < 2$

$\Leftarrow$ Supongamos ahora que $0 < \beta < 2$ entonces consideremos γ_k como se definió anteriormente. Como $0 < \beta < 2$ entonces se tiene que $0 < \gamma_k \le 1$ por lo que si consideramos

$$\sum_i^\infty \gamma_k$$

Se tiene que

$$\sum_{i=1}^\infty \gamma_k = \infty$$

Ya que se suman términos positivos mayores a 0

Lo que de acuerdo al Teorema 2.3 nos dice que la sucesión $\{\mathbf{x}_k\}$ converge a $\mathbf{x}^*$ para todo punto inicial $\mathbf{x}_0$. Probaremos ahora que $\mathbf{x}^* = \mathbf{Q}^{-1}\mathbf{b}$. Como $\mathbf{g}_k = \mathbf{Q}\mathbf{x}_k - \mathbf{b}$ entonces se tiene que

$$\begin{aligned} \lim_{k\to\infty}(\mathbf{x}_k - \mathbf{x}^*) = 0 \\ &= \mathbf{Q}^{-1}(\mathbf{Q}\mathbf{x}^* - b) \\ &= \mathbf{x}^* - \mathbf{Q}^{-1}\mathbf{b} \end{aligned}$$

Por lo tanto $\mathbf{x}^* = \mathbf{Q}^{-1}\mathbf{b}$ que es lo que se buscaba probar
■

REFERENCIAS

[1] Nocedal Jorge, *Numerical Optimization*, Segunda edición, Springer, 2006