



Universidad Nacional Autónoma de México

Escuela Nacional de Estudios Superiores

Unidad Juriquilla

Computación III

Tarea 1

Método de Newton para Optimización en Dos Dimensiones

Bruno Arturo López Pacheco

Dr. Ulises Olivares Pinto

No. De Cuenta 317347140

Licenciatura en Tecnología

Objetivo

Utilizar el método de optimización en dos dimensiones de Newton y compararlo en eficiencia con el método de optimización en dos dimensiones por búsqueda ingenua.

Marco teórico

Método de optimización en dos dimensiones por búsqueda ingenua

Dada una función $f(x)$, podemos tomar un rango lo suficientemente grande de búsqueda en donde se puede evaluar un valor de x , de la cual su evaluación sea el punto mayor o menor de dicha función.

El método plantea un rango de búsqueda de $[-10000, 10000]$ dando pasos de 0.1. Para así tener 200,000 evaluaciones (puntos de búsqueda) y compararlas para conocer en dónde se encuentra el punto mayor o menor de la función $f(x)$.

Método de optimización en dos dimensiones de Newton

La optimización en dos dimensiones se basa en el criterio de la primera derivada. Se utiliza principalmente para encontrar los mínimos y máximos relativos que se pueden encontrar en una función, puntos importantes para la optimización. El teorema del valor máximo y mínimo dice:

“Sea c un punto crítico de una función f que es continua en un intervalo abierto I que contiene a c . Si f es derivable en el intervalo, excepto posiblemente en c , entonces $f(c)$ puede clasificarse como sigue:

- 1.- Si $f'(c) > 0$ en algún intervalo a la izquierda de c y $f'(c) < 0$ en algún intervalo a la derecha de c entonces f tiene un máximo relativo en $(c, f(c))$.
- 2.- Si $f'(c) < 0$ en algún intervalo a la izquierda de c y $f'(c) > 0$ en algún intervalo a la derecha de c entonces f tiene un mínimo relativo en $(c, f(c))$.
- 3.- Si $f'(c) > 0$ en ambos lados de c o $f'(c) < 0$ en ambos lados de c entonces $f(c)$ no es ni un mínimo ni un máximo relativo.”

Metodología

Implementación

El pseudocódigo para implementar ambos métodos se muestra a continuación.

```
def minmax(func):
    X=arreglo(-10000,10000.1,0.1)#rango a, rango b y paso
    print("Rango de búsqueda: [-10000,10000]")
    maximo=func(X[0])
    minimo=func(X[0])
    indiceMaximo=X[0]
    indiceMinimo=X[0]

    Para i en rango(0,tamaño(X)):
        Si func(X[i])>maximo:
            maximo=func(X[i])
```

```

        indiceMaximo=X[i]
    Si func(X[i])<minimo:
        minimo=func(X[i])
        indiceMinimo=X[i]

    Regresar indiceMaximo,maximo,indiceMinimo,minimo

```

Para la implementación del método de optimización de Newton, se realizaron dos funciones, una de búsqueda de raíces de la derivada, y otro que analiza si las raíces encontradas son puntos máximos o mínimos de la función original.

```

def newton(x2, f, tolerancia, maximaIteracion):
    fp = derivada(f)
    iteracion=1
    errorAbsoluto=np.inf

    mientras (errorAbsoluto>=tolerancia o iteracion==maximaIteracion):
        x1=x2
        Si (fp(x1) ==0):
            "Error matemático"
            break
        x2=x1- (f(x1)/fp(x1))
        errorAbsoluto=absoluto(x2-x1)
        iteracion+=1

    Si (iteracion>maxIter):
        "No converge a ninguna solución"
    SiNo:
        Imprimir ("Solución:", x2)
    Regresar x2

```

```

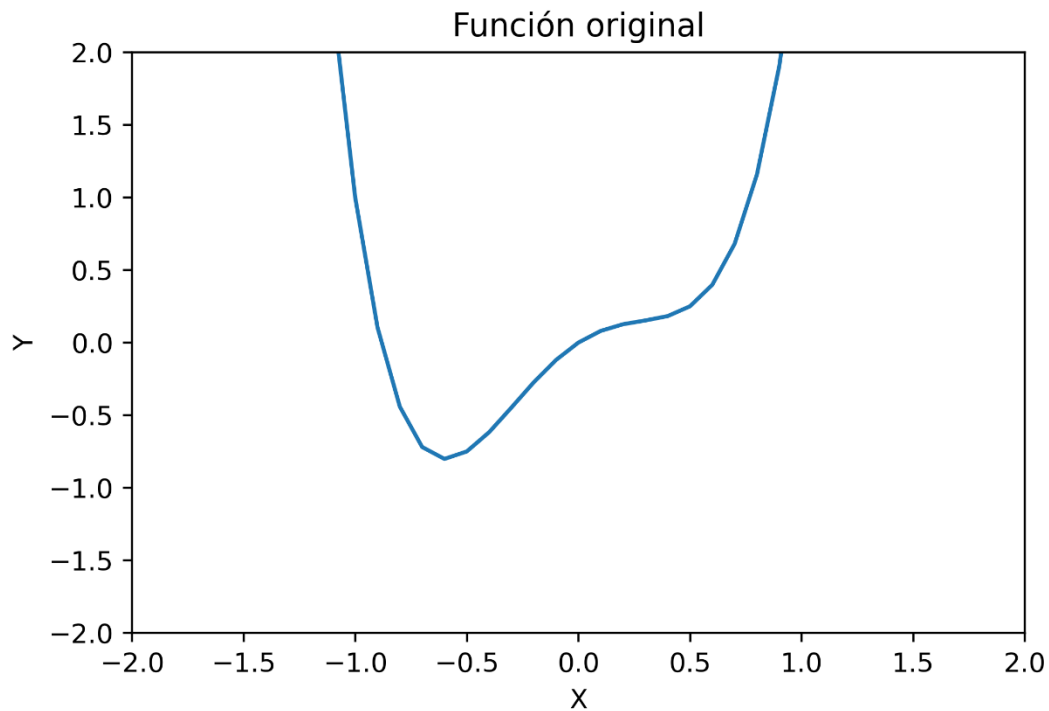
def checkRoot(f, fp, raiz):
    #Criterio de la primer derivada
    Si (fp(raiz-0.1)>0 y fp(raiz+0.1)<0):
        Imprimir("La raíz [raiz] de la derivada es un máximo")
        Imprimir("El punto máximo de la función se encuentra en [raiz,f(raiz
))]
    Si (fp(raiz-0.1)<0 y fp(raiz+0.1)>0):
        Imprimir("La raíz [raiz] de la derivada es un mínimo")
        Imprimir("El punto mínimo de la función se encuentra en [raiz,f(raiz
))]

```

Pruebas

Para probar ambos métodos y compararlos se utilizó la función $f(x) = 4x^4 - 2x^2 + x$. Como se explicó, se utilizó un rango de búsqueda $[-10000, 10000]$. Se buscaba encontrar un punto crítico en $x = -0.5957$, dándonos cuenta de que se trata de una raíz irracional.

La graficación de la función es la siguiente.

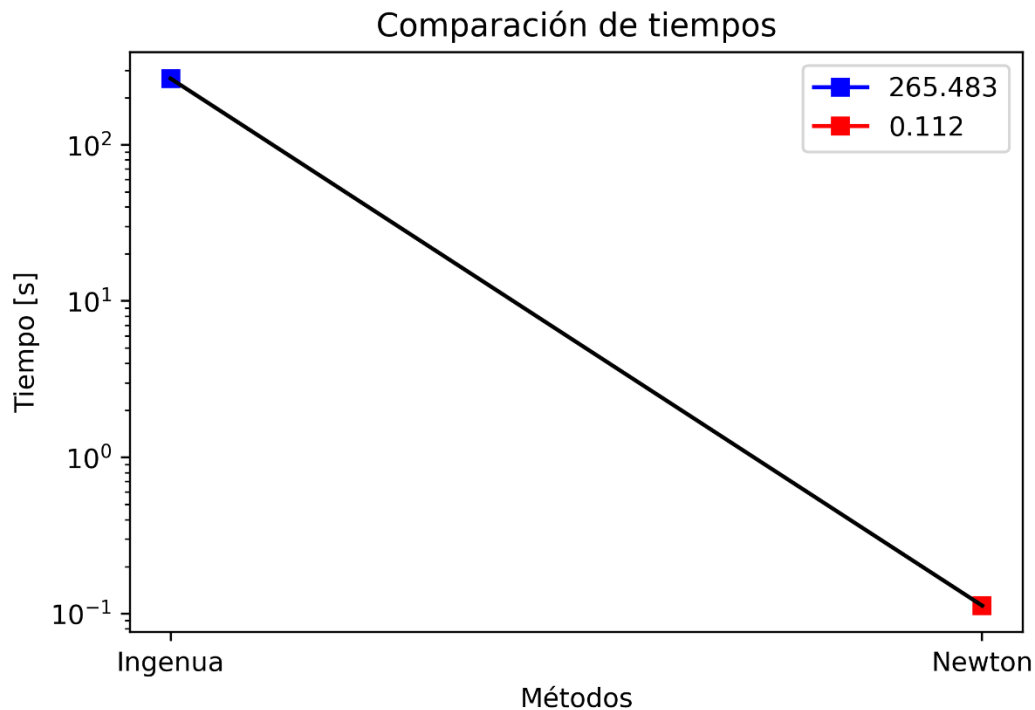


Resultados

Para el caso de la búsqueda ingenua, se pudo encontrar el punto máximo y mínimo de la función. El punto máximo (que no es un punto crítico de la función) se encuentra en $[10000, 4 \times 10^{16}]$. Y el punto mínimo (que es un punto crítico de la función) se encontró en $[-0.599, -0.801]$. Y su tiempo de ejecución fue de 278.84 segundos (4 minutos y medio).

Para el caso del método de Newton para optimización, se pudo encontrar un punto crítico en la función. Este punto crítico se clasificó como punto mínimo de la función, y se encontró en $[-0.595, -0.8017]$. Su tiempo de ejecución fue de 0.1129 segundos.

La comparación de tiempos se muestra a continuación.



Conclusión

El método de Newton para optimización se muestra efectivo para encontrar puntos críticos de la función, esto quiere decir que aunque se tengan puntos máximos o mínimos en la función, si este punto no es un punto crítico, será complicado encontrarlo. Como es el caso de esta función de ejemplo, en donde solo se tiene un punto crítico, y corresponde a un mínimo. Pero para este caso (con la implementación actual) no se pudo encontrar un punto máximo.

Aún teniendo esta diferencia, se pudo acortar demasiado el tiempo de ejecución, dando cuenta que el método de Newton tiene una convergencia muy rápida, y muestra ser efectiva cuando se conozca que la función tiene puntos críticos.

Referencias

Chapra, S., & Canale, R. (2011). METODOS NUMERICOS PARA INGENIEROS (Spanish Edition) (6th ed.). McGraw-Hill Interamericana de España S.L.