Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

Docente: Ing. Fred Torres Cruz
Estudiante: Luis Angel Quenaya Loza
Código: 241411

Actividad N°06

Método de la Secante

Introduccion

El **método de la secante** es un procedimiento iterativo para aproximar las raíces de una función continua $f(x)$. A diferencia de Newton-Raphson, no necesita derivadas, ya que la pendiente se estima usando dos puntos cercanos a la raíz.

Se basa en la idea de reemplazar la tangente por una *secante*, la línea que une los puntos $(x_{n-1}, f(x_{n-1}))$ y $(x_n, f(x_n))$. La intersección con el eje x nos da la siguiente aproximación:

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

Importancia de graficar la función

Antes de iniciar, es recomendable graficar $f(x)$ para ubicar aproximadamente dónde cruza el eje x y elegir dos valores iniciales x_0 y x_1 cercanos a la raíz, aumentando la probabilidad de convergencia.

Ventajas y limitaciones

- No requiere derivadas analíticas.
- Puede ser más rápida que la bisección.
- Convergencia superlineal si los puntos iniciales son adecuados.
- Puede divergir si los puntos iniciales no se eligen bien.
- No garantiza convergencia si la función no es continua.

Codigo en Python

```
import numpy as np
import matplotlib.pyplot as plt

func_str = input("Ingrese la funci n f(x): ")

def f(x):
    return eval(func_str, {"np": np, "x": x})

xmin = float(input("Ingrese el valor m nimo de x: "))
xmax = float(input("Ingrese el valor m ximo de x: "))

x = np.linspace(xmin, xmax, 400)
y = f(x)

plt.plot(x, y, color='blue', label=f"f(x)={func_str}")
plt.axhline(0, color='black', linestyle='--')
plt.axvline(0, color='black', linestyle='--')
plt.title("Gr fico de la funci n")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.legend()
plt.grid(True)
plt.show()

x0 = float(input("Ingrese x0: "))
x1 = float(input("Ingrese x1: "))

tol = 1e-6
max_iter = 100

print("Iter | x0         | x1         | f(x0)       | f(x1)       | x2         | Error")
print("-------------------------------------------------------------------------")

for i in range(1, max_iter+1):
    f0 = f(x0)
    f1 = f(x1)
    if f1 - f0 == 0:
        print(f"Divisi n por cero en iteraci n {i}")
        break
    x2 = x1 - f1 * (x1 - x0) / (f1 - f0)
    error = abs(x2 - x1)
    print(f"{i:4d} | {x0:10.6f} | {x1:10.6f} | {f0:10.6f} | {f1:10.6f} | {x2:10.6f} | {error:10.6f}")
```

```python
    if error < tol:
        print(f"Ra z aproximada: {x2:.6f} en {i} iteraciones")
        break
    x0, x1 = x1, x2
else:
    print("No se alcanz  convergencia")
```

```
= RESTART: C:/Users/LENOVO/AppData/Local/Programs/Python/Python312/metodo secante.py
Ingrese la función f(x): x**3-x-4
Ingrese el valor mínimo de x: -3
Ingrese el valor máximo de x: 3
Ingrese x0: 1
Ingrese x1: 4
Iter | x0        | x1        | f(x0)      | f(x1)      | x2        | Error
--------------------------------------------------------------------------
   1 |  1.000000 |  4.000000 |  -4.000000 | 56.000000 |  1.200000 |  2.800000
   2 |  4.000000 |  1.200000 |  56.000000 | -3.472000 |  1.363465 |  0.163465
   3 |  1.200000 |  1.363465 |  -3.472000 | -2.828733 |  2.082294 |  0.718829
   4 |  1.363465 |  2.082294 |  -2.828733 |  2.946427 |  1.715555 |  0.366739
   5 |  2.082294 |  1.715555 |   2.946427 | -0.666455 |  1.783206 |  0.067651
   6 |  1.715555 |  1.783206 |  -0.666455 | -0.112925 |  1.797007 |  0.013801
   7 |  1.783206 |  1.797007 |  -0.112925 |  0.005953 |  1.796316 |  0.000691
   8 |  1.797007 |  1.796316 |   0.005953 | -0.000049 |  1.796322 |  0.000006
   9 |  1.796316 |  1.796322 |  -0.000049 | -0.000000 |  1.796322 |  0.000000
Raíz aproximada: 1.796322 en 9 iteraciones
```

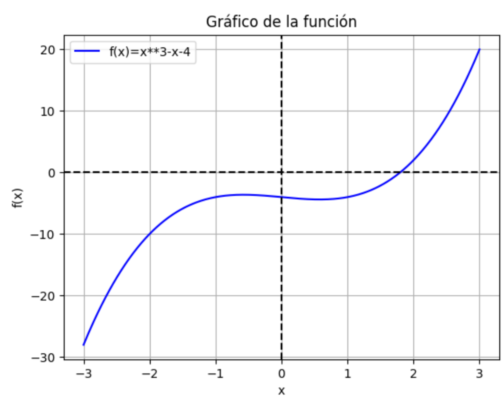

Figura 1: Compilador y Grafica del metodo secante.