

UNIVERSIDAD NACIONAL DEL ALTIPLANO

FACULTAD DE INGENIERÍA ESTADÍSTICA E
INFORMÁTICA

Método de la Secante

Estudiante: Nexu Yohan Mamani Yucra

Docente: Fred Cruz Torres

1. Introducción

En el campo de la ingeniería estadística, encontrar las raíces de ecuaciones no lineales es fundamental para la estimación de parámetros y la optimización de modelos. El **Método de la Secante** surge como una alternativa eficiente al método de Newton-Raphson, eliminando la necesidad de calcular derivadas complejas.

2. Marco Teórico

2.1. Definición

El método de la secante es un algoritmo de aproximación numérica que utiliza la pendiente de una línea que pasa por dos puntos de la función para proyectar la siguiente aproximación de la raíz.

A diferencia del método de bisección, no requiere que la raíz esté .^{en}cerrada.^{entre} los dos puntos iniciales, aunque esto ayuda a la convergencia.

2.2. Deducción Matemática

Partiendo de la aproximación de la derivada por diferencias finitas:

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \quad (1)$$

Sustituyendo en la fórmula de Newton-Raphson, obtenemos la relación recursiva del método de la secante:

$$x_{n+1} = x_n - f(x_n) \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \quad (2)$$

3. Algoritmo y Código en Python

La implementación en Python se ha realizado buscando eficiencia y claridad, utilizando un bucle for para controlar el número máximo de iteraciones y evitar bucles infinitos.

```

1 import math
2
3 def secante(f, x0, x1, tol, max_iter):
4     print(f"{'Iter':<5} | {'x_n':<10} | {'f(x_n)':<10} | {'Error':<10}")
5     print("-" * 45)
6
7     for i in range(max_iter):
8         fx0 = f(x0)
9         fx1 = f(x1)
10
11         if fx1 - fx0 == 0:
12             return None, "Error: División por cero"
13
14         # Formula de la secante
15         x_sig = x1 - fx1 * (x1 - x0) / (fx1 - fx0)
16         error = abs(x_sig - x1)
17
18         print(f"{'i+1':<5} | {'x1':<10.6f} | {'fx1':<10.6f} | {'error':<10.6e}")
19
20         if error < tol:
21             return x_sig, i+1
22
23         x0 = x1
24         x1 = x_sig
25
26     return x1, max_iter
27
28 # Ejemplo: f(x) = x^2 - 2
29 func = lambda x: x**2 - 2
30 raiz, it = secante(func, 1.0, 2.0, 1e-5, 20)
31 print(f"\nRaiz aproximada: {raiz} encontrada en {it} iteraciones.")

```

Listing 1: Script del Método de la Secante en Python

4. Conclusiones

1. El método de la secante es computacionalmente más económico que Newton-Raphson, ya que solo requiere una evaluación de la función por iteración tras los dos puntos iniciales.