MÉTODO DE LA SECANTE

Reporte Académico

Autor: Job Edward Apaza Curtihuanca

1. Introducción

El método de la secante es un método numérico iterativo utilizado para aproximar raíces de ecuaciones no lineales. A diferencia del método de Newton-Raphson, no requiere el cálculo de derivadas, lo que lo convierte en una alternativa eficiente cuando la derivada de la función es difícil o costosa de obtener.

Este método utiliza una aproximación lineal basada en dos puntos iniciales para construir una recta secante cuya intersección con el eje x proporciona una mejor aproximación de la raíz.

2. Objetivos

2.1. Objetivo general

Aplicar el método de la secante para aproximar raíces de ecuaciones no lineales mediante un proceso iterativo eficiente.

2.2. Objetivos específicos

- Analizar el fundamento matemático del método de la secante.
- Describir el algoritmo iterativo.
- Evaluar la convergencia del método.
- Resolver un problema práctico.

3. Fundamentación teórica

Sea una función $f(x)$ continua. Dados dos valores iniciales x_0 y x_1 cercanos a la raíz, el método de la secante aproxima la derivada mediante una diferencia finita:

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

Sustituyendo esta aproximación en el método de Newton-Raphson se obtiene la fórmula iterativa del método de la secante.

4. Fórmula iterativa

La expresión general del método de la secante está dada por:

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

Este proceso se repite hasta que se alcance la tolerancia deseada.

5. Algoritmo del método

1. Elegir dos valores iniciales x_0 y x_1.
2. Calcular $f(x_0)$ y $f(x_1)$.
3. Aplicar la fórmula iterativa.
4. Verificar el criterio de parada.
5. Repetir el proceso.

6. Criterios de parada

El método se detiene cuando se cumple al menos uno de los siguientes criterios:

- $|x_{n+1} - x_n| < \varepsilon$
- $|f(x_{n+1})| < \varepsilon$
- Se alcanza el número máximo de iteraciones.

7. Convergencia

El método de la secante presenta una convergencia superlineal, con un orden aproximado de:

$$p \approx 1{,}618 \tag{3}$$

Este valor corresponde al número áureo, lo que indica una convergencia más rápida que el método de la bisección, pero menor que el método de Newton-Raphson.

8. Ventajas y desventajas

8.1. Ventajas

- No requiere el cálculo de derivadas.
- Convergencia rápida.
- Fácil de implementar.

8.2. Desventajas

- No garantiza convergencia.
- Sensible a los valores iniciales.
- Puede fallar si $f(x_n) = f(x_{n-1})$.

9. Ejemplo de aplicación

Sea la función:

$$f(x) = x^3 - x - 2 \tag{4}$$

Con valores iniciales $x_0 = 1$ y $x_1 = 2$, se obtiene la siguiente aproximación iterativa:

Iteración	x_{n-1}	x_n	x_{n+1}
1	1.0	2.0	1.333
2	2.0	1.333	1.462
3	1.333	1.462	1.521

La raíz aproximada es:

$$x \approx 1{,}521$$

10. Implementación computacional

A continuación se deja el espacio reservado para la implementación del método de la secante en un lenguaje de programación (por ejemplo, Python):

```
# M todo de la Secante
def secante(f, x0, x1, tol=1e-6, max_iter=100):
    """
    Encuentra la ra z usando el m todo de la secante
```

```
    f: funci n
    x0, x1: dos valores iniciales
    tol: tolerancia
    max_iter: n mero m ximo de iteraciones
    """
    print("\n          M TODO DE LA SECANTE        ")
    print(f"{'Iter':<6} {'x_n-1':<15} {'x_n':<15} {'x_n+1':<15} {'f(x_n+1)':<15} {'Error':<15}")
    print("-" * 95)

    for i in range(max_iter):
        fx0 = f(x0)
        fx1 = f(x1)


        if abs(fx1 - fx0) < 1e-12:
            print("\nError: Divisi n por cero")
            return x1


        x2 = x1 - fx1 * (x1 - x0) / (fx1 - fx0)
        fx2 = f(x2)
        error = abs(x2 - x1)

        print(f"{i+1:<6} {x0:<15.8f} {x1:<15.8f} {x2:<15.8f} {fx2:<15.8e} {error:<15.8e}")

        if error < tol or abs(fx2) < tol:
            print(f"\nConvergi  en {i+1} iteraciones")
            print(f"Ra z aproximada: x = {x2:.8f}")
            return x2


        x0 = x1
        x1 = x2

    print(f"\nNo convergi  en {max_iter} iteraciones")
    return x2

# Encontrar ra z de x^2 - 2 = 0
def f(x):
```

```
    return x**2 - 2

# Ejecutar
raiz = secante(f, 1, 2)
```

11. Aplicaciones

El método de la secante es ampliamente utilizado en ingeniería, física, economía y ciencias computacionales para la resolución de ecuaciones no lineales.

12. Conclusiones

El método de la secante constituye una herramienta eficiente para la aproximación de raíces de ecuaciones no lineales. Su principal ventaja radica en que no requiere el cálculo de derivadas y presenta una convergencia superior al método de la bisección, aunque su éxito depende en gran medida de la elección de los valores iniciales.