Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

Docente:

Ing. Torres Cruz Fred

Alumno:

Ticona Miramira Roberto Angel

Trabajo 5 - Método de Newton-Raphson

» Descripción

El método de Newton-Raphson es un procedimiento iterativo utilizado para encontrar aproximaciones sucesivas de las raíces reales de una función continua y derivable. Su fundamento radica en el desarrollo del polinomio de Taylor de primer orden alrededor de un punto inicial x_0, lo cual permite estimar el punto donde la función corta el eje x mediante la siguiente fórmula recursiva:

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

En cada iteración, el método utiliza la pendiente de la tangente en el punto x_n para aproximarse progresivamente hacia la raíz. Si la función es suficientemente suave y el punto inicial se elige adecuadamente, la convergencia suele ser rápida y cuadrática.

Importancia de la visualización previa

Antes de aplicar el método, es recomendable graficar la función $f(x)$ en un intervalo apropiado. Esta visualización permite identificar los puntos donde la función cruza el eje x, estimar posibles raíces reales y seleccionar un valor inicial x_0 cercano a ellas. De esta forma se incrementan las probabilidades de convergencia del método y se evita iniciar en zonas problemáticas donde la pendiente $f'(x)$ sea nula o la función presente discontinuidades.

Restricciones y limitaciones

El método de Newton-Raphson presenta ciertas limitaciones importantes:

- No debe aplicarse en puntos donde $f'(x) = 0$, ya que produce una indeterminación en la fórmula.
- Su convergencia no está garantizada si el punto inicial x_0 se encuentra lejos de la raíz.
- En el caso de raíces múltiples, la convergencia puede ser muy lenta o incluso inestable.
- Se requiere que tanto $f(x)$ como su derivada $f'(x)$ sean continuas en el entorno de la raíz buscada.

» Aplicación del método en Python

El objetivo fue desarrollar en el lenguaje de programación Python un programa que permita al usuario ingresar una función $f(x)$ y graficarla en un intervalo definido. Esta etapa inicial tiene como propósito ayudar al usuario a identificar visualmente las posibles raíces y decidir si desea aplicar el método de Newton-Raphson.

En caso afirmativo, el programa solicita un valor inicial x_1 basado en la observación de la gráfica y ejecuta el algoritmo iterativo de Newton-Raphson hasta que la diferencia entre dos aproximaciones sucesivas sea menor a una tolerancia predefinida. Finalmente, el programa muestra en pantalla la raíz aproximada encontrada y el número de iteraciones necesarias para alcanzarla.

Este enfoque combina la interpretación gráfica con el análisis numérico, promoviendo una comprensión más completa del comportamiento de la función y de la eficacia del método iterativo.

» Entrada

Una cadena de texto que representa una función matemática.

$$f(x) = x^3 - x - 1$$

» Salida

- Gráfica para evaluar si realizar o no el método.
- La raíz encontrada.
- Número de iteraciones realizadas hasta encontrar la raíz.

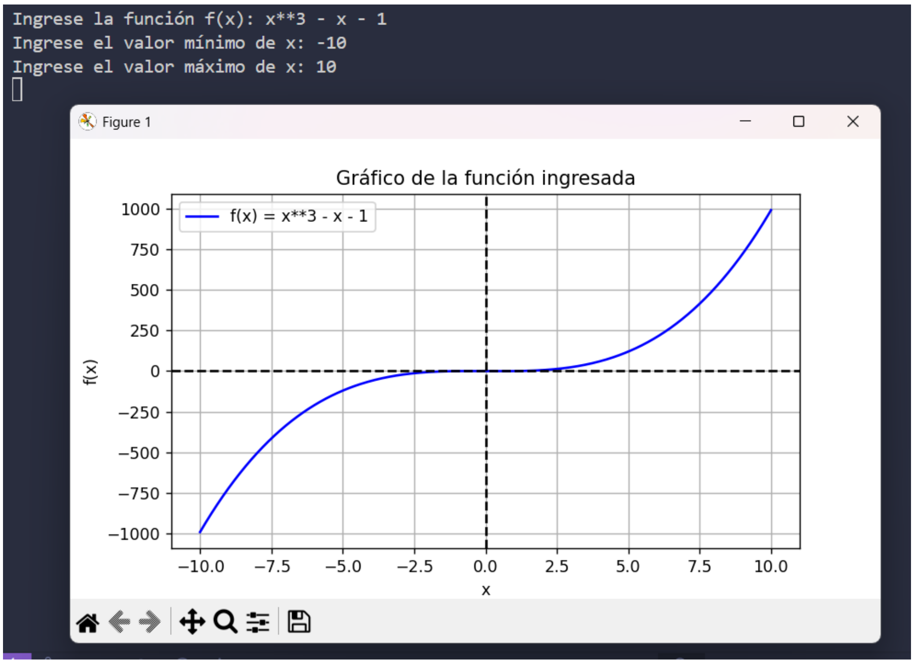
» Restricciones

Código

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

# === 1. Ingreso de la función ===
func_str = input("Ingrese la función f(x): ")  # Ejemplo: x**3 - x - 1

# Definimos la función y su derivada (usando derivada numérica)
def f(x):
    return eval(func_str)

def f_prime(x, h=1e-6):  # derivada numérica
    return (f(x + h) - f(x - h)) / (2 * h)

# === 2. Graficar la función ===
xmin = float(input("Ingrese el valor mínimo de x: "))
xmax = float(input("Ingrese el valor máximo de x: "))

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

plt.figure(figsize=(8, 5))
plt.plot(x, y, label=f"f(x) = {func_str}", color='blue')
plt.axhline(0, color='black', linestyle='--')
plt.axvline(0, color='black', linestyle='--')
plt.title("Gráfico de la función ingresada")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.legend()
plt.grid(True)
plt.show()

# === 3. Pregunta si desea aplicar Newton-Raphson ===
op = input("¿Desea encontrar una raíz con el método de Newton-Raphson? (s/n):
    ↪ ").lower()

```

```python
if op == "s":
    # === 4. Ingreso de punto inicial ===
    x0 = float(input("Basado en la gráfica, ingrese el valor inicial x1: "))

    # Parámetros del método
    tol = 1e-6
    max_iter = 100

    # Iteraciones
    for i in range(1, max_iter + 1):
        fx = f(x0)
        fpx = f_prime(x0)

        if fpx == 0:
            print(f"La derivada es cero en x = {x0}. El método no puede continuar.")
            break

        x1 = x0 - fx / fpx

        # Verificar convergencia
        if abs(x1 - x0) < tol:
            print(f"\n Raíz aproximada encontrada: {x1:.6f}")
            print(f"Iteraciones realizadas: {i}")
            break

        x0 = x1
    else:
        print("\n No se alcanzó la convergencia después de", max_iter, "iteraciones.")

else:
    print("No se aplicó el método de Newton-Raphson.")
```

Ejecución



```
Ingrese la función f(x): x**3 - x - 1
Ingrese el valor mínimo de x: -10
Ingrese el valor máximo de x: 10
¿Desea encontrar una raíz con el método de Newton-Raphson? (s/n): s
Basado en la gráfica, ingrese el valor inicial x1: 1

✅ Raíz aproximada encontrada: 1.324718
Iteraciones realizadas: 5
```