Método de Regula Falsi (Regla Falsa)

Autor: Henry Ccoarite Dueñas
Curso: Programación Numérica
Docente: Fred Torres Cruz
Universidad Nacional del Altiplano - Puno

1. Descripción del método

El **método de Regula Falsi** o **Regla Falsa** es un método numérico iterativo utilizado para encontrar las raíces reales de una función continua $f(x)$ dentro de un intervalo $[a, b]$ donde se cumple que $f(a)$ y $f(b)$ tienen signos opuestos, es decir:

$$f(a) \cdot f(b) < 0$$

La idea del método es similar a la del método de bisección, pero en lugar de tomar el punto medio del intervalo, se traza una **recta secante** entre $(a, f(a))$ y $(b, f(b))$, y se toma su punto de intersección con el eje x como nueva aproximación de la raíz.

Fórmula general

$$c = a - f(a) \cdot \frac{b - a}{f(b) - f(a)}$$

Luego se evalúa $f(c)$:

- Si $f(a) \cdot f(c) < 0$, la raíz está entre a y $c \Rightarrow$ se actualiza $b = c$.
- Si $f(b) \cdot f(c) < 0$, la raíz está entre c y $b \Rightarrow$ se actualiza $a = c$.

El proceso se repite hasta que el error sea menor que una tolerancia o se alcance el número máximo de iteraciones.

Ventajas

- Garantiza la convergencia si $f(a)$ y $f(b)$ tienen signos opuestos.
- Tiene una convergencia más rápida que el método de bisección.

Desventajas

- Puede converger lentamente si la función es muy curvada.
- Requiere que la función cambie de signo en el intervalo inicial.

2. Código implementado en Python

Listing 1: Implementación del método de Regula Falsi en Python

```
import math

def regula_falsi(f, a, b, tol=1e-6, max_iter=100):

    if f(a) * f(b) > 0:
        raise ValueError("f(a) y f(b) deben tener signos opuestos
           ")

    print(f"{'Iter':<6} {'a':<18} {'b':<18} {'c':<18} {'f(c)
       ':<18} {'Error':<18}")
    print("-" * 96)

    c_ant = a

    for i in range(max_iter):
        fa = f(a)
        fb = f(b)

        c = a - fa * (b - a) / (fb - fa)
        fc = f(c)

        if i > 0:
            error = abs(c - c_ant)
        else:
            error = abs(b - a)

        print(f"{i+1:<6} {a:<18.10f} {b:<18.10f} {c:<18.10f} {fc
           :<18.10e} {error:<18.10e}")

        if abs(fc) < tol or error < tol:
            print("-" * 96)
            print(f"Ra z encontrada: {c:.12f}")
            print(f"Iteraciones: {i+1}")
            return c, i+1

        if fa * fc < 0:
            b = c
        else:
            a = c

        c_ant = c

    print(f"\nSe alcanz  el m ximo de iteraciones")
    print(f" ltima  aproximaci n: {c:.12f}")
    return c, max_iter


if __name__ == "__main__":
```

```
46
47      print("M todo de Regula Falsi (Regla Falsa)")
48      print("="*96)
49      print("Ejemplo: f(x) = x   - 4 (Intervalo [1, 3])")
50
51      def funcion(x):
52          return x**2 - 4
53
54      try:
55          raiz, iteraciones = regula_falsi(funcion, 1, 3, tol=1e-6)
56          print(f"\nVerificaci n: f({raiz:.12f}) = {funcion(raiz)
               :.12e}")
57      except ValueError as e:
58          print(f"Error: {e}")
59      except Exception as e:
60          print(f"Error de ejecuci n: {e}")
```

3. Salida del programa (Output)

```
Método de Regula Falsi (Regla Falsa)
================================================================================================
Ejemplo: f(x) = x² - 4 (Intervalo [1, 3])

Iter   a              b              c              f(c)           Error
------------------------------------------------------------------------------------------------
1      1.0000000000   3.0000000000   2.3333333333   1.4444444444e+00  2.0000000000e+00
2      1.0000000000   2.3333333333   2.0666666667   4.4444444444e-01  2.6666666667e-01
3      1.0000000000   2.0666666667   2.0179487179   7.2242900723e-02  4.8717948718e-02
4      1.0000000000   2.0179487179   2.0027223230   1.0889289335e-02  1.5226394852e-02
5      1.0000000000   2.0027223230   2.0004156735   1.6626979842e-03  2.3066495591e-03
6      1.0000000000   2.0004156735   2.0000636440   2.5457598606e-04  3.5202952389e-04
7      1.0000000000   2.0000636440   2.0000097654   3.9087001264e-05  5.3878554382e-05
------------------------------------------------------------------------------------------------
Raíz encontrada: 2.000009765362
Iteraciones: 7
Verificación: f(2.000009765362) = 3.906154e-05
```

Figura 1: Resultados de la ejecución del método de Regula Falsi para $f(x) = x^2 - 4$ con tol $= 10^{-6}$.

4. Descripción del funcionamiento del código

1. Se define una función `regula_falsi()` que recibe la función $f(x)$, el intervalo $[a, b]$, una tolerancia y el número máximo de iteraciones.

2. Se verifica que $f(a)$ y $f(b)$ tengan signos opuestos para asegurar la existencia de una raíz.

3. Se calcula el nuevo punto c aplicando la fórmula de Regula Falsi.

4. Se evalúa el error entre las iteraciones consecutivas y la función en el nuevo punto.
5. Si el error o $|f(c)|$ son menores que la tolerancia, el método finaliza mostrando la raíz y el número de iteraciones.
6. Finalmente, se ejecuta un ejemplo con $f(x) = x^2 - 4$ y el intervalo $[1, 3]$.

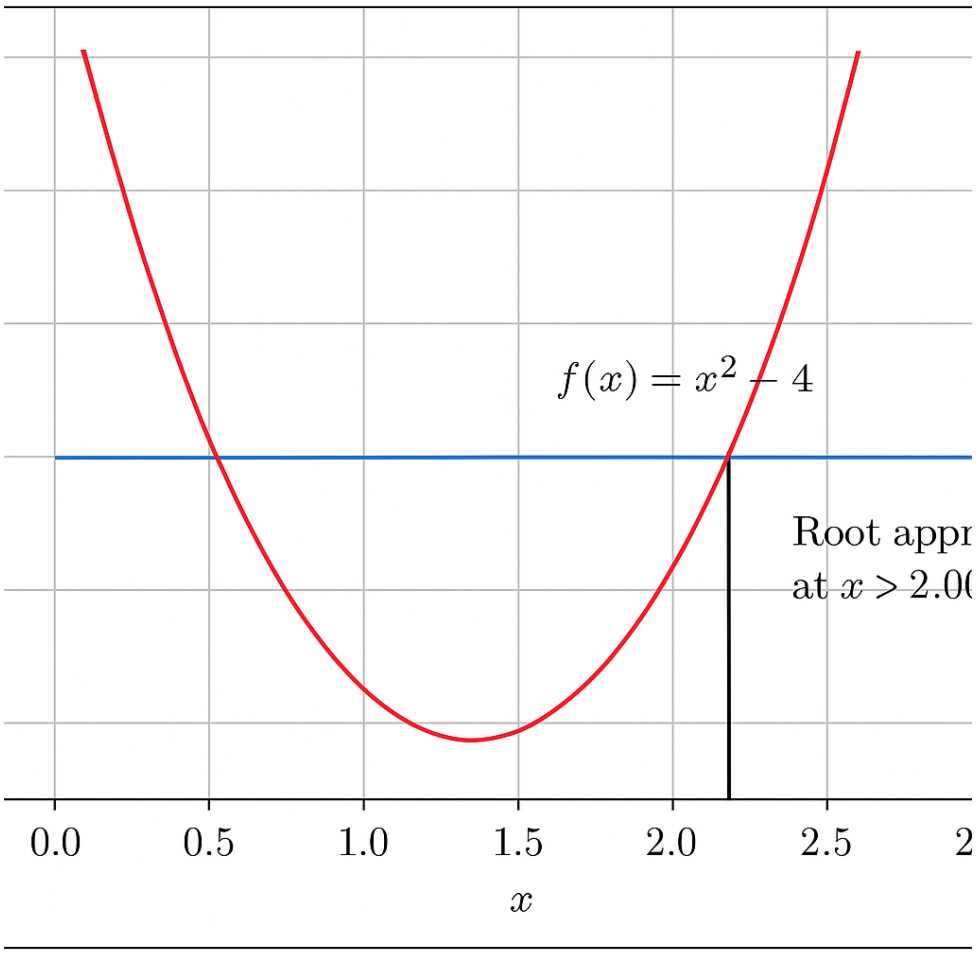

Figura 2: Gráfico de la función $f(x) = x^2 - 4$ mostrando la raíz aproximada obtenida.

5. Conclusión

El método de Regula Falsi combina las ventajas del método de bisección y de la secante, manteniendo la seguridad del primero y mejorando la velocidad de convergencia. En este ejemplo, aplicado a $f(x) = x^2 - 4$, el método encuentra con precisión la raíz $x \approx 2{,}0000$ en pocas iteraciones, demostrando su eficacia en funciones continuas que cambian de signo en un intervalo dado.