Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

E.P. de Ingeniería Estadística e Informática

Docente: Ing. Torres Cruz Fred
Presentado por: Quispe Ito Luz Leidy

Metodo de Regula Falsi

1. ¿Que es el Método de Regula Falsi?

El método de Regula Falsi combina las características del método de la bisección y del método de la secante. Se mantiene siempre un intervalo donde la función cambia de signo, pero la aproximación del nuevo punto se realiza mediante **interpolación lineal**.

Fórmula de interpolación

Dados a_n y b_n tales que:

$$f(a_n)\, f(b_n) < 0,$$

el nuevo punto c_n se calcula mediante:

$$c_n = b_n - f(b_n)\, \frac{b_n - a_n}{f(b_n) - f(a_n)} = \frac{a_n f(b_n) - b_n f(a_n)}{f(b_n) - f(a_n)}.$$

Algoritmo

1. Calcular c_n usando la fórmula de interpolación:

$$c_n = b_n - f(b_n) \frac{b_n - a_n}{f(b_n) - f(a_n)}.$$

2. Si $f(a_n)\, f(c_n) < 0$, entonces el nuevo intervalo es:

$$[a_{n+1}, b_{n+1}] = [a_n, c_n].$$

3. Si $f(c_n)\, f(b_n) < 0$, entonces el nuevo intervalo es:

$$[a_{n+1}, b_{n+1}] = [c_n, b_n].$$

Lectura de los valores iniciales

El usuario ingresa los puntos x_1 y x_2. Posteriormente se verifica la condición esencial del método:

$$f(x_1)\, f(x_2) < 0.$$

Si no existe cambio de signo, el programa termina porque el método no puede aplicarse.

Iteración principal del método

realiza el proceso del método de Regula Falsi:

1. Se calcula una nueva aproximación:
$$x_r = x_2 - f(x_2)\frac{x_1 - x_2}{f(x_1) - f(x_2)}.$$
2. Se evalúa $f(x_r)$.
3. Se calcula el error:
$$E = |x_r - x_{\text{anterior}}|.$$
4. Se decide qué subintervalo conservar:
$$f(x_1)f(x_r) < 0 \Rightarrow x_2 = x_r,$$
$$f(x_r)f(x_2) < 0 \Rightarrow x_1 = x_r.$$
5. Se almacena el valor calculado para la gráfica.
6. Se imprime la tabla iterativa.

2. Implementación del código en Python

```
import math, re, matplotlib.pyplot as plt

print("=== M TODO DE REGULA FALSI ===")

def preparar_funcion(expr):
    expr = expr.lower().strip()
    expr = expr.replace("^", "**").replace("sen", "sin").replace("ln", "
       log").replace("  ", "pi")
    expr = re.sub(r'(\d)([a-zA-Z\(])', r'\1*\2', expr)
    expr = re.sub(r'([a-zA-Z\)])(\d)', r'\1*\2', expr)
    return expr

```

```
funcion_str = input("Ingrese la funci n f(x): ")
funcion_str = preparar_funcion(funcion_str)

def f(x):
    try: return eval(funcion_str, {"__builtins__": None}, math.__dict__ |
       {"x": x})
    except: return None

x1 = float(input("Ingrese x1 = "))
x2 = float(input("Ingrese x2 = "))
f1, f2 = f(x1), f(x2)
if f1 * f2 > 0:
    print("No hay cambio de signo entre x1 y x2.")
    exit()

tol, max_iter = 1e-10, 100
itera, error, xr = 0, 1, x1
valores = []

print(f"\n{'Itera':<6} {'xr':<15} {'error':<15} {'f(xr)':<15}")
print("-" * 55)

while itera < max_iter and error > tol:
    xant = xr
    xr = x2 - f2 * (x1 - x2) / (f1 - f2)
    fxr = f(xr)
    if fxr is None: break
    itera += 1
    error = abs(xr - xant)
    valores.append(xr)
    print(f"{itera:<6} {xr:<15.10f} {error:<15.10f} {fxr:<15.10f}")
    if f1 * fxr < 0:
        x2, f2 = xr, fxr
    else:
        x1, f1 = xr, fxr

print("\n--- RESULTADOS ---")
print(f"xr = {xr:.10f}")
print(f"f(xr) = {f(xr):.10f}")
print(f"error = {error:.10f}")
print(f"iteraciones = {itera}")

# --- GRAFICAR ---
xs = [x1 + i*(x2-x1)/200 for i in range(201)]
ys = [f(x) for x in xs]
plt.plot(xs, ys, label="f(x)")
plt.axhline(0, color='black')
plt.scatter(valores, [f(v) for v in valores], color='red', label='
   Aproximaciones')
plt.title("M todo de Regula Falsi")
plt.legend()
plt.show()
```

```
=== MÉTODO DE REGULA FALSI ===
Ingrese la función f(x): x^3 - x - 1
Ingrese x1 = 1
Ingrese x2 = 2

Itera  xr             error          f(xr)
-----------------------------------------------------
1      1.1666666667   0.1666666667   -0.5787037037
2      1.2531120332   0.0864453665   -0.2853630296
3      1.2934374019   0.0403253687   -0.1295420928
4      1.3112810215   0.0178436196   -0.0565884873
5      1.3189885036   0.0077074821   -0.0243037472
6      1.3222827175   0.0032942139   -0.0103618501
7      1.3236842939   0.0014015764   -0.0044039499
8      1.3242794617   0.0005951679   -0.0018692584
9      1.3245319866   0.0002525248   -0.0007929592
10     1.3246390933   0.0001071067   -0.0003363010
11     1.3246845152   0.0000454219   -0.0001426137
12     1.3247037765   0.0000192613   -0.0000604750
13     1.3247119441   0.0000081676   -0.0000256438
14     1.3247154075   0.0000034634   -0.0000108739
15     1.3247168760   0.0000014686   -0.0000046109
16     1.3247174988   0.0000006227   -0.0000019552
17     1.3247177628   0.0000002641   -0.0000008291
18     1.3247178748   0.0000001120   -0.0000003516
19     1.3247179223   0.0000000475   -0.0000001491
20     1.3247179424   0.0000000201   -0.0000000632
21     1.3247179510   0.0000000085   -0.0000000268
22     1.3247179546   0.0000000036   -0.0000000114
23     1.3247179561   0.0000000015   -0.0000000048
24     1.3247179568   0.0000000007   -0.0000000020
25     1.3247179570   0.0000000003   -0.0000000009
26     1.3247179572   0.0000000001   -0.0000000004
27     1.3247179572   0.0000000000   -0.0000000002

--- RESULTADOS ---
xr = 1.3247179572
f(xr) = -0.0000000002
error = 0.0000000000
iteraciones = 27
```

Figure 1