Escuela Politécnica Nacional

Nombre: Dany Molina

Fecha: 18/05/2025

Tarea 5 - Método de Newton y la Secante

Ejercicio 7

La función $f(x) = x^{1/3}$ tiene raíz en $x = 0$.Usando el punto de inicio de $x = 1 \quad y \quad p_0 = 5, p_1 = 0.5$ para el método de secante, compare los resultados de los métodos de la secante y de Newton.

Vamos a comparar los métodos de Newton y Secante para encontrar la raíz de la función:

$$f(x) = x^{1/3}$$

con raíz en $x = 0$, usando los siguientes puntos iniciales:

Para **Newton**, $x_0 = 1$

Para **Secante**, $p_0 = 5$,$p_1 = 0.5$

$$f(x) = x^{1/3} \Rightarrow f'(x) = \frac{1}{3}x^{-2/3}$$

In [7]:
```
import numpy as np

# Definimos la función y su derivada
def f(x):
    return np.cbrt(x)  # x^(1/3)

def df(x):
    if x == 0:
        return np.inf  # Evitar división por 0
    return (1/3) *x**(-2/3)

def newton(f, df, x0, tol=1e-8, max_iter=20):
    for i in range(max_iter):
        if df(x0) == 0 or np.isinf(df(x0)):
            print(f"Derivada infinita o cero en iteración {i}.")
            return x0
        x1 = x0 - (f(x0)/df(x0))
        if abs(x1 - x0) < tol:
            break
        x0 = x1
    return x0

def secante(f, p0, p1, tol=1e-8, max_iter=20):
    for i in range(max_iter):
        f0 = f(p0)
        f1 = f(p1)
        if f1 - f0 == 0:
```

```python
            print(f"Dif. cero en iteración {i}")
            return p1
        p = p1 - f1 * (p1 - p0) / (f1 - f0)
        if abs(p - p1) < tol:
            break
        p0, p1 = p1, p
    return p
# Parámetros iniciales
x0_newton = 1
p0_secante = 5
p1_secante = 0.5

# Calcular las aproximaciones
raiz_newton = newton(f, df, x0_newton)
raiz_secante = secante(f, p0_secante, p1_secante)

# Mostrar resultados
print(f"Raíz (Newton-Raphson): {raiz_newton:.10f}")
print(f"Raíz (Secante):        {raiz_secante:.10f}")
print(f"Raíz exacta:           0.0000000000")
```

```
Raíz (Newton-Raphson): nan
Raíz (Secante):        0.8203606202
Raíz exacta:           0.0000000000
```

```
C:\Users\DANY M\AppData\Local\Temp\ipykernel_9516\2560083454.py:10: RuntimeWarnin
g: invalid value encountered in scalar power
  return (1/3) *x**(-2/3)
```

CONCLUSIÓN

El método de Newton no es adecuado cuando la derivada se vuelve infinita o muy inestable, como en este caso.

El método de la secante no requiere la derivada, y ofrece una mejor estabilidad numérica para funciones con derivadas problemáticas cerca de la raíz.