Método de Newton-Raphson

Resolución de ecuaciones no lineales

Profesor: Diego Valencia Enríquez

Universidad Mariana

17 de septiembre de 2025

¿Qué es el método de Newton-Raphson?

Concepto básico

El método de Newton-Raphson es un algoritmo iterativo para encontrar raíces de funciones no lineales mediante aproximaciones sucesivas.

Fundamento matemático

Se basa en la **linealización** de la función usando series de Taylor:

$$f(x_{n+1}) \approx f(x_n) + f'(x_n)(x_{n+1} - x_n) = 0$$

- Método de **convergencia rápida** (cuadrática)
- Requiere el cálculo de la **derivada** de la función
- Sensible a la elección del punto inicial

Fórmula iterativa

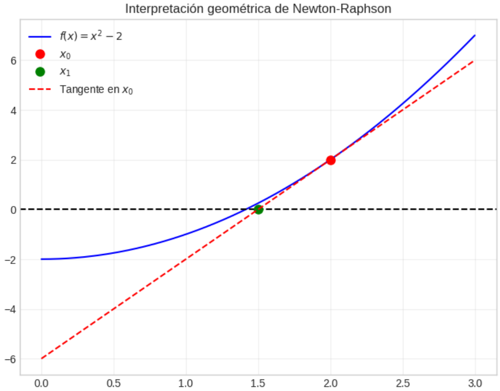
Derivación del método

La fórmula de iteración de Newton-Raphson es:

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

Interpretación geométrica

- Se traza la tangente en x_n
- La intersección con el eje x da x_{n+1}
- Proceso iterativo hasta convergencia



Algoritmo del método de Newton-Raphson

Pasos a seguir

1. Elegir un punto inicial x_0
2. Calcular $f(x_n)$ y $f'(x_n)$
3. Aplicar la fórmula: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$
4. Verificar criterio de convergencia:
   - $|x_{n+1} - x_n| < \epsilon$ (error absoluto)
   - $|f(x_{n+1})| < \epsilon$ (valor de la función)
5. Repetir hasta alcanzar la tolerancia o máximo de iteraciones

Condiciones de convergencia

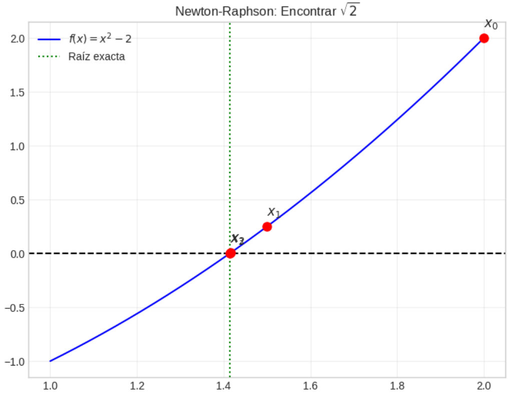
- $f'(x) \neq 0$ en la región de interés
- x_0 suficientemente cercano a la raíz
- Función suficientemente suave

Ejemplo 1: $f(x) = x^2 - 2$

Encontrar $\sqrt{2}$

- $f(x) = x^2 - 2$
- $f'(x) = 2x$
- Fórmula: $x_{n+1} = x_n - \frac{x_n^2 - 2}{2x_n}$
- $x_0 = 2{,}0$

Iter	x_n	$f(x_n)$	Error
0	2.000000	2.000000	-
1	1.500000	0.250000	0.500000
2	1.416667	0.006944	0.083333
3	1.414216	0.000006	0.002451
4	1.414214	0.000000	0.000002



Resultado

$\sqrt{2} \approx 1{,}414214$

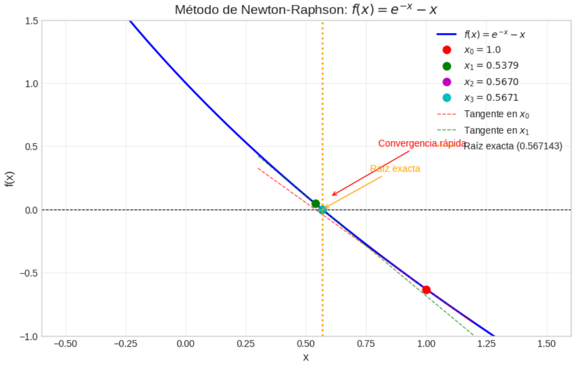
Solo 4 iteraciones vs 10+ en bisección

Ejemplo 2: $f(x) = e^{-x} - x$

Raíz en $[0, 1]$

- $f(x) = e^{-x} - x$
- $f'(x) = -e^{-x} - 1$
- Fórmula: $x_{n+1} = x_n - \frac{e^{-x_n} - x_n}{-e^{-x_n} - 1}$
- $x_0 = 1{,}0$

Iter	x_n	$f(x_n)$	Error
0	1.000000	-0.632121	-
1	0.537883	0.046051	0.462117
2	0.566987	-0.000300	0.029104
3	0.567143	0.000000	0.000156
4	0.567143	0.000000	0.000000



Resultado

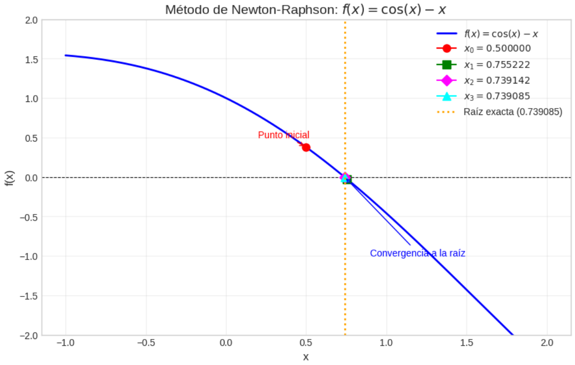
Raíz: $x \approx 0{,}567143$
Convergencia extremadamente rápida

Ejemplo 3: $f(x) = \cos(x) - x$

Raíz en $[0, 1]$

- $f(x) = \cos(x) - x$
- $f'(x) = -\sin(x) - 1$
- Fórmula: $x_{n+1} = x_n - \frac{\cos(x_n) - x_n}{-\sin(x_n) - 1}$
- $x_0 = 0{,}5$

Iter	x_n	$f(x_n)$	Error
0	0.500000	0.377583	-
1	0.755222	-0.027102	0.255222
2	0.739142	0.000304	0.016080
3	0.739085	0.000000	0.000057
4	0.739085	0.000000	0.000000



Resultado

Raíz: $x \approx 0{,}739085$
Solución de $\cos(x) = x$

Ejercicios propuestos

Resuelve las siguientes ecuaciones usando Newton-Raphson con $\epsilon = 10^{-6}$:

1. $f(x) = x^3 - 2x - 5 = 0$, $x_0 = 2{,}0$
2. $f(x) = \ln(x) - 1 = 0$, $x_0 = 2{,}5$
3. $f(x) = \sin(x) - 0{,}5x = 0$, $x_0 = 1{,}5$
4. $f(x) = e^x - 3x = 0$, $x_0 = 1{,}0$

Preguntas para analizar

- ¿Qué pasa si elegimos $x_0 = 0$ para $f(x) = x^2 - 2$?
- ¿Cómo afecta la elección del punto inicial a la convergencia?
- Compare la velocidad de convergencia con el método de bisección

Ventajas y desventajas

Ventajas:

- Convergencia muy rápida (cuadrática)
- Fácil de implementar
- Precisión alta en pocas iteraciones
- Eficiente computacionalmente

Desventajas:

- Requiere calcular la derivada
- Sensible al punto inicial
- Puede divergir si $f'(x) \approx 0$
- No siempre converge

Aplicaciones comunes

- Optimización numérica
- Solución de ecuaciones no lineales
- Ingeniería Civil
- Análisis de circuitos eléctricos
- Ingeniería de sistemas

Comparación: Newton-Raphson vs Bisección

Característica	Newton-Raphson	Bisección
Convergencia	Cuadrática	Lineal
Derivada requerida	Sí	No
Garantía de convergencia	No	Sí
Velocidad	Muy rápida	Lenta
Complejidad	Media	Baja
Punto inicial	Crítico	No crítico
Estabilidad	Variable	Muy estable
Iteraciones (ejemplo)	3-5	10-20
Precisión	Muy alta	Alta

Recomendación

- Usar Newton-Raphson cuando se pueda calcular la derivada
- Usar bisección como método de respaldo
- Combinar ambos métodos (bisección para aproximación inicial)

Implementación en Python

Código básico y ejemplo de uso

```
def newton_raphson(f, df, x0, tol=1e-6, max_iter=100):
    "
    f: función
    df: derivada de f
    x0: valor inicial
    tol: tolerancia
    max_iter: máximo de iteraciones
    "
    x = x0
    for i in range(max_iter):
        fx = f(x)
        if abs(fx) < tol:
            return x
        dfx = df(x)
        if abs(dfx) < 1e-10:  # Evitar división por cero
            raise ValueError("Derivada cercana a cero")
        x = x - fx / dfx
    return x

# Ejemplo: f(x) = x^2 - 2
raiz = newton_raphson(lambda x: x**2 - 2,
                      lambda x: 2*x, 2.0)
print(f"sqrt(2)  raiz:.6f")
```

¡Gracias por su atención!

¿Preguntas?