

Método de Newton-Raphson

Alumno: Henry Ccoarite Dueñas

Docente: Fred Torres Cruz

Universidad Nacional del Altiplano

Octubre 2025

1. Idea general

El método de Newton-Raphson es un procedimiento numérico iterativo que permite encontrar raíces de ecuaciones no lineales de la forma:

$$f(x) = 0$$

La idea geométrica se basa en usar la **recta tangente** en un punto cercano a la raíz para aproximarla mediante intersecciones sucesivas con el eje x .

Fórmula de iteración

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

2. Derivación del método

A partir de la expansión de Taylor de $f(x)$ en torno a x_n :

$$f(x) \approx f(x_n) + f'(x_n)(x - x_n)$$

Si imponemos $f(x) = 0$, despejando x se obtiene:

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

que es la fórmula iterativa básica del método de Newton-Raphson.

3. Convergencia

Si $f(r) = 0$, $f'(r) \neq 0$ y $f \in C^2$, se cumple que:

$$e_{n+1} \approx C e_n^2, \quad \text{con} \quad C = \frac{|f''(r)|}{2|f'(r)|}$$

donde e_n es el error en la iteración n . Esto indica que la convergencia es **cuadrática**: los dígitos correctos se duplican en cada iteración.

4. Condiciones que cumple el programa

El método de Newton-Raphson necesita que se cumplan algunas condiciones para que funcione correctamente y el proceso de aproximación llegue a la raíz deseada. A continuación se explican las principales condiciones teóricas y cómo se reflejan en el programa elaborado en Python.

Condiciones teóricas

- La función $f(x)$ debe ser continua y derivable cerca de la raíz.
- La derivada $f'(x)$ no debe ser igual a cero en los puntos donde se aplica el método.
- El valor inicial x_0 debe elegirse lo más cercano posible a la raíz real.
- La tolerancia (tol) debe ser pequeña para asegurar buena precisión.
- Debe limitarse el número de iteraciones para evitar bucles infinitos en caso de no converger.

En el programa

En el código desarrollado, estas condiciones se cumplen de la siguiente manera:

Condición	Cómo se cumple en el programa	Cumple
$f(x)$ continua y derivable	Se usa <code>sympy.symbols()</code> y <code>sympy.diff()</code> para definir y derivar la función.	Sí
$f'(x) \neq 0$	El programa evalúa $f'(x)$ numéricamente; si se elige mal x_0 , puede dar división por cero.	Parcial
Punto inicial adecuado	El usuario ingresa el valor de x_0 ; si está lejos de la raíz, puede no converger.	Parcial
Tolerancia pequeña (tol)	Se define como 10^{-6} , lo cual permite obtener varios decimales correctos.	Sí
Número máximo de iteraciones	Se controla con el parámetro n (por ejemplo, 10 iteraciones).	Sí

5. Implementación en Python

```
1 import sympy as sp
2 from math import *
3
4 def NewtonRaphson(x0, tol, n):
5     x = sp.symbols('x')
6     f = input('Digite la funcion (con variable x): ')
7     f = sp.sympify(f)
8     df = sp.diff(f)
9
10    f = sp.lambdify(x, f)
11    df = sp.lambdify(x, df)
```

```

12
13     for k in range(n):
14         x1 = x0 - f(x0) / df(x0)
15
16         if abs(x1 - x0) < tol:
17             print('x', k+1, '=', x1, end=' ')
18             print('    Es una buena aproximacion de la raiz')
19             return
20
21         x0 = x1
22         print('x', k+1, '=', x1)
23
24     print('La raiz aproximada despues de', n, 'iteraciones es:', x1)
25
26 NewtonRaphson(pi, 0.000001, 10)

```

Listing 1: Implementación del método de Newton-Raphson

5. Ejemplo de ejecución

Para la función:

$$f(x) = x^3 - x - 2$$

y con condiciones iniciales $x_0 = 1,5$, $tol = 10^{-6}$ y $n = 10$, el método converge a:

$$x \approx 1,5213797068$$

```

Digite la función (con variable x): x**3-x-2
x 1 = 2.237511668372433
x 2 = 1.7407350739176706
x 3 = 1.551133335317269
x 4 = 1.5220379868676062
x 5 = 1.5213800393134478
x 6 = 1.5213797068046524 → Es una buena aproximación de la raíz

```

Figura 1: Gráfica del método de Newton-Raphson mostrando la raíz aproximada.

Iteraciones del método

Para la función:

$$f(x) = x^3 - x - 2$$

con un valor inicial $x_0 = 1,5$, una tolerancia de 10^{-6} y un máximo de 10 iteraciones, el proceso de Newton-Raphson se desarrolla de la siguiente forma:

Iteración	x_n	$f(x_n)$	x_{n+1}
0	1.5000000000	-0.1250000000	1.5217391304
1	1.5217391304	0.0021369278	1.5213798059
2	1.5213798059	0.0000020656	1.5213797068
3	1.5213797068	0.0000000000	—

A partir de la tercera iteración, la diferencia entre x_{n+1} y x_n es menor que la tolerancia establecida, por lo que se considera que el método ha convergido.

$$x \approx 1,5213797068$$

El resultado cumple con el criterio de precisión definido y representa una buena aproximación de la raíz real de la ecuación $x^3 - x - 2 = 0$.