Método de Newton

Juan F. Olivares Pacheco*

Índice

1 História del método de Newton

El **método de Newton**, también conocido como método de Newton-Raphson, es una técnica iterativa utilizada para aproximar las raíces de una función real. Se destaca por su eficiencia y rapidez de convergencia, especialmente cuando se compara con otros métodos numéricos. Este método es fundamental en el análisis numérico y tiene aplicaciones que abarcan desde la ingeniería y la física hasta la economía y las ciencias computacionales.

El método de Newton tiene sus raíces en el siglo XVII, una época marcada por avances significativos en matemáticas y ciencias naturales. Isaac Newton (1642-1727), matemático y físico inglés, es ampliamente reconocido por sus contribuciones fundamentales en cálculo, óptica y mecánica clásica. Aunque Newton es el nombre más asociado con este método, es importante reconocer que el desarrollo de técnicas para encontrar raíces de ecuaciones precede a su trabajo y ha sido una preocupación de matemáticos desde la antigüedad.

Civilizaciones como la babilónica y la egipcia desarrollaron técnicas rudimentarias

*jfolivar@uda.cl

para resolver ecuaciones cuadráticas y cúbicas. Sin embargo, carecían de una formulación general y sistemática. Con el renacimiento de las matemáticas en Europa, hubo un interés renovado en resolver ecuaciones polinomiales de grados superiores. Matemáticos como François Viète y René Descartes hicieron contribuciones significativas en álgebra y geometría.

Newton desarrolló el cálculo diferencial e integral, al que llamó "método de las fluxiones". Este marco teórico permitió una comprensión más profunda de las tasas de cambio y las tangentes a curvas, conceptos esenciales para el método que lleva su nombre. Aunque Newton desarrolló su método alrededor de 1669, no lo publicó inmediatamente. Fue Joseph Raphson quien, en 1690, publicó una versión del método en su obra "Analysis aequationum universalis". Más tarde, en 1740, el matemático inglés Thomas Simpson popularizó el método en la forma en que se conoce hoy.

Aunque menos conocido, Raphson refinó el método y presentó una versión más generalizada que no requería recalcular ciertos términos en cada iteración. A lo largo de los siglos XVIII y XIX, matemáticos como Euler y Cauchy analizaron y extendieron el método, estudiando su convergencia y aplicabilidad.

El método de Newton representó un avance significativo porque proporcionó una herramienta sistemática para resolver ecuaciones que no podían abordarse analíticamente. Esto fue crucial en una época en la que las matemáticas comenzaban a aplicarse de manera más intensa en la física y la ingeniería.

Algunas ventajas del método de Newton son:

- **Convergencia rápida:** Cuando se inicia cerca de la raíz verdadera, el método suele converger muy rápidamente debido a su naturaleza cuadrática.
- **Aplicabilidad general:** Puede adaptarse para funciones de varias variables y sistemas de ecuaciones no lineales.
- **Facilidad de implementación**: Es relativamente sencillo de programar y utilizar en cálculos computacionales.

Por otro lado, tenemos limitaciones y consideraciones a tener en cuenta en la aplicación de método:

- **Necesidad de la derivada**: Requiere el cálculo de la derivada de la función, lo que puede ser complicado para funciones muy complejas.
- **Sensibilidad a la aproximación inicial**: Una elección inadecuada del valor inicial puede conducir a divergencia o a convergencia hacia una raíz no deseada.

- **Problemas con derivadas cero**: Si la derivada en algún punto es cero, el método puede fallar o producir errores numéricos.

2 Derivación del método de Newton

El método de Newton es una herramienta poderosa en el análisis numérico para encontrar aproximaciones sucesivas de las **raíces de funciones reales**. Su derivación se basa en la aproximación de Taylor y permite transformar un problema no lineal en una serie de problemas lineales más manejables. En esta sección, detallaremos cómo se desarrolla el método a partir de la aproximación de Taylor y deduciremos la fórmula iterativa que es el núcleo del algoritmo.

La **serie de Taylor** es una representación de una función diferenciable $f(x)$ alrededor de un punto $x = x_n$ como una suma infinita de términos que involucran las derivadas de la función en ese punto:

$$f(x) = f(x_n) + f'(x_n)(x - x_n) + \frac{f''(x_n)}{2!}(x - x_n)^2 + \frac{f'''(x_n)}{3!}(x - x_n)^3 + \dots$$

Para el método de Newton, nos interesa la aproximación de primer orden, es decir, truncamos la serie después del término lineal:

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

Esta expresión es la **aproximación lineal** de la función $f(x)$ cerca del punto x_n. Supone que, en un intervalo pequeño alrededor de x_n, la función puede aproximarse razonablemente por su tangente en ese punto.

La idea detrás de utilizar la aproximación lineal es simplificar el problema de **encontrar una raíz de una función no lineal** a la resolución de una ecuación lineal. Dado que estamos buscando el valor de x tal que $f(x) = 0$, reemplazamos $f(x)$ por su aproximación lineal:

$$0 = f(x_n) + f'(x_n)(x - x_n)$$

Al resolver esta ecuación lineal para x, obtenemos una nueva aproximación, que podemos llamar x_{n+1} de la raíz. Es decir, partiendo de la ecuación obtenida:

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

Despejamos x_{n+1}:

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

Esta es la <u>**fórmula iterativa del método de Newton**</u>, que nos proporciona una secuencia de aproximaciones $x_0, x_1, x_2, \dots$ convergiendo hacia la raíz buscada.

En el contexto de la optimización, un punto x^* es óptimo cuando:

$$\nabla f(x^*) = f'(x^*) = 0$$

Entonces, podemos usar el método de Newton para encontrar la "raíz" del gradiente de $f(x)$.

Si hacemos:

$$g(x) = \nabla f(x) = f'(x)$$

y aplicamos la serie de Taylor de primer orden sobre $g(x)$, tenemos:

$$g(x) = g(x_n) + g'(x_n)(x_{n+1} - x_n)$$

Si tomamos $g(x) = 0$ y despejando x_{n+1}:

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

Si reemplazamos la función $g(x)$ por $f'(x)$:

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

Esta fórmula iterativa del método de Newton, nos proporcionará una secuencia de aproximaciones convergiendo al valor óptimo buscado.

3 Método de Newton

En el contexto general, el método de Newton para la minimización de funciones utiliza la información de la primera y segunda derivada de la función (el gradiente y el Hessiano) para iterativamente aproximar el mínimo de la función.

Algunos conceptos claves de este método son:

1. **Función objetivo:** La función $f(\mathbf{x})$ que se desea optimizar (minimizar o maximizar).

2. **Derivada (gradiente en varias dimensiones):** La primera derivada de la función objetivo, que indica la pendiente de la función en un punto dado. En el caso de funciones de varias variables, el gradiente $\nabla f(\mathbf{x})$ es un vector que apunta en la dirección de mayor aumento de la función.

3. **Segunda derivada (Hessiano en varias dimensiones):** La segunda derivada de la función objetivo, proporciona información sobre la curvatura de la función. En el caso de funciones de varias variables, la matriz Hessiana $\mathbf{H}_f(\mathbf{x})$ es una matriz de derivadas parciales de segundo orden.

4. **Iteración:** El método de Newton utiliza una fórmula iterativa que actualiza la estimación de la raíz o del punto óptimo hasta que se alcanza la convergencia.

El método de Newton para la minimización de una función $f(\mathbf{x})$ en varias dimensiones se puede describir con los siguientes pasos:

1. **Inicialización:** Seleccionamos un punto inicial, $\mathbf{x}_0$, en el dominio de la función como punto de partida para la optimización, cerca de la solución óptima.

2. **Iteración:** Para $k = 0, 1, 2, \ldots$:

   - Calcular el gradiente $\nabla f(\mathbf{x}_k)$
   - Calcular la matriz Hessiana $\mathbf{H}_f(\mathbf{x}_k)$
   - Actualizar el punto:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_f^{-1}(\mathbf{x}_k) \cdot \nabla f(\mathbf{x}_k)$$

3. **Verificación de la convergencia:** Se finalizan las iteraciones si se cumple algún criterio de convergencia. Si es así, se retorna $\mathbf{x}_{k+1}$ como solución aproximada.

El método de Newton converge rápidamente hacia el mínimo local cuando la función es convexa y su Hessiana es positiva definida. Sin embargo, puede no funcionar bien si la Hessiana no es definida positiva o si el punto inicial está lejos de la solución óptima.

Ejemplo: Supongamos que queremos encontrar el mínimo de la siguiente función:

$$f(x) = x^2,\ x \in \mathbb{R}$$

el siguiente código muestra la implementación en `R` del método de Newton. Comenzamos definiendo la función objetivo $f(x)$.

```
f <- function(x) {
  return(x^2)
}
```

También definimos el gradiente de f (primera derivada) y el hessiano de f (segunda derivada).

```
grad_f <- function(x) {
  return(2 * x)
}
```

```
hess_f <- function(x) {
  return(2)
}
```

Ajustamos los parámetros del algoritmo.

```
max_iter  <- 1000
tol <- 1e-6
```

Definimos el punto inicial x_{actual}.

```
x_actual <- 5
```

Ahora realizamos las iteraciones del método de Newton.

```
for (i in 1:max_iter) {
  x_nuevo <- x_actual - grad_f(x_actual) / hess_f(x_actual)
  if (norm(grad_f(x_nuevo), "2") < tol) {
    break
  }
  x_actual <- x_nuevo
```

```
}
```

Mostramos los resultados.

```
resultados <- list(minimo = f(x_nuevo),
                   objetivo = x_nuevo,
                   iteraciones = i)
print(resultados)
```

```
## $minimo
## [1] 0
##
## $objetivo
## [1] 0
##
## $iteraciones
## [1] 1
```

Ejemplo: Supongamos que queremos encontrar el mínimo de la siguiente función:

$$f(x, y) = x^2 + y^2,\ x,\ y \in \mathbb{R}^2$$

el siguiente código muestra la implementación en `R` del método de Newton.

```
# Definimos la función objetivo
f <- function(param) {
  x <- param[1]
  y <- param[2]
  return(x^2 + y^2)
}

# Definimos el gradiente de la función
grad_f <- function(param) {
  x <- param[1]
  y <- param[2]
  return(c(2 * x, 2 * y))
}

# Definimos la matriz hessiana de la función
hess_f <- function(param) {
  x <- param[1]
```

```r
  y <- param[2]
  return(matrix(c(2, 0, 0, 2),
              ncol = 2, byrow = TRUE))
}

# Establecemos los parámetros del algoritmo
max_iter <- 1000
tol <- 1e-6

# Establecemos un punto inicial
x_actual <- c(1, 1)

# Algoritmo de Newton
for (i in 1:max_iter) {
  x_nuevo <- x_actual - solve(hess_f(x_actual)) %*% grad_f(x_actual)
  if (norm(grad_f(x_nuevo), "2") < tol) {
    break
  }
  x_actual <- x_nuevo
}

# Mostramos los resultados
resultados <- list(minimo = f(x_nuevo),
                   objetivo = as.vector(x_nuevo),
                   iteraciones = i)
print(resultados)
```

```
## $minimo
## [1] 0
##
## $objetivo
## [1] 0 0
##
## $iteraciones
## [1] 1
```

La implementación del método de Newton requiere varias consideraciones adicionales para asegurar su eficacia y estabilidad:

- **Condición de convergencia:** Es crucial establecer criterios claros de convergencia. Generalmente, esto se hace comprobando si la norma del gradiente es menor que un umbral pequeño ($\|\nabla f(\mathbf{x}_{k+1})\| < \epsilon$). Además de la norma del gradiente, otros criterios incluyen la variación en el valor de la función objetivo ($\|f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k)\| < \epsilon$) o el cambio en las variables de decisión entre iteraciones consecutivas ($\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \epsilon$).

- **Condición inicial:** La elección del punto inicial puede afectar significativamente la convergencia. En algunos casos, un mal punto inicial puede llevar a convergencia lenta o a puntos de silla. A veces, se requiere una buena intuición o conocimiento del problema para elegir un punto inicial adecuado.
- **Positividad definida del Hessiano:** El método de Newton requiere que la matriz Hessiana sea positiva definida para asegurar que el paso de actualización apunte en la dirección correcta (es decir, hacia un mínimo). Si el Hessiano no es positivo definido, puede ser necesario modificarlo. Una técnica común es la **regularización**, donde se agrega una pequeña constante diagonal a la matriz Hessiana.

```
if (any(eigen(hessian)$values <= 0)) {
  hessian <- hessian + diag(rep(c, length(x)))
}
```

  donde `c` es una constante.

Ejemplo: Supongamos que queremos encontrar el mínimo de la siguiente función:

$$f(x, y) = (1-x)^2 + 100(y-x^2)^2,\ x,\ y \in \mathbb{R}^2$$

el siguiente código muestra la implementación en `R` del método de Newton. Aquí, usaremos la librería `Deriv` para calcular las derivadas necesarias.

```
# Definimos la función objetivo
f <- function(param) {
  x <- param[1]
  y <- param[2]
  (1 - x)^2 + 100 * (y - x^2)^2
}

# Cargamos la librería "Deriv"
library(Deriv)

# Definimos el gradiente de la función
```

```r
grad_f <- Deriv(f, c("x", "y"))

# Definimos la matriz hessiana de la función
hess_f <- Deriv(f, c("x", "y"), n = 2)

# Establecemos los parámetros del algoritmo
max_iter <- 1000
tol <- 1e-6

# Establecemos el punto inicial
x_actual <- c(-1.2, 1)

# Algoritmo de Newton
for (i in 1:max_iter) {
  hessian <- matrix(hess_f(x_actual), nrow = 2, byrow = TRUE)
  if (any(eigen(hessian)$values <= 0)) {
    hessian <- hessian + diag(abs(min(eigen(hessian)$values)), 2)
  }
  x_nuevo <- x_actual - solve(hessian) %*% grad_f(x_actual)
  if (norm(grad_f(x_nuevo), "2") < tol) {
    break
  }
  x_actual <- x_nuevo
}

# Mostramos los resultados
resultados <- list(minimo = f(x_nuevo),
                   objetivo = as.vector(x_nuevo),
                   iteraciones = i)
print(resultados)
```

```
## $minimo
## [1] 3.432646e-20
##
## $objetivo
## [1] 1 1
##
## $iteraciones
## [1] 6
```

4 Ejercicios

1. Encuentre el mínimo de la siguiente función:

$$f(x) = x^2 - 4x + 4$$

2. Encuentre el mínimo local de la función en el intervalo $[-2, 2]$:

$$f(x) = \sin(x) + \frac{x^2}{4}$$

3. Encuentre el mínimo de la función:

$$f(x, y) = (x - 3)^2 + (y + 1)^2$$

4. Encuentre el mínimo de la función:

$$f(x, y) = x^2 + y^2 + 4xy$$

5. Encuentre el mínimo de la función:

$$f(x, y) = e^{x+y} + x^2 + y^2$$

6. Encuentre los mínimos locales y verifique la positividad definida del Hessiano en esos puntos:

$$f(x, y) = x^4 + y^4 - 4x^2 - 4y^2$$

7. Encuentre el mínimo de la función:

$$f(x, y, z) = x^2 + y^2 + z^2$$

8. Encuentre el mínimo de la función:

$$f(x, y, z) = x^2 + 2y^2 + 3z^2 + 4xy + 5yz + 6zx$$

9. Encuentre un mínimo local de la función. Asegúrese de elegir un buen punto inicial para la convergencia.

$$f(x, y, z) = \sin(x) + \cos(y) + \tan(z)$$

10. Encuentre el mínimo de la función:

$$f(x, y, z) = e^{x+y+z} + x^2 + y^2 + z^2$$