Integración numérica

Uriel Paluch

9/11/2021

Método

```
newtonCotesCerradas <- function(limiteInferior, limiteSuperior, funcion, n){

  h <- (limiteSuperior - limiteInferior)/n
  fx0 <- eval(funcion, list(x = limiteInferior))
  fxn <- eval(funcion, list(x = limiteSuperior))

  if(n > 1){
    fx <- rep(NA, times = (n-1))
    for (i in 1:(n-1)) {
      fx[i] <- eval(funcion, list(x = limiteInferior + i*h))
    }
  }

  else if (n == 1){
    return((h/2) * (fx0 + fxn))
  }
  else if (n == 2){
    return((h/3) * (fx0 + 4*fx[1] + fxn))
  }
  else if(n == 3){
    return((3/8)*h*(fx0 + 3*fx[1] + 3*fx[2] + fxn))
  }
  else if(n == 4){
    return((2/45) * h * ( 7 * fx0 + 32 * fx[1] + 12 * fx[2] + 32 * fx[3] + 7 * fxn))
  }

}

# n = 1. Regla del trapecio.
# n = 2. Regla de Simpson.
# n = 3. Regla de tres octavos de Simpson.
# n = 4 regla de NC cerrada con n = 4.
# Poner la funcion con "x" como incognita
newtonCotesCerradas(limiteInferior = 0, limiteSuperior = 0.35, funcion = expression(2/(x^2-4)), n = 1)
```

```
## [1] -0.1777643
```

```
newtonCotesAbiertas <- function(limiteInferior, limiteSuperior, funcion, n){
  h <- (limiteSuperior - limiteInferior)/(n+2)

  fx <- rep(NA, times = (n+1))
```

```r
  for (i in 1:(n+1)) {
    fx[i] <- eval(funcion, list(x = limiteInferior + i*h))
  }

  if (n == 0){
    return(2 * h * fx[1])
  }
  else if (n == 1){
    return((3/2)* h * (fx[1] + fx[2]))
  }
  else if(n == 2){
    return((4/3)*h*(2*fx[1] - fx[2] + 2*fx[3]))
  }
  else if(n == 3){
    return((5/24) * h * ( 11 * fx[1] + fx[2] + fx[3] + 11 * fx[4]))
  }

}

# Poner la funcion con "x" como incognita
newtonCotesAbiertas(limiteInferior = 0, limiteSuperior = pi/4, funcion = expression(sin(x)), n = 3)
```

```
## [1] 0.2928692
```

A menudo surge la necesidad de evaluar una integral definida de una función que es difícil de obtener. El método básico asociado con la aproximación de $\int_a^b f(x)dx$ recibe el nombre ed *cuadratura numérica*. Éste utiliza una suma $\sum_{i=0}^{n} a_i f(x_i)$ para aproximar $\int_a^b f(x)dx$. Los métodos de cuadratura que analizamos en esta sección se basan en los polinomios explicados en la semana 8.

La *regla trapezoidal* y la *regla de Simpson* surgen del polinomio de Lagrange con valores equiespaciados.

La regla trapezoidal o regla del trapecio

Sean $x_0 = a, x_1 = b, h = b - a$

$$\int_a^b f(x)dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3}{12}f''(\varepsilon) \tag{1}$$

Esto recibe el nombre de regla trapezoidal porque cuando f es una función con valores positivos, $\int_a^b f(x)dx$ se aporxima mediante el área de un trapecio.

El termino de error implica la derivada segunda, por lo tanto, dara resultados exactos cuando se aplique a polinomio de grado 1 o cero.

Regla de Simpson

Sea $x_0 = a, x_2 = by x_1 = a + h$, en donde $h = (b - a)/2$ La regla de simpson resulta de la integración sobre $[a, b]$ del segundo polinomio de Lagrange con nodos equiespaciados.

$$\int_a^b f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90}f^{(4)}(\varepsilon) \tag{2}$$

El término de error en la regla de Simpson implica la cuarta derivada de f, por lo que da resultados exactos cuando se aplica a cualquier polinomio de grado tres o menos.

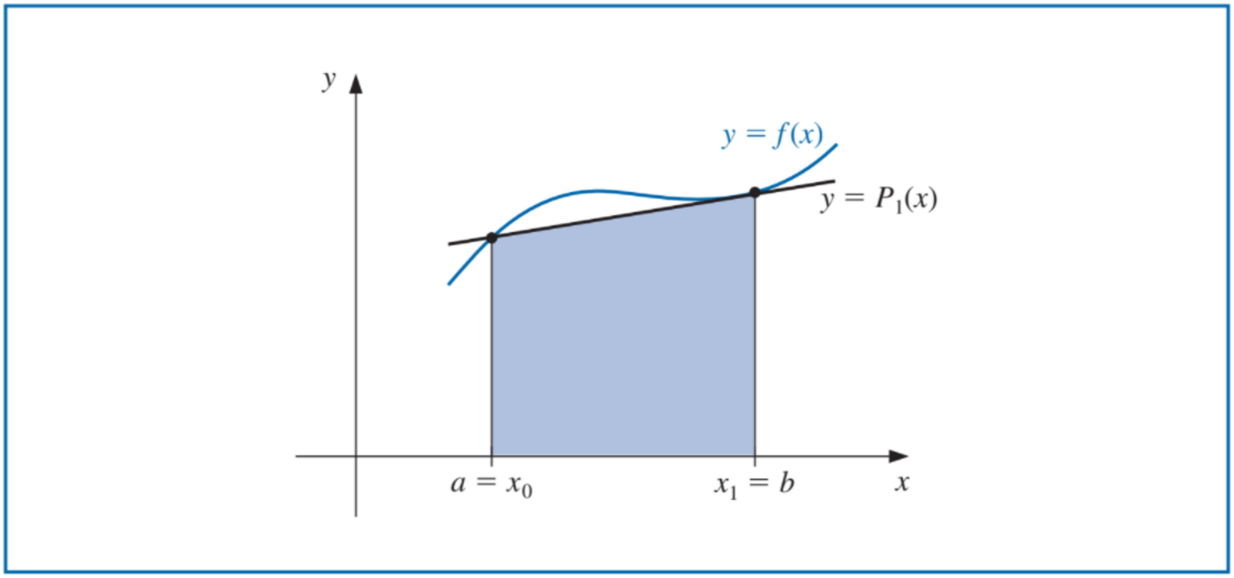

Figure 1: regla del trapecio

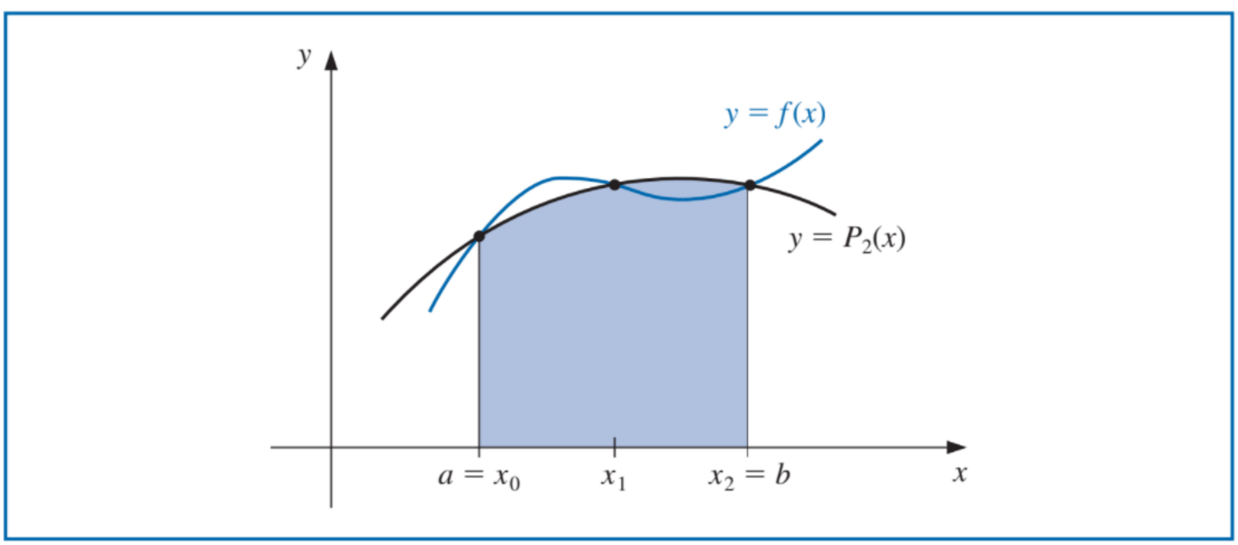

Figure 2: regla de Simpson

Fórmulas de Newton-Cotes cerradas

La fórmula cerrada de $n+1$ puntos de Newton-Cotes utiliza nodos $x_i = x_0 + i * h$, para $i = 0, 1, ..., n$, donde $x_0 = a$, $x_n = b$, $h = (b-a)/n$. Recibe el nombre de cerrada porque los extremos del intervalo $[a, b]$ se incluyen como nodos.

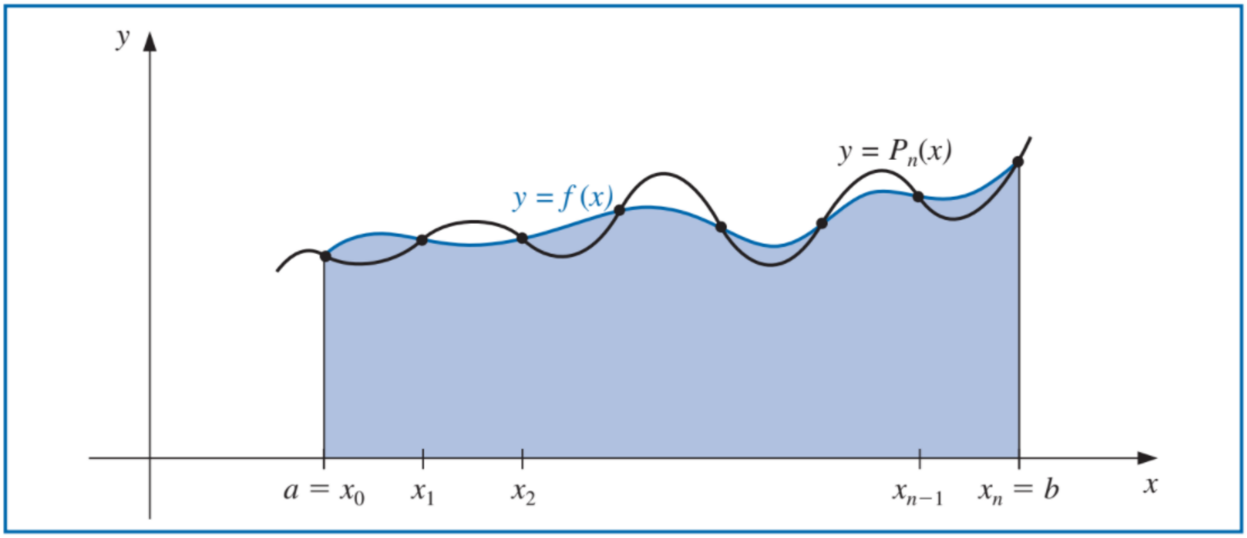

Figure 3: Newton-Cotes cerrada

Fórmulas de Newton-Cotes abiertas

Las fórmulas de Newton-Cotes abiertas no incluyen a los extremos de $[a, b]$ como nodos.

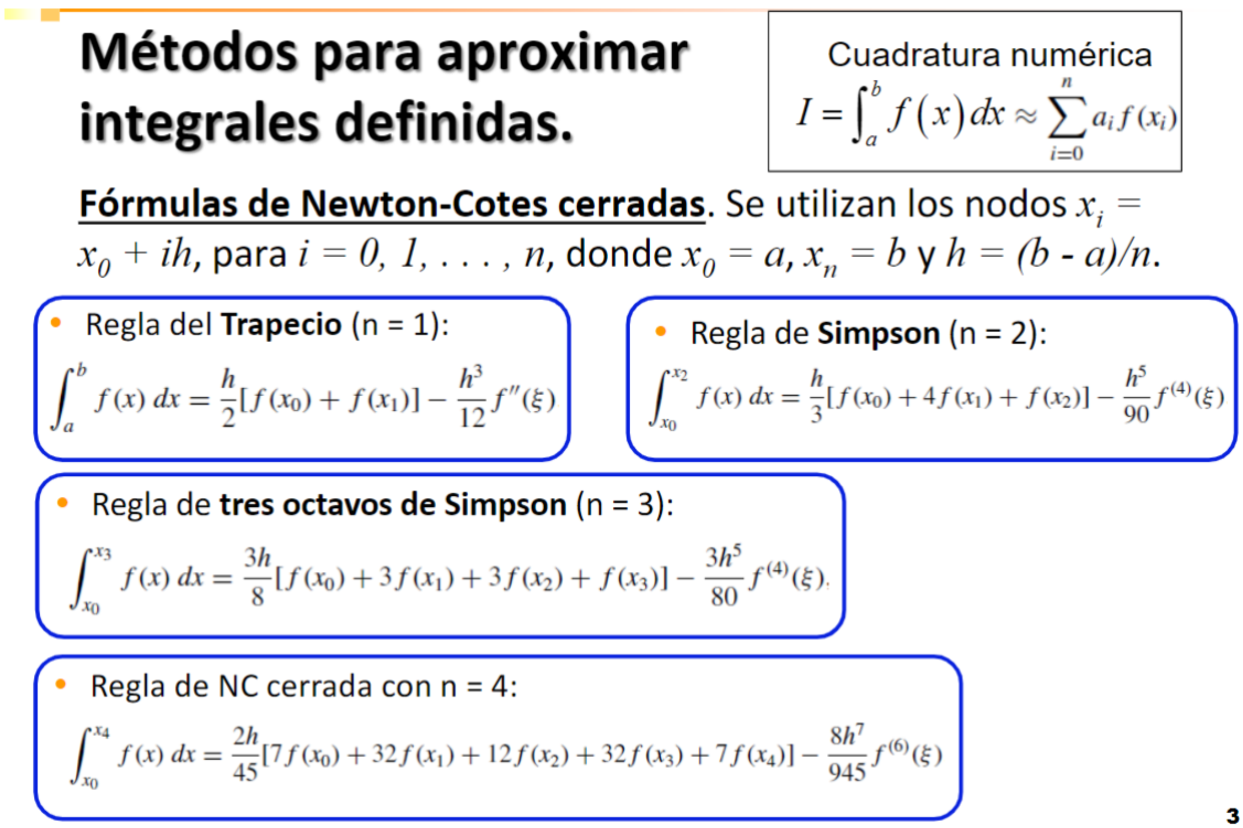

Figure 4: Newton-Cotes formulas