Actividad 2 – Método de Secante y Newton

Métodos Numéricos

Ingeniería en Desarrollo de Software

Tutor: Miguel Ángel Rodríguez Vega.

Alumno: Uziel Abisai Martinez Oseguera.

Fecha: 20/10/2023.

Índice

Introducción

En el campo del análisis numérico, los métodos de secante y Newton-Raphson son herramientas esenciales para encontrar soluciones aproximadas a problemas matemáticos complejos. Estos métodos son particularmente útiles cuando se trata de resolver ecuaciones no lineales que no tienen soluciones analíticas directas o cuyas soluciones son difíciles de obtener. La presente actividad tiene como objetivo explorar y comprender la aplicación de estos métodos en la búsqueda de soluciones numéricas.

Descripción

Esta actividad se centra en la implementación y análisis de dos métodos numéricos ampliamente utilizados: el método de la secante y el método de Newton-Raphson. Ambos métodos se aplicarán para resolver dos ecuaciones no lineales específicas: la ecuación 1, que corresponde al método de la secante, y la ecuación 2, que se resolverá utilizando el método de Newton-Raphson. Utilizaremos el entorno de programación RStudio y los archivos proporcionados para llevar a cabo estas resoluciones.

Justificación

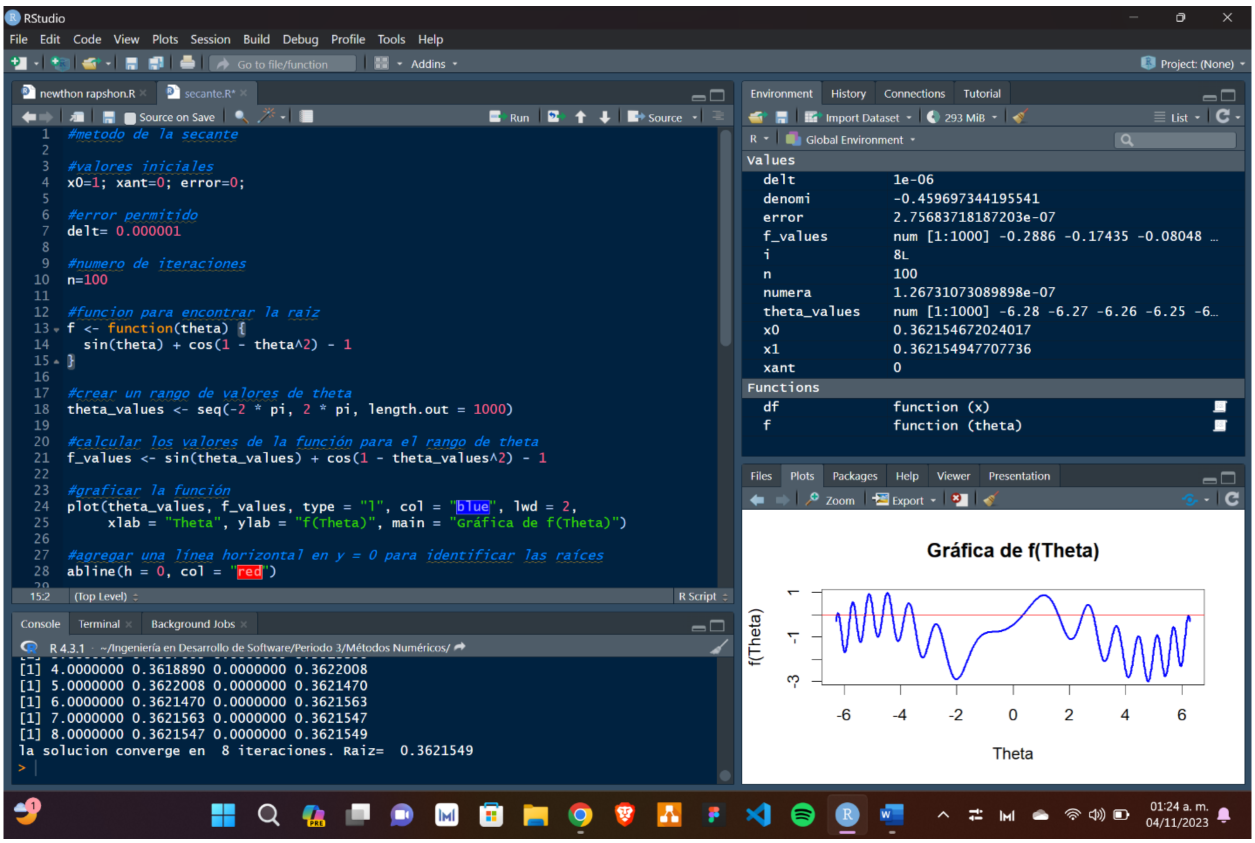
La justificación de emplear estos métodos radica en su utilidad en la vida cotidiana y en el campo laboral. Muchas situaciones reales se pueden modelar como ecuaciones no lineales, desde cálculos financieros y diseño de ingeniería hasta problemas científicos. Resolver estas ecuaciones de manera analítica a menudo es complicado o incluso imposible. Los métodos numéricos, como la secante y Newton-Raphson, ofrecen una solución eficiente para aproximarse a las soluciones de estos problemas. En el ámbito laboral, el conocimiento de estos métodos puede ser una habilidad valiosa, especialmente en matemáticas aplicadas, ciencia de datos, ingeniería y finanzas.

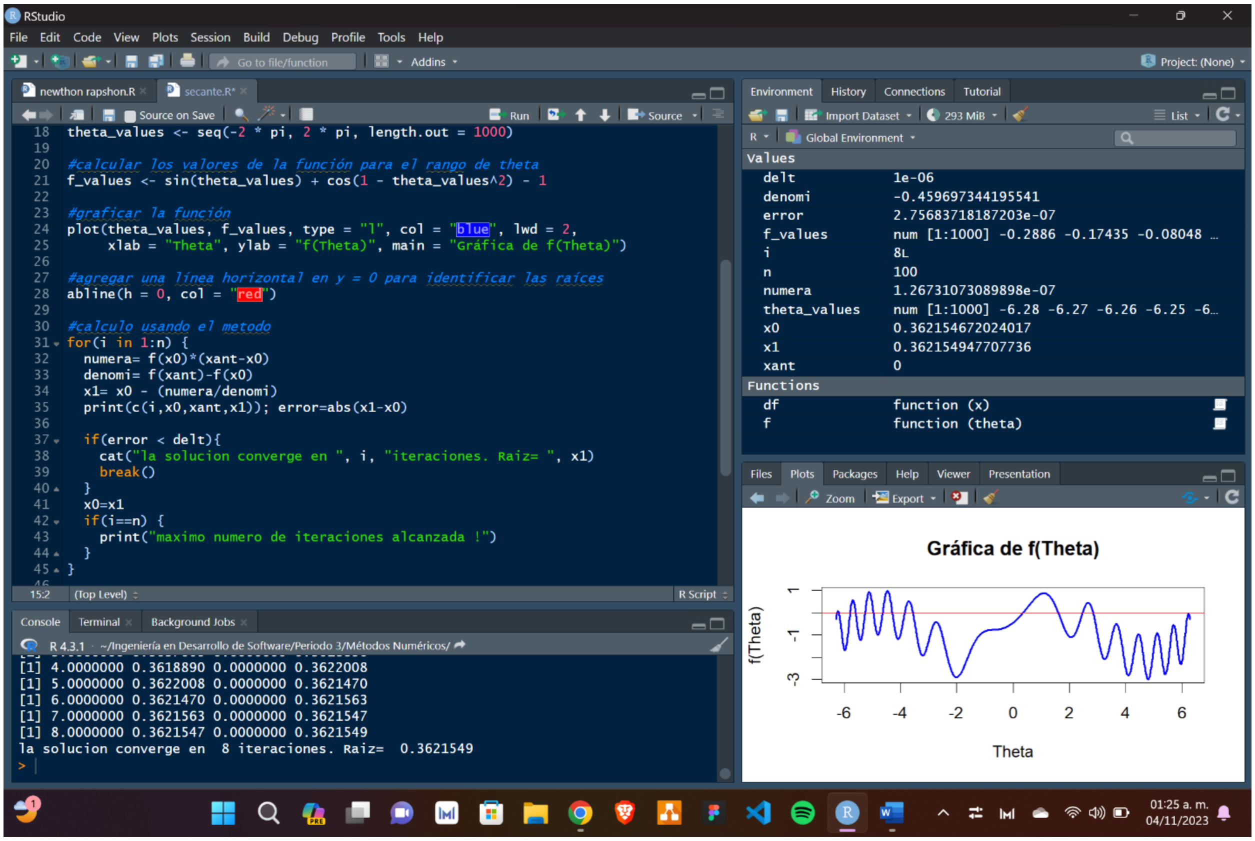
Desarrollo

Ecuación método secante

En este apartado mostraremos las capturas de la ecuación resuelta por el método de la secante.

f <- function(theta) { sin(theta) + cos(1 - theta^2) – 1 }

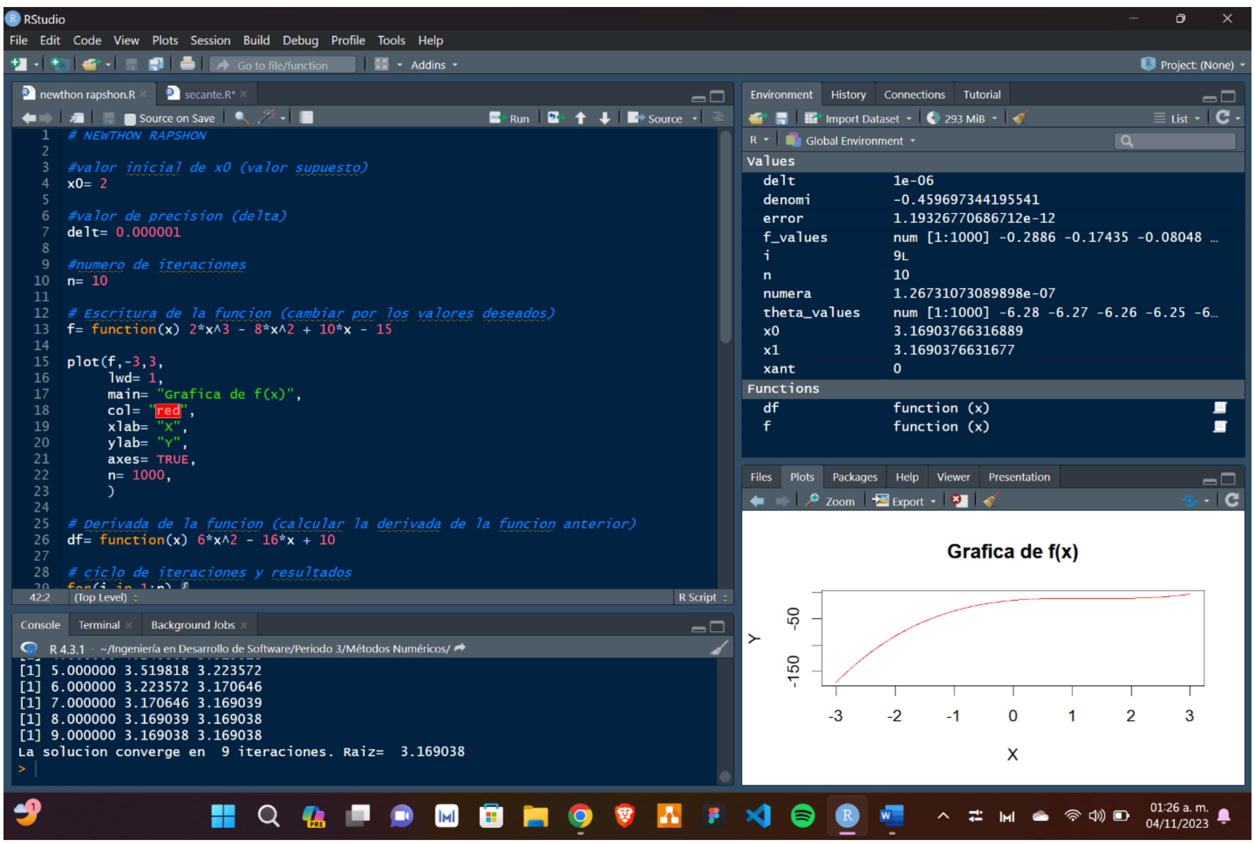


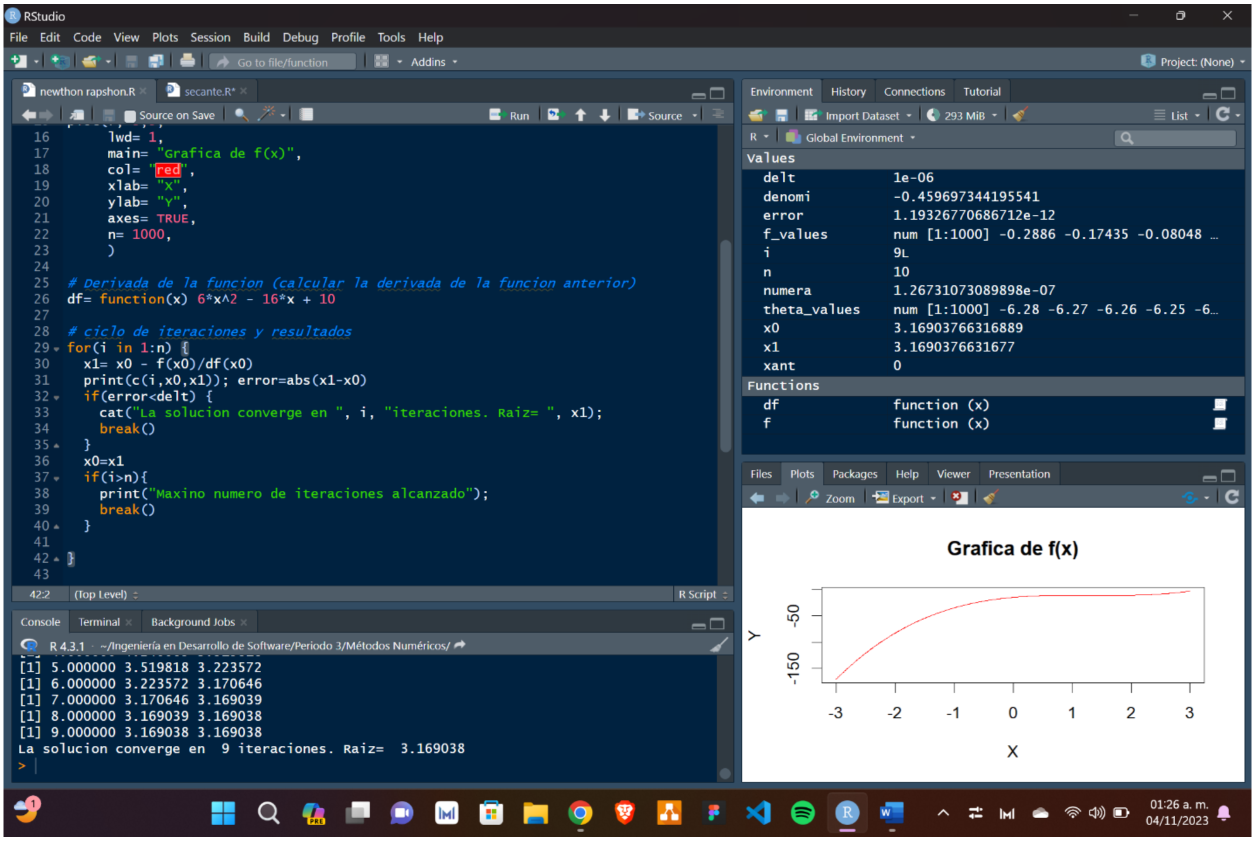

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
newthon rapshon.R
secante.R*
theta_values <- seq(-2 * pi, 2 * pi, length.out = 1000)
#calcular los valores de la función para el rango de theta
f_values <- sin(theta_values) + cos(1 - theta_values^2) - 1
#graficar la función
plot(theta_values, f_values, type = "l", col = "blue", lwd = 2,
xlab = "Theta", ylab = "f(Theta)", main = "Gráfica de f(Theta)")
#agregar una línea horizontal en y = 0 para identificar las raíces
abline(h = 0, col = "red")
#calculo usando el metodo
for(i in 1:n) {
numera= f(x0)*(xant-x0)
denomi= f(xant)-f(x0)
x1= x0 - (numera/denomi)
print(c(i,x0,xant,x1)); error=abs(x1-x0)
if(error < delt){
cat("la solucion converge en ", i, "iteraciones. Raiz= ", x1)
break()
}
x0=x1
if(i==n) {
print("maximo numero de iteraciones alcanzada !")
}
}
Console Terminal Background Jobs
R 4.3.1 · ~/Ingeniería en Desarrollo de Software/Periodo 3/Métodos Numéricos/
[1] 4.0000000 0.3618890 0.0000000 0.3622008
[1] 5.0000000 0.3622008 0.0000000 0.3621470
[1] 6.0000000 0.3621470 0.0000000 0.3621563
[1] 7.0000000 0.3621563 0.0000000 0.3621547
[1] 8.0000000 0.3621547 0.0000000 0.3621549
la solucion converge en 8 iteraciones. Raiz= 0.3621549
Environment History Connections Tutorial
Global Environment
Values
delt 1e-06
denomi -0.459697344195541
error 2.75683718187203e-07
f_values num [1:1000] -0.2886 -0.17435 -0.08048 …
i 8L
n 100
numera 1.26731073089898e-07
theta_values num [1:1000] -6.28 -6.27 -6.26 -6.25 -6…
x0 0.362154672024017
x1 0.362154947707736
xant 0
Functions
df function (x)
f function (theta)
Files Plots Packages Help Viewer Presentation
Gráfica de f(Theta)
f(Theta)
Theta
-6 -4 -2 0 2 4 6
01:25 a. m.
04/11/2023

Ecuación método Newton-Raphson

En este apartado mostramos las capturas de evidencia sobre la ecuación resuelta por el método de Newton-Raphson.

f = function(x) 2*x^3 - 8*x^2 + 10*x - 15




RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
newthon rapshon.R
secante.R*
lwd= 1,
main= "Grafica de f(x)",
col= "red",
xlab= "X",
ylab= "Y",
axes= TRUE,
n= 1000,
)
# Derivada de la funcion (calcular la derivada de la funcion anterior)
df= function(x) 6*x^2 - 16*x + 10
# ciclo de iteraciones y resultados
for(i in 1:n) {
x1= x0 - f(x0)/df(x0)
print(c(i,x0,x1)); error=abs(x1-x0)
if(error<delt) {
cat("La solucion converge en ", i, "iteraciones. Raiz= ", x1);
break()
}
x0=x1
if(i>n){
print("Maxino numero de iteraciones alcanzado");
break()
}
}
Environment History Connections Tutorial
Global Environment
Values
delt 1e-06
denomi -0.459697344195541
error 1.19326770686712e-12
f_values num [1:1000] -0.2886 -0.17435 -0.08048 …
i 9L
n 10
numera 1.26731073089898e-07
theta_values num [1:1000] -6.28 -6.27 -6.26 -6.25 -6…
x0 3.16903766316889
x1 3.1690376631677
xant 0
Functions
df function (x)
f function (x)
Console Terminal Background Jobs
R 4.3.1 · ~/Ingeniería en Desarrollo de Software/Periodo 3/Métodos Numéricos/
[1] 5.000000 3.519818 3.223572
[1] 6.000000 3.223572 3.170646
[1] 7.000000 3.170646 3.169039
[1] 8.000000 3.169039 3.169038
[1] 9.000000 3.169038 3.169038
La solucion converge en 9 iteraciones. Raiz= 3.169038
Files Plots Packages Help Viewer Presentation
Grafica de f(x)
Y
-50
-150
-3 -2 -1 0 1 2 3
X
01:26 a. m.
04/11/2023

Interpretación de resultados método secante

- Valores Iniciales: Se comienza con dos valores iniciales x0 y xant, que son dos aproximaciones iniciales en el dominio de la función.
- Error Permitido y Número de Iteraciones: El método se ejecuta hasta que el error entre dos iteraciones consecutivas sea menor que el valor permitido delt o hasta que se alcance un número máximo de iteraciones n. delt controla cuán cerca se debe estar de la raíz, y n establece un límite en el número de iteraciones.
- Función a Analizar: La función $f(\theta)=\sin(\theta)+\cos(1-\theta2)-1$ es la función que se está analizando, y el objetivo es encontrar el valor de theta para el cual esta función es igual a cero.
- Gráfica de la Función: Se crea una gráfica de la función para visualizar su comportamiento y las raíces de la función. Las raíces son los puntos en los que cruza el eje horizontal (y = 0).
- Cálculo Usando el Método de la Secante: El bucle for implementa el método de la secante. En cada iteración, se calcula un nuevo valor x1 basado en las dos iteraciones anteriores x0 y xant. El proceso se repite hasta que el error entre dos iteraciones consecutivas sea menor que el valor permitido delt.
- Resultado de Convergencia: Si el error es menor que delt, se muestra un mensaje indicando que la solución ha convergido, y se presenta la raíz encontrada junto con la cantidad de iteraciones requeridas.
- Iteraciones Máximas Alcanzadas: En caso de que se alcance el número máximo de iteraciones n sin converger, se muestra un mensaje informando que se ha alcanzado el límite máximo de iteraciones.

Interpretación de resultados método Newton-Raphson

- Valor Inicial y Precisión: Se comienza con un valor inicial x0 de 2 y una precisión delt de 0.000001, que determina cuán cerca se debe estar de la raíz.
- Número de Iteraciones Máximas: El método se ejecuta hasta que el error entre dos iteraciones consecutivas sea menor que delt o hasta que se alcance un número máximo de iteraciones n, que en este caso es 10.
- Función a Analizar: La función $f(x)$=2x3−8x2+10x−15 es la función que se está evaluando para encontrar su raíz.
- Gráfica de la Función: Se crea una gráfica de la función para visualizar su comportamiento. Esto ayuda a identificar la ubicación de la raíz.
- Derivada de la Función: La derivada de la función, $f'(x)$=6x2−16x+10, es necesaria para aplicar el método de Newton-Raphson.
- Cálculo Usando el Método de Newton-Raphson: El bucle for implementa el método de Newton-Raphson. En cada iteración, se calcula un nuevo valor x1 utilizando la fórmula del método. Se verifica el error entre las iteraciones y se compara con la precisión delt.
- Resultado de Convergencia: Si el error es menor que delt, se muestra un mensaje que indica que la solución ha convergido y se presenta la raíz encontrada junto con la cantidad de iteraciones requeridas.
- Iteraciones Máximas Alcanzadas: Si no se alcanza la convergencia dentro de las 10 iteraciones, se muestra un mensaje que informa que se ha alcanzado el límite máximo de iteraciones.

Conclusión

La aplicación de los métodos de la secante y Newton-Raphson es esencial en el análisis numérico y en la resolución de problemas que involucran ecuaciones no lineales. A través de esta actividad, hemos podido experimentar su utilidad y eficacia al resolver ecuaciones específicas. Estos métodos permiten obtener soluciones aproximadas con relativa facilidad, lo que puede ser fundamental en la toma de decisiones en el mundo laboral y en la vida cotidiana. El uso de RStudio y la comprensión de estos métodos proporcionan una valiosa base para abordar problemas matemáticos y científicos que requieren soluciones numéricas. En un entorno laboral cada vez más impulsado por los datos, estas habilidades pueden marcar la diferencia en la resolución de problemas complejos. En resumen, esta actividad nos ha brindado una sólida introducción a herramientas matemáticas esenciales que tienen aplicaciones significativas en diversos campos.

Referencias

Link del archivo en GitHub.