Actividad 2 – Método de secante y Newton.

Métodos Numéricos.

Ingeniería en Desarrollo de Software

Tutor: Miguel Ángel Rodríguez Vega.

Alumnos: Carlos Ariel Nicolini.

Fecha: 22/10/2023

Índice

Introducción

Los métodos numéricos son un procedimiento mediante el cual se puede obtener de manera aproximada una solución realizando cálculos aritméticos y lógicos. El procedimiento consiste en una lista de instrucciones que especifican una secuencia de operaciones algebraicas y algoritmos, que resultan en una aproximación de la solución de dicho problema. El resultado dependerá del algoritmo utilizado y las herramientas con las que se resuelta. El objetivo principal del análisis numérico es encontrar soluciones aproximadas para problemas complejos.

En este ejercicio aplicaremos dos de esos métodos: El método de Newton-Raphson y de la secante

Método de Newton-Raphson:

Este método es un algoritmo para encontrar aproximaciones de los ceros o raíces de una función real. También puede ser usado para encontrar el máximo o mínimo de una función, encontrando los ceros de su primera derivada.

Este es un método abierto. La única manera de alcanzar la convergencia es seleccionar un valor inicial lo suficientemente cercano a la raíz buscada. Así, se ha de comenzar la iteración con un valor razonablemente cercano al cero. La relativa cercanía del punto inicial a la raíz depende mucho de la naturaleza de la propia función; si esta presenta múltiples puntos de inflexión o pendientes grandes en el entorno de la raíz, entonces las probabilidades de que el algoritmo diverja aumentan, lo cual exige seleccionar un valor supuesto cercano a la raíz. Una vez se ha hecho esto, el método linealiza la función por la recta tangente en ese valor supuesto. La abscisa en el origen de dicha recta será, según el método, una mejor aproximación de la raíz que el valor anterior. Se realizaran sucesivas iteraciones hasta que el método haya convergido lo suficiente.

Método de la secante:

Este método es para encontrar los ceros de una función de forma iterativa.

Es una variación del método de Newton-Raphson donde en vez de calcular la derivada de la función en el punto de estudio, teniendo en mente la definición de derivada, se aproxima la pendiente a la recta que une la función evaluado en el punto de estudio y en el punto de la iteración anterior. Este método es de especial interés cuando el coste computacional de derivar la función de estudio y evaluarla es demasiado elevado.

Descripción

En esta actividad utilizaremos el programa Rstudio para interpretar dos archivos con instrucciones, donde pondremos en práctica y resolveremos dos ecuaciones, una con el método de la secante y la otra con el método de Newton-Raphson.

Descargaremos dos archivos con el lenguaje R (Archivo Met_Secante.R y Met_Newton2.R), ejecutaremos dichos método con las siguientes funciones:

Para el método de la secante:

$f(\theta) = \sin(\theta) + \cos(1 - \theta 2) - 1$

Para el método de Newton-raphson:

$f(x) = 2x 3 - 8x 2 + 10x - 15$

Realizaremos dichas ecuaciones en el programa Rstudio e interpretaremos sus resultados.

Justificación

Esta actividad es complemento y continuación de la actividad anterior. Seguiremos haciendo uso del programa Rstudio y de los métodos numéricos para llegar a encontrar una solución a dos ecuaciones.

En este caso en particular haremos uso de los métodos de la secante y de Newton-Raphson para resolver dos ecuaciones con unas funciones específicas para cada método, funciones que nos son proporcionadas en el mismo documento.

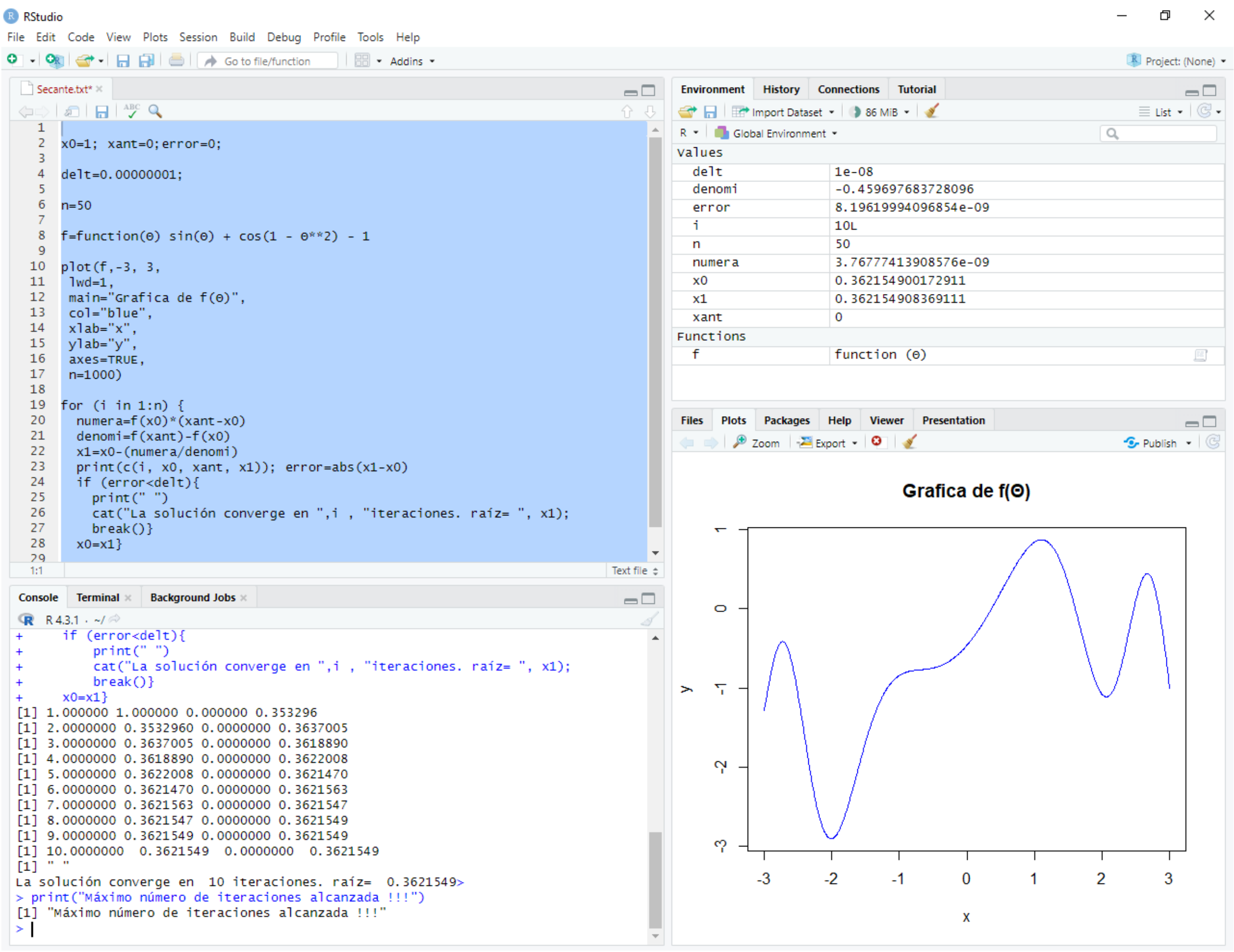
Este ejercicio me gusto por la explicación de la clase, el uso del Rstudio el cual me estoy familiarizando, es una herramienta muy noble y a la cual estoy aprendiendo a usar.

Espero que dicha presentación sea del agrado y que cumpla con las expectativas deseadas.

Desarrollo

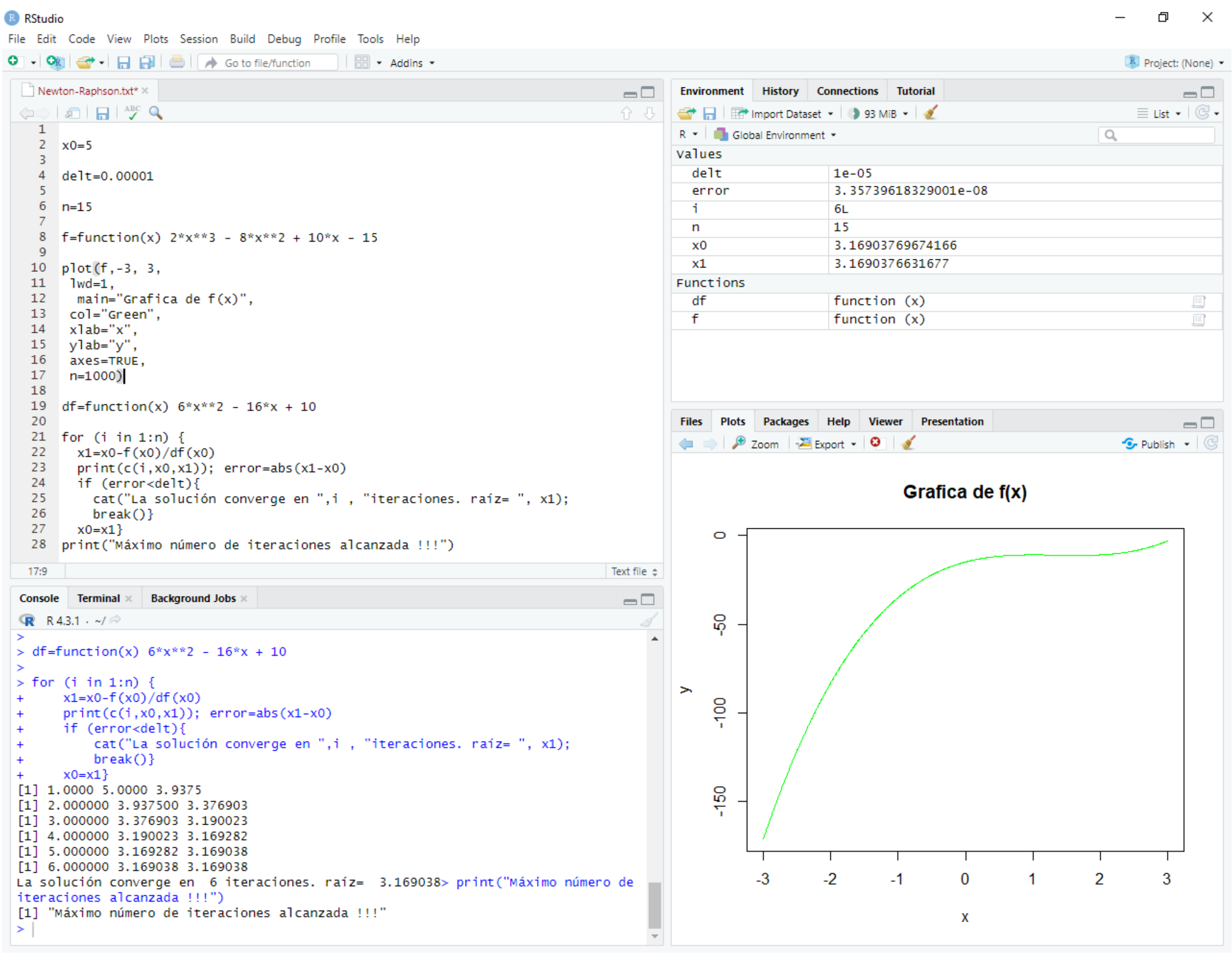
Ecuación de método Secante

En esta ecuacion que debemos resolver con el metodo de la secante, se descargo el archivo, se realizo la ecuacion con la funcion f(θ) = sin(θ) + cos(1 − θ 2) – 1, se adjunta la evidencia de su ejecución y resultado.



Ecuación método Newton-Raphson

En esta ecuacion que debemos resolver con el metodo de Newton-Raphson, se descargo el archivo, se realizo la ecuacion con la funcion $f(x) = 2x\ 3 - 8x\ 2 + 10x - 15$, se adjunta la evidencia de su ejecución y resultado.



Interpretación de resultados

Pasamos a describir la ecuaciones con su solucion de acuerdo a el metodo aplicado

Metodo de la secante:

Metodo de Newton-Raphson:

x0=1; xant=0;error=0; La variable inicial (x0) donde le asignamos un valor en esta caso igual a 1, xant es igual a 0 (x anterior) y el error empezamos en 0.

delt=0.00000001; Variable donde definimos el valor de la presicion (en este caso se define como Delta) o error permitido donde nuestro programa va a parar.

n=50 En esta variable (n) definimos el numero de iteraciones en este caso 15.

f=function(Θ) sin(Θ) + cos(1 - Θ2) – 1** En la variable f definimos la funcion (function(Θ)) proporcionada en el ejercicio. Θ es la variable que vamos a estar manipulando.

plot(f,-3, 3, Usamos la funcion plot para graficar la variable f y le asignamos en valor mas bajo y el valor mas alto de la grafica.

lwd=1, Aquí definimos el grosor de la linea en 1.

main="Grafica de f(Θ)", Definimos el nombre de la grafica.

col="blue", Definimos su color en rojo.

xlab="x", La etiqueta del eje x.

ylab="y", La etiqueta del eje y.

axes=TRUE, Para ver los ejes este campo debe ponerse en true.

n=1000) Se define el numero de puntos que formaran nuestra grafica.

```
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){
  print(" ")
  cat("La solución converge en ",i , "iteraciones. raíz= ", x1);
  break()}
x0=x1}
```

print("Máximo número de iteraciones alcanzada !!!") Usamos un ciclo for para realizar la iteraciones, las cuales van a comenzar en uno hasta encontrar el resultado. Realizara un maximo de 50 iteraciones. Se utiliza el metodo de la secante en la funcion para calcular el resultado, si dicho resultado se encuentra nos indicara en que iteracion fue hallado y su resultado el cual es 0.3621549 y se encontro en la decima iteracion.

Si transcurren las 50 iteraciones y no encuentra el valor, nos indicara que el numero de iteraciones total fue alcanzado, lo cual nos indica que debemos aumentarlo para poder encontrar la raiz.

La grafica nos muestra que hay multiples soluciones para Θ.

Metodo de Newton-Raphson:

x0= 5 La variable inicial donde le asignamos un valor en esta caso igual a 5.

delt=0.00001 Variable donde definimos el valor de la presicion (en este caso se define como Delta) o error permitido donde nuestro programa va a parar.

n=15 En esta variable (n) definimos el numero de iteraciones en este caso 15.

f=function(x) 2*x3 - 8*x**2 + 10*x – 15** En la variable f definimos la funcion (function(x)) proporcionada en el ejercicio. X es la variable que vamos a estar manipulando.

plot(f,-3, 3, Usamos la funcion plot para graficar la variable f y le asignamos en valor mas bajo y el valor mas alto de la grafica.

lwd=1, Aquí definimos el grosor de la linea en 1.

main="Grafica de f(x)", Definimos el nombre de la grafica.

col="red", Definimos su color en rojo.

xlab="x", La etiqueta del eje x.

ylab="y", La etiqueta del eje y.

axes=TRUE, Para ver los ejes este campo debe ponerse en true.

n=1000) Se define el numero de puntos que formaran nuestra grafica.

df=function(x) 6*x2 - 16*x + 10** En esta variable definimos la derivada de la funcion (2*x**3 - 8*x**2 + 10*x – 15).

```
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 solución converge en ",i , "iteraciones. raíz= ", x1);
```

break()}

x0=x1}

print("Máximo número de iteraciones alcanzada !!!") Usamos un ciclo for para realizar la iteraciones, las cuales van a comenzar en uno hasta encontrar el resultado. Realizara un maximo de 15 iteraciones. Se utiliza el metodo de Newton-raphson en la funcion para calcular el resultado, si dicho resultado se encuentra nos indicara en que iteracion fue hallado y su resultado el cual es 3.169038 y se encontro en la sexta iteracion.

Si transcurren las 15 iteraciones y no encuentra el valor, nos indicara que el numero de iteraciones total fue alcanzado, lo cual nos indica que debemos aumentarlo para poder encontrar la raiz.

La grafica nos muestra que hay una soluciones para x.

El ejercicio realizado fue subido al repositorio de github en el siguiente enlance

https://github.com/CarlosNico/M-todos-Num-ricos

Conclusión

El uso del método de la secante y el de Newton-Raphson son métodos muy efectivos que nos van a permitir encontrar una solución aproximada, lo cual se va a traducir en un ahorro de tiempo, esfuerzo y dinero.

La aplicación en práctica fue muy interesante y educativo, no solo me ayudo a seguir aprendiendo en el uso del Rstudio y el manejo del lenguaje R, sino en el uso de algoritmos además en el entendimiento y uso de estos dos métodos numéricos.

Son métodos que destacan por su simplicidad, convergencia razonable, menos costosos computacionalmente, precisión (mas el método de Newton).

.

Referencias

Wikipedia contributors. (n.d.). *Método de la secante*. Wikipedia, The Free Encyclopedia. https://es.wikipedia.org/w/index.php?title=M%C3%A9todo_de_la_secante&oldid=148607107

Wikipedia contributors. (n.d.-b). *Método de Newton*. Wikipedia, The Free Encyclopedia. https://es.wikipedia.org/w/index.php?title=M%C3%A9todo_de_Newton&oldid=149567022

(N.d.). Edu.Pe. Retrieved October 27, 2023, from https://www.urp.edu.pe/pdf/id/2555/n/app#:~:text=M%C3%A9todo%20de%20la%20secante&text=Es%20una%20variaci%C3%B3n%20del%20m%C3%A9todo,punto%20de%20la%20iteraci%C3%B3n%20anterior.

Métodos Numéricos 3: Raíces de ecuaciones: Métodos de Newton-Raphson y de la secante. (n.d.). Ulpgc.Es. Retrieved October 27, 2023, from https://estadistica-dma.ulpgc.es/FCC/05-3-Raices-de-Ecuaciones-2.html