Zaira Carolina Martinez Vargas

Tarea 2: Método de Newton y método de Bisección

2022-08-29

Resumen

Se implementó el método de bisección y de Newton-Raphson para la búsqueda de raíces de una ecuación, en específico, se utilizaron para encontrar las raíces de las funciones: $f(x) = ax^3 - 21x^2 + 120x - 100$ donde a tiene valores de 1, 1.01 y 0.99; $g(x) = 2 - x^{-1}In(x)$; y $h(x) = In(x^2 + 1) - e^{0,4x}cos(\pi x)$.

1. Introducción

La búsqueda de raíces consiste en obtener una raíz x de una ecuación de la forma f(x) = 0 para una función dada f. Supongamos que f es una función continua definida en el intervalo [a, b] con f(a) y f(b) de signos diferentes.

2. Metodología

Sea f continua definida en el intervalo $[a, b]$ tal que $f(a)f(b) < 0$, entonces de acuerdo al teorema del valor intermedio se puede aseverar la existencia de un número p en (a, b) tal que $f(p) = 0$. Confirmada la existencia de dicho valor p, es al que se conoce como raíz de la función.

2.1. Método de bisección

El método requiere dividir varias veces a la mitad los subintervalos de $[a, b]$ y, en cada paso, localizar la mitad que contenga p. Para empezar supongamos que a_1 = a y b_1 = b, sea p_1 el punto medio de $[a, b]$; es decir,

$$p_1 = \frac{1}{2}(a_1 + b_1).$$

Si $f(p_1) = 0$, entonces $p = p_1$; de no ser así, entonces $f(p_1)$ tiene mismo signo que $f(a_1)$ o $f(b_1)$. Si $f(p_1)$ y $f(a_1)$ tienen el mismo signo, entonces $p \in (p_1, b_1)$, y tomamos $a_2 = p_1$ y $b_2 = b_1$. Si $f(p_1)$ y $f(a_1)$ tienen signos opuestos, entonces $p \in (a_1, p_1)$ y tomamos $a_2 = a_1$ y $b_2 = p_1$. Después volvemos a aplicar el proceso al intervalo $[a_2, b_2]$. Veremos el pseudo-código para este método [1].

Algorithm 1 Bisección

Entrada: Extremos a, b; tolerancia TOL; número máximo de iteraciones N_0
Salida: Solución aproximada p o mensaje de fracaso

1: Tome $i = 1$;
   $FA = f(a)$
2: Mientras $i \leq N_0$ haga pasos 3-6 {
3: Tome $p = a + (b - a)/2$;
   $FP = f(p)$.
4: Si $FP = 0$ o $(b - a)/2 < TOL$ entonces
   SALIDA: p ,
   Pare.

5: i = i + 1
6: Si FA*FP ¿0 entonces tome a = p; FA = FP
   si no tome b=p. }
7: SALIDA: (el método fracaso después de N_0 iteraciones)
   Pare.

2.2. Método de Newton-Raphson

Esté método es uno de los más poderosos conocidos para resolver raices de ecuaciones. Supongamos que $f \in C^2[a, b]$. Sea $\bar{x} \in [a, b]$ una aproximación de p tal que $f'(x) \neq 0$ y $|\bar{x} - p|$ es pequeño. Consideremos el primer pokinomio de Taylor para $f(x)$ expandida alrededor de x,

$$f(x) = f(\bar{x}) + (\bar{x} - x)f'(\bar{x}) + \frac{(x - \bar{x})^2}{2} f''(\xi(x)),$$

donde ξ está entre x y $\bar{x}$. Dado qu e $f(p)=0$, da

$$0 = f(\bar{x}) + (p-\bar{x})f'(\bar{x}) + \frac{(p-\bar{x})^2}{2} f''(\xi(p))$$

Derivamos el método de Newton suponiendo que, como $|p-\bar{x}|$ es tan pequeño, el término que contiene $(p-\bar{x})$ es mucho menor y que

$$0 \approx f(\bar{x}) + (p-\bar{x})f'(\bar{x}).$$

Despejando p de esta ecuación obtenemos

$$p \approx \bar{x} - \frac{f(x)}{f'(x)}.$$

Esto nos prepara para introducir el método de Newton-Raphson, el cual comienza con una aproximación inicial p_0, y genera la sucesión $\{p_0\}$ definida por

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}, \quad \text{para } n \geq 1.$$

Algorithm 2 Método Newton-Raphson

Entrada: Aproximación inicial p_0; tolerancia TOL; número máximo de iteraciones N_0
Salida: Solución aproximada de p o mensaje de fracaso
1: Tome $i = 1$.
2: Mientras $i \leq N_0$ haga pasos 3-6 {
3: Tome $p = p_0 - \frac{f(p_0)}{f'(p_0)}$;
4: Si $|p - p_0| < TOL$ entonces
SALIDA: p ,
Pare.

5: i = i + 1
6: Tome $P_0 = p$. }
7: SALIDA: (el método fracaso después de N_0 iteraciones)
Pare.

3. Resultados

Se graficaron todas las funciones con el objetivo de saber si en un intervalo dado [a,b] existían raíces y observar su continuidad. Además, los intervalos donde se buscarán las raíces están dados por el usuario, al igual que la cantidad de iteraciones.

Función	Raíz real	Raíz real	Raíz real	Bisección	Newton	Error relativo Bisección	Error relativo Newton
f(x), a= 0.99	1.00	9.043	11.1707	0.5,10,12	1.17012, NE, NE	0.5, 0.1058, 0.074	0.17012
f(x), a= 1.00	1	10		1.1699, 10.5	1.1699, NE		
f(x), a=1.01	0.999			1.16989	1.16989	0.1710	0.1710
g(x)	No aplica						
h(x)	3.7090	9.55	12.82	3.5, 10.4773, 15.5035	3.70, 9.53183, 14.4948	0.05, 0.097, 0.2093	0.0024, 0.1306

Cuadro 1: Resultados para algunos valores de x.

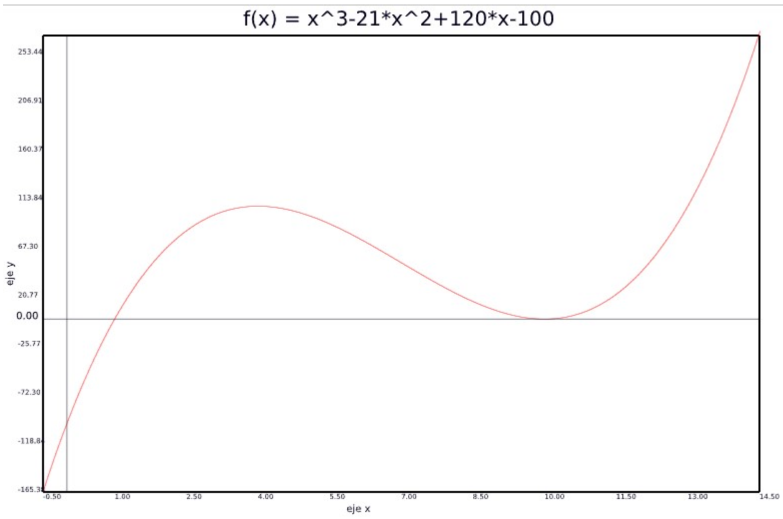

Figura 1: f(x) para a = 1.0

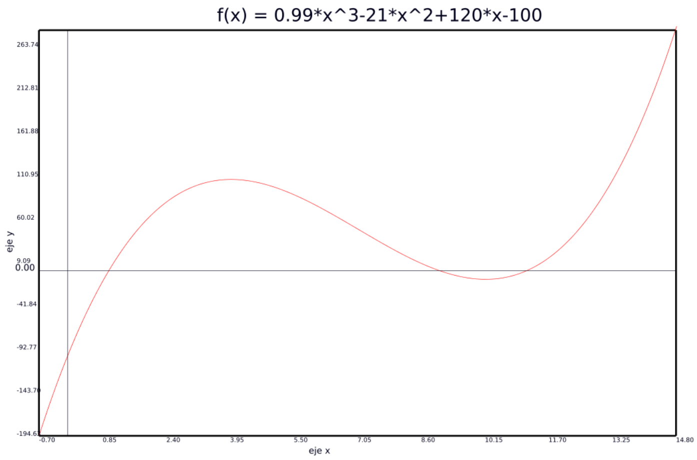

Figura 2: f(x) para a = 0.99

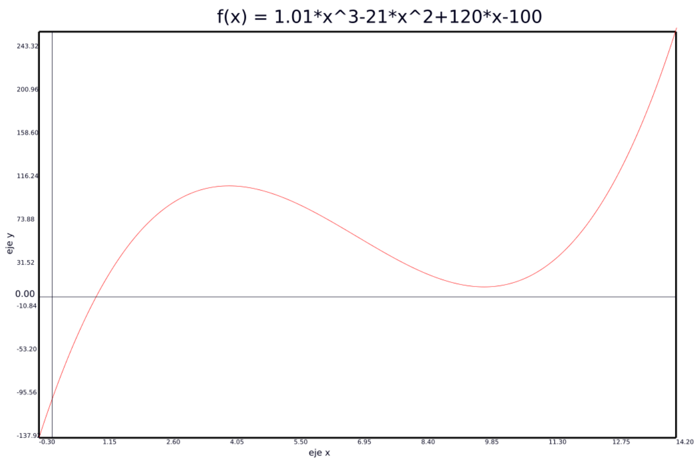

Figura 3: f(x) para a = 1.01

Como podemos ver de la fig. 4, la función g(x) no tiene raíces reales.

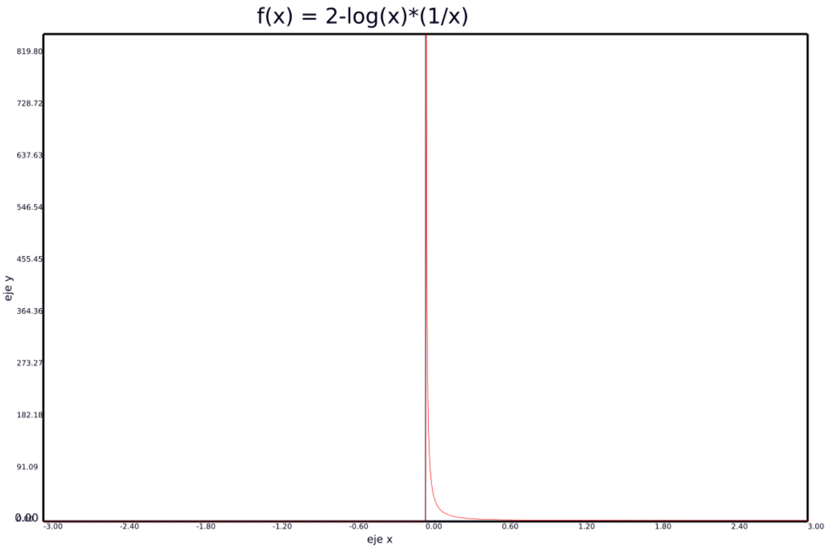

Figura 4: g(x)

En cambio, la función h(x) tiene muchas raíces, por ello, es que el intervalo es importante.

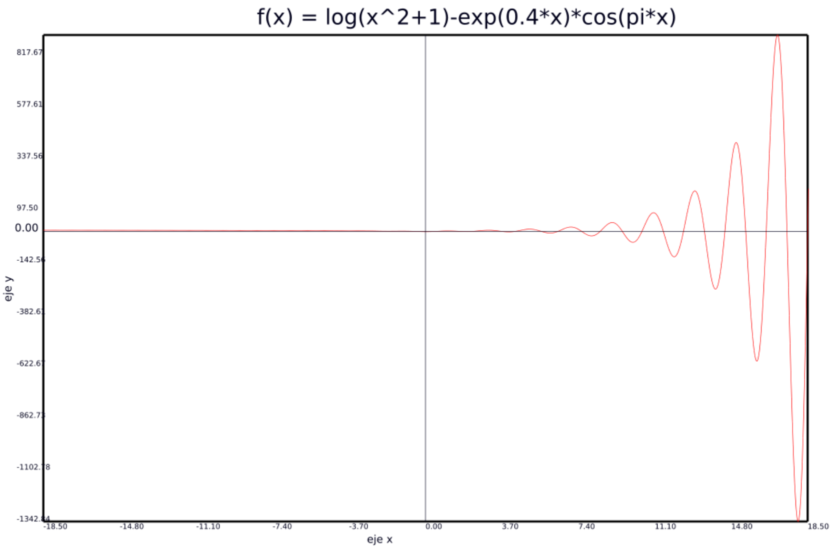

Figura 5: h(x)

Además, como se mencionó anteriormente, el usuario decide sus intervalos y cantidad de iteraciones (como se muestra en la fig.6).

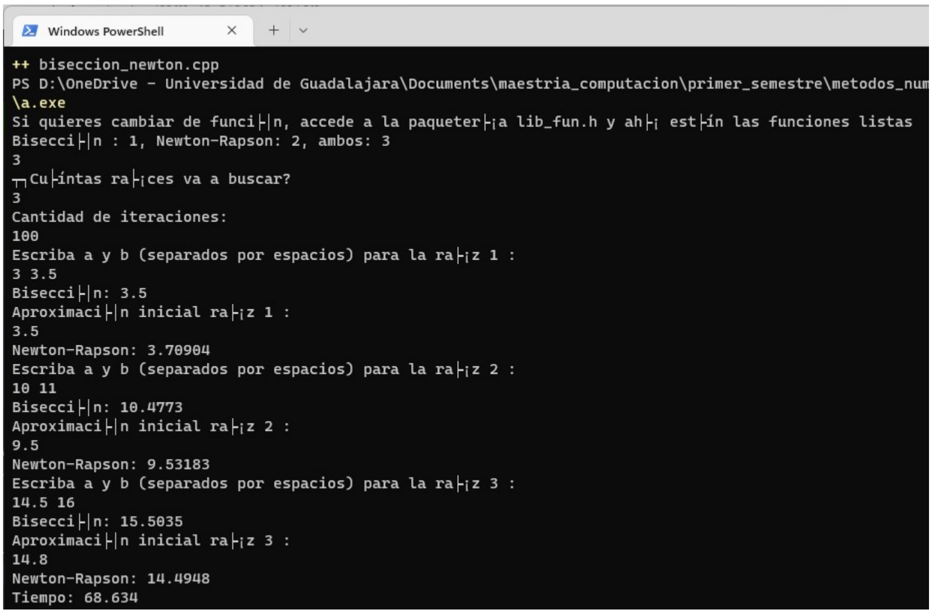

Figura 6: Encontrar raíces de la función h(x)

4. Conclusiones

Una de las ventajas que tienen estos métodos, es su facilidad de implementación, sólo es necesario conocer un poco sobre los ciclos y los condicionales para implementarlos. Sin embargo, como se observó en la tabla, el método de Newton no funciona bien en ciertos intervalos, aunque se intentaba cambiar de intervalo para mejorar la aproximación, la pendiente era un punto importante. Además, se observó que aunque se aumentara el número de iteraciones, no mejoraba mucho el error, como se mejora para otros métodos, teniendo un error relativo muy alto cuando no se escogian adecuadamente los intervalos de estudio.

Referencias

[1] R. Burden, J. Faires, and A. Burden, *Numerical Analysis*. Cengage Learning, 2015. [Online]. Available: https://books.google.com.mx/books?id=9DV-BAAAQBAJ