Pontificia Universidad Javeriana

Facultad de Ingenieria

Problema de las raíces: Método Newton-Raphson

Miguel Angel Ávila Santos
Juan Andres Martínez Amado
Juan Sebastián Herrera Guaitero
Jorge Luis Esposito Albornoz

Agosto 2021

1. Introducción

El método de Newton es una fórmula iterativa eficiente para encontrar la raíz de una función dada. Es un caso especial del método del punto fijo en el que la ecuación f(x) = 0 se reescribe en la forma x = g(x) eligiendo g de tal manera que la convergencia sea de segundo orden.

2. Ejercicios

1. $f(x) = \cos(x)^2 - x^2$
2. $f(x) = e^x - x - 1$
3. $f(x) = x^3 - 2x^2 + \frac{4}{3x} - \frac{8}{27}, x^3 - 2x - 5 = 0$ en particular, esta última ecuación tiene un valor histórico. Fue la ecuación que usó John Wallis para presentar por primera vez el método de Newton a la academia francesa de ciencias en el siglo XV.
4. Determinar el coeficiente de arrastre W necesario para que un paracaidista de masa m=68.1 kg tenga una velocidad de 40 m/s después de una caída libre de t=10 s.
5. Sean $f(t) = 3\sin^3(t) - 1; g(t) = 4\sin(t)\cos(t); t \geq 0$ las ecuaciones paramétricas que describe el movimiento en dos partículas en un mismo plano de coordenadas.Determinar el instante t donde coinciden y muestre gráficamente la solución

3. Gráficas de las funciones

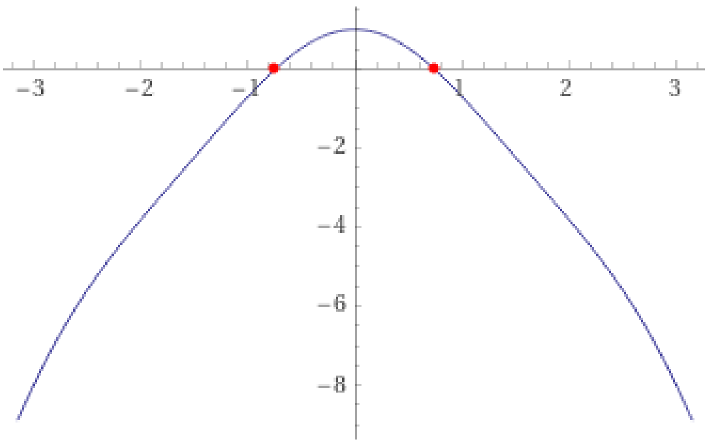


Figura 1: Gráfica $f(x) = \cos(x)^2 - x^2$

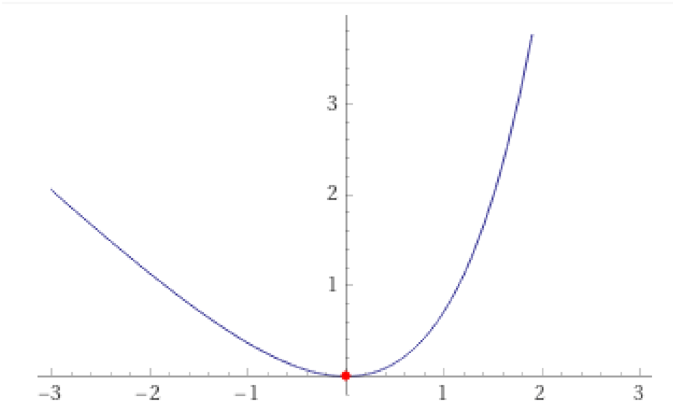


Figura 2: Gráfica $f(x) = e^x - x - 1$

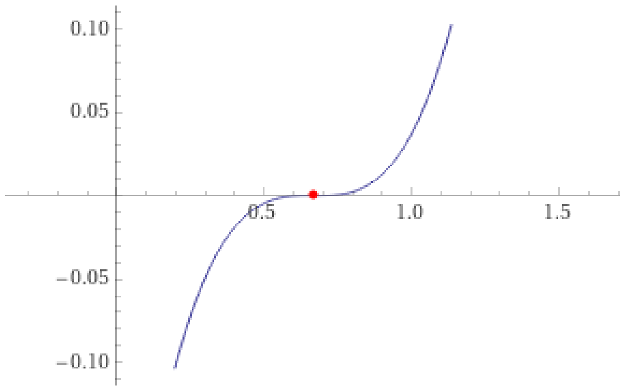


Figura 3: Gráfica $f(x) = x^3 - 2x^2 + \frac{4}{3x} - \frac{8}{27}$

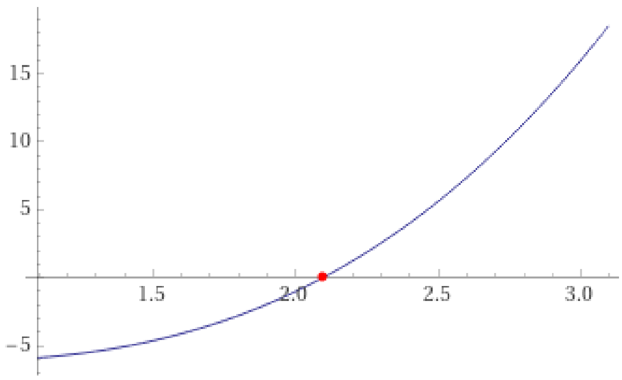


Figura 4: Gráfica $f(x) = x^3 - 2x - 5 = 0$

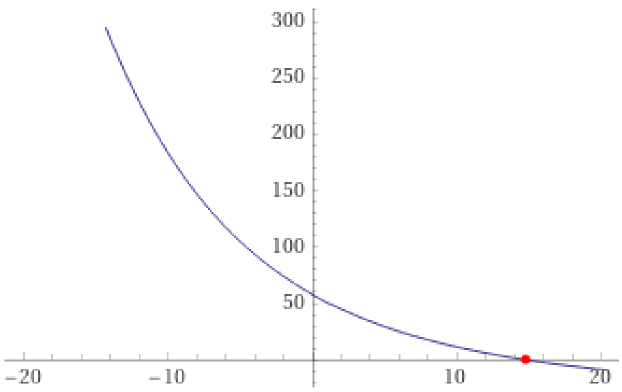


Figura 5: Gráfica $f(x) = \frac{667{,}38}{x}(1 - e^{-0{,}146843x}) - 40$

Para encontrar la solución de este problema se hizo uso de la siguiente función:

$f(c) = \frac{gm}{c}(1 - e^{-\frac{c}{m}t} - v)$, sustituyendo c por x, tomando el valor de gravedad como 9.8 y reemplazando la masa y el tiempo ya dados. Así encontrando la raíz de está función se encontrará a su vez el coeficiente de arrastre del problema.

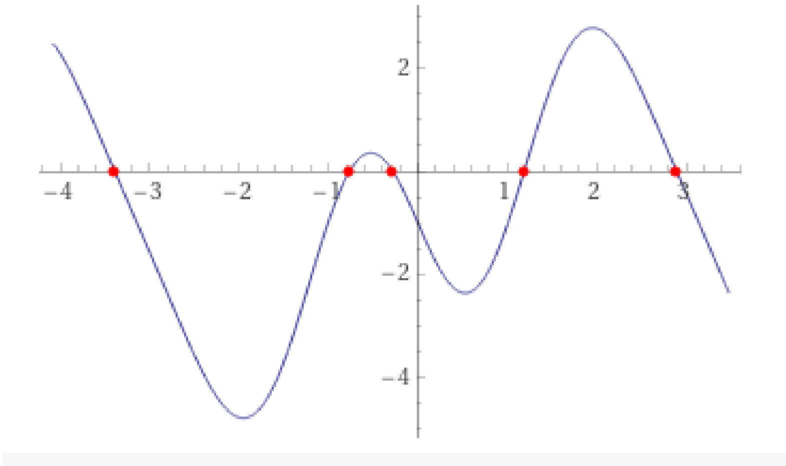


Figura 6: Gráfica $f(x) = 3\sin(x)^3 - 1 - 4\sin(x)\cos(x)$

Para encontrar la solución de este problema se igualaron las funciones dadas, así al encontrar la raíz de está función a su vez se encontraría el instante t donde coninciden ambas particulas.

4. Preguntas

1. Cuales son condiciones para aplicar el método y señalar donde quedan implementadas en su codificación.

   La condiciones para aplicar el método de Newton-Raphson son que la función sea continua y derivable al menos dos veces y que cumpla que dado un punto x en el que f(x)=0, f'(x) != 0 , asi el metodo de Newton es local y cuadráticamente convergente.

   Implementación en codigo:

```
 def NewtonRaphson (x0, tol, n):
     x = sp.symbols('x') #Crea la variable x
     f= input('Digite') #Donde f es una funcion continua
     # Validar covergencia
     df = sp.diff(f) #Calcula la derivada de f, df no puede ser 0
     d2f = sp.diff(df) #Calcula la segunda derivada de f
     f=sp.lambdify(x,f)
     df = sp.lambdify(x,df)
     if(d2f == 0):
         print('No se puede realizar el metodo de NewtonRaphson porque la funcion no es derivable al menos dos veces')
     return
     if(f(x0) == 0 and df(x0) == 0):
         print('No se puede realizar el metodo de NewtonRaphson porque la funcion no es cuadraticamente convergente en el valor inicial')
     return

```

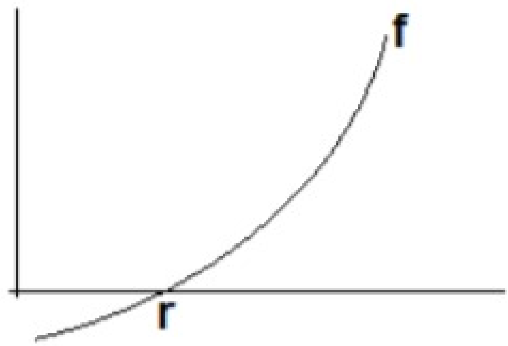
Code Listing 1: Newton Raphson

2. Proporcione una explicación geométrica del algoritmo.

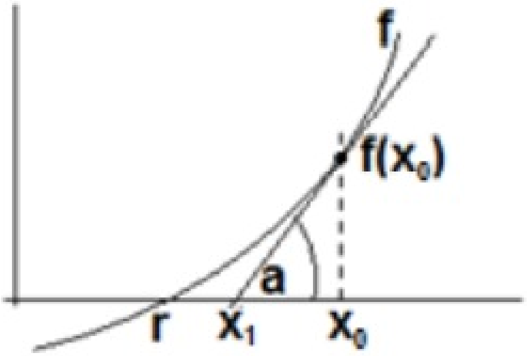
   Sean:

   f(x)=0: Ecuación de la que se pretende calcular una raíz real r
   Se supondrá que f es diferenciable y tiene forma creciente
   x_0 : Valor inicial elegido para calcular la raíz r
   x_1 : Valor calculado, más cercano a r que el valor inicial x_0

Para calcular x_1 se utilizará un procedimiento geométrico, considerando que f tiene la forma descrita en el siguiente gráfico:



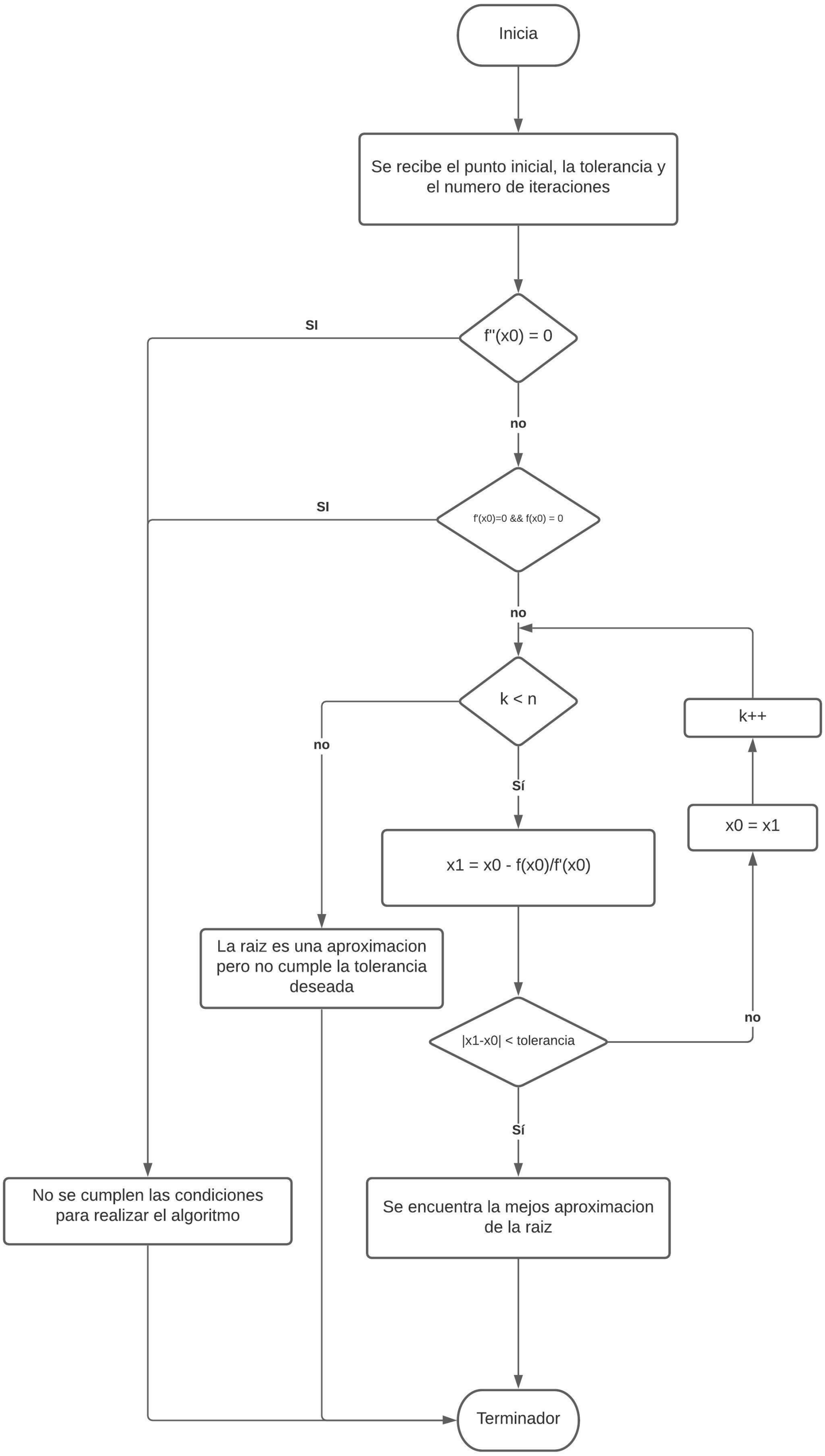
Se elige el valor inicial x_0. Se traza una tangente a f en el punto $(x_0, f(x_0))$. El punto de intersección de la recta con el eje horizontal estará más cerca de r, y se lo designa x_1:



Si a es el ángulo de intersección, el valor de x1 se lo puede encontrar con la definición: $tan(a) = f'(x_0) = \frac{f(x_0)}{x_0 - x_1} \rightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$

Esta fórmula iterativa se puede utilizar repetidamente, tomando como nuevo valor inicial el valor x_1 para calcular x_2. Se generará una secuencia de valores que esperamos converja a la raíz r.

3. Realice un diagrama de flujo que muestre como se debe operar el algoritmo.



4. Aplicar el método y determinar la solución incluyendo la validación del resultado.

Resultados

Tolerancia $10E-08$

Función	N.Iteraciones	Resultado	Validación(Wolfram)
$f(x) = \cos(x)^2 - x^2$	5	0.73908513	0.73908513321516064166
$f(x) = e^x - x - 1$		No converge	0
$f(x) = x^3 - 2x^2 + 4/3x - 8/27$	33	0.66667086	0.66666666666667
$f(x) = x^3 - 2x - 5 = 0$	5	2.09455148	2.09455148154233
$f(x) = (667{,}38/x)(1 - exp(-0{,}146843x)) - 40$	4	14.78020859	14.7802085936794677513..
$f(x) = 3sin(x)^3 - 1 - 4sin(x)cos(x)$	3	2.89265125	2.89265124527496..

Tolerancia $10E-16$

Función	N.Iteraciones	Resultado	Validación (Wolfram)
$f(x) = \cos(x)^2 - x^2$	6	0.7390851332151607	0.7390851332151606
$f(x) = e^x - x - 1$		No converge	0
$f(x) = x^3 - 2x^2 + 4/3x - 8/27$	33	0.6666708672256998	0.66666666666667
$f(x) = x^3 - 2x - 5 = 0$	6	2.0945514815423265	2.09455148154233
$f(x) = (667{,}38/x)(1 - exp(-0{,}146843x)) - 40$	5	14.780208593679466	14.780208593679467..
$f(x) = 3sin(x)^3 - 1 - 4sin(x)cos(x)$	3	2.89265124527495	2.89265124527496..

Tolerancia $10E-32$

Función	N.Iteraciones	Resultado	Validación(Wolfram)
f(x) = $\cos(x)^2 - x^2$	6	0.7390851332151607	0.7390851332151606
$f(x) = e^x - x - 1$		No converge	0
$f(x) = x^3 - 2x^2 + 4/3x - 8/27$	33	0.6666708672256998	0.66666666666667
$f(x) = x^3 - 2x - 5 = 0$	6	2.0945514815423265	2.09455148154233
$f(x) = (667{,}38/x)(1 - exp(-0{,}146843x)) - 40$	5	14.780208593679466	14.780208593679467..
$f(x) = 3sin(x)^3 - 1 - 4sin(x)cos(x)$	3	2.89265124527495	2.89265124527496..

Tolerancia $10E-56$

Función	N.Iteraciones	Resultado	Validación(Wolfram)
$f(x) = \cos(x)^2 - x^2$	6	0.7390851332151607	0.7390851332151606
$f(x) = e^x - x - 1$		No converge	0
$f(x) = x^3 - 2x^2 + 4/3x - 8/27$	33	0.6666708672256998	0.66666666666667
$f(x) = x^3 - 2x - 5 = 0$	6	2.0945514815423265	2.09455148154233
$f(x) = (667{,}38/x)(1 - exp(-0{,}146843x)) - 40$	5	14.780208593679466	14.780208593679467..
$f(x) = 3sin(x)^3 - 1 - 4sin(x)cos(x)$	3	2.89265124527495	2.89265124527496..

Los resultados anteriores reflejan el desarrollo de cada problema en el algoritmo de Newton-Raphson, con su respectivo número de iteraciones y raíz aproximada para cada tolerancia además de la validación de esta última utilizando Wolfram, lamentablemente, el épsilon de la máquina únicamente permite llegar a una tolerancia de 10E-16 por lo que los valores no varían en las tolerancias mayores. Adicionalmente, como se puede observar en la segunda función el resultado no converge, esto se debe a que al resolverla con el valor inicial 0 el algoritmo, en su validación de convergencia,muestra en pantalla que la función no se puede realizar con este método ya que no es cuadráticamente convergente en el valor inicial y finaliza la ejecucción del algoritmo.

5. Qué pasa con el método cuando hay más de dos raíces, explique su respuesta, encontrar la multiplicidad.

   Cuando se tiene una función con dos raíces y se aplica este método para encontrarla, el algoritmo hallará la raíz más cercana al punto de partida seleccionado.Si se quiere encontrar la o las otras raíces, si existen, se debe cambiar el punto inicial a uno que se encuentre más cerca de la otra raíz que se quiere hallar. En cuanto al problema de multiplicidad la versión de Newton-Raphson relajado lo soluciona al añadir en una variable m la multiplicidad,agregandola como producto del numerador de la fracción de la función del método de Newton-Raphson original.

6. ¿Qué pasa con el método cuando la función es periódica, par o impar, estas características influyen?.

   Este tipo de características que pueden tener las funciones no afectan la implementación del algoritmo de Newton-Raphson, ya que en los diferentes tipos de funciones en las que se aplicó si era par, como la primera, o periódica, como la última, no afectó en la obtención de la raíz ni en el número de iteraciones.

7. Realice una gráfica que muestra la relación entre $Ei + 1$ y E, qué representa esa gráfica encuentre una relación de la forma $Ei + 1 = f(Ei)$.

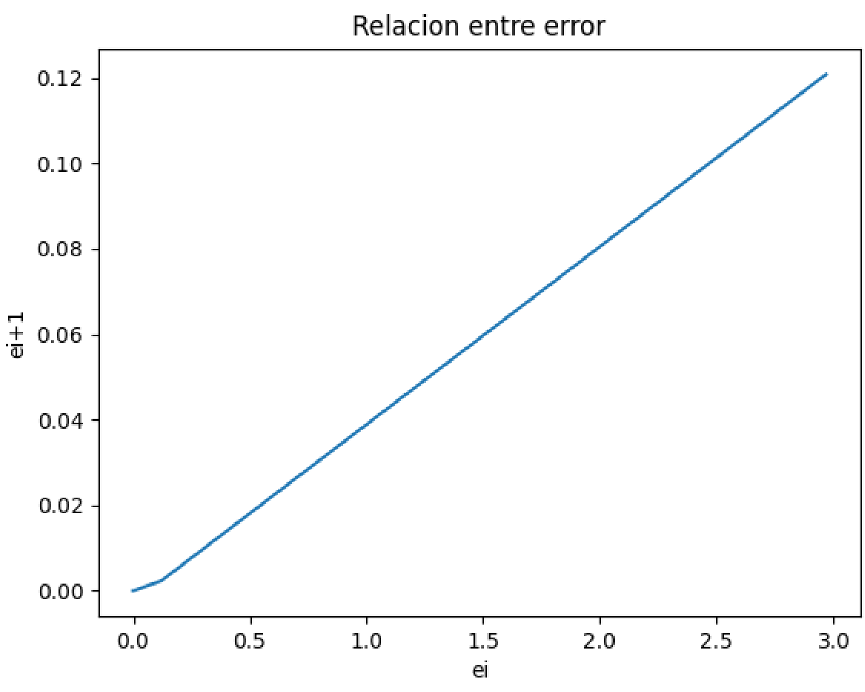


Figura 7: Relación entre $Ei + 1$ y E $f(x) = \cos(x)^2 - x^2$

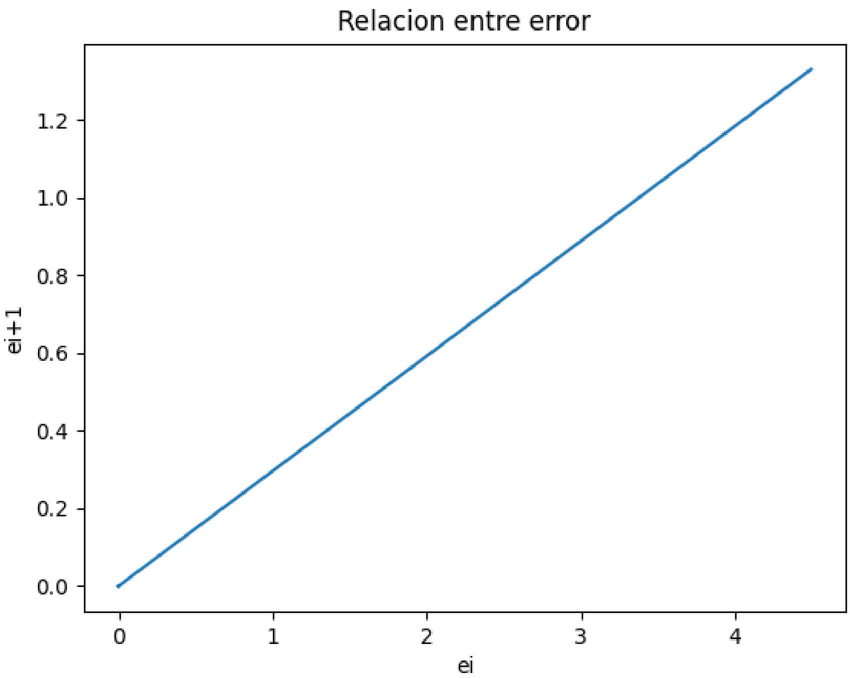


Figura 8: Relación entre $Ei + 1$ y E $f(x) = x^3 - 2x^2 + 4/3x - 8/27$

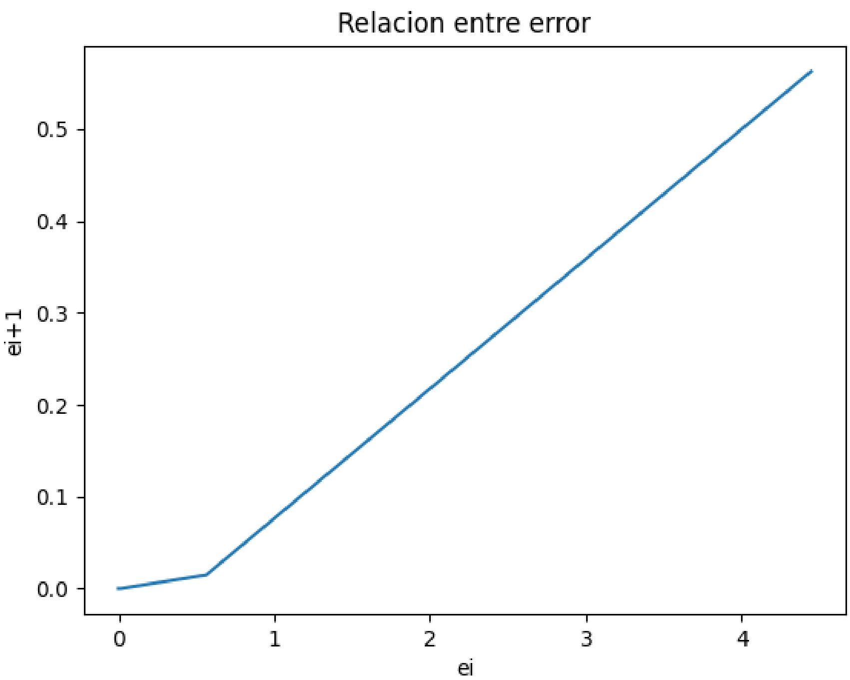


Figura 9: Relación entre $Ei + 1$ y E $f(x) = x^3 - 2x - 5 = 0$

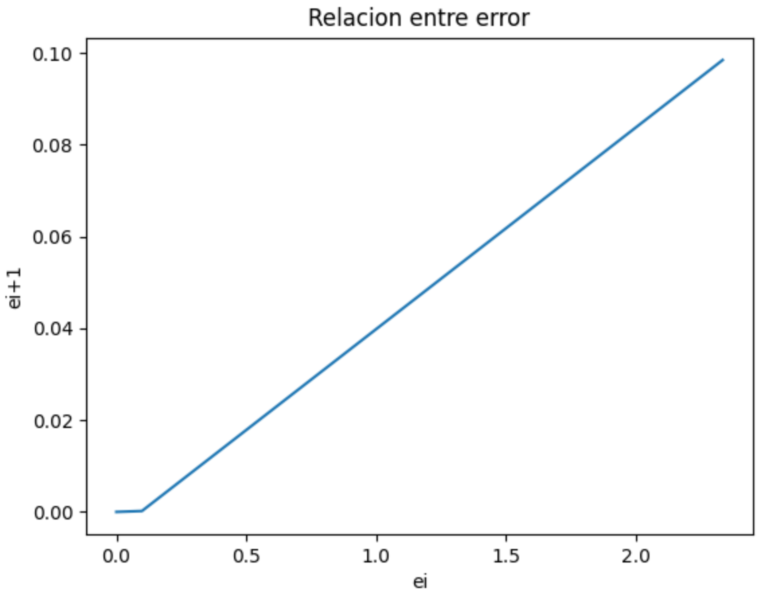


Figura 10: Relación entre $Ei + 1$ y E $f(x) = (667{,}38/x)(1 - exp(-0{,}146843x)) - 40$

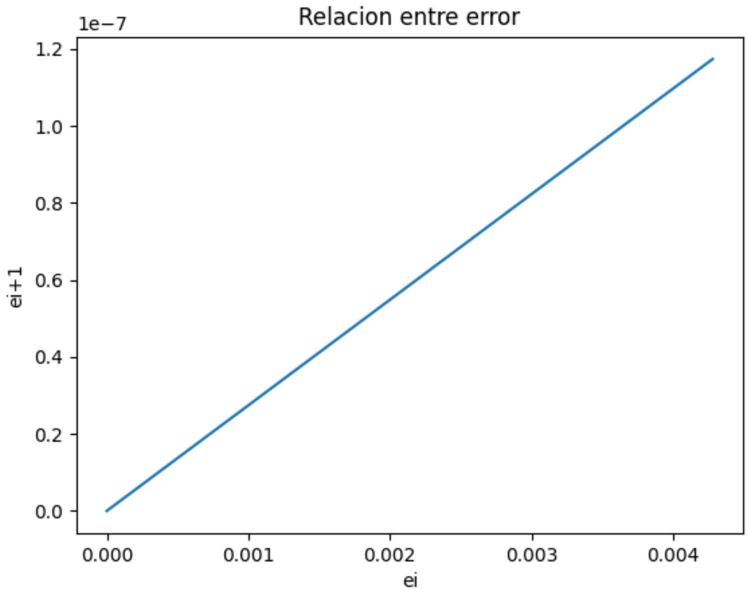


Figura 11: Relación entre $Ei + 1$ y E $f(x) = 3sin(x)^3 - 1 - 4sin(x)cos(x)$

Gracias a las anteriores graficas podemos observar que el metodo de Newton-Rapshon coverge linealmente en todos los problemas pues los concientes entre Ei+1 y Ei tienden a a estabilizarse en un numero menor a 1.

8. Realice una gráfica que muestre cómo se comporta el método en cada caso con respecto a la tolerancia y al número de iteraciones.

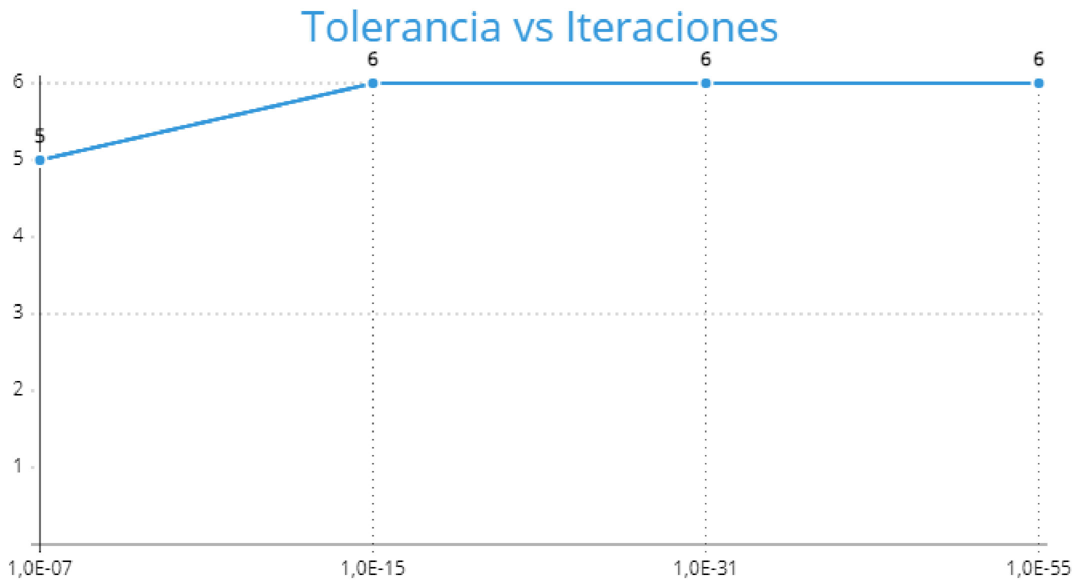


Figura 12: Tolerancia vs iteraciones$f(x) = \cos(x)^2 - x^2$

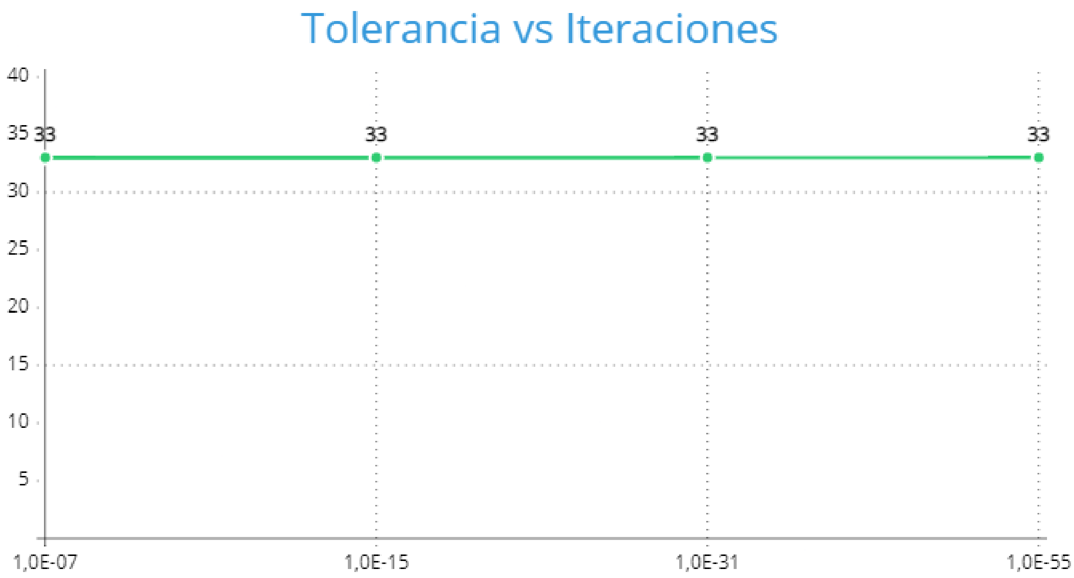


Figura 13: Tolerancia vs iteraciones $f(x) = x^3 - 2x^2 + 4/3x - 8/27$

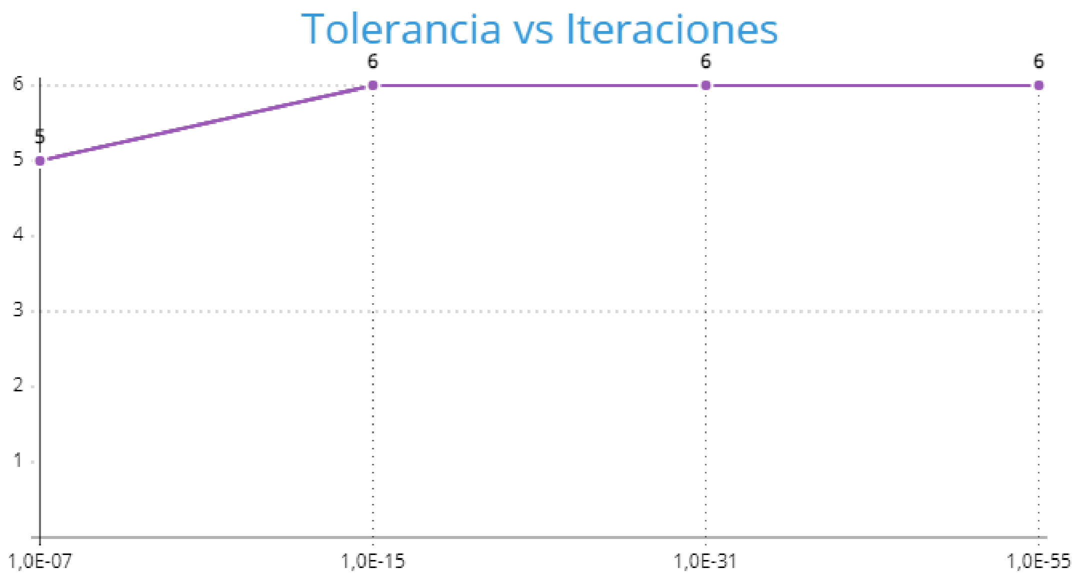


Figura 14: Tolerancia vs iteraciones $f(x) = x^3 - 2x - 5 = 0$

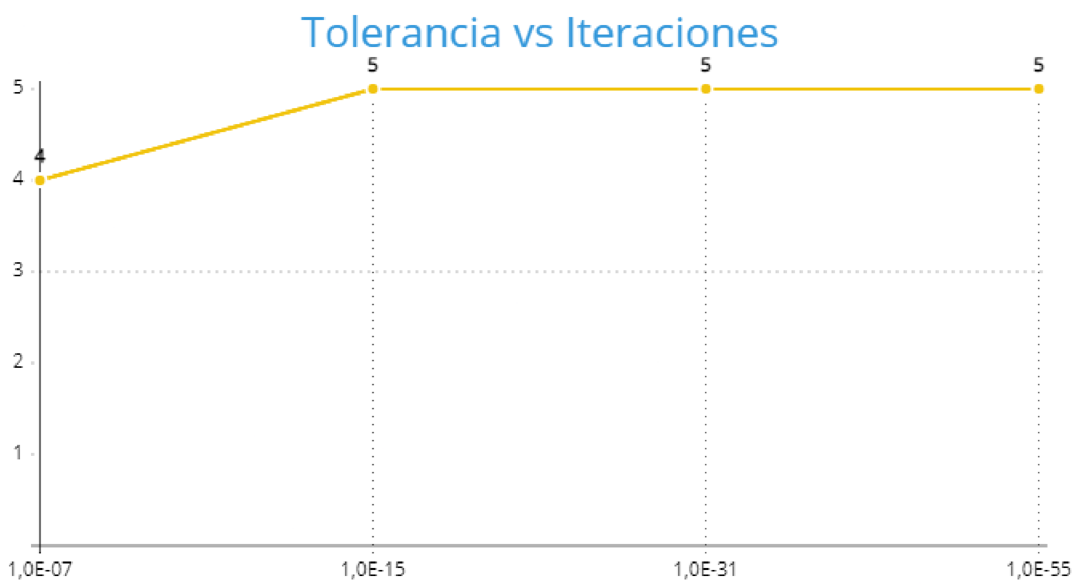


Figura 15: Tolerancia vs iteraciones $f(x) = (667{,}38/x)(1 - exp(-0{,}146843x)) - 40$

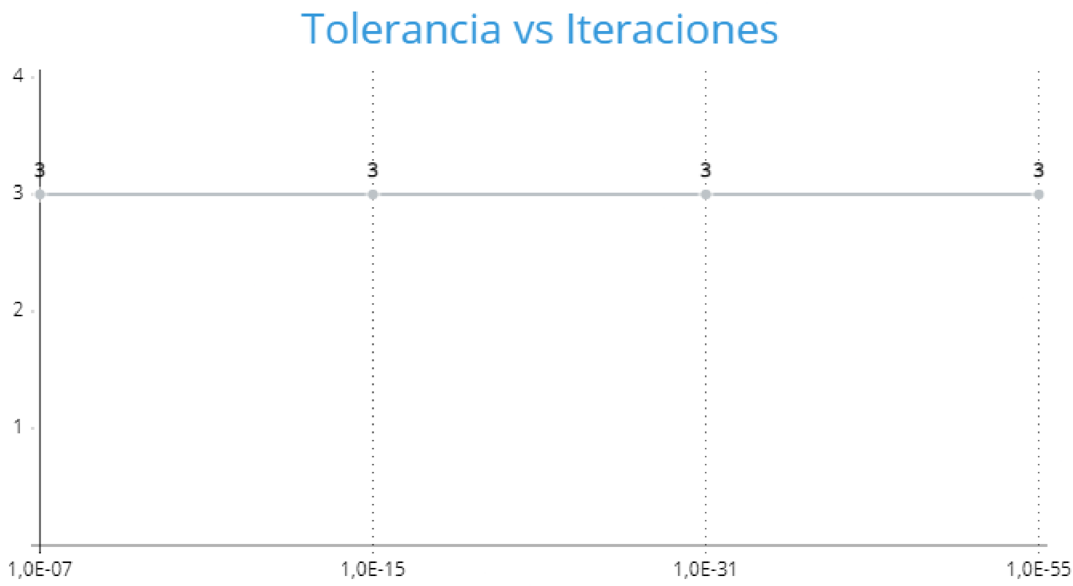


Figura 16: Tolerancia vs iteraciones $f(x) = 3sin(x)^3 - 1 - 4sin(x)cos(x)$

En la comparación de las iteraciones frente a la tolerancia, las gráficas muestran que el algoritmo de Newton-Raphson es eficiente a la hora de encontrar la raíz en una función, ya que el número de iteraciones necesarias para hallar el valor aproximado fue bajo.Sin embargo, hay una excepción,en la primera función del punto 3, ya que en está el algoritmo realizo 33 iteraciones, esto sucedió ya que el ángulo de intersección de la recta con el eje horizontal es pequeño en esta función cuando se acerca a la raíz, requiriendo un mayor número de iteraciones.

9. Cada ejercicio debe ser comparado con el método de bisección

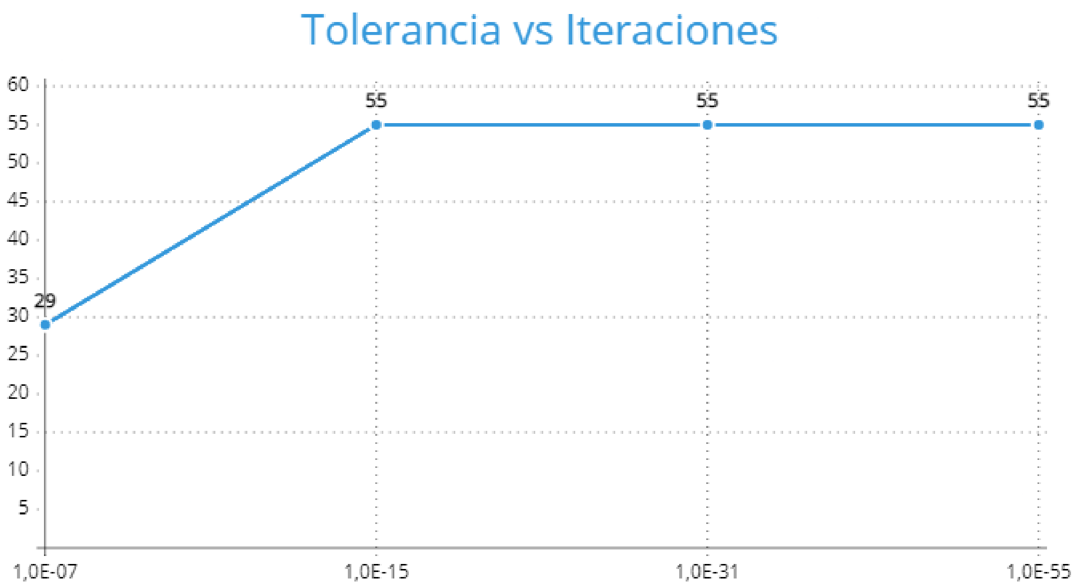


Figura 17: Tolerancia vs iteraciones(B) $f(x) = \cos(x)^2 - x^2$

Para esta problema el método de bisección es claramente inferior al método de Newton-Raphson, esto debido a que el primero requiere de un mayor número de iteraciones para llegar al valor aproximado de la raíz, la diferencia es de 24 iteraciones con una tolerancia de 10E-08 y de 49 iteraciones para el resto de tolerancias, dándonos a entender que a medida que la tolerancia aumente, el método de bisección para esta caso será todavía más ineficiente frente al método de Newton-Raphson.

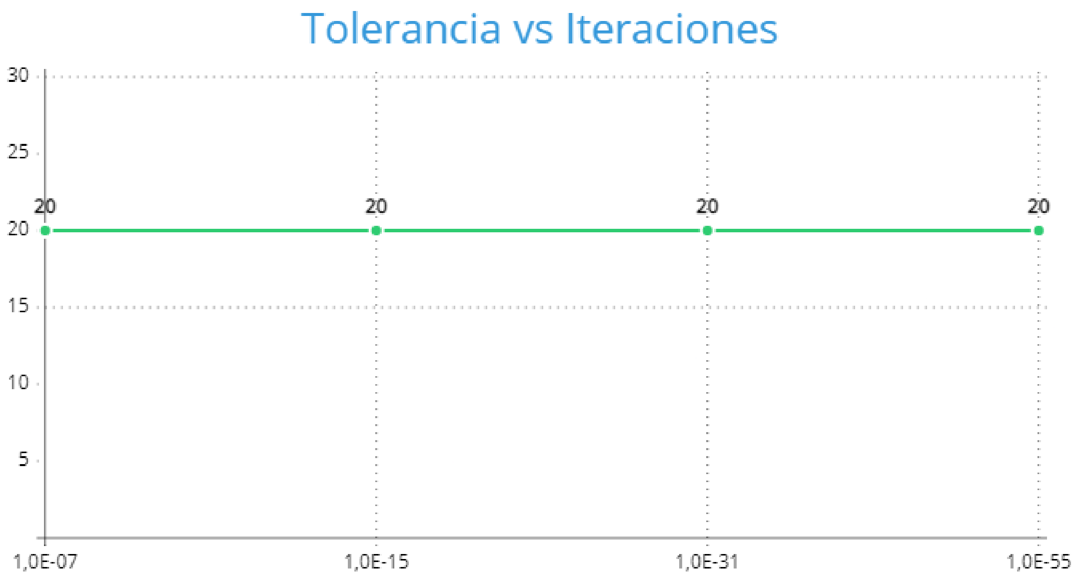


Figura 18: Tolerancia vs iteraciones(B) $f(x) = x^3 - 2x^2 + 4/3x - 8/27$

Para esta problema el método de bisección es más eficiente que el método de Newton-Raphson, esto debido a que el primero requiere de un menor número de iteraciones para llegar al valor aproximado de la raíz, la diferencia es de 13 iteraciones para todas las tolerancias, dándonos a entender que pese a que el método de bisección no llego en pocas iteraciones a la raíz aproximada (20) , el método de Newton-Raphson es todavía más ineficiente para esta función en particular requiriendo de 33 iteraciones.

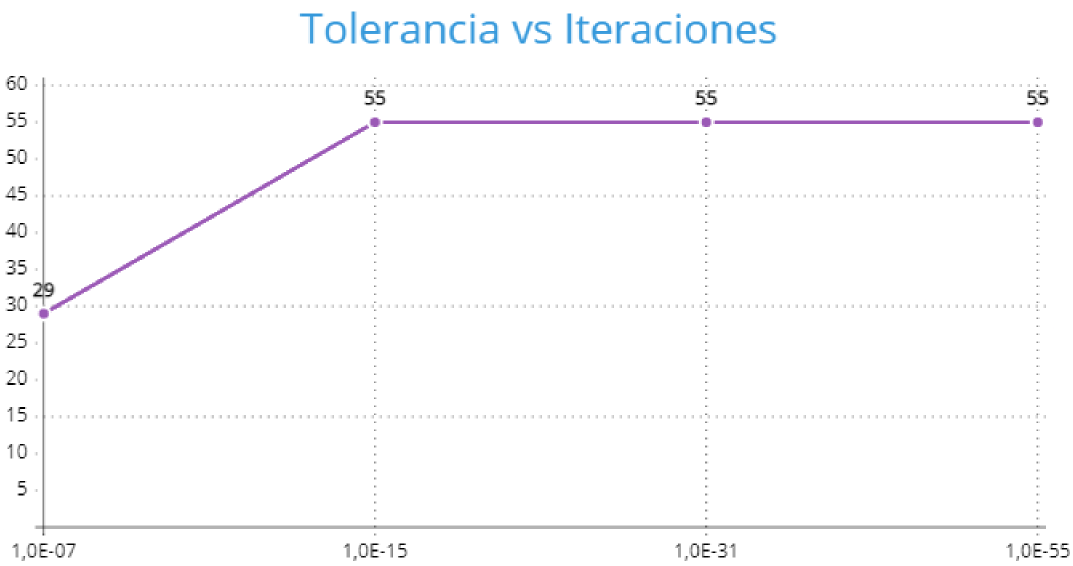


Figura 19: Tolerancia vs iteraciones(B) $f(x) = x^3 - 2x - 5 = 0$

Para esta problema el método de bisección es claramente inferior al método de Newton-Raphson, esto debido a que el primero requiere de un mayor número de iteraciones para llegar al valor aproximado de la raíz, la diferencia es de 24 iteraciones con una tolerancia de 10E-08 y de 49 iteraciones para el resto de tolerancias, dándonos a entender que a medida que la tolerancia aumente, el método de bisección para esta caso será todavía más ineficiente frente al método de Newton-Raphson.

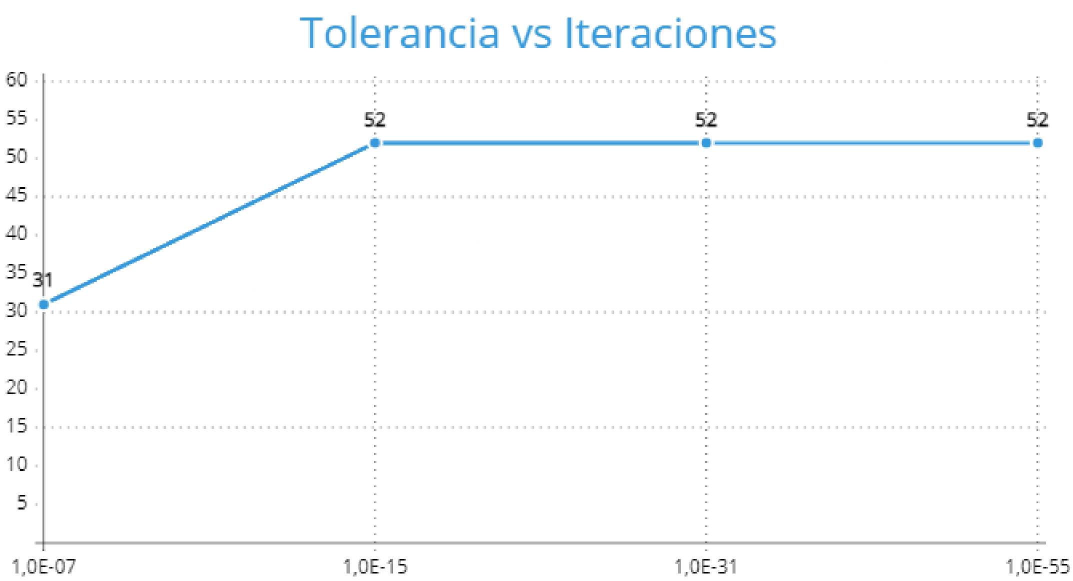


Figura 20: Tolerancia vs iteraciones(B) $f(x) = (667{,}38/x)(1 - exp(-0{,}146843x)) - 40$

Para esta problema el método de bisección es claramente inferior al método de Newton-Raphson, esto debido a que el primero requiere de un mayor número de iteraciones para llegar al valor aproximado de la raíz, la diferencia es de 27 iteraciones con una tolerancia de 10E-08 y de 47 iteraciones para el resto de tolerancias, dándonos a entender que a medida que la tolerancia aumente, el método de bisección para esta caso será todavía más ineficiente frente al método de Newton-Raphson.

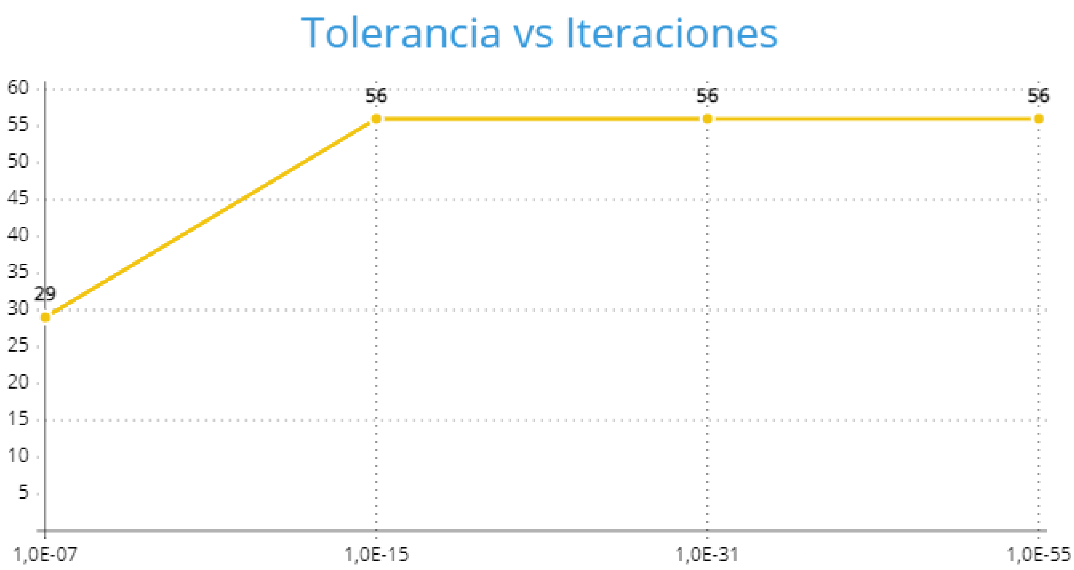


Figura 21: Tolerancia vs iteraciones $f(x) = 3sin(x)^3 - 1 - 4sin(x)cos(x)$

Para esta problema el método de bisección es claramente inferior al método de Newton-Raphson, esto debido a que el primero requiere de un mayor número de iteraciones para llegar al valor aproximado de la raíz, la diferencia es de 26 iteraciones con una tolerancia de 10E-08 y de 53 iteraciones para el resto de tolerancias, dándonos a entender que a medida que la tolerancia aumente, el método de bisección para esta caso será todavía más ineficiente frente al método de Newton-Raphson.

10. Como se comporta el método con respecto a la solución con Taylor

    Hemos concluido a partir de la definición del método de Taylor y la implementación en Python, que el método de Newton-Raphson es mejor que usar el método de Taylor. Para hacer una aproximación haciendo uso del método de Taylor con la misma tolerancia que con el método de Newton-Raphson, deben realizarse un número mucho más alto de iteraciones ya que deben calcularse más términos de la serie de Taylor y cada vez que se busca calcular un término más, se deben calcular derivadas mucho más altas por lo que los cálculos se hacen cada vez más inexactos y la pérdida de cifras en cada iteración se hace mayor.

5. Conclusiones

Tras haber revisado la teoría e implementación aplicada del método de Newton-Raphson, se puede concluir que este es un método altamente eficiente en la mayoría de los casos, sobrepasando claramente a métodos como el de bisección y el de Taylor, además, fue linealmente convergente en todos los problemas resueltos. Sin embargo, no es un método perfecto, esto debido a que como pudimos ver con la primera función del tercer problema propuesto, el método de Newton-Raphson requirió de más iteraciones que incluso el método de bisección, asimismo, no pudo dar respuesta al segundo problema propuesto dadas sus limitaciones por convergencia. Adicionalmente, este puede presentar problemas por la multiplicidad de ciertas funciones, no obstante, el método posee variaciones como el método de Newton-Raphson relajado, el cual soluciona el problema de multiplicidad.

Referencias

[1] Rodríguez Ojeada, L. (2016). ANÁLISIS NUMÉRICO BÁSICO Un enfoque algorítmico con el soporte de python (pp. 52–54). Guayaquil.

[2] de la Fuente O'Connor, J. L. (2017). Ingeniería de los Algoritmos y Métodos Numéricos Un acercamiento práctico avanzado a la computación científica e ingenieril con MATLAB (2.ª ed., pp. 40–46). España: Circulo Rojo.