



Pontificia Universidad Javeriana

Asignatura:

Análisis Numérico

Docente:

Eddy Herrera Daza

TALLER 1

Integrantes:

Heyling Burgos Algarín

burgosaheyling@javeriana.edu.co

Fabian Andres Olarte Vargas

olarte.fabian@javeriana.edu.co

Johan Mateo Rosero Quenguan

roseroq - j@javeriana.edu.co

Andrés Felipe Vásquez Rendón

af.vasquezr@javeriana.edu.co

Agosto 2021

Índice

1. Introducción	2
2. Método de la secante	2
2.1. Explicación Geométrica del Algoritmo	2
2.2. ¿Qué condiciones existen para aplicar el método?	3
2.3. Diagrama de Flujo	3
2.4. Código Fuente	4
2.5. Solución del Problema	5
2.6. Gráficas	8
2.6.1. Gráficas de Funciones	8
2.6.2. Gráficas Relación Tolerancia Vs Número de iteraciones	11
2.6.3. Gráficas de errores	14
3. Conclusiones	18

1. Introducción

En el presente documento se plantea el método de la secante como una solución a distintos problemas numéricos. Asimismo, se analizará la forma correcta para usar de manera eficiente dicho algoritmo, en cuanto a: las condiciones iniciales, los órdenes de convergencia y la relación entre la tolerancia y las iteraciones. Del mismo modo, se compara contra otros algoritmos, tales como, de Bisección y Aitken para analizar el cambio de la raíz en cada uno de sus resultados.

Conceptos Clave: Algoritmo de la Secante.

Palabras Clave: Algoritmo, Iteraciones, Tolerancia, Convergencia, Significancia.

2. Método de la secante

2.1. Explicación Geométrica del Algoritmo

El método de la Secante es una variación del método de Newton-Raphson, ya que en este no es necesario calcular la derivada, sino que se aproxima una recta secante a la curva, siendo la pendiente de esta una aproximación a la derivada de la función. A continuación, se presentan los pasos para llegar de Newton-Raphson al método de la Secante. [1]

$$f'(x) \approx \frac{\Delta y}{\Delta x} = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (1)$$

Aproximación de la derivada.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (2)$$

Método de Newton Rapshon.

$$x_{n+1} = x_n - f(x_n) * \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \quad (3)$$

Proceso iterativo de la secante.

Después de la explicación histórica del método de la secante, a continuación, se presenta la explicación geométrica del método:

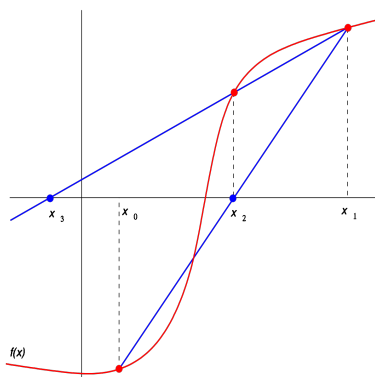


Figura 1: Cálculo del Segundo Valor Provisional

Como podemos ver en la anterior figura, se suponen dos valores iniciales como x_0 y x_1 para trazar una recta secante por encima de estas. Luego de obtener esa recta secante se obtiene el punto x_2 que es el punto de corte cuando la función es igual a 0. Al hacer esta iteración se proyecta este punto en la gráfica para poder reemplazar este nuevo valor en la función y trazar una recta secante entre el valor inicial x_1 y x_2 , obteniendo así el valor de x_3 siendo el punto de corte cuando la función es 0. Así se continua iterativamente hasta lograr la tolerancia deseada. [2]

2.2. ¿Qué condiciones existen para aplicar el método?

El algoritmo de la secante debe tener unas condiciones iniciales para que pueda funcionar correctamente, sin ellas no se conseguiría el resultado deseado. Estas condiciones son:

1. La función a evaluar debe ser continua en cada punto para el intervalo dado entre $[a,b]$. [3]
2. Si una función contiene una raíz en el intervalo $[a,b]$ si ambos contienen un signo opuesto, esto quiere decir que sigue la condición:

$$f(a) \cdot f(b) < 0 \quad (4)$$

Esto es conocido como el Teorema de Bolzano. [4]

3. Del mismo modo, se debe cumplir con la condición

$$|f(b)| < |f(a)| \quad (5)$$

tal que b sea la mejor candidata que a para ser raíz. [5]

2.3. Diagrama de Flujo

En la Figura 2, se puede observar el diagrama de flujo del Algoritmo Secante:

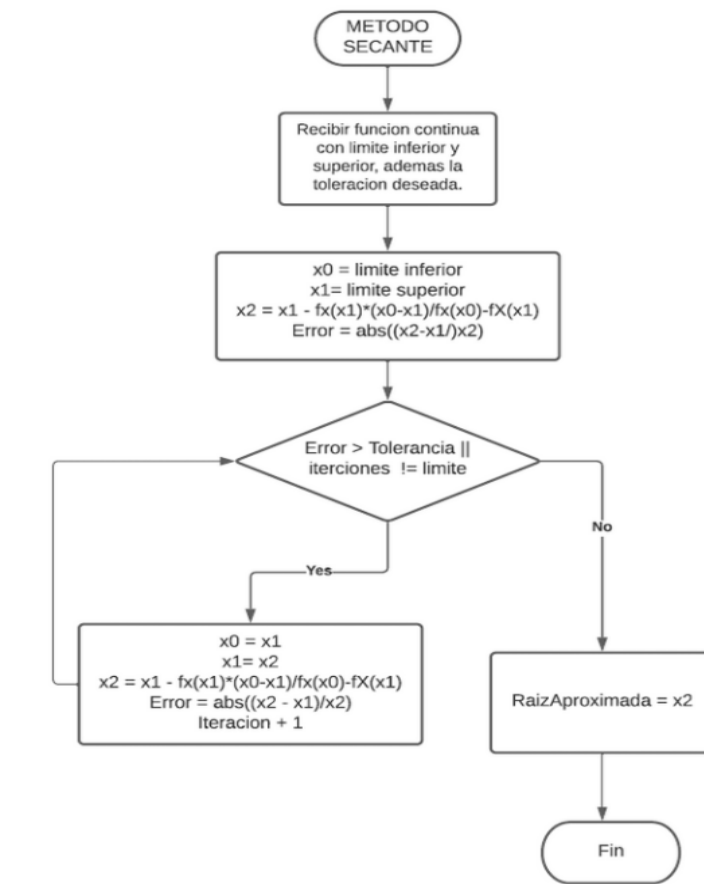


Figura 2: Diagrama de flujo método secante.

2.4. Código Fuente

En el listing 1, se puede ver la implementación del Algoritmo Secante en Python:

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import warnings
4 warnings.filterwarnings("ignore", category=DeprecationWarning)
5
6 def Secante (fx, t, a, b): #Recibe una funcion continua
7
8     i = 0 #Iteraciones
9     aux = 0
10    puntos = []
11
12    # Algoritmo
13    x0 = a
14    x1 = b
15    #dlt = 0.1
16    x2 = np.longdouble((x1) - ((fx(x1))*(x0-x1))/np.longdouble(fx(x0) - fx(x1))) #Algoritmo Inicial
17    #x2 = x0 - ((dlt*fx(x0))/(fx(x0+dlt) - fx(x0))) #Metodo modificado
  
```

```

18     i += 1
19     errorR = abs(np.longdouble((x2-x1)/x2))
20
21     while abs(np.longdouble((x2-x1)/x2)) > t:
22
23         aux = errorR
24         x0 = x1
25         x1 = x2
26         x2 = np.longdouble((x1) - ((fx(x1))*(x0-x1))/np.longdouble(fx(x0) -
fx(x1))) #Algoritmo Inicial
27         #x2 = x0 - ((dlt*fx(x0))/(fx(x0+dlt) - fx(x0)))
28         i += 1
29
30         if i>2000000:
31             aux = 1
32             break
33
34         errorR = abs(np.longdouble((x2-x1)/x2))
35         if i > 5:
36             puntos.append((aux, errorR))
37
38     print("\nMETODO SECANTE:\nLa raiz aproximada es: {} / Numero de iteraciones
: {} / Tolerancia: {}".format(x1, i, t))
39     if aux == 1:
40         print("Diverge")
41
42     # Grafico
43     plt.xlabel("X")
44     plt.ylabel("Y")
45     x = np.arange(a, b, 0.1)
46     plt.figure(0)
47     plt.plot(x,fx(x))
48     xs = np.linspace(a,b,100)
49     horiz_line_data = np.array([0 for i in range(len(xs))])
50     plt.plot(xs, horiz_line_data, 'r--')
51     plt.plot(x1, fx(x1), "ro")
52
53     return puntos

```

Listing 1: Código en Python intersección entre dos curvas

2.5. Solución del Problema

Dada las ecuaciones siguientes se les aplica el método de la secante para la obtención de las raíces cuando la función es igual a 0. Cabe resaltar que para la aplicación del algoritmo, se empleo un tipo de dato de tipo doble precisión extendida (numpy.longdouble) y junto con el épsilon de la maquina se logra la mejor exactitud que la maquina puede alcanzar, permitiendo así obtener la mayor precisión posible en los resultados propuestos a continuación.

$$f(x) = \cos^2(x) - x^2 \quad (6)$$

Para la ecuación 6, se obtuvieron los siguientes resultados:

Tolerancia	Secante	Iter. Secante	Bisección	Iter. Bisección
$1e^{-8}$	-0.7390851332300535	10	-0.73908512853086	30
$1e^{-16}$	-0.7390851332151607	11	-0.7390851332151607	54
$1e^{-32}$	-0.7390851332151607	11	-0.7390851332151607	54
$1e^{-56}$	-0.7390851332151607	11	-0.7390851332151607	54

Tabla 1: Resultados obtenidos para la función 6.

Como podemos observar en la tabla 1, la primera tolerancia correspondiente a $1e^{-8}$, ambos métodos contienen un número de iteraciones mínimo con respecto a las otras. En el caso del método secante se puede observar que realiza 10 iteraciones con un resultado significancia menor al método de bisección que contiene 30 iteraciones. Asimismo, las demás tolerancias contienen el mismo número de iteraciones (secante 11, bisección 54), los resultados de esas iteraciones entre ambos métodos resultan con un mismo número de significancia.

$$f(x) = e^x - x - x^2 \quad (7)$$

Para la ecuación 7, se obtuvieron los siguientes resultados:

Tolerancia	Secante	Iter. Secante	Bisección	Iter. Bisección
$1e^{-8}$	-1,95741586578854E+08	40	-1,49011611938476E+08	27
$1e^{-16}$	-1,95741586578854E+08	40	-1,49011611938476E+08	27
$1e^{-32}$	-1,95741586578854E+08	40	-1,49011611938476E+08	27
$1e^{-56}$	-1,95741586578854E+08	40	-1,49011611938476E+08	27

Tabla 2: Resultados obtenidos para 7.

Como se puede observar en la tabla 2, todas las iteraciones del método secante dan como resultado 40 y en el método bisección 27 sin importar la tolerancia usada. Del mismo modo, el resultado de la secante contiene más significancia que el método de bisección.

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

Para la ecuación 8, se obtuvieron los siguientes resultados:

Tolerancia	Secante	Iter. Secante	Bisección	Iter. Bisección
$1e^{-8}$	0.6666722860337118	43	0.6666660308837891	19
$1e^{-16}$	0.6666722860337118	42	0.6666660308837891	19
$1e^{-32}$	0.6666722860337118	42	0.6666660308837891	19
$1e^{-56}$	0.6666722860337118	42	0.6666660308837891	19

Tabla 3: Resultados obtenidos para la función 8.

Como se puede observar en la tabla 3, el método de la secante en la primera tolerancia contiene mayores iteraciones que las otras, sin embargo, todos los resultados contienen el mismo número de significancia. Por otro lado, todas las tolerancias en el método de la bisección contienen el mismo número de iteraciones al igual que su significancia en el resultado. En este caso, la significancia es menor en el resultado de la bisección versus al método secante.

$$f(x) = x^3 - 2x - 5 \quad (9)$$

Para la ecuación 9, se obtuvieron los siguientes resultados:

Tolerancia	Secante	Iter. Secante	Bisección	Iter. Bisección
$1e^{-8}$	2.094551478163650	8	2.094551481306550	29
$1e^{-16}$	2.0945514815423265	10	2.094551481542327	55
$1e^{-32}$	2.0945514815423265	10	2.094551481542327	55
$1e^{-56}$	2.0945514815423265	10	2.094551481542327	55

Tabla 4: Resultados obtenidos para la función 9.

Como podemos observar en la tabla 4, la primera tolerancia correspondiente a $1e^{-8}$, ambos métodos contienen un número de iteraciones mínimo con respecto a las otras. Asimismo, en el caso de la secante se puede observar que realiza 8 iteraciones con un resultado de significancia menor a la de bisección que contiene 29 iteraciones. Por otra parte, las demás tolerancias contienen el mismo número de iteraciones, en el caso de la secante 10 y en el de bisección 55, la cual sus resultados, en cuanto a iteraciones, resultan distintas con un número diferente de significancia, siendo el método de la secante el que mayor significancia tiene.

$$f(x) = \left(\frac{667.38}{x}\right) * (1 - e^{-146843x}) - 40 \quad (10)$$

Para la ecuación 10, se obtuvieron los siguientes resultados:

Tolerancia	Secante	Iter. Secante	Bisección	Iter. Bisección
$1e^{-8}$	14.78020859368090	8	14.780208585085300	33
$1e^{-16}$	14.780208593679400	9	14.780208593679400	53
$1e^{-32}$	14.780208593679400	9	14.780208593679400	53
$1e^{-56}$	14.780208593679400	9	14.780208593679400	53

Tabla 5: Resultados obtenidos para la función 10.

Como podemos observar en la tabla 5, la primera tolerancia correspondiente a $10e^{-8}$, ambos métodos contienen un número de iteraciones mínimo con respecto a las otras. Asimismo, en el caso de la secante se puede observar que realiza 8 iteraciones con un resultado de significancia mayor al método de bisección, el cual contiene 33 iteraciones. Por otra parte, las demás tolerancias contienen el mismo número de iteraciones (secante 9, bisección 53), los resultados de las iteraciones entre ambos procesos resultan con un mismo número de significancia.

$$f(t) = e * \text{sen}^3(t) - 1 - 4 * \text{sen}(t) * \cos(t) \quad (11)$$

Para la ecuación 11 se obtuvieron los siguientes resultados:

Tolerancia	Secante	Iter. Secante	Bisección	Iter. Bisección
$1e^{-8}$	6,0028385908143	5	5,51754284052537	30
$1e^{-16}$	6,002838590814260	7	5,17542840528378	54
$1e^{-32}$	6,002838590814260	7	5,17542840528378	54
$1e^{-56}$	6,002838590814260	7	5,17542840528378	54

Tabla 6: Resultados obtenidos para la función 10.

Para el desarrollo de la función 11, se igualaron ambas funciones a cero, esto debido a que cuando son igualadas a cero, el punto obtenido es donde ambas funciones colisionan y, por ende, la raíz de las funciones.

Como podemos en la tabla 6 la primera tolerancia correspondiente a $10e^{-8}$, ambos métodos contienen un número de iteraciones mínimo con respecto a las otras. Asimismo se logra observar que las otras tres tolerancias, ambos métodos presentan una misma cantidad de iteraciones (siete en el método de la secante y 54 en el método de la bisección); además se logra evidenciar que la raíz es diferente en el método de la secante, ya que cuando evaluamos el punto x_0 y x_1 en la función, esta puede desbordarse del rango y empezaría a operar con valores fuera de este, por lo que este hallaría una raíz fuera de los límites iniciales, caso contrario en el método de la bisección en el cual es imposible que suceda este fenómeno.

2.6. Gráficas

2.6.1. Gráficas de Funciones

A continuación, se presentan las gráficas de cada una de las funciones propuestas anteriormente. Cabe resaltar que cuando una función tiene una relación par, impar o periódica con respecto a sus valores, este método no tiene afectaciones al momento de encontrar raíces sin importar el límite inferior y superior que es dado de parámetro. Del mismo modo, solo es posible encontrar una raíz cuando se usa el algoritmo de la secante, ya que esta se limita a encontrar solo una relación cuando la función tiende a cero.

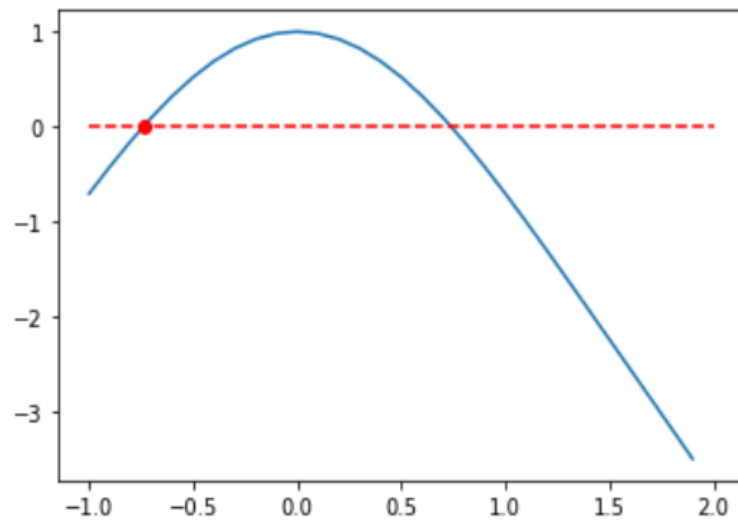


Figura 3: Gráfica de la función $\cos^2(x) - x^2$

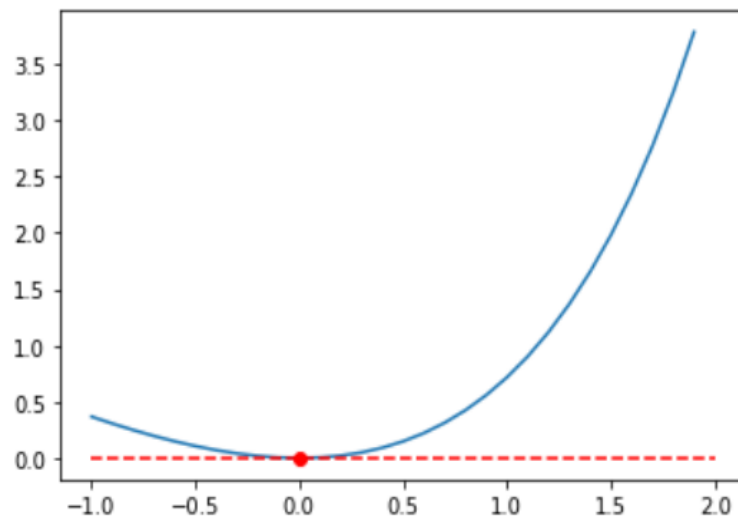


Figura 4: Gráfica de la función $f(x) = e^x - x - x^2$

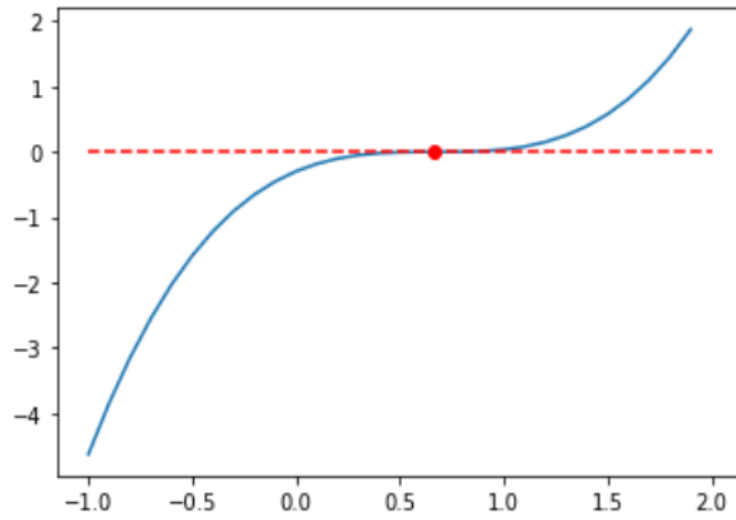


Figura 5: Gráfica de la función $f(x) = x^3 - 2x^2 + \frac{4x}{3} - \frac{8}{27}$

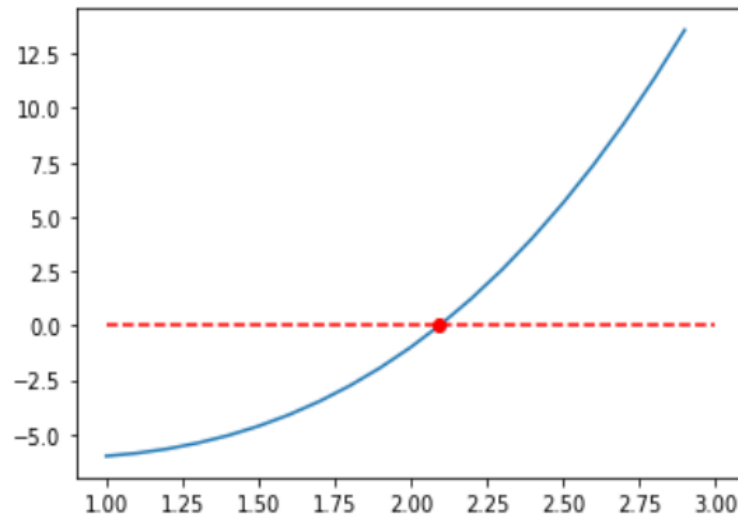


Figura 6: Grafica de la función $f(x) = x^3 - 2x - 5$

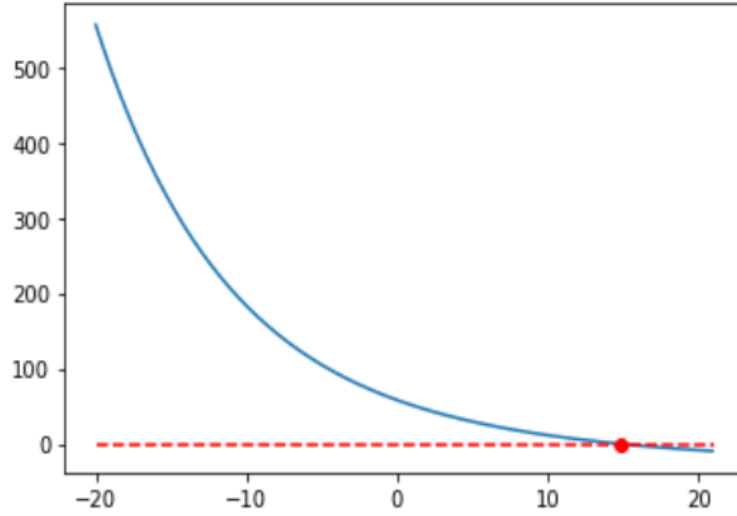


Figura 7: Gráfica de la función $f(x) = ((\frac{667.38}{x}) * (1 - e^{-146843x})) - 40$

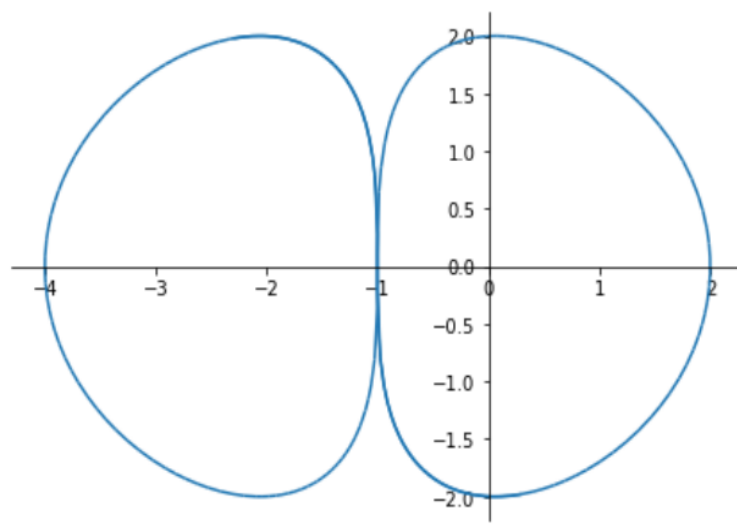


Figura 8: Gráfica de la función $f(t) = e * \text{sen}^3(t) - 1 - 4 * \text{sen}(t) * \text{cos}(t)$

2.6.2. Gráficas Relación Tolerancia Vs Número de iteraciones

A continuación, se presentan las gráficas que relacionan la tolerancia con el número de iteraciones que se emplearon para desarrollar cada una de las funciones propuestas con el método de la secante.

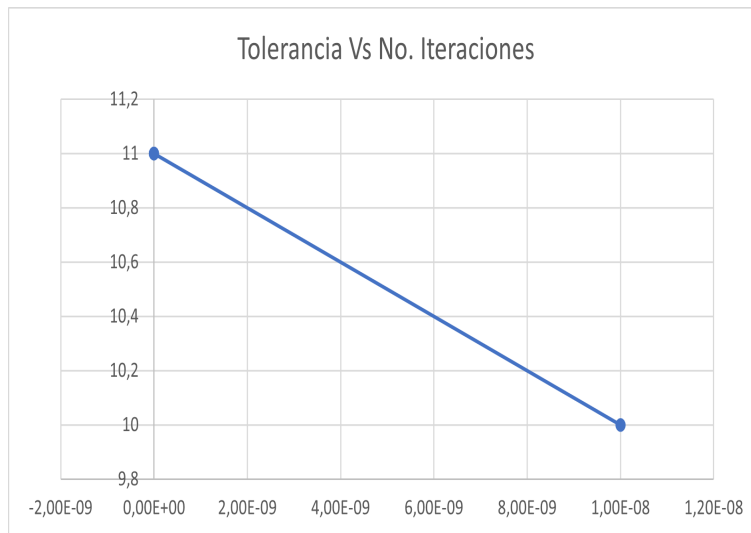


Figura 9: Gráfica de la función $\cos^2(x) - x^2$

Como se puede observar en la gráfica, la primera tolerancia $1e^{-8}$ requiere 10 iteraciones para llegar a un resultado, mientras que el resto de tolerancias, requieren de una iteración más para poder llegar a un resultado.

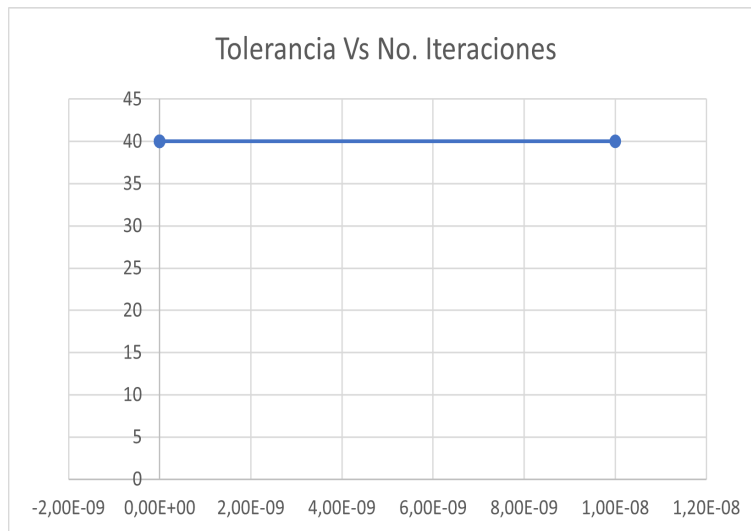


Figura 10: Gráfica de la función $f(x) = e^x - x - x^2$

Como se puede observar en la gráfica, todas las tolerancias requieren la misma cantidad de iteraciones, que en este caso son 40 iteraciones.

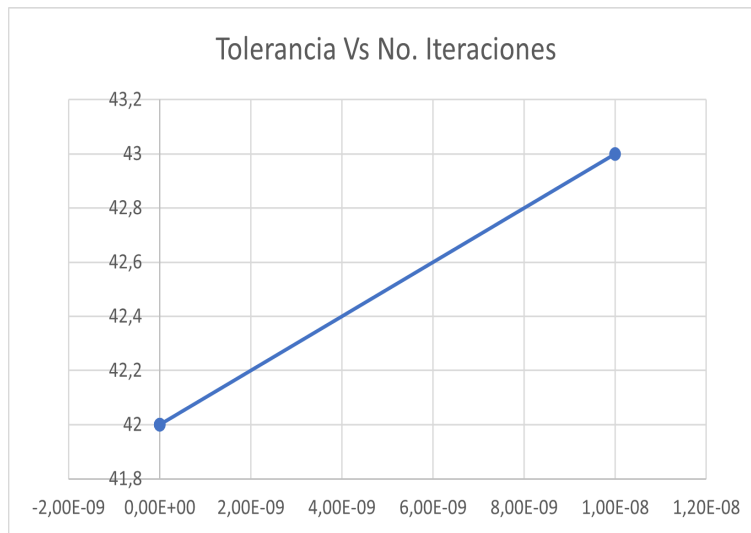


Figura 11: Gráfica de la función $f(x) = x^3 - 2x^2 + \frac{4x}{3} - \frac{8}{27}$

Como se puede observar en la gráfica, la primera tolerancia $1e^{-8}$ requiere de una iteración más (43 iteraciones) que el resto de tolerancias, las cuales emplean 42 iteraciones para obtener la raíz.

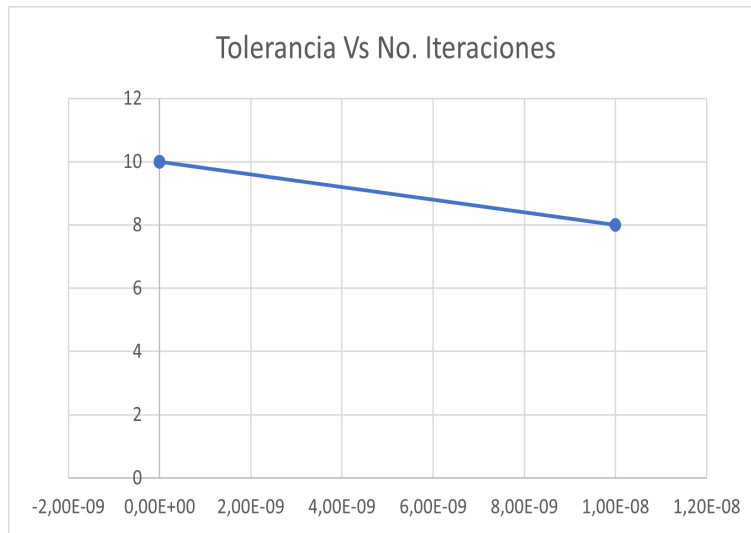


Figura 12: Gráfica de la función $f(x) = x^3 - 2x - 5$

Como se puede observar en la gráfica, la primera tolerancia $1e^{-8}$ requiere dos iteraciones menos (8 iteraciones) que el resto de tolerancias, las cuales emplean 10 iteraciones para obtener la raíz.

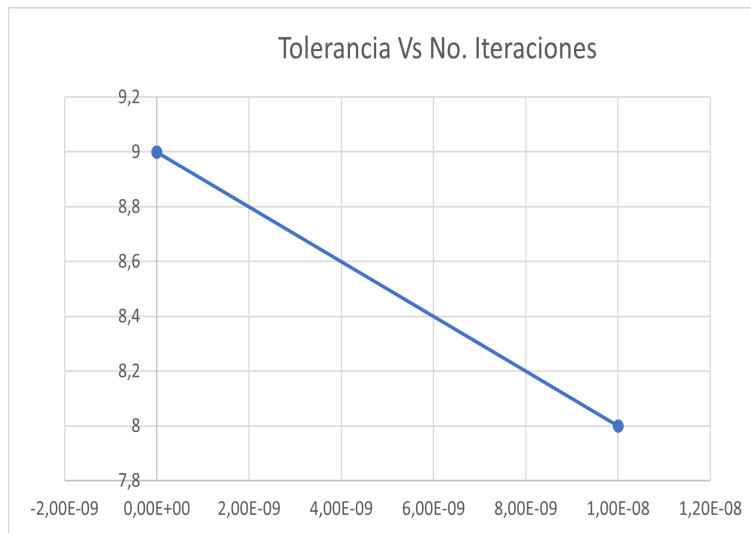


Figura 13: Gráfica de la función $f(x) = ((\frac{667.38}{x}) * (1 - e^{-146843x})) - 40$

Como se puede observar en la gráfica, la primera tolerancia $1e^{-8}$ requiere una iteración menos (8 iteraciones) que el resto de tolerancias, las cuales emplean 9 iteraciones para obtener la raíz.

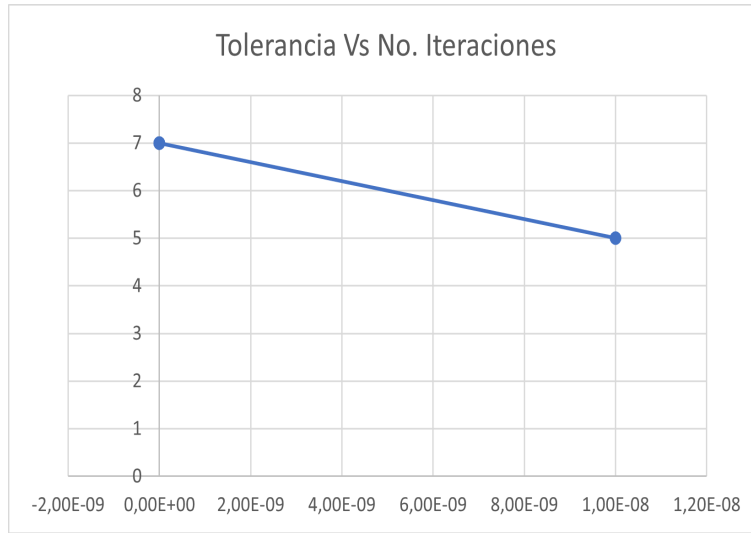


Figura 14: Gráfica de la función $f(t) = e * \text{sen}^3(t) - 1 - 4 * \text{sen}(t) * \text{cos}(t)$

Como se puede observar en la gráfica, la primera tolerancia $1e^{-8}$ requiere dos iteraciones menos (5 iteraciones) que el resto de tolerancias, las cuales emplean 7 iteraciones para obtener la raíz.

2.6.3. Gráficas de errores

A continuación se presentan los errores relativos correspondientes a cada una de las funciones en forma de gráfica.

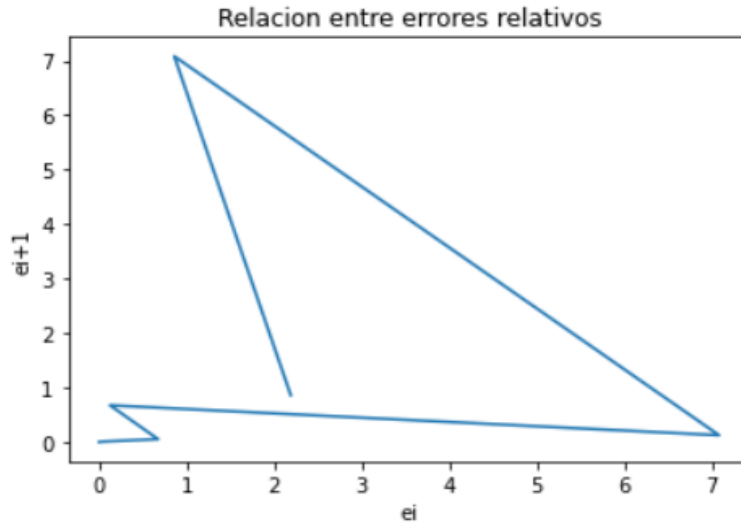


Figura 15: Gráfica de la función $\cos^2(x) - x^2$

Como se logra observar en la gráfica, el error al inicio es un error disperso, con valores cercanos a uno y con picos de hasta siete, pero es después de varias iteraciones que el error empieza a tender a cero.

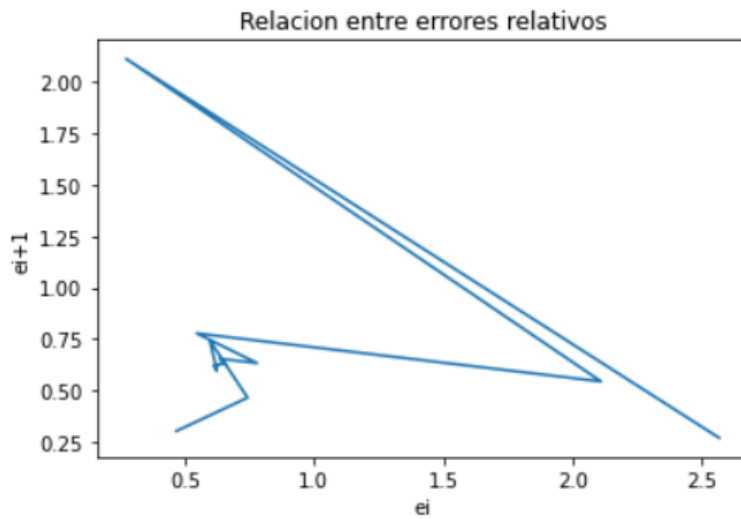


Figura 16: Gráfica de la función $f(x) = e^x - x - x^2$

Como se observa en la gráfica, el comportamiento del error empieza a ser disperso siguiendo un orden lineal ascendente y descendente hasta luego mantenerse en un punto fijo hasta minimizar su valor y acercarse a cero.

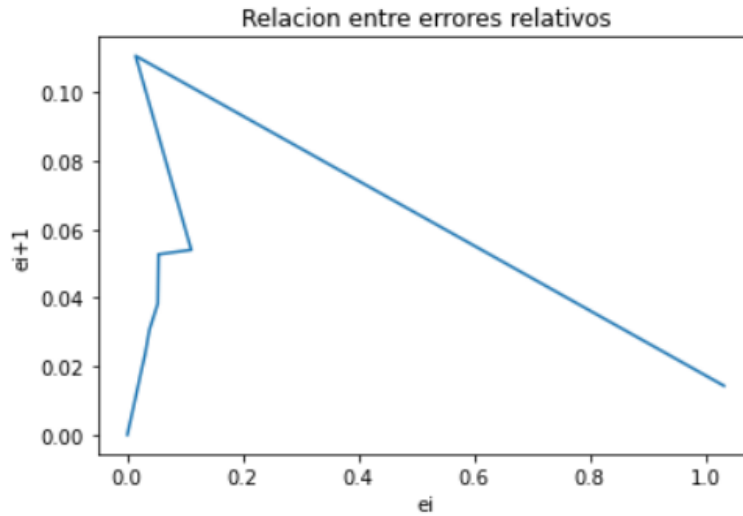


Figura 17: Gráfica de la función $f(x) = x^3 - 2x^2 + \frac{4x}{3} - \frac{8}{27}$

Como se ve reflejado en la gráfica, con un valor cercano a uno la tendencia de la gráfica empieza a aumentar mientras su valor va disminuyendo, teniendo una relación inversa, esto hasta llegar a un punto que desciende en picada para obtener el valor cero.

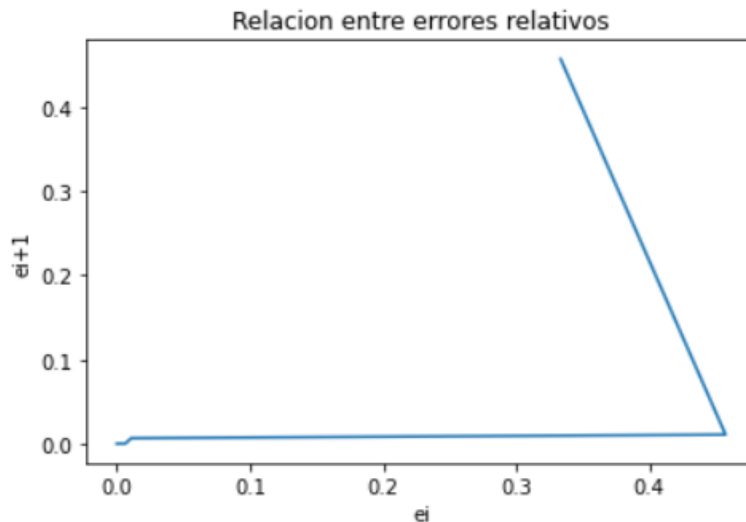


Figura 18: Gráfica de la función $f(x) = x^3 - 2x - 5$

Como se puede observar en la gráfica, no contiene una variación entre sus valores manteniendo un error constante, a excepción del punto cercano a 0.4 que obtiene una dispersión con respecto a los otros valores.

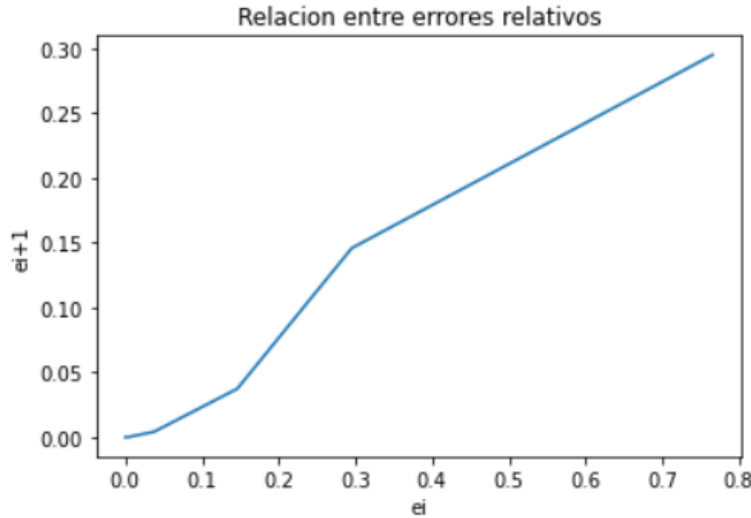


Figura 19: Gráfica de la función $f(x) = ((\frac{667.38}{x}) * (1 - e^{-146843x})) - 40$

Como se puede observar en la gráfica, el error desciende de una forma lineal y continua, empe-
zando por un punto cercano a 0.30 y terminando finalmente en cero.

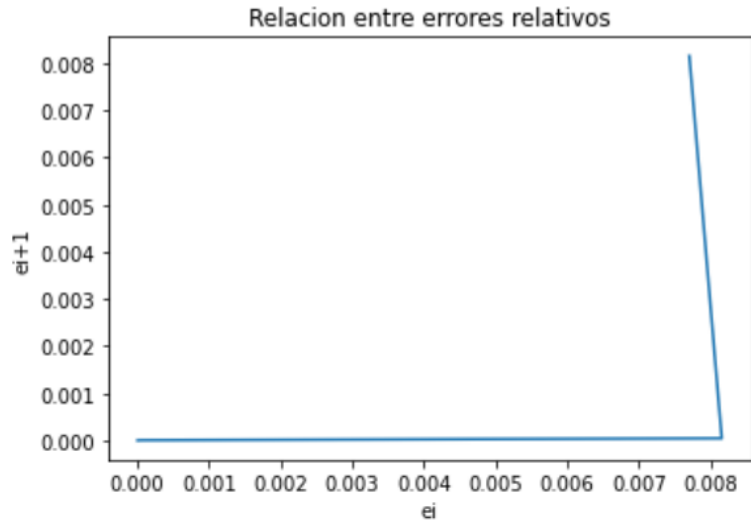


Figura 20: Gráfica de la función $f(t) = e * \text{sen}^3(t) - 1 - 4 * \text{sen}(t) * \text{cos}(t)$

Como se observa en la gráfica, el comportamiento del error empieza a ser disperso siguiendo un
orden lineal ascendente y descendente hasta luego mantenerse en un punto fijo hasta minimizar su
valor y acercarse a cero.

Además de lo ya descrito anteriormente, es necesario mencionar que la convergencia del método
de la secante es lineal, esto lo podemos notar en la figura 17, en la cual se ve de manera clara como
el error disminuye linealmente, mientras que en las demás graficas esta convergencia no se ve de
forma clara, es por esto que se presenta a continuación, una gráfica de los errores de la función
 $f(x) = e^x - x - x^2$, en la cual se puede observar de como esta converge linealmente; esta fue
ensayada en un rango de -1 a 1 y realizo un total de 42 iteraciones.

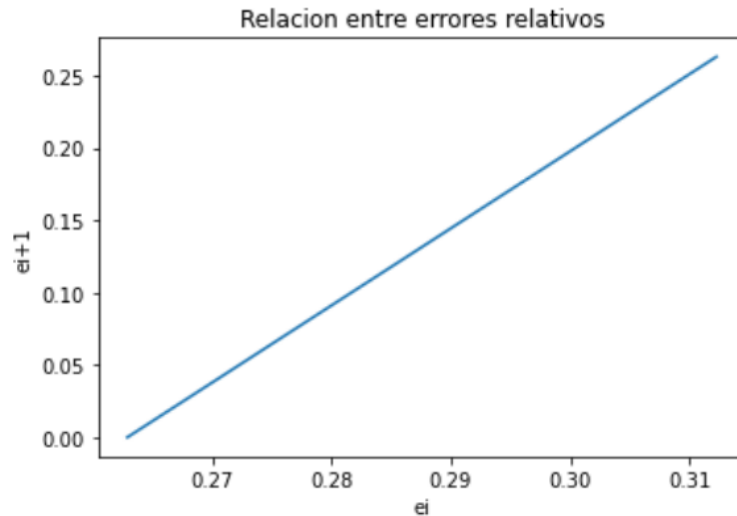


Figura 21: Gráfica de la función $f(x) = e^x - x - x^2$

3. Conclusiones

- Se puede concluir que tanto el método de la secante como el método de bisección logran llegar a la raíz de una forma rápida, pero en funciones complejas, siembargo el método de la secante es mucho más eficaz que el método de la bisección.
- La precisión de un valor es algo fundamental para cuando se necesita obtener una clara razón de cambio entre valores, por ello para conseguir la raíz exacta se aproxima al valor más preciso que el programa puede hallar.
- La función dada como parámetro obligatoriamente debe ser continua en el intervalo establecido ya que al realizar las rectas secantes que pasan entre los puntos debe existir un valor real para lograr la iteración correcta.
- La gráfica de errores es tan dispersa, debido a que, como se mencionó en la explicación gráfica del método de la secante, se parte de un límite inferior y superior dónde al restar estos límites, el valor dado es muy lejano de la raíz, y es hasta cierto punto del proceso iterativo, cuando los valores empiezan a tender cada vez más hacia la raíz.
- Cuando la tolerancia es un valor mínimo, el resultado obtenido suele tener una diferencia significativa con respecto a las demás iteraciones siendo la menos aproximada a la raíz que se desea encontrar.
- Como se pudo observar en las tablas de resultados, llega un punto en que las iteraciones son constantes después de una tolerancia determinada, esto es debido a que, por la épsilon de la máquina, el algoritmo es incapaz de llegar a un valor más aproximado al resultado real, es por ello que cuando se alcanzan unas iteraciones constantes significa que a pesar del cambio de tolerancia no se puede llegar a un valor más exacto o en dado caso ya se llegó a el valor real.

- El método de bisección en algunos casos es posible que se desborde, es decir, cuando la raíz de la función es muy cercana o es cero, estas pueden cambiar de signo muy fácilmente, por ello el rango de a y b podría cambiar y dejar por fuera al valor aproximado de la raíz, lo cual generaría el retorno de una raíz igual a alguno de los dos límites (a o b), generando así información incorrecta. Este problema es uno de los inconvenientes con el método de la bisección, por eso cabe mencionar que en la tabla de resultados en la función $e^x - x - x^2$ se usaron límites diferentes entre el método de la secante y bisección con el fin de comparar el número de iteraciones (rendimiento) de cada método.
- Cada raíz es el valor más aproximado logrado mediante una precisión doble extendida, pero puede ser mínimamente diferente al valor real por cuestiones de la epsilon de la máquina.
- La convergencia del método de la secante no asegura si la primera aproximación a la raíz es lo suficientemente cercana a ella, ni tampoco se asegura cuando la raíz es múltiple. Esto no quiere decir que no se pueda usar el método en esos casos, significa que al usarlo entramos en un riesgo de que este no converja y no podamos hallar la raíz.

Referencias

- [1] Morante, F. FdeT: METODOS NUMERICOS 4/4: METODO DE LA SECANTE, 2015. Recuperado de: https://www.youtube.com/watch?v=gvXptKRHXPA&ab_channel=canalfdet [1].
- [2] Madhav, M. Secant Method of Numerical analysis, 2020. Recuperado de: <https://www.geeksforgeeks.org/secant-method-of-numerical-analysis/> [6].
- [3] N.N. THE SECANT METHOD, S.F. Recuperado de: <http://homepage.math.uiowa.edu/~whan/3800.d/S3-3.pdf> [5].
- [4] Marta. ¿Para que sirve el Teorema de Bolzano?, 2021. Recuperado de: <https://www.superprof.es/apuntes/escolar/matematicas/calculo/derivadas/teorema-de-bolzano.html> [2].
- [5] Russo, R. Método de la Secante, S.F. Recuperado de: http://www3.fi.mdp.edu.ar/metodos/apuntes/secante_rodrigo.pdf [3].