Departamento de Ingeniería de sistemas
Facultad de Ingeniería
Pontificia Universidad Javeriana

Metodos para encontrar las Raíces de una función f(x)

Cuestas Samy, Escobar Carlos, Hernández Camilo, Mendieta Juan Camilo

---

1. Método de Bisección

1.1. La condición para poder utilizar el método de bisección es darle un intervalo [a , b], donde F(a) * F(b) es menor a 0, esto garantiza que F(a) y F(b) tienen distinto signo. En la implementación esto se tiene que verificar, por lo que para llevar a cabo el método. primero se utiliza un condicional que compare si F(a)*F(b) es menor a 0 , en caso de no ser así, se pide cambiar el intervalo de la función.

Otra condición que debe garantizarse es que la respuesta arrojada por el método cumpla con la tolerancia indicada. Para esto se utiliza un ciclo While que seguirá iterando hasta que el valor absoluto de la diferencia de los extremos del intervalo sea menor a la tolerancia.

1.2. El método de bisección consiste en utilizar un intervalo en el cual se encuentra la raíz, para garantizar esto los extremos del intervalo deben tener signos opuestos al ser evaluados en la función. Lo que se busca con el método es encontrar el punto medio de este intervalo y ver cual de los dos nuevos intervalos que se forman entre los extremos y el punto medio siguen cumpliendo con la condición de que los extremos evaluados en la función tenga diferente signo, después se sigue iterando sobre el intervalo que cumpla la condición.

1.3. Diagrama de flujo:

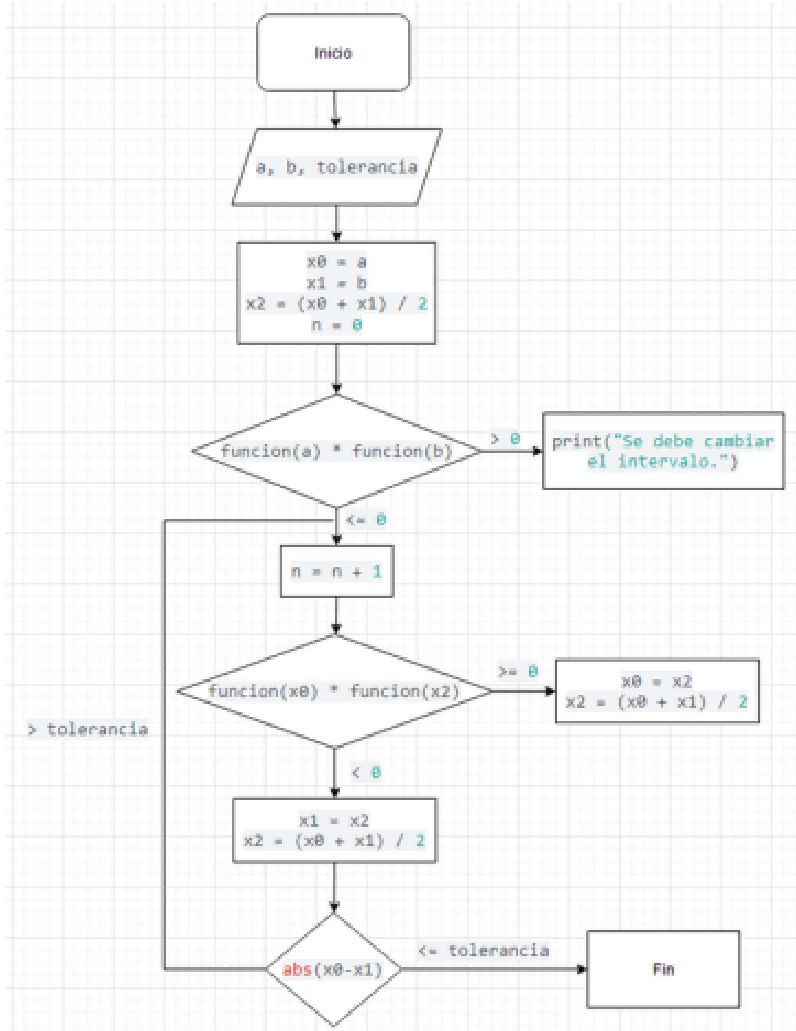


Figura 1: Diagrama de flujo del método de bisección.

1.4. Resultados esperados (Wolfram Alpha):

Resultados de Wolfram Alpha	
f(x)=$cos^2(x) - x^2$	x≈0.73908513321516064165531208767387340401 34 x≈-0.7390851332151606416553120876738734040134
f(x)=$x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$	x≈1.6550042270670163149579881380139462789011 2331
f(x)=$x^3 - 2x - 5$	x≈2.0945514815423265914823865405793030
f(x)=$e^x - x - 1$	x = 0
f(x)=$(\frac{667.38}{x})(1-e^{-0.146843x})-40$	x ≈ 14.780208593679467751317122645834539885 1

Figura 2: Resultados Esperados.

Aplicación del método y resultados obtenidos:

$f(x) = cos(x)^2 - x^2$		
Raíces	Tolerancia	Iteraciones
-0.7390851341187954	10^{-8}	27
-0.7390851332151605	10^{-16}	54
-0.7390851332151605	10^{-32}	54
-0.7390851332151605	10^{-56}	54

Figura 3: Resultados punto a.

$f(x) = e^x - x - 1$		
Raíces	Tolerancia	Iteraciones
x: -3.725290298461914e-09 f(x):0.0	10^{-8}	27
x: -2.7755575615628914 e-17 f(x) : 0.0	10^{-16}	54
x: -3.0814879110195774e -33	10^{-32}	107
x: -2.5489470578119236e -57	10^{-56}	187

Figura 4: Resultados punto b.

$f(x) = x^3 - 2x^2 + 4/3 * x - 8/27$		
Raíces	Tolerancia	Iteraciones
0.6666717492043972	10^{-8}	28
0.6666717529296875	10^{-16}	55
0.6666717529296875	10^{-32}	55
0.6666717529296875	10^{-56}	55

Figura 5: Resultados punto c.

$f(x) = x^3 - 2x - 5$		
Raíces	Tolerancia	Iteraciones
2.094551481865346	10^{-8}	28
---	10^{-16}	---
---	10^{-32}	---
---	10^{-56}	---

Figura 6: Resultados punto c_2.

1.5. Se puede apreciar claramente en todas las funciones evaluadas que el número de iteraciones aumenta a medida que disminuye la tolerancia, obviamente en los casos en los que se logró obtener un número más exacto a medida que se disminuye la tolerancia ya que hubo casos en los que no fue posible obtener números diferentes a pesar de que disminuir la tolerancia. Al aumentar las iteraciones aumenta la posibilidad de encontrar un número periódico por lo que el computador redondea dicho número al no ser almacenable, incurriendo en pérdida de significancia.

1.6. Se podría solucionar la pérdida de significancia aplicando tolerancias incluso menores a las presentadas ya que así se tienen en cuenta más decimales para entregar una respuesta, sumando a esto se podrían utilizar tipos de variable que permitan almacenar más información.

1.7. A medida de que se aumenta la cantidad de raíces, la cantidad de situaciones en las cuales pide cambiar el intervalo aumenta cuando no se le da un intervalo lo suficientemente grande, es decir que entre más raíces hay, más grande debe ser el intervalo para cumplir con todas las raíces.

1.8. Cuando la función es par, el método es capaz de encontrar los valores en tolerancias cada vez más bajas mientras que cuando es impar resulta bastante común que no pueda procesar los resultados pasada cierta disminución de tolerancia, comparada con funciones par.

1.12. Como el método de Taylor implementado solo trabaja con polinomios de segundo grado, se puede observar que comparado con el método de bisección, el método de Taylor es bastante menos exacto. El caso en donde los algoritmos llegan a los resultados más cercanos entre sí es el caso con la función f(x)=x3-2x2+4/3*x -8/27 donde igualmente existe una diferencia de exactamente 0.1031283555273666 pero también observamos casos como el de la función f(x)=cos(x)2-x2 donde la diferencia es tan grande como 1.260914866784839.

$f(x) = cos(x)^2 - x^2$	
Bisección	Taylor
-0.7390851332151609, 0.7390851332151609	-2.0, 2.0
$f(x) = e^x - x - 1$	
0	0.0
$f(x) = x^3 - 2x^2 + 4/3 * x - 8/27$	
0.6666717529296875	0.5635433974413834
$f(x) = x^3 - 2x - 5$	
2.094551481865346	2^{-16}

Figura 7

Comportamiento del método de bisección con respecto a la solución con Taylor.

2. Método de Muller

2.1. Es un método para encontrar raíces, el cual es un refinamiento del método de la secante, solo que en lugar de utilizar rectas secantes con dos puntos, se crean parábolas con tres puntos. La condición para que se pueda aplicar el método de Muller es que la función en la que se utiliza sea continua, otra condición es que al momento de realizar la operación para encontrar el siguiente punto de corte debido a que los números con los que se trabajan son muy pequeños puede ocurrir que la resta en el denominador sea cero, por lo que debe garantizarse que esto no ocurra. Para cumplir con esta condición en el código se valida primero si la resta que se realiza en el denominador es igual a cero, si es el caso se realiza la resta pero se le suma la tolerancia de modo que el denominador no quede como cero, en caso de que no sea posible llegar a la tolerancia deseada y el algoritmo no avance se coloco como medida un parámetro que define un número máximo de iteraciones, cuando estas se superan el algoritmo se detiene y arroja el resultado que tenga en ese momento.

2.2. El método de Muller es similar al método de la secante solo que en lugar de utilizar 2 puntos utiliza 3 puntos para determinar una parábola que se intersecta con la función en los 3 puntos seleccionados, esos puntos son p0, p1 y p2 y se determina la siguiente aproximación p3 considerando la intersección entre el eje x y la parábola que pasa por p0, f (p0)), (p1, f (p1)), y (p2, f (p2)).

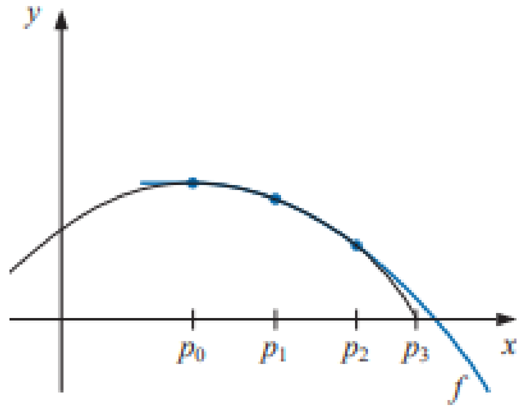


Figura 8: Numerical Analysis Ninth Edition - Muller's Method

Como ya tenemos 3 puntos iniciales ahora necesitamos encontrar a, b y c,que son parámetros que determinamos del polinomio de la parábola utilizando la ecuación:

$$P(x) = a(x - x_2)^2 + b(x - x_2) + c \tag{1}$$

Como sabemos que los 3 puntos pasan por la parabola pero al mismo tiempo por la función podemos deducir el siguiente sistema de ecuaciones:

$$f(x_0) = a(x_0 - x_2)^2 + b(x_0 - x_2) + c \tag{2}$$

$$f(x_1) = a(x_1 - x_2)^2 + b(x_1 - x_2) + c \tag{3}$$

$$f(x_2) = a(x_2 - x_2)^2 + b(x_2 - x_2) + c \tag{4}$$

Para resolver este sistema se utilizan las siguientes fórmulas:

$$h_0 = x_1 - x_0 \tag{5}$$

$$h_1 = x_2 - x_1 \tag{6}$$

$$d_0 = \frac{f(x_1) - f(x_2)}{x_1 - x_0} \tag{7}$$

$$d_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1} \tag{8}$$

Una vez con estas variables ya se pueden identificar a,b y c con las siguientes ecuaciones:

$$a = \frac{d_1 - d_0}{h_1 + h_0} \tag{9}$$

$$b = ah_1 + d_1 \tag{10}$$

$$c = f(x_2) \tag{11}$$

Una vez que se tienen estas 3 variables ya se puede encontrar el punto de intersección de la parabola con el eje x con la ecuación cuadratica por lo que se halla ya que este será el siguiente punto de aproximación a la raiz en la siguiente iteración del método:

$$x_3 = x_2 + \frac{-2c}{b \pm \sqrt{b^2 - 4ac}} \tag{12}$$

Finalmente se elige el signo de la raíz para que coincida con el signo de b, de modo que el denominador sea más grande y p3 sea más cercano al punto de corte.

Una vez que se determina p3, el procedimiento se reinicia usando p1, p2 y p3 en lugar de p0,p1 y p2 para determinar la siguiente aproximación, p4. El método continúa hasta un resultado satisfactorio.

2.3. Diagrama de flujo:

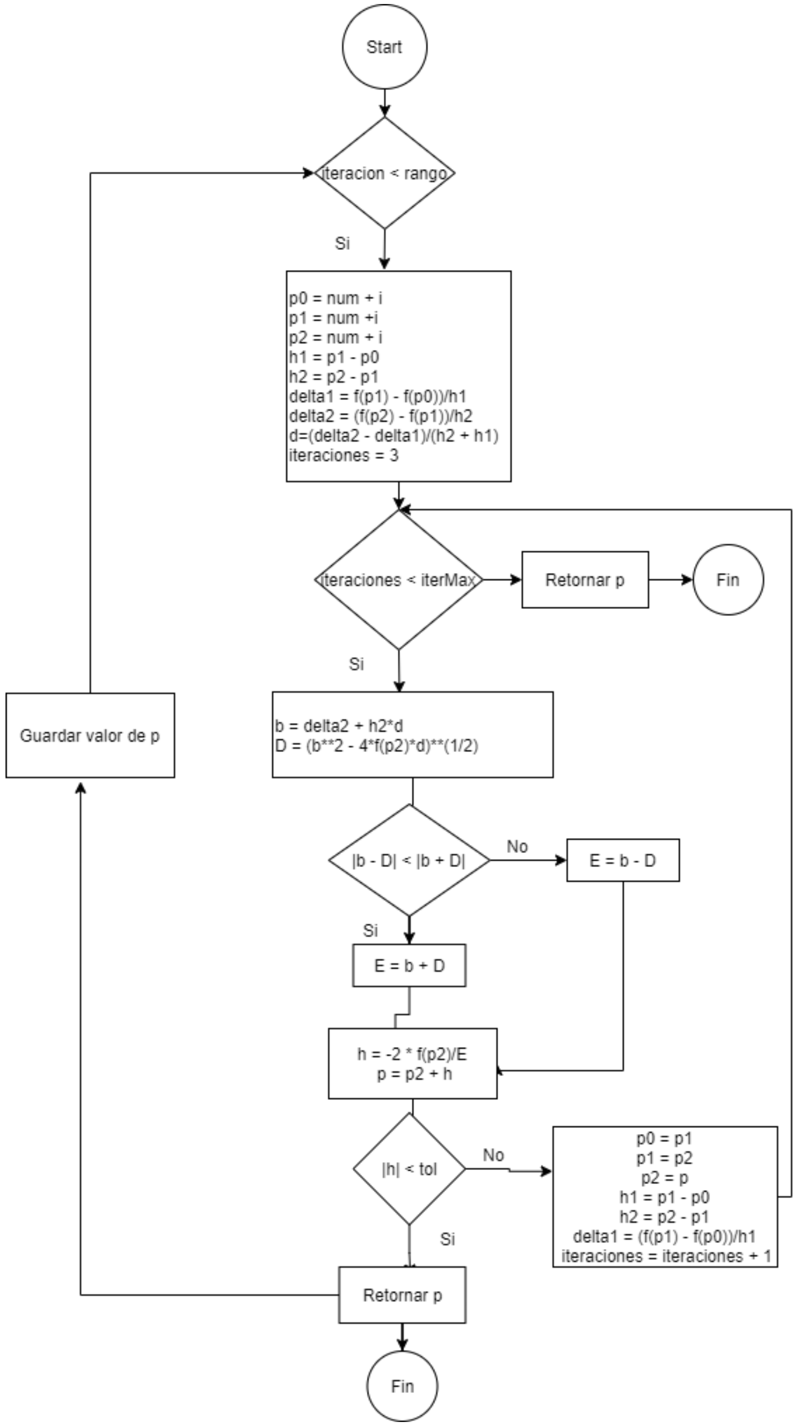


Figura 9: Diagrama de flujo del método de Muller.

2.4. Resultados esperados (Wolfram Alpha):

Resultados de Wolfram Alpha	
f(x)=$cos^2(x) - x^2$	x≈0.7390851332151606416553120876738734040134 x≈-0.7390851332151606416553120876738734040134
f(x)=$x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$	x≈1.65500422706701631495798813801394627890112331
f(x)=$x^3 - 2x - 5$	x≈2.0945514815423265914823865405793030
f(x)=$e^x - x - 1$	x = 0
f(x)=$(\frac{667.38}{x})(1\text{-}e^{-0.146843x})$-40	x ≈ 14.7802085936794677513171226458345398851

Figura 10: Resultados Esperados.

Aplicación del método y resultados obtenidos:

$f(x) = cos(x)^2 - x^2$		
Raíces	Tolerancia	Iteraciones
-0.73908513 0.73908513	10^{-8}	7
0.7390851332151607 -0.7390851332151607	10^{-16}	8
0.7390851332151606722 9310920083662 -0.7390851332151606722 9310920083662	10^{-32}	8
-0.7390851332151606722 931092008366249501705 1696777343750000 0.7390851332151606722 931092008366249501705 1696777343750000	10^{-56}	8

Figura 11: Resultados punto a - Muller.

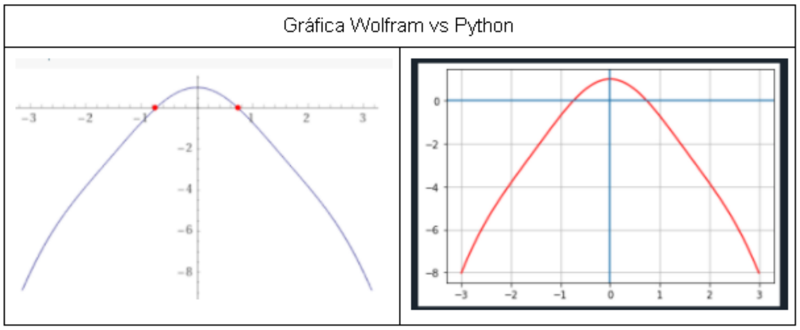


Figura 12: Gráficas punto a - Muller.

$f(x) = e^x - x - 1$		
Raíz	Tolerancia	Iteraciones
0.00000000	10^{-8}	4
0.0000000000000000	10^{-16}	61
0.000000000000000000000 00000000000	10^{-32}	61
0.000000000000000000000 0000000000000000000000 0000000000000	10^{-56}	3

Figura 13: Resultados punto b - Muller.

Figura 14: Gráficas punto b - Muller.

$f(x) = x^3 + 2x^2 + 4/3 * x + 8/27$		
Raíces	Tolerancia	Iteraciones
{'0.66666466+0.00000350j',	10^{-8}	50
{'0.6666697656691868+0.0000000000000000j',	10^{-16}	61
{'0.66666976566918678681616938774823+0.000000000000000000000000000000j',	10^{-32}	183
{'0.66666465408692632266252076078671962022781372070312500000+0.000000000000000000000000000000000000000000000000000j',	10^{-56}	356

Figura 15: Resultados punto c - Muller.

Gráfica Wolfram vs Python

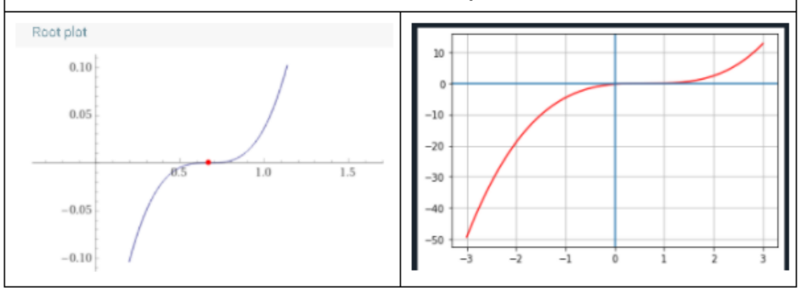


Figura 16: Gráficas punto c - Muller.

$f(x) = x^3 + 2x + 5$		
Raíz	Tolerancia	Iteraciones
{'-1.04727574+1.13593989j' ' '2.09455148'}	10^{-8}	8 11 7
{'-1.0472757407711633+1.1359398890889283j', '2.0945514815423265'}	10^{-16}	11 15 8
{(-1.0472757407711635+1.1359398890889283j), (2.0945514815423265+0j)}	10^{-32}	10000+
{(-1.0472757407711635+1.1359398890889283j), (2.0945514815423265+3.7199990403638925e-20j)}	10^{-56}	10000+

Figura 17: Resultados punto c$_2$ − *Muller*.

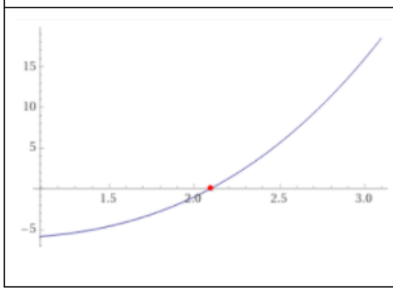
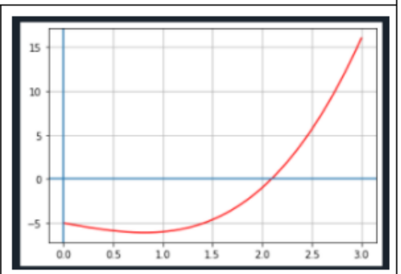
Gráfica Wolfram vs Python	
	

Figura 18: Gráficas punto c$_2$ − *Muller*.

$\frac{667.38}{x}\left(1-e^{-0.146843\,x}\right)-40$		
Raíz	Tolerancia	Iteraciones
14.78020859	10^{-8}	6
14.7802085936794665	10^{-16}	7
14.78020859367946471252 253104466945	10^{-32}	7
14.78020859367946648887 93704449199140071868896 4843750000000	10^{-56}	7

Figura 19: Resultados punto d - Muller.

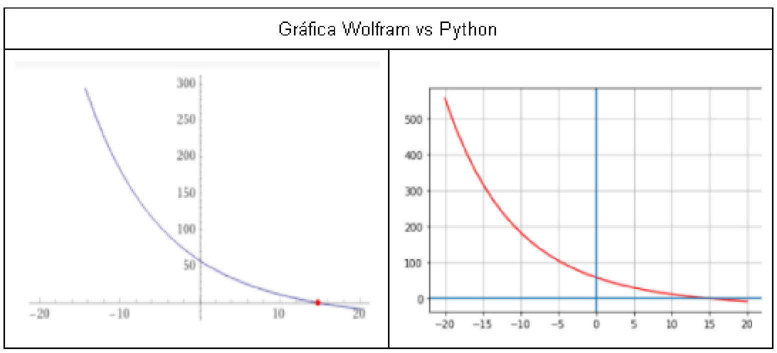


Figura 20: Gráficas punto d - Muller.

2.5. Se puede observar que a medida que se disminuye la tolerancia también incrementan las iteraciones realizadas, pero a su vez es valor obtenido a través del método es más cercano a los valores que arrojan otras herramientas de cálculo por lo que puede determinarse que la respuesta obtenida es más cercana al valor real, por lo que se tiene una mayor significancia en la respuesta.

2.6. Al igual que en anteriores casos, en algunos ejercicios donde la raíz es un número periodico debido al error que se trabaja y su propagación en las diferentes operaciones al realizar el método, nunca se va a llegar a la tolerancia deseada ya que no se llegara a un número suficientemente cercano al valor real, esto provocará que el algoritmo se mantenga en un ciclo. En este caso para que se pueda obtener una respuesta se tiene que tomar una medida como detener el algoritmo y retornar la respuesta que este tenga en esa iteración aunque esto causa una pérdida de significancia ya que el valor obtenido está más alejado del valor real. Aunque se puede obtener una respuesta esta no cumpliría con la tolerancia deseada.

2.7. Cuando existen más de dos raíces es necesario realizar un mayor número de iteraciones ya que se deben encontrar un mayor número de parábolas que cumplan con la condición de interceptar en 3 puntos la función, ya que se utiliza una parábola para encontrar cada una de las raíces.

En el caso de que haya una raíz con una multiplicidad mayor a 1 nuestro método va a encontrarla ya que este se utiliza en un intervalo de la función, y aunque va a encontrar la misma raiz nuevamente, se utiliza el contenedor "set" que solo permite almacenar un valor una única vez para guardar el resultado, por lo que no se guardan las raíces duplicadas.

2.8. En este método podemos observar que estas características si influyen ya que las funciones cuyas raíces son periódicas tienen un alto impacto en este ya que al no llegar a la tolerancia deseada se realizan infinitas iteraciones por lo que es necesario tomar medidas como limitar el número de iteraciones.

Por otro lado si influye que la función sea par ya que esto permite hallar raíces de una forma más rápida debido a que la característica de estas funciones es que f(x) = f(-x) , por lo que si se encuentra una raíz y la función es par, no es necesario aplicar el método nuevamente para hallar la otra raíz sino que basta con multiplicar el "x" donde está la raíz por -1.

2.9.
2.10. Tolerancia (8, 16, 32, 56) vs número de iteraciones:

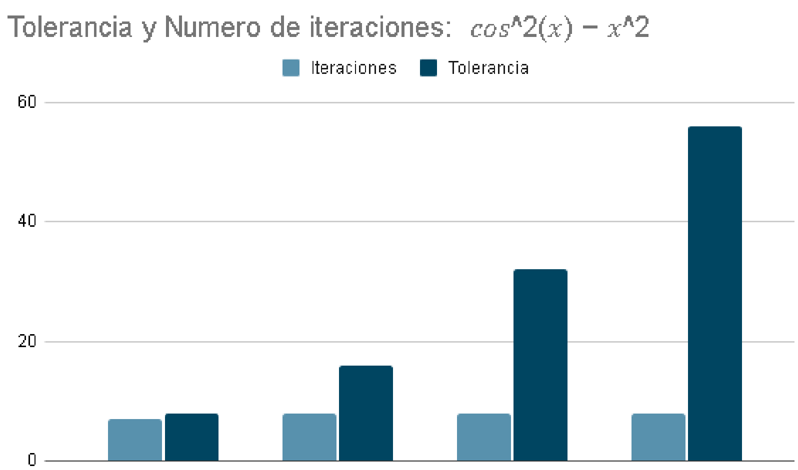


Figura 21: Punto A Tolerancia e iteraciones

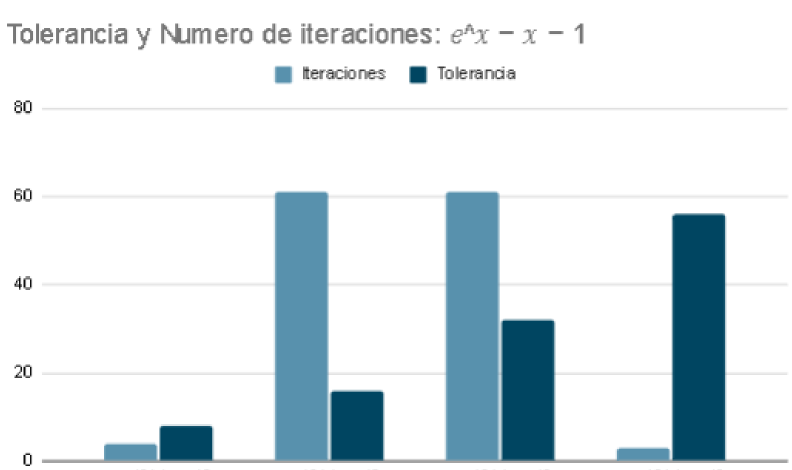


Figura 22: Punto B Tolerancia e iteraciones

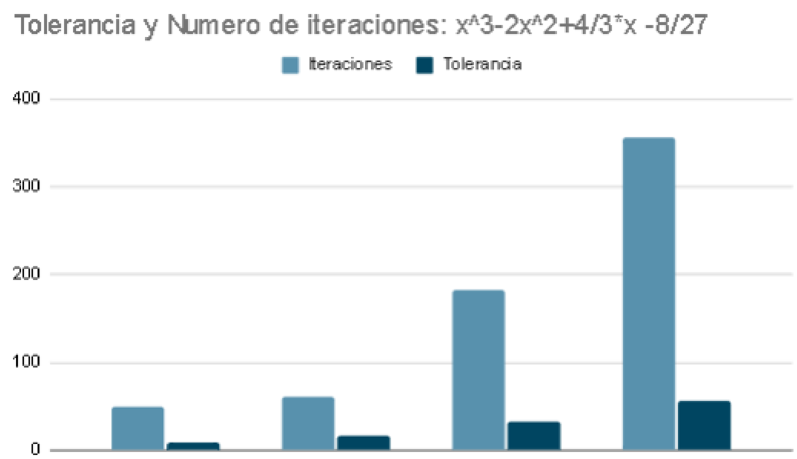


Figura 23: Punto C1 Tolerancia e iteraciones

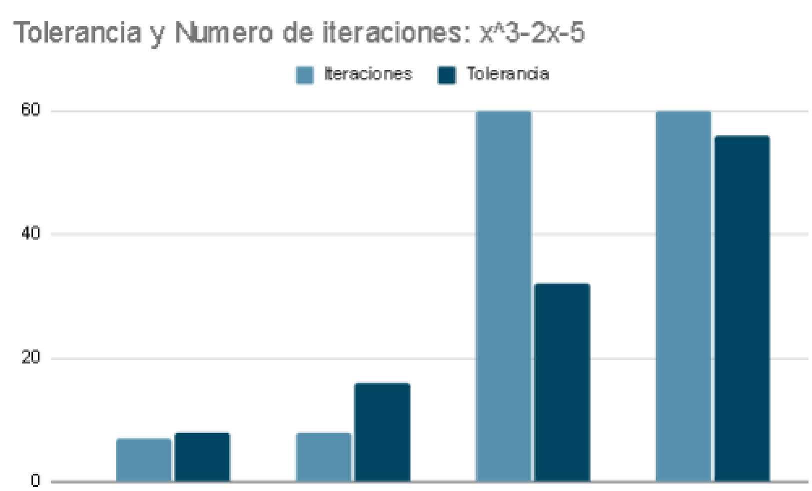


Figura 24: Punto C2 Tolerancia e iteraciones

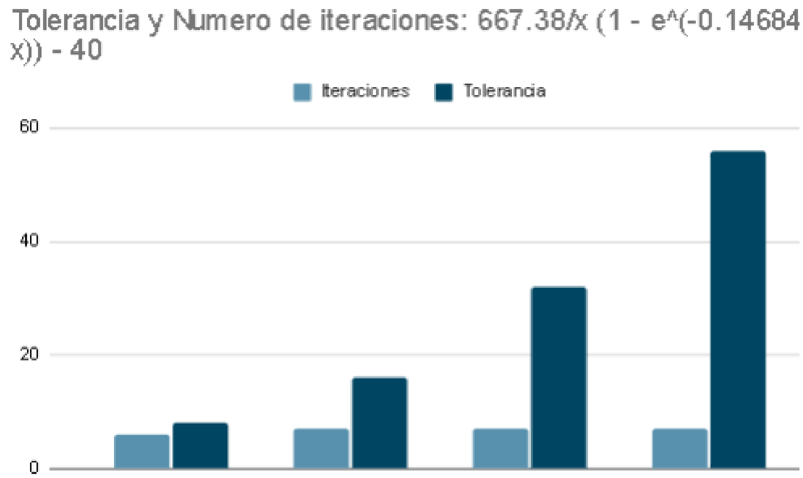


Figura 25: Punto D Tolerancia e iteraciones

2.11. Comparación con bisección:

$f(x) = cos(x)^2 - x^2$	
Iteraciones Muller	Iteraciones Bisección
7	54
$f(x) = e^x - x - 1$	
15	213
$f(x) = x^3 - 2x^2 + 4/3 * x - 8/27$	
50	213
$f(x) = x^3 - 2x - 5$	
8	187
667.38/x (1 - e^(-0.146843 x)) - 40	
6	52

Figura 26: Muller y Biseccion

2.12. Comparación de comportamiento con Taylor

$f(x) = cos(x)^2 - x^2$	
Muller	Taylor
-0.73908513 0.73908513	-2.0, 2.0
$f(x) = e^x - x - 1$	
0.00000000	0.0
$f(x) = x^3 - 2x^2 + 4/3 * x - 8/27$	
0.66666466+0.00000350j	0.5635433974413834
$f(x) = x^3 - 2x - 5$	
2.09455148	2^{-16}
667.38/x (1 - e^(-0.146843 x)) - 40	
14.78020859	NaN

Figura 27: Muller y Taylor

3. Algoritmo delta cuadrado de Aitken

3.1. Para la aplicación del método se deben de tener como entradas un punto elegido por la persona que ejecuta el método esto va influir en el número de iteraciones ya que si está más cerca de la raíz van a ser menos iteraciones, y la otra entrada es la tolerancia que se indique.

3.2. No debe, es un método acelerado.

3.3. Diagrama de flujo:

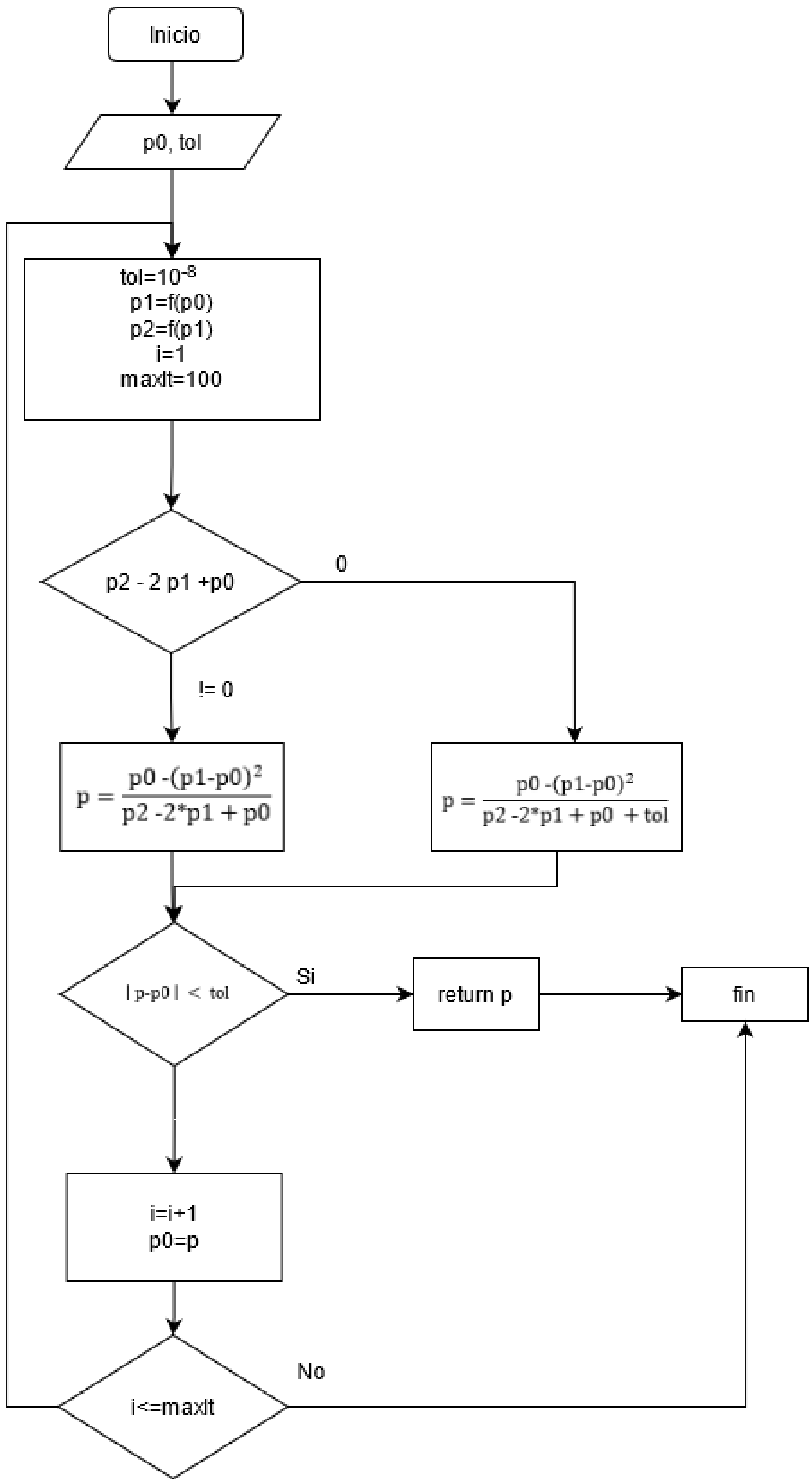


Figura 28: Diagrama de flujo del método de Aitken

3.4. Resultados esperados (Wolfram Alpha):

Resultados de Wolfram Alpha	
f(x)=$cos^2(x) - x^2$	x≈0.73908513321516064165531208767387340401340 x≈-0.73908513321516064165531208767387340401340
f(x)=$x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$	x≈1.65500422706701631495798813801394627890112331
f(x)=$x^3 - 2x - 5$	x≈2.0945514815423265914823865405793030
f(x)=$e^x - x - 1$	x = 0
f(x)=$(\frac{667.38}{x})(1\text{-}e^{-0.146843x})$-40	x ≈ 14.7802085936794677513171226458345398851

Figura 29: Resultados Esperados.

Aplicación del método y resultados obtenidos:

f(x)=$cos^2(x)$-x^2		
Raíz	Tolerancia	iteraciones
0.73908513, -0.73908513	10^{-8}	5
-0.7390851332151607, 0.7390851332151607	10^{-16}	6
-0.73908513321516067229310920083662, 0.73908513321516067229310920083662	10^{-32}	6
- 0.73908513321516067229310920083662495017051696777343750000, 0.73908513321516067229310920083662495017051696777343750000	10^{-56}	6

Figura 30: Resultados punto a - Aitken

f(x)=e^x-x-1		
Raíz	Tolerancia	iteraciones
0	10^{-8}	1
4.0476675319963296^{-6}	10^{-16}	20
x	10^{-32}	x
0	10^{-56}	22

Figura 31: Resultados punto b - Aitken

$f(x)=x^3-2x^2+\frac{4}{3}x-\frac{8}{27}$		
Raíz	Tolerancia	iteraciones
0.66630621	10^{-8}	21
	10^{-16}	
	10^{-32}	
	10^{-56}	

Figura 32: Resultados punto c - Aitken

$f(x)=x^3-2x-5$		
Raíz	Tolerancia	iteraciones
2.09455148	10^{-8}	7
2.0945514815423265	10^{-16}	8
2.09455148154232650981043661886360	10^{-32}	8
2.094551481542326509810436618863604962825775146484375000000	10^{-56}	8

$f(x)=(\frac{667.38}{x})(1-e^{-0.146843x})-40$		
Raíz	Tolerancia	iteraciones
14.78020859	10^{-8}	6
14.7802085936794665	10^{-16}	7
14.78020859367946826523620984517038	10^{-32}	7
14.7802085936794664888793704449199140071868896484375000000000	10^{-56}	7

Figura 33: Resultados punto d - Aitken

3.5. Al revisar el número de iteraciones con respecto a la tolerancia deseada observamos que al disminuir la tolerancia sus iteraciones aumentan haciendo que la significancia mejore, pero en alguno de los casos en donde se encuentra una raíz pero su valor es periodico el problema es que no se podrá almacenar por la condición del número haciendo que alguno de sus números se redondee obteniendo un valor erróneo.

3.6. Una posible solución al problema de la pérdida de significancia puede ser el uso de una tolerancia muy baja ya que esto hace que el programa tengan en cuenta una mayor cantidad de decimales al momento de realizar la validación de la respuesta, además se puede utilizar variables que almacenen una mayor cantidad de información, pero si está destinado al fracaso en donde se debe almacenar un número periodico ya que por su condición no se va a poder manipular con exactitud haciendo que se pierda significancia.

3.7. Lo que pasa con el método cuando existen más de dos raíces es que con cada nuevo valor con el que se evalúa y que es ingresado, lo que hace es encontrar la raíz más próxima al numero suministrado, por lo tanto hay que hacer varias iteraciones con un rango de valores de entrada para encontrar las raíces para encontrar las raíces que existen en ese espacio.

3.8.Con el método encontramos que las funciones periódicas influyen sobre este ya que con este método cada vez se busca una raíz depende del valor tomado hacía la raíz más cercana al valor, además influyen las funciones par e impar ya que si se toman valores muy grandes y si se hacen pocas iteraciones no se podra encontrar la raiz de estas funciones.

3.9.

3.10. A continuación se muestran las gráficas donde se ve el comportamiento del método con respecto a la Tolerancia e iteraciones

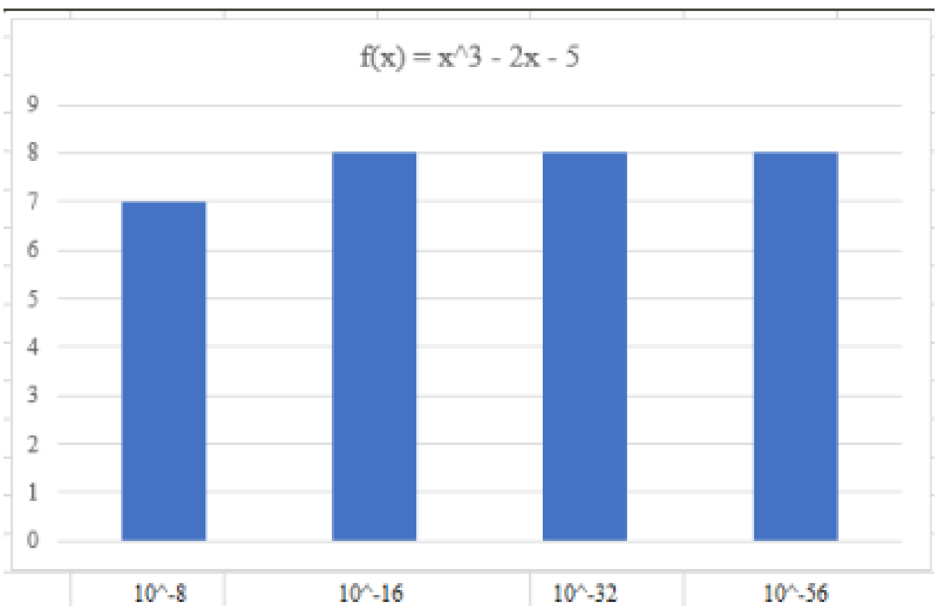


Figura 34: Tolerancia e iteraciones Aitken Punto a

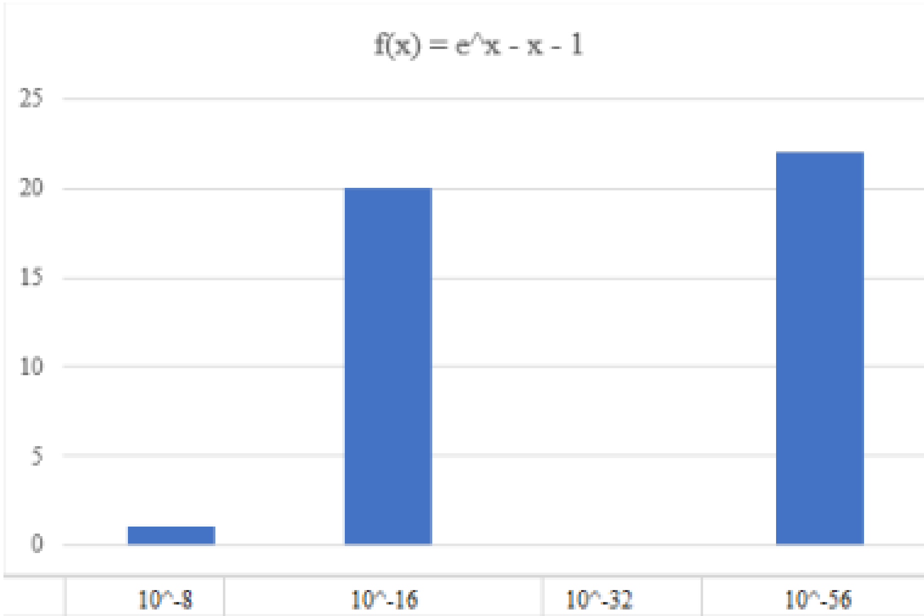


Figura 35: Tolerancia e iteraciones Aitken Punto b

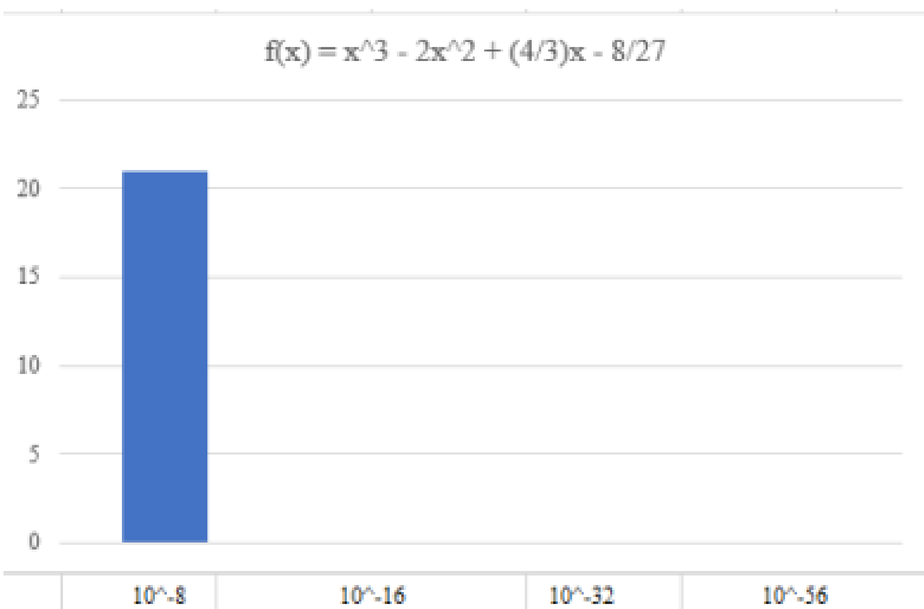


Figura 36: Tolerancia e iteraciones Aitken Punto c

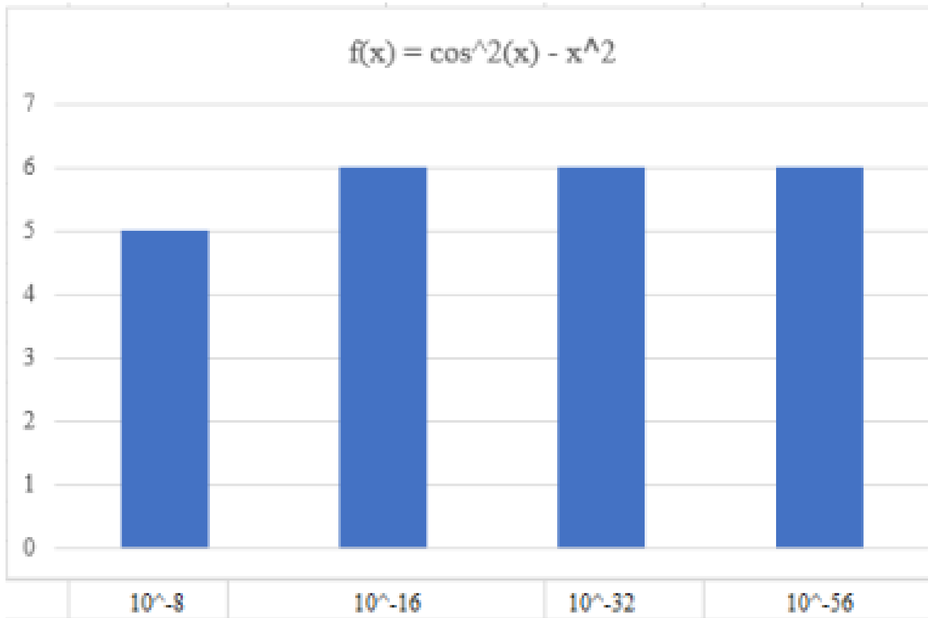


Figura 37: Tolerancia e iteraciones Aitken Punto c_2

3.11. Comparación con bisección:

Comparación	
Iteraciones Aitken	Iteraciones Bisección
f(x)=$cos^2(x) - x^2$	
6	54
f(x)=$x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$	
21	213
f(x)=$x^3 - 2x - 5$	
7	213
f(x)=$e^x - x - 1$	
22	187
f(x)=$(\frac{667.38}{x})(1-e^{-0.146843x})-40$	
7	52

Figura 38: Aitken vs Bisección

3.12.La forma en la que se comporta el método comparado con Taylor es que encontramos que por el método de taylor es menos exacto ya que la implementación realizada solo se realizan polinomios de grado 2 y por este motivo el encontrar las raíces puede llegar a ser más inexacto en comparación con el método de Aitken, esto se puede evidenciar en la tabla que los resultados de una función que las raíces tienen una separación de aproximada de 1.3, además encontramos que al realizar la comparacion con el problema d el método de Taylor es incapaz de encontrar la raíz ya que puede ser por lo que no puede hacer una función de grado 2 para encontrar la raíz de este problema.

Comparación raíces	
Aitken	Taylor
f(x)=$cos^2(x) - x^2$	
0.73908513, -0.73908513	-2.0, 2.0
f(x)=$x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$	
0.66630621	0.563532974413834
f(x)=$x^3 - 2x - 5$	
2.0945514815423265	2e-16
f(x)=$e^x - x - 1$	
0	0.0
f(x)=$(\frac{667.38}{x})(1\text{-}e^{-0.146843x})$-40	
14.7802085936794665	NaN

Figura 39: Aitken y Taylor

4. Bibliografía

[1] Burden Richard L y Faires J. Douglas, Numerical Analysis. Cengage Learning, 9ª. 2012.
[2] O'Connor José Luis, Ingeniería de los Algoritmos y Métodos Numéricos. 2ª edición. 2017.