Análisis Numérico: Taller 1

Diana Sofía Carrillo   Nelson Alejandro Mosquera   Sergio Esteban Triana

Pontificia Universidad Javeriana
{ds-carrillog, nelson.mosquera, triana_se}@javeriana.edu.co
23 de agosto de 2021

Índice

Índice de figuras

This work is shared under a CC BY-SA 4.0 license unless otherwise noted

1. Introducción

En el análisis numérico, existen varios tipos de algoritmos de búsqueda de la raíz o raíces de una función. Estos algoritmos o métodos numéricos principalmente se utilizan para encontrar una solución aproximada de la ecuación dada por la expresión f (x) = 0 para una función matemática dada f. Este articulo se enfocara primordialmente en los algoritmos de Bisección, Aitken y Regula falsi, teniendo un énfasis principal en este ultimo.

2. Bisección

El método de bisección, también conocido como búsqueda binaria, se basa en el teorema del valor intermedio. Teniendo una función continua f definida en el intervalo [a, b] con f(a) y f(b) de signos diferentes. De acuerdo al Teorema del Valor Intermedio, existe un número P en (a, b) tal que f(p) = 0. Este método se aplica aunque exista más de una raíz en el intervalo (a, b), pero por razones de simplicidad se asume que la raíz en el intervalo es única. El método requiere de un intervalo inicial (a, b), se procede a hacer uso de la fórmula: cuyo valor será el siguiente valor de b o de a (dependiendo de en cuál de los dos intervalos esté la raíz), para después aplicar de nuevo la fórmula del valor intermedio:

$$f(a)f(b) < 0$$

y confirmar la existencia de la raíz en el nuevo intervalo. Estas iteraciones se repiten hasta que se haya encontrado el valor real con la tolerancia esperada o hasta que se haya realizado el número máximo de iteraciones. Al final se tendrá un intervalo tan pequeño que cualquier valor será una buena aproximación al valor real de la raíz.

A continuación una representación geométrica del método:

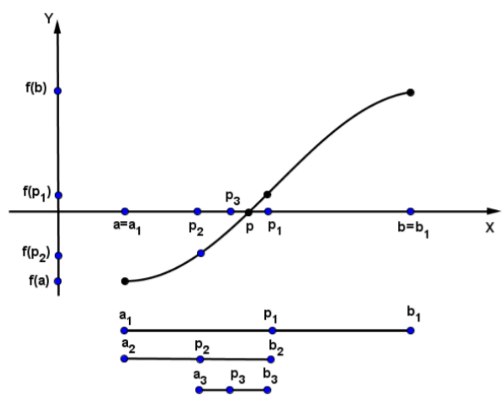


Figura 1: Diagrama Método de Bisección

El método de bisección es un método robusto que converge en todos los casos, mas no es un método eficiente ya que no usa ninguna información sobre el comportamiento de la función.

3. Algoritmo 2 de Aitken

El método 2 de Aitken es un método de aceleración de la convergencia, lo que nos dice que es muy útil para acelerar la convergencia de una sucesión que converge linealmente. Cuando este se aplica a una sucesión obtenida mediante una iteración de punto fijo se conoce como Método de Steffensen.

This work is shared under a CC BY-SA 4.0 license unless otherwise noted

3.1. Definición

Supóngase que:

$$x = (x_n)_{n\in\mathbb{N}}$$

Se calcula la nueva sucesión definida por:

$$\hat{x}_{n+2} = x_{n+2} - \frac{(x_{n+2} - x_{n+1})^2}{x_{n+2} - 2\,x_{n+1} + x_n}$$

Si se emplea el operador de las diferencias progresivas definido como:

$$\Delta x_n = x_{n+1} - x_n$$

$$\Delta^2 x_n = \Delta(\Delta x_n) = x_{n+2} - 2x_{n+1} + x_n$$

También puede escribirse de la siguiente forma:

$$\hat{x}_{n+2} = x_{n+2} - \frac{(\Delta x_{n+1})^2}{\Delta^2 x_n}$$

3.2. Propiedades

El proceso ² de Aitken es un método de aceleración de la convergencia, y en particular un caso de transformación no lineal de una sucesión.

x converge linealmente a $\ell siexisteunnúmero (0,1) talque$

$$\lim_{n\to\infty} \frac{|x_{n+1} - \ell|}{|x_n - \ell|} = \mu$$

El método de Aitken acelerará la sucesión

$$\lim_{n\to\infty} \frac{\hat{x}_n - \ell}{x_n - \ell} = 0$$

Aunque la nueva sucesión no converge en general de forma cuadrática, se puede demostrar que para un método de punto fijo, es decir, para una sucesión

$$x_{n+1} = f(x_n) x_{n+1} = f(x_n)$$

para alguna función iterada f, convergiendo hacia un punto fijo, la convergencia es cuadrática. En este caso, la técnica se conoce como método de Steffensen.

4. Posición Falsa

4.1. Método numérico

Regula falsi o posición falsa, es un método muy parecido al método de la bisección. La diferencia es que no toma el punto medio del intervalo, sino que parte de dos puntos $[x_i, x_j]$ y consiste en encontrar X_r como el punto que corta el eje x resultado de la linea formada por los punto $(x_i, f(x_i))$ y $(x_f, f(x_f))$. Esta linea, es la recta secante a la curva que pasa por estos dos puntos, la cual se puede escribir como $y = mx + p$, siendo

$$p = \frac{f(x_f) - f(x_i)}{x_f - x_i}$$

$$m = \frac{f(x_i)x_f - f(x_f)x_i}{x_f - x_i}$$

This work is shared under a CC BY-SA 4.0 license unless otherwise noted

$$mx_2 + p = 0$$

Al despejar esta tercera ecuación podemos obtener como resultado la ecuación principal del método regula falsi.

$$x_2 = -\frac{p}{m} = \frac{f(x_f)x_i - f(x_i)x_f}{f(x_f) - f(x_i)} = x_f - \frac{f(x_f)}{f(x_f) - f(x_i)}(x_f - x_i)$$

Cabe destacar que el método de Regula falsi converge más rápidamente que el de bisección debido a que al permanecer uno de sus valores iniciales fijo el número de cálculos se reduce mientras que el otro valor inicial converge hacia la raíz.

4.2. Condiciones

Para que el método de regula falsi pueda iniciar, se debe de validar la siguiente condición:

$$f(x_i) * f(x_f) < 0$$

Si esta condición llega a ser falsa significara que la solución no se encuentra en el intervalo propuesto y no se podrá utilizar este método. Así mismo durante todas las iteraciones este método exige que se mantenga el horquillado. Para poder mantener esta condición, se debe revisar en cada iteración la siguiente condición, para así poder determinar que punto se moverá.

$$f(x2) * f(x_f) < 0$$

Si esta condición llega a ser verdadera, se le asignara a x_f el valor de $x2$ y a x_i el valor de x_f. De lo contrario se le asignara a x_f el valor de $x2$ y a x_i el valor de x_i

4.3. Explicación geométrica

Se tiene la función $f(x)$ y los puntos x_u y x_l. Si se gráfica una linea entre estos dos puntos, el punto donde esta linea corte con el eje x va a ser nuestra solución x_r. Como se puede observar en la siguiente figura.

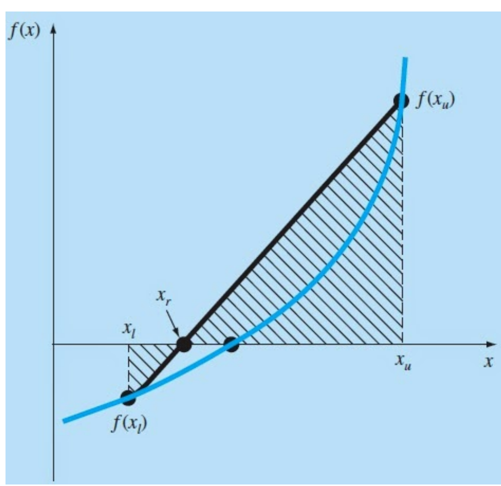


Figura 2: diagrama de regula falsi

Para encontrar esta solución se utilizara la propiedad de los triángulos semejantes, en donde nos dicen que la razón entre lados homólogos va a ser igual.

$$\frac{a}{b} = \frac{c}{d}$$

This work is shared under a CC BY-SA 4.0 license unless otherwise noted

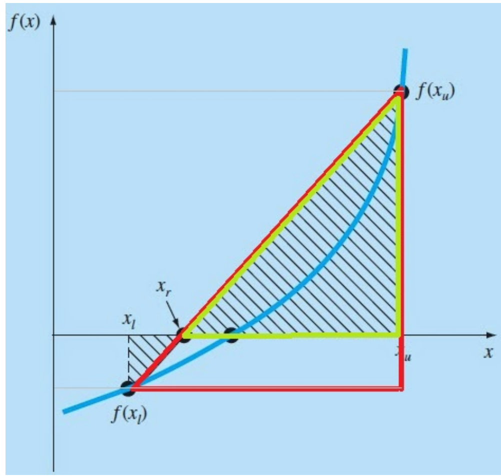


Figura 3: diagrama de regula falsi

Con esto presente y guiándonos de la figura 2, podemos observar que los lados de los triangulo van a ser:

$$h_r = f(x_u) - f(x_l)$$

$$b_r = x_u - x_l$$

$$h_v = f(x_u)$$

$$b_v = x_u - x_r$$

Ya teniendo los lados estipulados y utilizando la propiedad previamente mencionada podemos llegar a la siguiente ecuación:

$$\frac{x_u - x_r}{x_u - x_l} = \frac{f(x_u)}{f(x_u) - f(x_l)}$$

Ahora despejando para x_r podemos llegar a nuestra ecuación principal.

$$x_r = x_u - \frac{f(x_u) * (x_u - x_l)}{f(x_u) - f(x_l)}$$

4.4. Validación de resultados

En esta sección se validaran los resultados del algoritmo de Regula falsi. Para esto se utilizara el algoritmo implementado en python en conjunto con la herramienta graficadora de Symbolab. Además, El calculo de los errores se realiza teniendo en cuenta los intervalos planteados para hallar las raíces y las siguientes formulas.

Se valida y verifica inicialmente que la raíz exista, seguido de esto vamos a calcular la aproximación por el método de regula falsi. Por lo que:

$$C_0 = Aproximación$$

Error hacia Adelante:

$$|f(C_0)|$$

Error hacia Atrás:

$$|raiz - C_0|$$

This work is shared under a CC BY-SA 4.0 license unless otherwise noted

4.4.1. $\cos(x)^2 - x^2; [0, 1]$

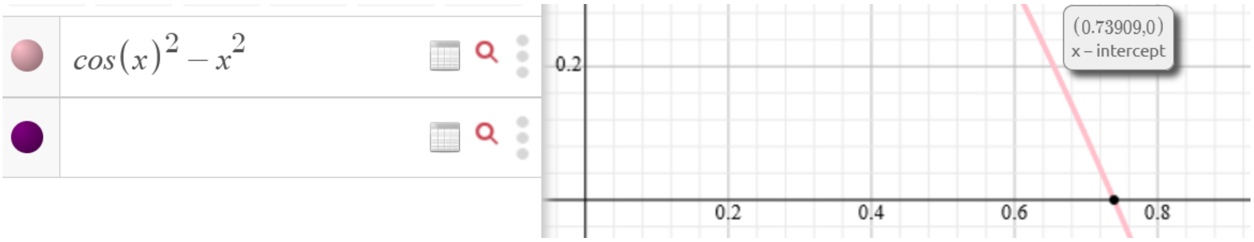


Figura 4: Resultados brindados por Symbolab para $\cos(x)^2 - x^2$

Tolerancia	Solucion
10ˆ-8	0.73908513
10ˆ-16	0.7390851332151607
10ˆ-32	0.73908513321516067229310920083662
10ˆ-56	0.73908513321516067229310920083662495017051696777343750000

Cuadro 1: Aproximaciones método Posición Falsa

Error Adelante	Error atrás
0.00000000795393	0.0
0.0	1.000*10ˆ-16
0.000000000000000007579	3.063779711316275*10ˆ-17
0.000000000000000007579	3.0637797113162751546157105208872680046*10ˆ-17

Cuadro 2: Calculo Errores

4.4.2. $x\sin(x) - 1; [-1, 2]$

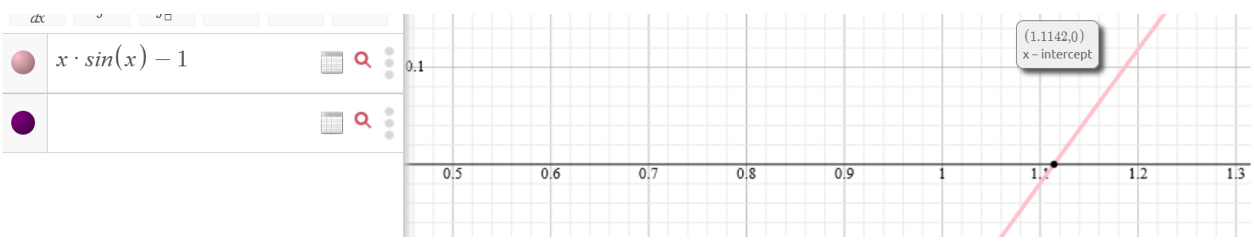


Figura 5: Resultados brindados por Symbolab para $x\sin(x) - 1$

Tolerancia	Solucion
10ˆ-8	1.11415714
10ˆ-16	1.1141571408719302
10ˆ-32	1.11415714087193018499988284020219
10ˆ-56	1.11415714087193018499988284020218998193740844726562500000

Cuadro 3: Aproximaciones método Posición Falsa

This work is shared under a CC BY-SA 4.0 license unless otherwise noted

Error Adelante	Error atrás
0.00000000121094468534	0.0
0.00000000000000015652	2*10^-16
0.00000000000000013569	9.769935766203299*10^-17
0.00000000000000013569	9.769935766203298607798330707121624940467*10^-17

Cuadro 4: Calculo Errores

4.4.3. $x^3 - 2x^2 + \frac{4x}{3} - \frac{8}{27}$; $[0, 1]$

Figura 6: Resultados brindados por Symbolab para $x^3 - 2x^2 + \frac{4x}{3} - \frac{8}{27}$

Tolerancia	Solucion
10^-8	0.68122222
10^-16	0.6713466451000216
10^-32	0.66815401758405701393428444134770
10^-56	0.66713773682366894757223008127766661345958709716796875000

Cuadro 5: Aproximaciones método Posición Falsa

Error Adelante	Error atrás
0.00000308	0.01455553...
0.0000001025018149	0.00467997843335493...
0.00000000329033666534949259512072	0.00148735091739034726761777468103...
0.00000000010453380905376732948486280765309618072551974193	0.000471070157002280905563414610999933...

Cuadro 6: Calculo Errores

4.4.4. $x^3 - 2x - 5$; $[2, 3]$

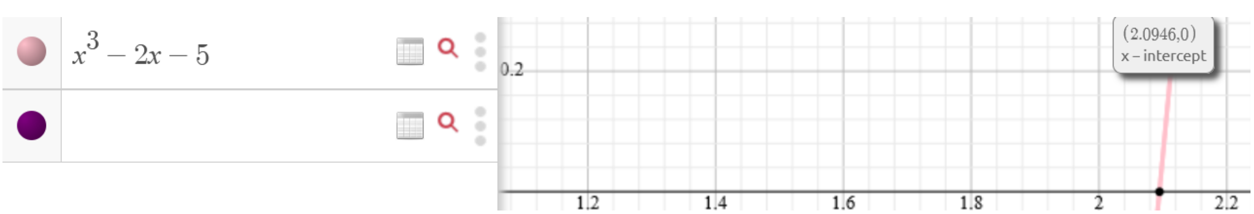


Figura 7: Resultados brindados por Symbolab para $x^3 - 2x - 5$

This work is shared under a CC BY-SA 4.0 license unless otherwise noted

Tolerancia	Solucion
10ˆ-8	2.09455148
10ˆ-16	2.0945514815423265
10ˆ-32	2.09455148154232650981043661886360
10ˆ-56	2.09455148154232650981043661886360496282577514648437500000

Cuadro 7: Aproximaciones método Posición Falsa

Error Adelante	Error atrás
0.00000001	0.0
0.0000000000000001	0.0
0.00000000000000091158	0.00000000000000008167
0.00000000000000091158	0.00000000000000008167194992171569800

Cuadro 8: Calculo Errores

Como se puede observar cada respuesta del algoritmo concuerda con la intersección del eje x en cada función. También cabe resaltar como llega a ser mas preciso que la herramienta graficadora y como en algunos casos se necesito una mayor tolerancia para poder llegar al resultado indicado.

4.5. Comportamiento del método

4.5.1. Perdida de Significancia

La perdida de Significancia se basa en que representa magnitudes mayores a cambio de reducir la precisión después del punto decimal, La perdida de Significancia en Posición Falsa se produce en las operaciones en la que el resultado es mayor al de los operandos. En el Método de Posición Falsa presentamos una perdida de significancia baja, exceptuando el caso de la función: $x^3 - 2x^2 + \frac{4x}{3} - \frac{8}{27}$, debido a que en los resultados se evidencia que a mayor tolerancia menor precisión. Con respecto a Aitken encontramos que la perdida de significancia es mucho menor en Posición Falsa.
Para el caso donde se presente el problema de significancia se podría decir que esta destinado al fracaso por que para lograr conseguir una precisión aceptable tenemos que aumentar la tolerancia a un numero enorme, en el punto donde nos da perdida de significancia es debido a que la raíz de función es periódica por lo que se harán infinitas iteraciones para obtener infinita cantidad de decimales repetidos.

4.5.2. Iteraciones

Con respecto a las iteraciones, se evidencio principalmente que con una tolerancia baja como lo es10^-8 se obtenía un menor numero de iteraciones a comparación de las tolerancias mas altas, además arroja un numero de iteraciones infinito como resultado en la función: $x^3 - 2x^2 + \frac{4x}{3} - \frac{8}{27}$, debido a que la raíz de la función es periódica dando $0,\overline{666}$.
con este método podemos decir que para las funciones pares se van a dar un numero menor de iteraciones con respecto a las impares que demuestran un numero mas alto de iteraciones, por otro lado cuando la función es periódica las iteraciones tenderán al infinito como anteriormente vimos.

4.5.3. Convergencia

El método de Posición Falsa conlleva un orden de convergencia lineal por lo que para nuestro caso podemos decir que el método es eficiente en cuanto a la convergencia. Podemos comprobar el orden de convergencia con la siguiente formula:

$$0 \leq \lim_{i \to \infty} \frac{|x_{i+1} - x^*|}{|x_i - x^*|^r} < \infty$$

This work is shared under a CC BY-SA 4.0 license unless otherwise noted

4.6. Número de Raíces

En caso de tener una función con dos o más raíces, se recomienda escoger el intervalo inicial de tal forma que solo contenga la raíz que se quiere encontrar. En caso de dar como intervalo inicial uno con dos o más raíces, el método encontrará una de ellas, más no habrá forma de especificar cuál, sino que se sabrá al correr el algoritmo. Para explicar esto se puede usar como ejemplo la función $f(x) = \cos(x)$, usando el intervalo $(a, b) = \frac{\pi}{4}, \frac{11\pi}{4}$.

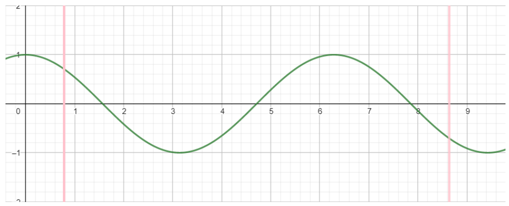


Figura 8: Gráfica función coseno de x

Como se puede observar, en este intervalo existen 3 raíces. Cuando se aplica el método de posición falsa converge en la raíz con valor 4.71, la razón de esto es que el método encuentra el intercepto con el eje x de la línea recta conectada por los puntos $((a, f(a), (b, f(b))$:

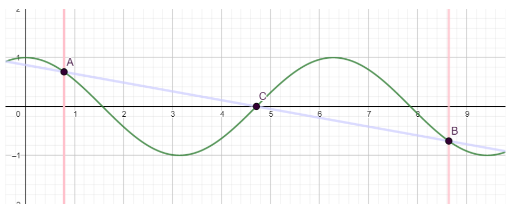
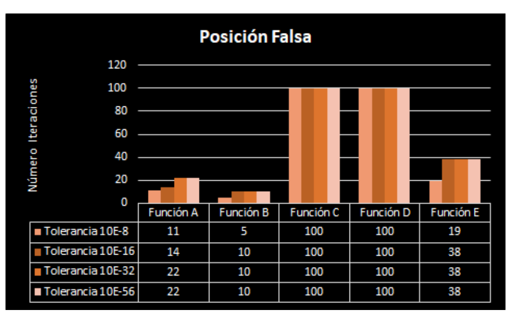


Figura 9: Gráfica función coseno de x

Como se observa, al hacer la intersección de la recta con el eje x, se halla el valor 4.7123889803847 en x. En conclusión, el método si acepta el intervalo una función con múltiples raíces, sin embargo, solo devuelve el valor de una de ellas.

4.7. Implementación

El método de la posición falsa se ejecutó con cinco funciones diferentes y con cuatro tolerancias distintas. Las funciones son: $f(x) = \cos x^2 - x^2$ (Función A), $f(x) = x \operatorname{sen} x - 1$ (Función B), $f(x) = x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$ (Función C), $f(x) = \frac{68{,}1 * 9{,}81}{x}(1-\epsilon)^{-\frac{10}{68{,}1}x} - 40$ (Función D) y $f(x) = x^3 - 2x - 5$ (Función E).



	Función A	Función B	Función C	Función D	Función E
Tolerancia 10E-8	11	5	100	100	19
Tolerancia 10E-16	14	10	100	100	38
Tolerancia 10E-32	22	10	100	100	38
Tolerancia 10E-56	22	10	100	100	38

Figura 10: Resultados Método Posición Falsa. El valor 100 corresponde al número máximo de iteraciones

En la figura anterior, vemos una representación gráfica de la implementación del método. Se evaluó con cada función y con cuatro tolerancias diferentes. Durante la implementación, las funciones con las que el

This work is shared under a CC BY-SA 4.0 license unless otherwise noted

método más se esforzó fueron la C y la D, sobretodo la D, en la cual no pudimos encontrar un número exacto de iteraciones en ninguna de las tolerancias.

4.8. Comparación

Para comparar la eficacia y desempeño del método de la posición falsa, ejecutamos los métodos de bisección y el algoritmo de Aitken con las mismas funciones y los mismos intervalos. A continuación las gráficas con los números de iteraciones.

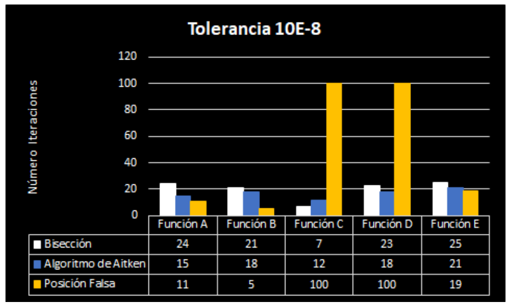


Figura 11: Número de iteraciones con tolerancia de E10-8

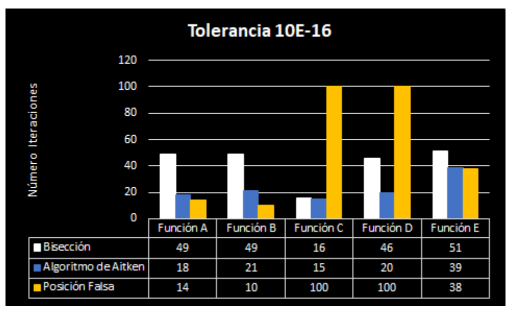


Figura 12: Número de iteraciones con tolerancia de E10-16

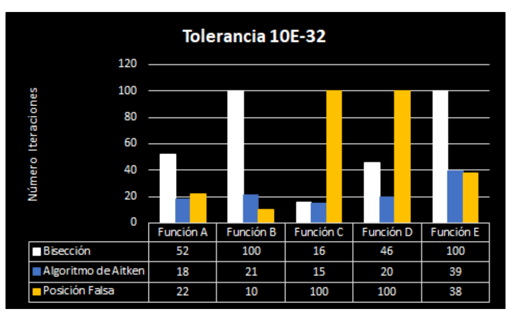


Figura 13: Número de iteraciones con tolerancia de E10-32

This work is shared under a CC BY-SA 4.0 license unless otherwise noted

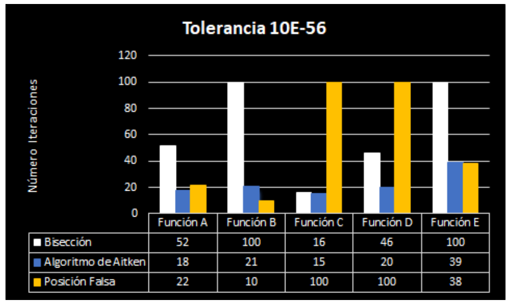


Figura 14: Número de iteraciones con tolerancia de E10-56

Analizando las gráficas, es evidente la eficacia del algoritmo de Aitken junto con su flexibilidad en términos de las funciones. El algoritmo de Aitken mantuvo el menor número de iteraciones durante la gran mayoría de las pruebas y este número no se vio tan afectado por el incremento de la tolerancia.

Sobre el método de bisección, podemos resaltar su flexibilidad al funcionar con el mismo esfuerzo independientemente del tipo de función. También es notoria la influencia del incremento de la tolerancia en el número de iteraciones, llegando al punto de superar un número máximo de iteraciones de más de 50.000.

Por último, sobre el método de posición falsa, podemos resaltar el poco impacto del aumento de la tolerancia en la implementación. Sin embargo, se queda atrás con algunos tipos de funciones, haciendo que métodos más simples, como el de bisección, lo superen.

Como conclusión de la comparación, el algoritmo de Aitken mostró tanto flexibilidad como eficiencia, mientras que los otros dos dejaron que desear en alguna de estas dos cualidades.
Adicionalmente, otro método de solución a polinomios es el de Taylor. Este método vendría siendo el menos flexible entre los antes mencionados debido a que solo trabaja con polinomios de grado dos. En algunos casos de prueba (en los que se utilizó coseno y seno), se puede observar una diferencia de hasta 1.2, indicando su baja fidelidad.

This work is shared under a CC BY-SA 4.0 license unless otherwise noted