

# Método de la secante

Luisa Contreras      Jairo Vanegas      Gabriel Forero      Nicolás Quintero

5 de septiembre de 2020

## 1. Introducción

El método secante es similar al método de Newton-Raphson, e incluso se puede llegar a pensar que es una derivación del mismo. No obstante, la diferencia radica en que, en vez de calcular la derivada de la función con el objetivo de encontrar la raíz, este se aproxima por una recta secante a la raíz. Por lo tanto, el método de la secante resulta más sencillo de realizar para las funciones cuya derivada no sea fácil de obtener. Una de sus características principales es que la pendiente de la recta es aproximadamente igual a la derivada del punto inicial. Para generar las rectas, el método de la secante necesita como condiciones iniciales dos puntos que pertenezcan a  $f(x)$  y que el dominio de la función sea derivable. Sean  $X_a$  y  $X_b$  pertenecientes a cierta  $f(x)$  se puede definir:

$$f'(x) \approx \frac{\Delta y}{\Delta x} = \frac{f(b) - f(x_a)}{x_b - x_a}$$

Si remplazamos en el método de Newton-Raphson, la ecuación quedaría de la siguiente manera:

$$x_{n+1} = x_n - f(x_n) \cdot \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

## 2. Explicación geométrica del algoritmo

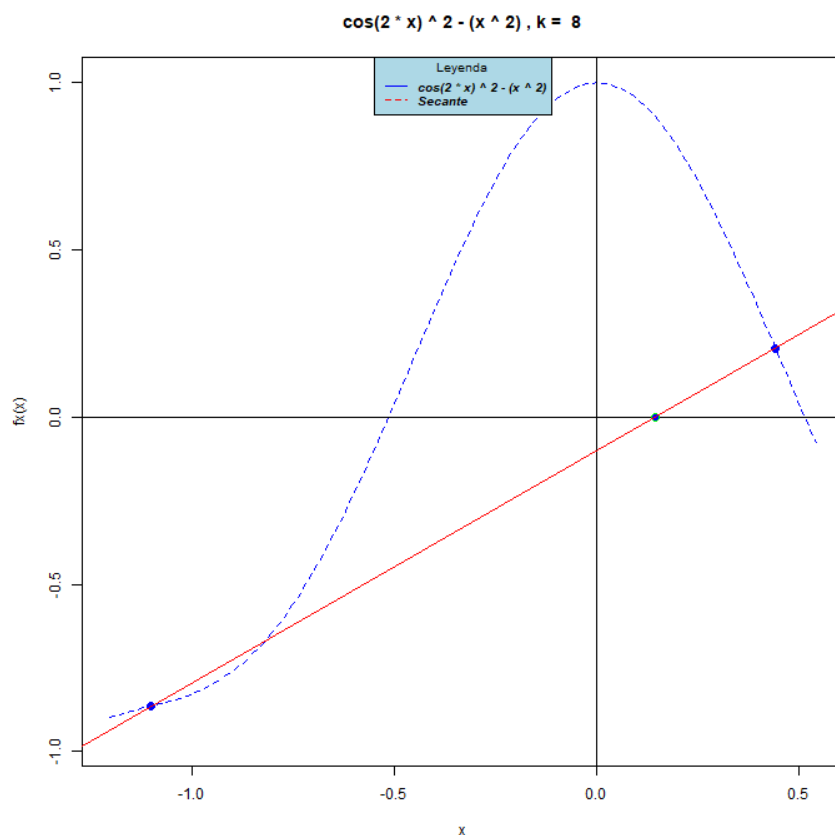


Figura 1: Interpretación geométrica método de la secante

Como podemos ver en la figura 1, el método de la secante trabaja con dos valores iniciales los cuales se aproximan por izquierda o derecha a la raíz independientemente de si están los dos a un lado o a ambos lados de la raíz. Se traza una recta entre las imágenes de los dos valores iniciales en la función a evaluar, esta secante corta al eje  $x$  en algún punto y este punto nos otorga un nuevo valor con el que aproximarnos a la raíz, reemplaza a uno de los valores iniciales y se repite el procedimiento hasta que la aproximación tenga un error menor al de la tolerancia necesaria.

## 3. Algoritmo

Se desarrolló un algoritmo en R-Studio con el fin de analizar el método de la secante y poder compararlo con otros métodos numéricos, junto con las especificaciones requeridas. El algoritmo secante toma dos valores iniciales,  $X_0$  y  $X_1$  con estos puntos establecidos se traza una recta secante, la cual busca cortar el eje  $X$ . Una vez evaluado el valor esperado con el mínimo error que en este caso sería la tolerancia administrada  $10^{-8}$ ,  $10^{-16}$ ,  $10^{-32}$ . El valor final se aprueba una vez la tolerancia sea igual o menor a la que se administra. En caso de

que no sea así el algoritmo será iterado 100 veces, y luego de ello, si aún no es óptima la tolerancia, devolveremos la mejor solución. En la ilustración 1 se encuentra el diagrama de flujo que muestra la explicación ya definida.

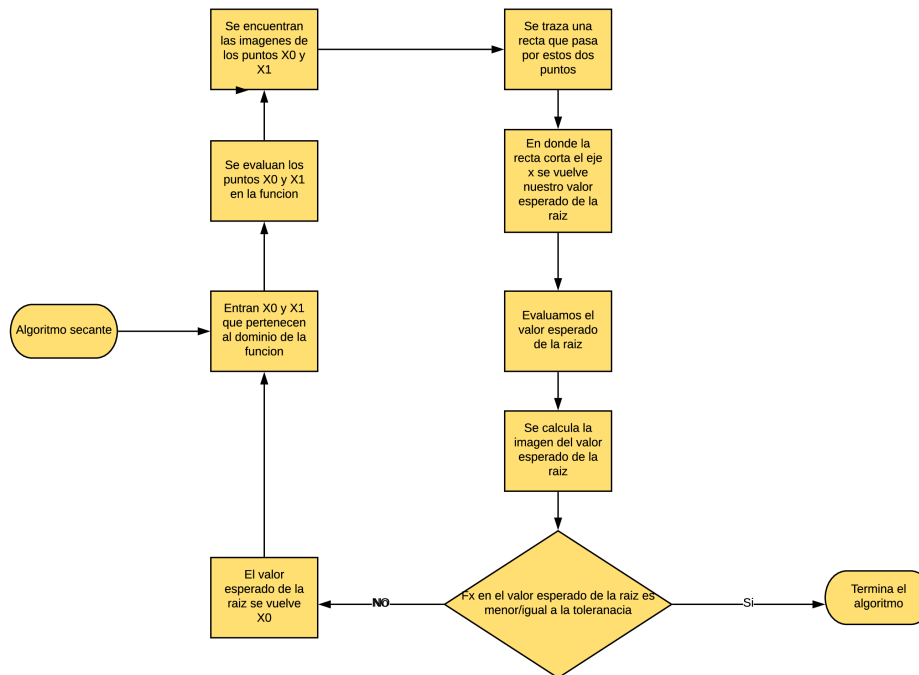


Figura 2: Diagrama de flujo del método.

Entre las funcionalidades que aplicamos realizamos la función de plasmar gráficamente, de qué manera se encuentra cada una de las raíces punto por punto. Podemos ver ese proceso por medio de las siguientes ilustraciones.

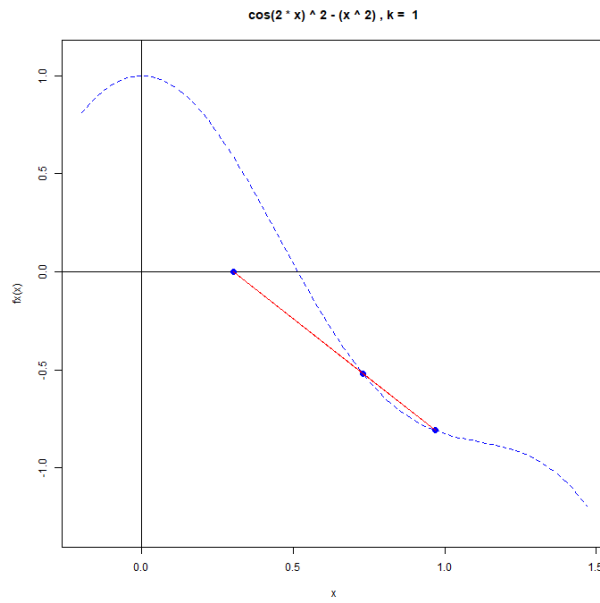


Figura 3: Iteración 1 - Búsqueda de convergencia.

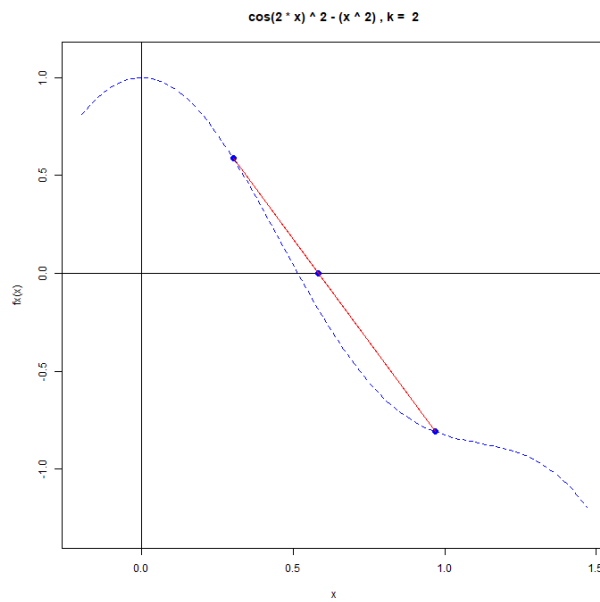


Figura 4: Iteración 2 - Búsqueda de convergencia.

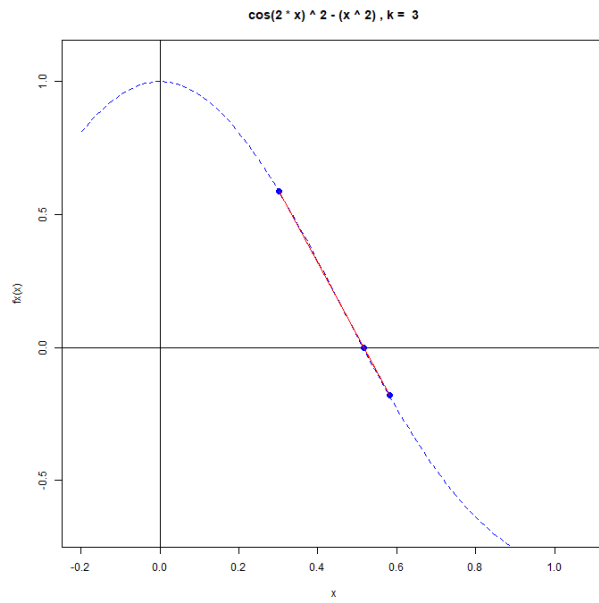


Figura 5: Iteración 3 - Búsqueda de convergencia.

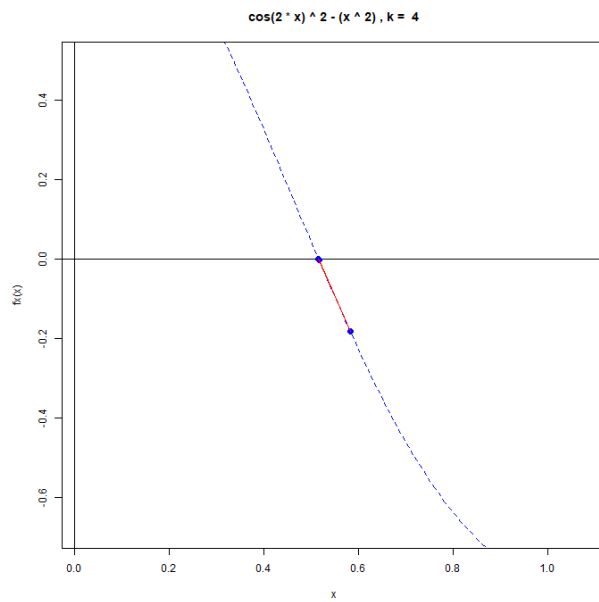


Figura 6: Iteración 4 - Búsqueda de convergencia.

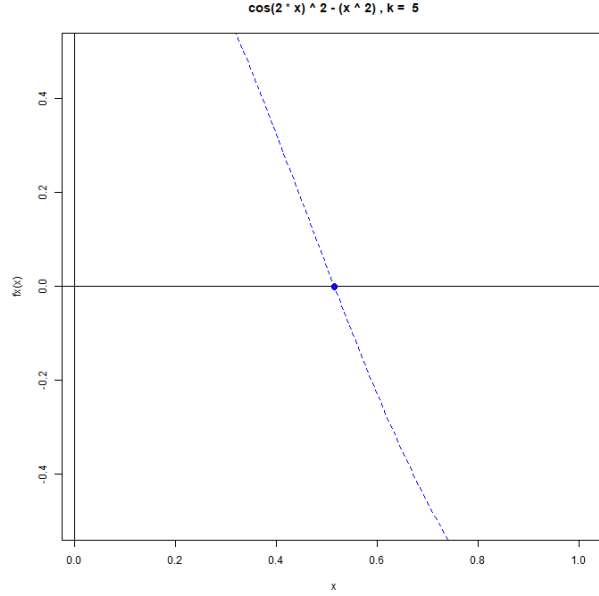


Figura 7: Iteración 5 - Búsqueda de convergencia.

Adicionalmente, graficamos que tan rápido el algoritmo converge por medio de tomar cada raíz con respecto a las iteraciones necesarias para encontrar la raíz más óptima, de esta forma logramos determinar cuando el algoritmo se aproxima a la raíz por derecha (arriba) o por izquierda (abajo) y que tan rápido lo hace. Otras funciones que realiza nuestro código es la comparación de que tan óptimo es el algoritmo de la secante con respecto al de la bisección, dándonos como resultado como se distinguen entre sí.

## 4. Resultados

### 4.1. Raíces

Para validar los valores arrojados por el algoritmo usamos la función uniroot de la librería estándar de R. Para cada función tenemos una misma raíz de uniroot ya que no recibe tolerancias.

#### 4.1.1. $\cos^2(2x) - x^2$

Valor de uniroot para esta función: 0.51493342197240011

Cuadro 1: Resultados de las raíces  $\cos^2(2x) - x^2$ .

$\cos^2(2x) - x^2$	
Magnitud tolerancia	Raíz
-8	0.5149332657
-16	0.514933264661129431
-32	-0.5149332646611294138010592584369104

#### 4.1.2. $x \cdot \sin(x) - 1$

Valor de uniroot para esta funcion: 1.114148296187552

Cuadro 2: Resultados de las raíces  $x \cdot \sin(x) - 1$ .

$x \cdot \sin(x) - 1$	
Magnitud tolerancia	Raíz
-8	1.114157140
-16	1.11415714087193041
-32	1.114157140871930087300525178169204

#### 4.1.3. $x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$

Valor de uniroot para esta funcion: 0.66666566840120922

Cuadro 3: Resultados de las raíces  $x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$ .

$x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$	
Magnitud tolerancia	Raíz
-8	0.6661825851
-16	0.666671189416643106
-32	0.6666698708192892000819652787612284

## 4.2. Comportamiento del método

Por la naturaleza del método es posible que converja o no, en nuestros experimentos el método siempre llego a converger, especulamos que es debido a los rangos que escogimos para los valores iniciales. La perdida de significancia se controló usando la mejor versión de la ecuación para encontrar el punto de intersección con el eje x como se ve en la siguiente ecuación:

$$x_2 = x_1 - f(x_1) \cdot \frac{x_1 - x_0}{f(x_1) - f(x_0)}$$

Aunque fue la mejor expresión que pudimos encontrar en la literatura, sigue presentando problemas ya que llegando cada vez mas cerca al valor del épsilon de la maquina la división puede caer en tres casos, ser 0, una forma indeterminada o llegar a un lado del infinito, para los tres casos implementamos un criterio de parada que devuelve la mejor aproximación hasta que se llega a estos casos.

### 4.2.1. Funciones con mas de una raíz

El método secante, sólo puede aproximarse a una raíz en especial con un error mínimo, en este caso dado por una tolerancia determinada. Por lo tanto, es imposible llegar a dos o más raíces por medio de este método y por lo tanto tampoco el alcance del algoritmo está limitado. A diferencia de las herramientas matemáticas más comunes como lo son los solucionadores matemáticos, como, por ejemplo, Wolfram Alpha, Geogebra y otros. Esto se debido a la

capacidad de cómputo, y la capacidad de desarrollo junto con distintas configuraciones que hacen posible este resultado. Como lo podría ser la modificación de secante llamada “Método secante en varias variables”, el cual se basa en encontrar ceros por medio de un jacobiano dado por:

$$\vec{F}(\vec{r}) = \begin{pmatrix} F_1(\vec{r}) \\ \vdots \\ F_n(\vec{r}) \end{pmatrix}$$

donde  $\vec{r} = (x_1, x_2, \dots, x_n)$ . Es decir se busca puntos en todos los  $\vec{r}_0$  en los que todas las funciones  $F_n(\vec{r})$  se anulan simultáneamente. Visto lo anterior, vemos como el método modificado se sale totalmente de contexto y por lo tanto es inviable su implementación como solución.

### 4.3. Convergencia

Como se dijo anteriormente el caso llego a converger en todos los casos, a continuación presentamos las ilustraciones para la función  $x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$ .

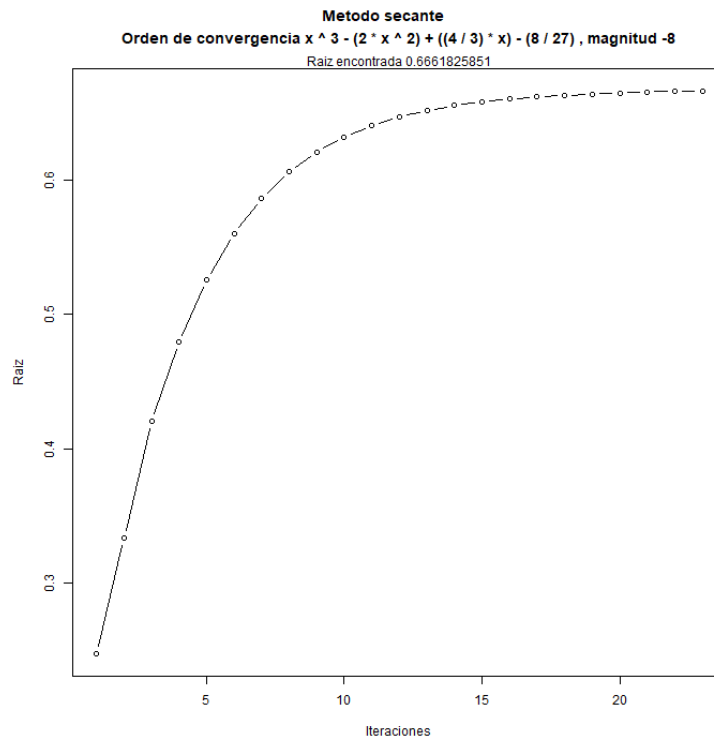


Figura 8: Método de la secante, magnitud -8,  $x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$ .



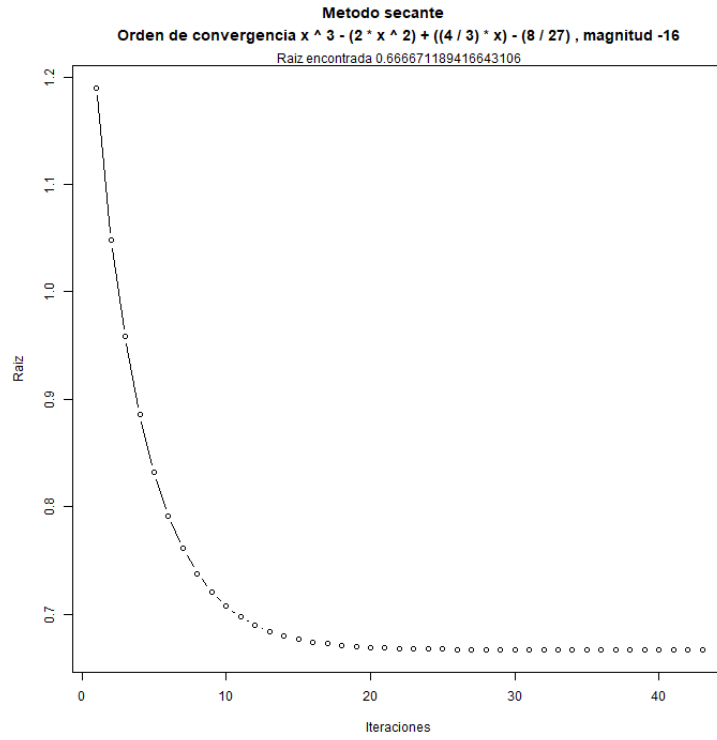


Figura 9: Método de la secante, magnitud -16,  $x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$ .

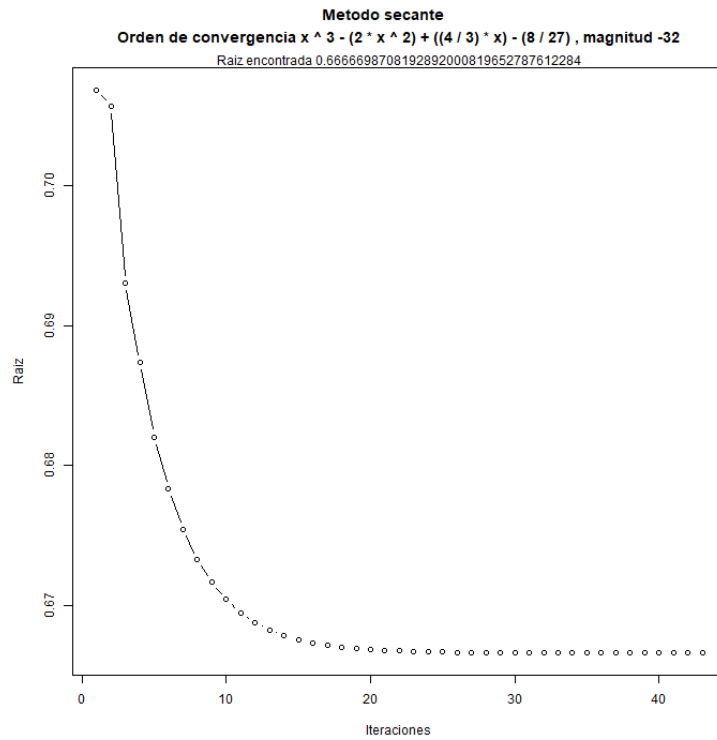


Figura 10: Método de la secante, magnitud -32,  $x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$ .

Como podemos ver en las imágenes el método converge en pocas iteraciones y puede llegar a converger tanto por arriba(por derecha) como por debajo(por izquierda) de la raíz, y tiene una aproximación muy pronunciada en las primeras iteraciones, siendo las ultimas las que refinan la aproximación dependiendo de la magnitud de la tolerancia.

#### 4.4. Función del error

En los archivos que genera el código en R, se pueden ver toda la información recolectada para todas las funciones y todas las magnitudes. Experimentalmente la única función que realizo suficientes iteraciones para poder analizar y hacer una regresión lineal sobre los datos fue  $x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$ , por lo que procedemos a analizar la siguiente ilustración.

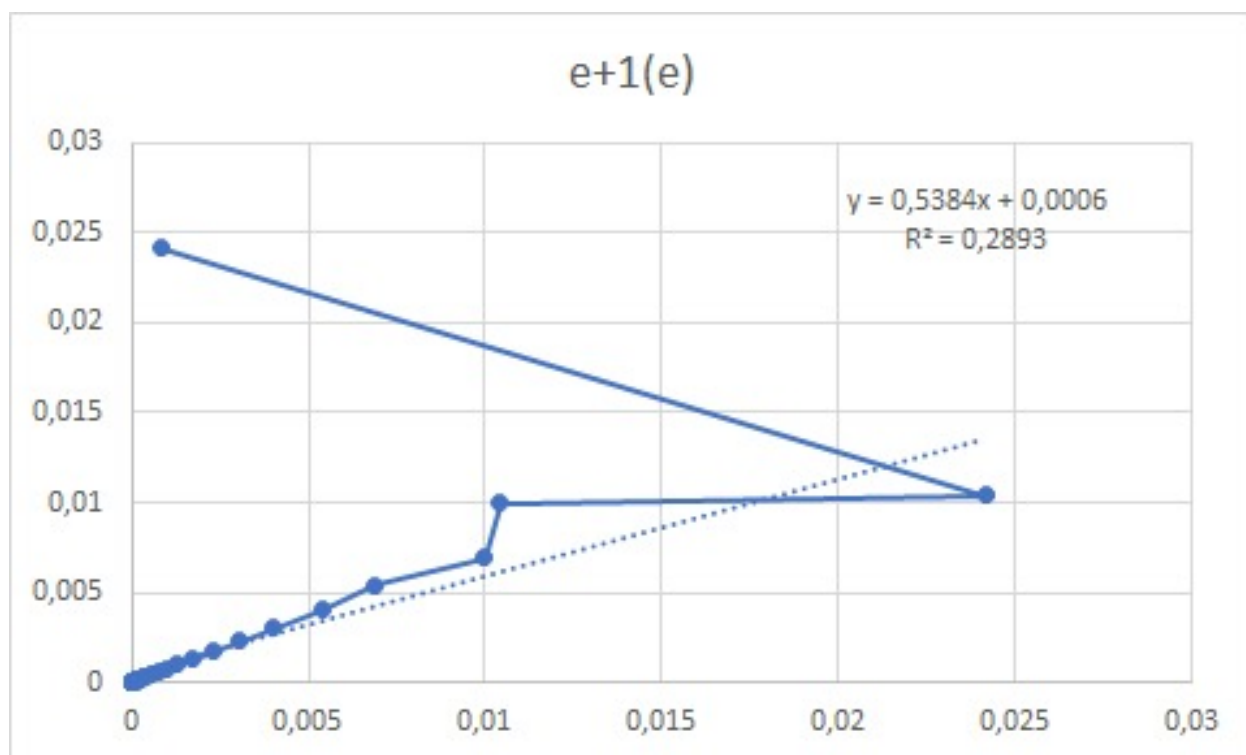


Figura 11:  $e_{i+1} = f(e_i) x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$ .

En la figura podemos observar como el error empieza comportándose de forma arbitraria pero con cada iteración tiende a acercarse a cero de forma cada vez mas lineal.

#### 4.5. Orden de convergencia

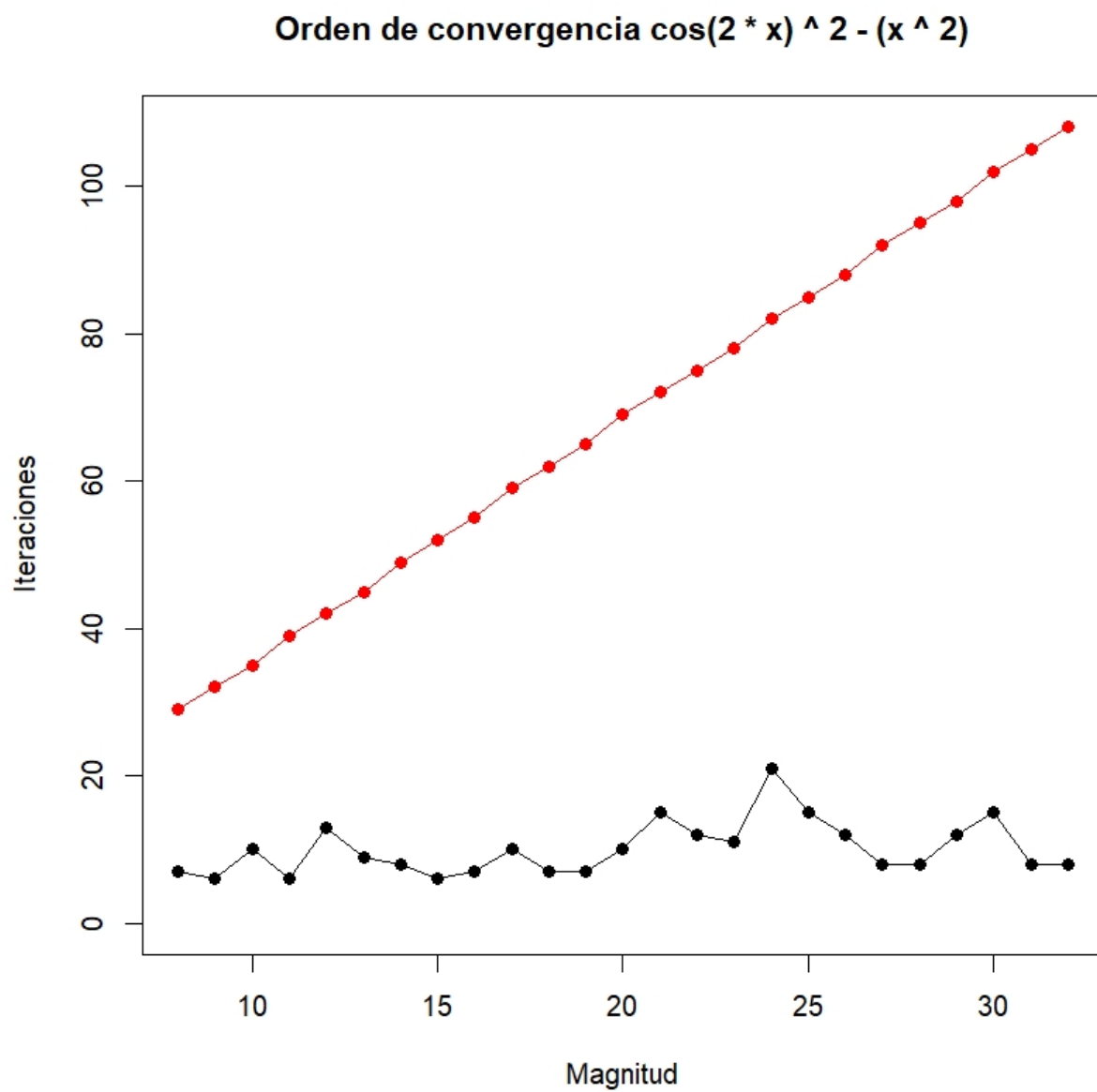


Figura 12: Comparación entre método de la secante(negro), y método de bisección(rojo)  $\cos^2(2x) - x^2$ .

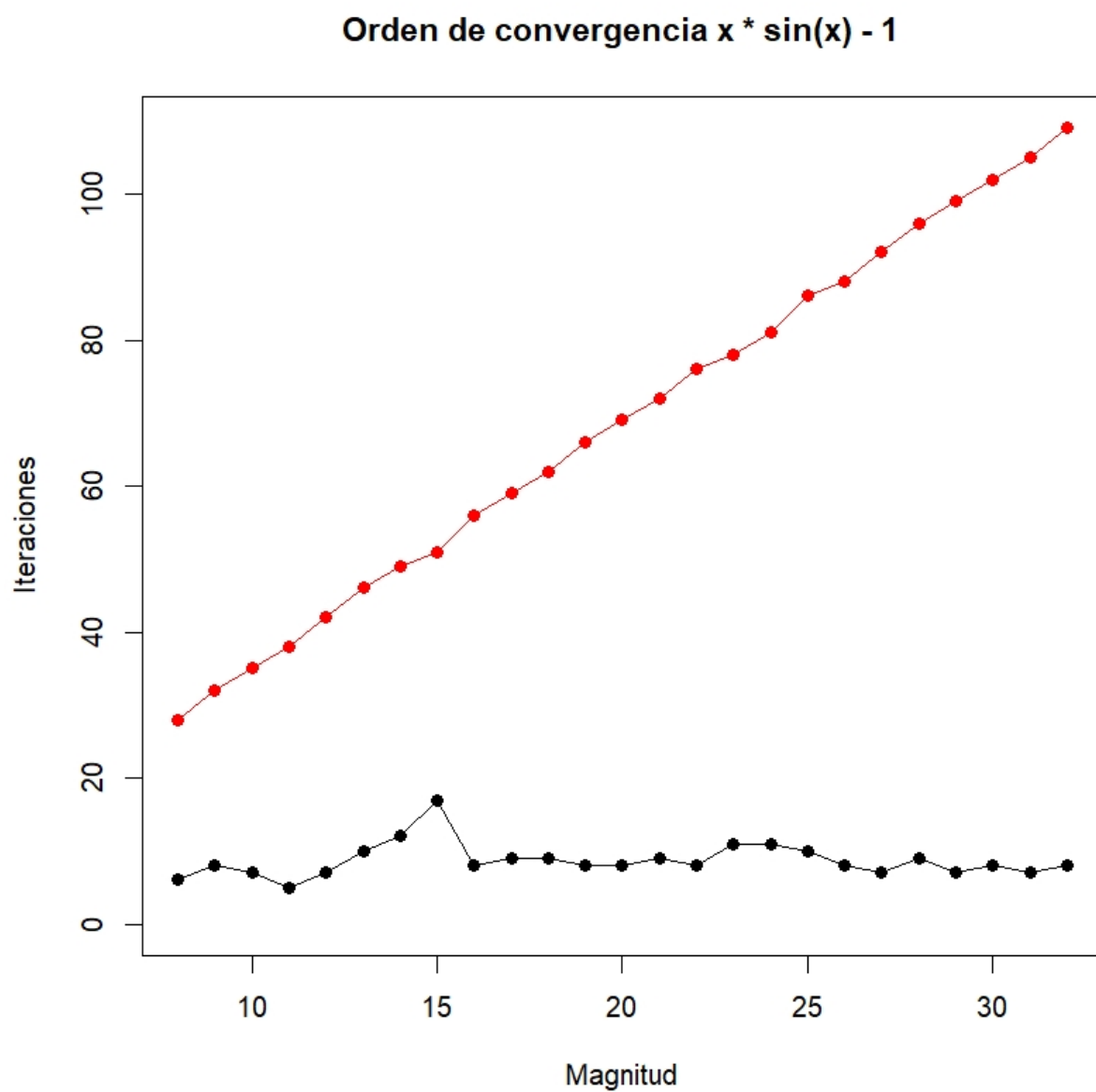


Figura 13: Comparación entre método de la secante(negro), y método de bisección(rojo)  $x \cdot \sin(x) - 1$ .

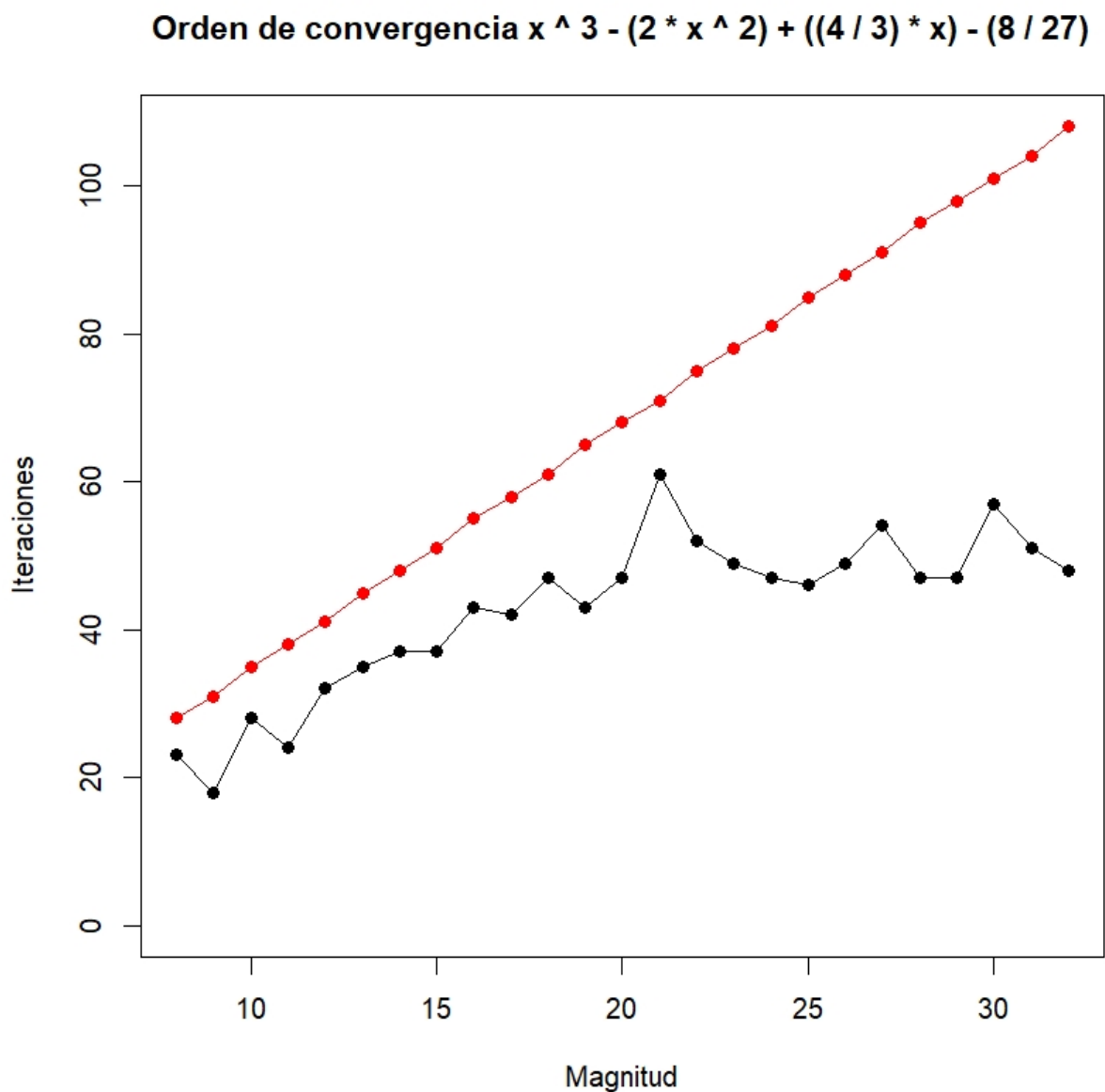


Figura 14: Comparación entre método de la secante(negro), y método de bisección(rojo)  $x^3 - 2x^2 + \frac{4}{3}x - \frac{8}{27}$ .

Como podemos observar en todas las gráficas, tenemos como cota superior el método de la bisección que acota al método de la secante. El método de la bisección tiene una convergencia lineal mientras que la secante se comporta caóticamente dependiendo de la calidad de los valores iniciales.