ALGORITMO ITERATIVO USANDO POLINOMIOS DE TAYLOR CUARTICOS PARA ENCONTRAR RAÍCES DE FUNCIONES

Pedro Fabricio Echeverría Briones

Universidad ECOTEC – pecheverria@dmgs.ecotec.edu.ec

RESUMEN

Las matemáticas aplicadas buscan soluciones para determinar raíces de funciones debido a las aplicaciones en los programas de software que poseen limites con los recursos del hardware en un tiempo viable. El algoritmo deducido en esta investigación es obtenido por unir dos teorías: la aproximación de Newton–Raphson y las resoluciones algebraicas de Ludovico Ferrari para polinomios cuártico, que en conjunto con un análisis de convergencias permite seleccionar la mejor raíz de las cuatro raíces, en el modelo de convergencia, sin que el punto de inicio lo afecte. El método de investigación utilizado es deductivo algebraico y utilizando supuestos cuando el número de iteraciones tiende a ser muy grande. El algoritmo quedaría generalizado siendo fácilmente aplicable en un cualquier lenguaje de programación y también el algoritmo indica que puntos iniciales deben podrían ser aplicados, debido a que las restricciones que se presentan por los diferentes radicales que deben de cumplirse para evitar respuestas en los números complejos y asegurar respuestas en los números reales.

Palabras Claves: iteración, raíz, cero de función, convergencia.

ABSTRACT

Applied mathematics seeks solutions to determine roots of functions due to applications in software programs that have limits with hardware resources in a viable time. The algorithm deduced in this research is obtained by joining two theories: the Newton – Raphson approximation and the algebraic resolutions of Ludovico Ferrari for quartic polynomials, which together with a convergence analysis allows to select the best root of the four roots, in the convergence model, without being affected by the starting point. The research method used is deductive algebraic and using assumptions when the number of iterations tends to be very large. The algorithm would remain generalized, being easily applicable in any programming language and also the

algorithm indicates that initial points should be applied, because the restrictions presented by the different radicals that must be met to avoid responses in complex numbers and ensure answers in real numbers.
Keywords: iteration, root, function zero, convergence.

INTRODUCCIÓN

Encontrar raíces para funciones ha sido explorado por diferentes métodos, entre ellos los iterativos. Partiendo del procedimiento de Newton- Raphson para un polinomio de grado uno, de donde formulamos preguntas que no han sido exploradas: ¿Es posible extender el método del Newton-Raphson hasta un polinomio de Taylor de cuarto grado? ¿Es posible deducir por medio de un algoritmo que indique cuál es la raíz que debe de ser seleccionada? ¿Es posible que este algoritmo sea fácilmente programable? ¿Consigue el algoritmo iterativo la convergencia a pesar de comenzar en un valor muy distantes de la raíz?

Las deducciones matemáticas a usarse para obtener por medio del método del matemático Ludovico Ferrari los valores de las cuatro raíces y por algoritmo de convergencia buscaremos la mejor opción para la siguiente iteración. El resultado es un algoritmo que cambia de signos en términos que tienen patrones repetidos las cuatro raíces. En la fase experimentación se puede verificar con algunas funciones que se consigue llegar a raíces a partir de diferentes puntos alejados de la raíz, que se muestran en figuras del logaritmo del valor absoluto de la función por iteración.

MARCO TEÓRICO

Teorema de Taylor

Para formular una función f(x) que es m veces derivable, en polinomios de Taylor utilizamos la ecuación(1) (Leithold, 1998) (Burden, Faires, & Burden, 2017) (Conte, 1980):

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \qquad (1)$$

El polinomio de Taylor definido como un polinomio cuártico será:

$$f(x_{n+1}) = f(x_n) + \frac{f'(x_n)}{1!}(x_{n+1} - x_n) + \frac{f''(x_n)}{2!}(x_{n+1} - x_n)^2 + \frac{f'''(x_n)}{3!}(x_{n+1} - x_n)^3 + \frac{f'^v(x_n)}{4!}(x_{n+1} - x_n)^4 \quad (2)$$

Despejando $f(x_{n+1})$:

$$f(x_{n+1}) = f(x_n) + \frac{f'(x_n)}{1!}(x_{n+1} - x_n) + \frac{f''(x_n)}{2}(x_{n+1} - x_n)^2 + \frac{f'''(x_n)}{6}(x_{n+1} - x_n)^3 + \frac{f'^v(x_n)}{24}(x_{n+1} - x_n)^4 \quad (3)$$

Resolución algebraica de Ferrari para polinomios cuárticos

Para el polinomio de orden cuartico, se debe de llevar al modelo de polinomio donde a, b, c y d (Ivorra, 2020)

$$y^4 + 2a\,y^3 + by^2 + 2cy + d = 0 \quad (4)$$

$$y^4 + 2a\,y^3 + by^2 + 2cy + d = (y^2 + ay + A)^2 - (By + C)^2 = 0 \quad (5)$$

$$y^4 + 2a\,y^3 + by^2 + 2cy + d = y^4 + 2ay^3 + (a^2 + 2A - B^2)y^2 + 2(aA - BC)y + (A^2 - C^2) \quad (6)$$

$$(y^2 + (a + B)y + (A + C))\,(y^2 + (a - B)y + (A - C)) = 0 \quad (7)$$

De lo que se pueden obtener 4 raíces en dos ecuaciones cuadráticas:

Raíces 1 y 2:

$$(y^2 + (a + B)y + (A + C)) = 0 \qquad y = \frac{-(a+B) \pm \sqrt{(a+B)^2 - 4(A+C)}}{2} \quad (8)$$

Raíces 3 y 4:

$(y^2 + (a - B)y + (A - C))=0 \quad y = \frac{-(a-B) \pm \sqrt{(a-B)^2 - 4(A-C)}}{2}$ (9)

Los valores de A, B y C, se los obtiene de las ecuaciones (4) y (7):

$$b = a^2 + 2A - B^2 \tag{10}$$

$$c = aA - BC \tag{11}$$

$$d = A^2 - C^2 \tag{12}$$

Al unir las ecuaciones (11) y (12)

$$B^2C^2 = (b - a^2 - 2A)(A^2 - C^2) = (c - aA)^2 \tag{13}$$

Llegamos la solución de un polinomio de orden cúbico a partir de despejar la ecuación (13).

$$A^3 - \frac{b}{2}A^2 + (ac - d)A + \frac{(bd - c^2 - a^2d)}{2} = 0 \tag{14}$$

Donde aplicamos la resolución de Tartaglia-Cartago para obtener el valor A del polinomio cúbico. Los valores de B y C deben de cumplir las ecuaciones

$$B = \pm\sqrt{a^2 + 2A - b} \tag{15}$$

$$C = \pm\sqrt{A^2 - d} \tag{16}$$

Lo que nos indica que se pueden conocer |B| y |C|, deberían de hacer permutaciones de signos para conocer los signos de B y de C.

$$BC = c - aA \tag{17}$$

Resolución algebraica de Tartaglia-Cartago para polinomios cúbicos

El polinomio de orden cúbico debe de cumplir el modelo de polinomio donde a, b y c (Ivorra, 2020):

$$z^3 + az^2 + bz + c = 0 \tag{18}$$

Donde una de las raíces reales cumplirá con:

$$z = \sqrt[3]{-\frac{q}{2} + \sqrt{\Delta}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\Delta}} - \frac{a}{3} \tag{19}$$

Donde los valores p, q y Δ:

$$p = \frac{3b - a^2}{3} \tag{20}$$

$$q = \frac{2a^3 - 9ab + 27c}{27} \tag{21}$$

$$\Delta = \left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3 \tag{22}$$

METODOLOGÍA DE DEDUCCIÓN MATEMÁTICA

Convergencia del algoritmo iterativo usando el polinomio de orden cuártico para funciones de una variable.

Al unir el polinomio de Taylor de la ecuación (3) con la resolución de Newton-Raphson $f(x_{n+1}) = 0$ (Burden, Faires, & Burden, 2017).

$$0 = f(x_n) + \frac{f'(x_n)}{1!}(x_{n+1} - x_n) + \frac{f''(x_n)}{2}(x_{n+1} - x_n)^2 + \frac{f'''(x_n)}{6}(x_{n+1} - x_n)^3 + \frac{f'^v(x_n)}{24}(x_{n+1} - x_n)^4 \tag{23}$$

La ecuación (10) la llevamos al modelo la ecuación (5), dividiendo para $\frac{f'^v(x_n)}{4!}$ a todos los términos del polinomio.

$$0 = \frac{24f(x_n)}{f'^v(x_n)} + \frac{24\,f'(x_n)}{f'^v(x_n)}(x_{n+1} - x_n) + \frac{12f''(x_n)}{f'^v(x_n)}(x_{n+1} - x_n)^2 + \frac{4f'''(x_n)}{f'^v(x_n)}(x_{n+1} - x_n)^3 + (x_{n+1} - x_n)^4 \tag{24}$$

$$0 = \frac{24f(x_n)}{f'^v(x_n)} + 2\frac{12f'(x_n)}{f'^v(x_n)}(x_{n+1} - x_n) + \frac{12f''(x_n)}{f'^v(x_n)}(x_{n+1} - x_n)^2 + 2\frac{2f'''(x_n)}{f'^v(x_n)}(x_{n+1} - x_n)^3 + (x_{n+1} - x_n)^4 \tag{25}$$

Encontrando los valores de a, b, c y d, de la ecuación (25):

$$a = \frac{2f'''(x_n)}{f'^v(x_n)} \tag{26}$$

$$b = \frac{12f''(x_n)}{f'^v(x_n)} \tag{27}$$

$$c = \frac{12f'(x_n)}{f'^v(x_n)} \tag{28}$$

$$d = \frac{24f(x_n)}{f'^v(x_n)} \tag{29}$$

Al aplicar a,b,c y d en las ecuaciones de las raíces (8) y (9), obtenemos las expresiones que obtienen las cuatro raices:

$$(x_{n+1} - x_n) = \frac{-(a+B) \pm \sqrt{(a+B)^2 - 4(A+C)}}{2} \tag{30}$$

$$(x_{n+1} - x_n) = \frac{-(a-B) \pm \sqrt{(a-B)^2 - 4(A-C)}}{2} \tag{31}$$

De las ecuaciones (15) y (16) se pueden obtener el valor absoluto de B y C, pero no su signo. Por lo que quedarian resumidas las ecuaciones (30) y (31) en una expresión que cubren las cuatro raíces.

$$(x_{n+1} - x_n) = \frac{-(a \pm B) \pm \sqrt{(a \pm B)^2 - 4(A \pm C)}}{2} \quad (32)$$

Despejando x_n:

$$x_{n+1} = x_n + \frac{-(a \pm B) \pm \sqrt{(a \pm B)^2 - 4(A \pm C)}}{2} \quad (33)$$

Supuestos de Convergencia

Primer Supuesto: Resolveremos la selección del signo de C para que el algoritmo pueda elegir los signos. Al remplazar $C = \pm\sqrt{A^2 - d}$ en $A \pm C$, resultando la expresión $A \pm \pm\sqrt{A^2 - d}$, en donde sabemos que d tiende a cero y reduciendo la expresión a $A \pm \pm|A|$, esta expresión tiene dos posibles valores:

- Si $A + signo(A)|A| = 2A$ estaríamos en una raíz que podría generar un número imaginario.
- Si $A - signo(A)|A|$=0 lo cual indicaría siguiendo este signo puede converger.

De la ecuación (33), remplazando $-signo(A)$, obtenemos:

$$x_{n+1} = x_n + \frac{-(a \pm B) \pm \sqrt{(a \pm B)^2 - 4(A - signo(A)|C|)}}{2} \quad (34)$$

Segundo Supuesto: Al aplicar los supuestos: $A - signo(A)|C| = 0$ y $x_{n+1} - x_n = 0$, obtenemos:

$$0 = \frac{-(a \pm B) \pm \sqrt{(a \pm B)^2 - 0}}{2} \quad (35)$$

$$0 = -(a \pm B) \pm \sqrt{(a \pm B)^2} \quad (36)$$

$$0 = -(a \pm B) \pm |(a \pm B)| \quad (37)$$

Debe de cumplirse con la igualdad con lo que podemos obtener:

$$0 = -(a \pm B) + signo(a \pm B)\sqrt{(a \pm B)^2} \tag{38}$$

Al aplicar el supuesto de la ecuación (38) en la ecuación (34), remplazando $signo(a \pm B)$ obtenemos:

$$x_{n+1} = x_n + \frac{-(a \pm B) + signo(a \pm B)\sqrt{(a \pm B)^2 - 4(A - signo(A)|C|)}}{2} \tag{35}$$

Tercer Supuesto: Queda pendiente determinar el símbolo B, donde debería $a \pm B = 0$, remplazando B, $a \pm B = a \pm \pm\sqrt{a^2 + 2A - b} = 0$. Donde $2A - b$ tenderá a cero, para que se produzca una forma de convertirse en cero, obteniendo $a \pm \pm\sqrt{a^2} = a \pm \pm|a| = 0$, donde se debe de cumplir $a - signo(a)|a| = 0$. Cumpliendo el último supuesto se obtendría el algoritmo generalizado:

$$x_{n+1} = x_n + \frac{-(a - signo(a)|B|) + signo(a - signo(a)|B|)\sqrt{(a - signo(a)|B|)^2 - 4(A - signo(A)|C|)}}{2} \tag{36}$$

ANÁLISIS DE RESULTADOS

VALIDACIÓN DE LA CONVERGENCIA DEL ALGORITMO ITERATIVO

Se plantearon pruebas del algoritmo con diferentes funciones diferenciables mínimo en grado 4, con la cuarta derivada diferente de cero y tomando diferentes valores de inicio.

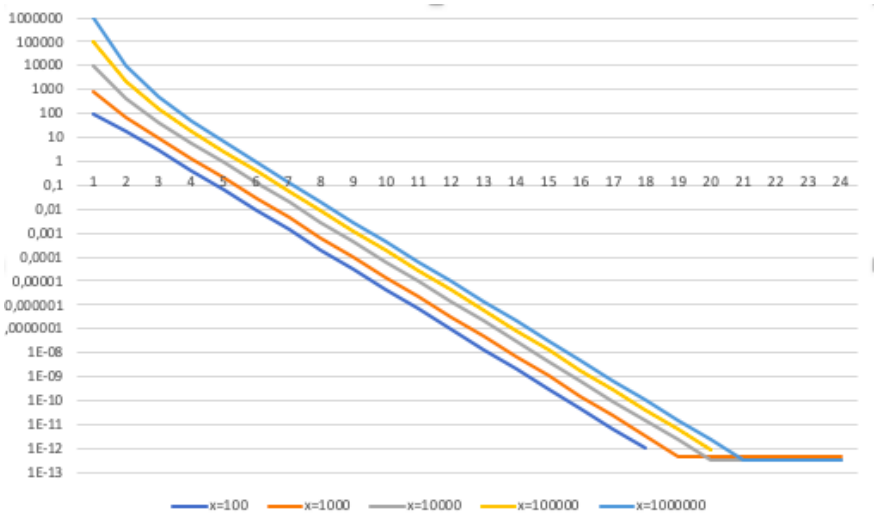

Figura 1: Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x) = \ln(\mathrm{x}^2) + \mathrm{x} - 200$ con diferentes puntos de inicios alejados de la raíz.

Fuente: Elaboración Propia

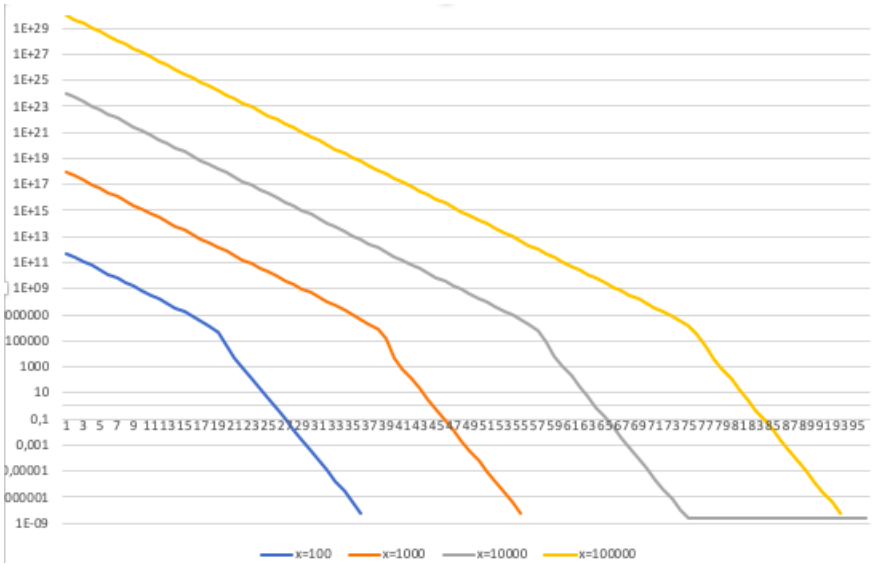

Figura 2: Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x) = (\mathrm{x} - 10)^6 - 10^6$ con diferentes puntos de inicios alejados de la raíz.

Fuente: Elaboración Propia

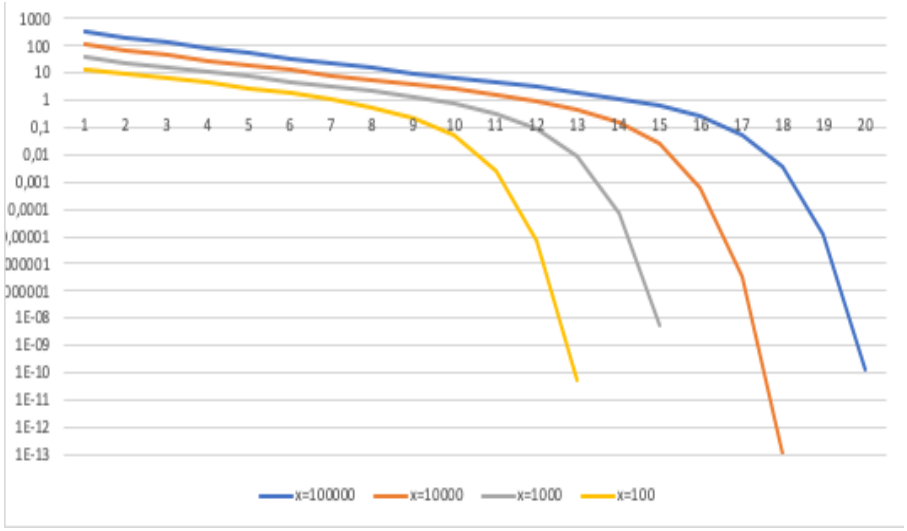

Figura 4: Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x) = \ln(\mathrm{x}) + \sqrt{x}$ con diferentes puntos de inicios alejados de la raíz.

Fuente: Elaboración Propia

CONCLUSIONES

1. El supuesto del método de Newton-Raphson se puede extender a polinomios cuarticos.
2. A pesar de utilizar puntos distantes a la raíz el algoritmo converge.
3. La programación del algoritmo iterativo es simple, debido a que utiliza elementos del algebra tradicional.
4. Los algoritmos iterativos tienen un potencial de investigación al utilizar diferentes áreas de las matemáticas que están poco exploradas, como en este caso unir ideas de dos grupos de matemáticos, que no fueron de la misma época en una aplicación computacional.

REFERENCIAS

Burden, R., Faires, D., & Burden, A. (2017). ANALISIS NUMERICO. MEXICO: Cengage Learning.

Conte, S. D. (1980). Elementary numerical analysis: algorithmic approach. New York: McGraw-Hill.

Ivorra, C. (01 de 10 de 2020). Las fórmulas de Cardano-Ferrari. Obtenido de Universidad de Valencia: https://www.uv.es/ivorra/Libros/Ecuaciones.pdf

Leithold, L. (1998). El Cálculo. México: Oxford University Press.