

Aplicación del método Newton-Raphson por polinomio cúbicos usando el método Tartaglia-Cardano

Application of the Newton-Raphson method by cubic polynomial using the Tartaglia-Cardano method

Pedro Fabricio Echeverría Briones¹ <https://orcid.org/0000-0002-4921-1111>

¹Universidad Tecnológica Empresarial de Guayaquil, Guayaquil, Ecuador
coordinadoracademicoposgrado03@uteg.edu.ec,
pecheverr@gmail.com

Enviado: 2021/01/12

Aceptado: 2021/10/18

Publicado: 2021/12/30

Resumen

Algunos de los métodos usados para encontrar raíces de funciones en las matemáticas aplicadas son los algoritmos interactivos, que constituyen un campo de interés muy importante dentro de la disciplina debido a que permiten encontrar soluciones para usar la menor capacidad de recursos dentro de los sistemas computacionales. La propuesta de estos algoritmos iterativos implica la relación de diferentes áreas de las matemáticas: álgebra, cálculo diferencial, cálculo vectorial y análisis numérico. Esta relación conjuga la teoría de cálculo diferencial y el análisis numérico realizado por matemáticos ingleses, como Newton y Raphson. Estos propusieron la primera forma de buscar una raíz de una función usando iteraciones, la que a su vez se sustentó en teorías algebraicas desarrolladas por algunos matemáticos italianos, como Tartaglia y Cardano, quienes anteriormente habían encontrado las soluciones a las raíces de un polinomio de orden cúbico. Dentro del proceso deductivo que se siguió en el proceso, se utilizaron las aproximaciones recomendadas que se generaron al aplicar las Series de Taylor. Por otro lado, durante el proceso de experimentación se han identificado determinados patrones repetitivos, especialmente, cuando el punto de inicio se aleja de la raíz. Esto permitió mejorar el algoritmo iterativo al identificar el lugar de la curva de convergencia. Finalmente, se generalizó la ecuación iterativa para muchas variables, a partir de las deducciones utilizadas de 1 y 2 variables.

Palabras clave: iteración, raíz, cero de función, convergencia, cúbico.

Sumario: Introducción, Desarrollo, Análisis de resultados y Conclusiones.

Como citar: Echeverría, P. (2021). Aplicación del método Newton-Raphson por polinomio cúbicos usando el método Tartaglia-Cardano. *Revista Tecnológica - Espol*, 33(3), 258-270.
<http://www.rte.espol.edu.ec/index.php/tecnologica/article/view/799>

Abstract

Some of the methods employed to find roots of functions in applied mathematics are interactive algorithms, which is a very important field of interest within the discipline because they allow finding solutions using the least number of resources within computational systems. The proposal of these iterative algorithms involves the relationship of different areas of mathematics being algebra, differential calculus, vector calculus, and numerical analysis. This relationship combines the theories of differential calculus and numerical analysis carried out by English mathematicians, such as Newton and Raphson. They proposed the first way to find a root of a function using iterations with the application of algebraic theories of Italian mathematicians, Tartaglia and Cardano, who had previously found the solutions to the roots of a cubic order polynomial. Along the deductive process, the recommended approximations were followed by using the Taylor series. On the other hand, the experimentation process determined repetitive patterns, especially when the starting point is far from the root. This allowed the improvement of the iterative algorithm by identifying the location of the convergence curve. Finally, the iterative equation was generalized for many variables using the deductions of 1 and 2 variables.

Keywords: iteration, root, function zero, convergence, cubic.

Introducción

El método de Newton-Raphson utiliza demostraciones algebraicas, cálculo diferencial, cálculo vectorial y análisis numérico para la aproximación de raíces. Cabe señalar que este método contiene errores de orden cuadrático. Por otro lado, para determinar las raíces de un polinomio de orden cúbico, se utiliza el método Tartaglia-Cardano. Esta es la razón por la que el algoritmo iterativo que se deduce tiene ventajas, pues, al encontrar la raíz de una función usando la raíz del polinomio cúbico, que se aproxima por series de Taylor, se obtiene un error de orden cuartico. Esto hace que la convergencia sea más segura, a pesar de que se usan puntos de inicio muy alejados con respecto al valor de la raíz. La forma de programación utilizada es sencilla porque el proceso de experimentación simplemente utiliza una hoja de cálculo. Algunas restricciones del modelo son que la función tenga una tercera derivada continua diferente de cero y que existan ciertos intervalos en donde se pueda tener un valor inicial.

Desarrollo

Definición de una función de una variable $f(x)$ en serie de Taylor

Para formular una función $f(x)$, que es m veces derivable, en polinomios por serie de Taylor, se utilizó la ecuación (1) (Leithold, 1998; Burden, Faires, y Burden, 2017):

$$f(x) = \sum_{a=0}^m \frac{f^{(a)}(x_0)}{a!} (x - x_0)^a \quad (1)$$

Al llevar la serie de Taylor hasta un polinomio cúbico se tiene:

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

Obteniendo:

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

Definición de una función de dos variables $f(x_1, x_2)$ en serie de Taylor

Para formular una función $f(x_1, x_2)$, la cual puede ser m veces derivable, se utiliza su descomposición en polinomios por serie de Taylor (Colley, 2012; Conte, 1980):

$$\begin{aligned} f(x_{1\ n+1}, x_{2\ n+1}) &= f(x_{1\ n}, x_{2\ n}) + \sum_{j=1}^2 \frac{\partial f(x_{1\ j}, x_{2\ j})}{\partial x_j} (x_{j\ n+1} - x_{j\ n}) \\ &+ \frac{1}{2} \sum_{j=1}^2 \sum_{k=1}^2 \frac{\partial^2 f(x_{1\ j}, x_{2\ j})}{\partial x_j \partial x_k} (x_{j\ n+1} - x_{j\ n})(x_{k\ n+1} - x_{k\ n}) \\ &+ \frac{1}{6} \sum_{j=1}^2 \sum_{k=1}^2 \sum_{l=1}^2 \frac{\partial^3 f(x_{1\ j}, x_{2\ j})}{\partial x_j \partial x_k \partial x_l} (x_{j\ n+1} - x_{j\ n})(x_{k\ n+1} - x_{k\ n})(x_{l\ n+1} - x_{l\ n}) \end{aligned} \quad (4)$$

Resolución algebraica de Tartaglia-Cardano para polinomios cúbicos

Para el polinomio de orden cúbico, se debe llevar al modelo de polinomio hacia donde este satisface a, b y c (Ivorra, 2020):

$$y^3 + ay^2 + by + c = 0 \quad (5)$$

Donde una de las raíces reales cumplirá con:

$$y = \sqrt[3]{-\frac{q}{2} + \sqrt{\Delta}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\Delta}} - \frac{a}{3} \quad (6)$$

Siendo:

$$p = \frac{3b - a^2}{3} \quad (7)$$

$$q = \frac{2a^3 - 9ab + 27c}{27} \quad (8)$$

$$\Delta = \left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3 \quad (9)$$

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

Al combinar la serie de Taylor de la ecuación (3) con la resolución de Newton-Raphson $f(x_{n+1}) = 0$ (Burden, Faires, & Burden, 2017), se obtiene:

$$0 = f(x_n) + \frac{f'(x_n)}{1!}(x_{n+1} - x_n) + \frac{f''(x_n)}{2!}(x_{n+1} - x_n)^2 + \frac{f'''(x_n)}{3!}(x_{n+1} - x_n)^3. \quad (10)$$

De la ecuación (10) se tiene que pasar al formato de la ecuación (5), dividiendo para $\frac{f'''(x_n)}{3!}$ todos los términos del polinomio.

$$0 = \frac{6f(x_n)}{f'''(x_n)} + \frac{6f'(x_n)}{f'''(x_n)}(x_{n+1} - x_n) + \frac{3f''(x_n)}{f'''(x_n)}(x_{n+1} - x_n)^2 + (x_{n+1} - x_n)^3 \quad (11)$$

Encontrando los valores de a, b y c:

$$a = \frac{3f''(x_n)}{f'''(x_n)} \quad (12)$$

$$b = \frac{6f'(x_n)}{f'''(x_n)} \quad (13)$$

$$c = \frac{6f(x_n)}{f'''(x_n)} \quad (14)$$

Determinando los valores de las ecuaciones (7), (8) y (9):

$$p = \frac{3 \frac{6f'(x_n)}{f'''(x_n)} - \left(\frac{3f''(x_n)}{f'''(x_n)} \right)^2}{3} \quad (15)$$

$$q = \frac{2 \left(\frac{3f''(x_n)}{f'''(x_n)} \right)^3 - 9 \frac{3f''(x_n)}{f'''(x_n)} \frac{6f'(x_n)}{f'''(x_n)} + 27 \frac{6f(x_n)}{f'''(x_n)}}{27} \quad (16)$$

$$\Delta = \left(\frac{3 \frac{6f'(x_n)}{f'''(x_n)} - \left(\frac{3f''(x_n)}{f'''(x_n)} \right)^2}{6} \right)^2 + \left(\frac{2 \left(\frac{3f''(x_n)}{f'''(x_n)} \right)^3 - 9 \frac{3f''(x_n)}{f'''(x_n)} \frac{6f'(x_n)}{f'''(x_n)} + 27 \frac{6f(x_n)}{f'''(x_n)}}{81} \right)^3 \quad (17)$$

Al aplicar los valores de las ecuaciones (15), (16) y (17) en la ecuación (5), se determina:

$$(x_{n+1} - x_n) = \sqrt[3]{-\frac{q}{2} + \sqrt{\Delta}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\Delta}} - \frac{a}{3} \quad (18)$$

Al despejar x_n se determina el algoritmo iterativo:

$$x_{n+1} = x_n + \sqrt[3]{-\frac{q}{2} + \sqrt{\Delta}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\Delta}} - \frac{a}{3} \quad (19)$$

Convergencia del algoritmo iterativo usando el polinomio de orden cúbico para funciones de dos variables

Aquí se usa el modelo de Newton-Raphson con el método de Tartaglia-Cardano para una función de dos variables. Para varias variables, se tiene que utilizar la serie de Taylor, en

donde se realiza una reducción de términos, asumiendo que $x_{n+1k} - x_{nk} = 0$ cuando $n \rightarrow \infty$, excepto cuando los índices de las sumatorias sean iguales. Reduciendo la ecuación (4):

$$0 = f(x_{1n}, x_{2n}) + \frac{\partial f(x_{1n}, x_{2n})}{\partial x_j} (x_{jn+1} - x_{jn}) + \frac{1}{2} \frac{\partial^2 f(x_{1n}, x_{2n})}{\partial x_j^2} (x_{jn+1} - x_{jn})^2 + \frac{1}{6} \frac{\partial^3 f(x_{1n}, x_{2n})}{\partial x_j^3} (x_{jn+1} - x_{jn})^3 \quad (20)$$

Donde: $j=\{1,2\}$, se obtienen los siguientes valores de la ecuación (5) :

$$a_{jn} = 3 \frac{\frac{\partial^2 f(x_{1n}, x_{2n})}{\partial x_j^2}}{\frac{\partial^3 f(x_{1n}, x_{2n})}{\partial x_j^3}} \quad (21)$$

$$b_{jn} = 6 \frac{\frac{\partial f(x_{1n}, x_{2n})}{\partial x_j}}{\frac{\partial^3 f(x_{1n}, x_{2n})}{\partial x_j^3}} \quad (22)$$

Donde: $j=\{1,2\}$, se obtienen los siguientes valores de las ecuaciones (6), (7) y (8) :

$$p_{jn} = \frac{3b_{jn} - a_{jn}^2}{3} \quad (23)$$

$$q_{jn} = \frac{2a_{jn}^3 - 9a_{jn}b_{jn} + 27c_{jn}}{27} \quad (24)$$

$$\Delta_{jn} = \left(\frac{q_{jn}}{2}\right)^2 + \left(\frac{p_{jn}}{3}\right)^3 \quad (25)$$

Al aplicar los valores de las ecuaciones (23), (24) y (25) en la ecuación (5), se determina:

$$(x_{jn+1} - x_{jn}) = \sqrt[3]{-\frac{q_{jn}}{2} + \sqrt{\Delta_{jn}}} + \sqrt[3]{-\frac{q_{jn}}{2} - \sqrt{\Delta_{jn}}} - \frac{a_{jn}}{3} \quad (26)$$

Al despejar x_n , se determina el algoritmo iterativo para una función de dos variables:

$$x_{jn+1} = x_{jn} + \sqrt[3]{-\frac{q_{jn}}{2} + \sqrt{\Delta_{jn}}} + \sqrt[3]{-\frac{q_{jn}}{2} - \sqrt{\Delta_{jn}}} - \frac{a_{jn}}{3} \quad (27)$$

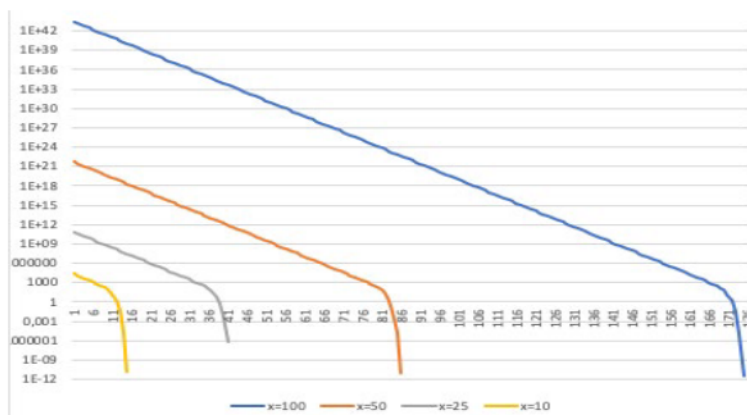
Análisis de resultados

Validación de la convergencia del algoritmo iterativo para funciones de una variable

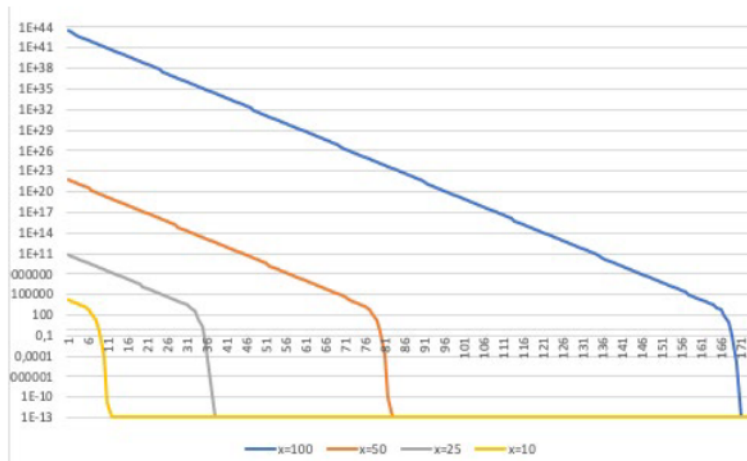
A continuación, en los siguientes gráficos, se presentan los resultados que se obtuvieron de las pruebas que se realizaron con diferentes funciones, cuyas condiciones fueron que estas tengan la tercera derivada diferente de cero y en diferentes puntos de inicio:

Figura 1

Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x) = e^x - 100$ con diferentes puntos de inicios alejados de la raíz

**Figura 2**

Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x) = e^x - 1000 - x^2 - x$ con diferentes puntos de inicios alejados de la raíz

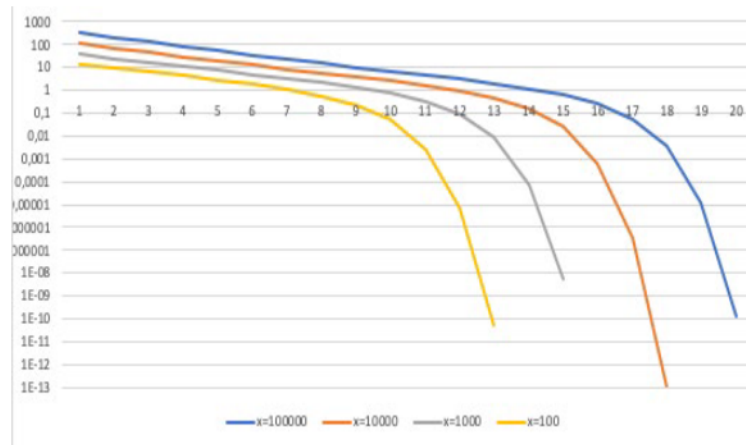
**Figura 3**

Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x) = -100 * \cos \cos(x) + e^{2x} + e^{-2x}$ con diferentes puntos de inicios alejados de la raíz

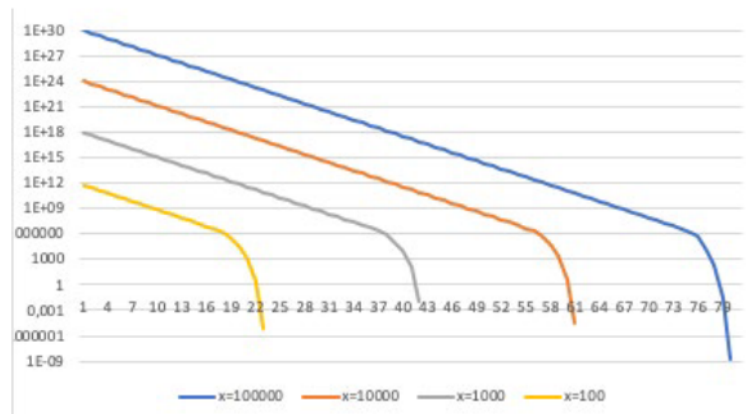


Figura 4

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

**Figura 5**

Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x) = (x - 10)^6 - 10^6$ con diferentes puntos de inicios alejados de la raíz



A partir de las representaciones gráficas de esta experimentación, se puede deducir que el algoritmo iterativo tiene dos formas de convergencia. Este patrón se puede observar visualmente cuando se aleja el valor de inicio de la raíz. En resumen, se puede concluir:

1. Al observar las curvas de convergencia, se verifica la formación de una línea recta decreciente entre $\log \log (|f(x_n)|)$ y n . Este proceso es lento. Lo que llama la atención es que para diferentes valores del punto inicial se obtienen rectas que son casi paralelas.
2. Cuando la curva se acerca a la raíz, su convergencia se acelera cambiando la pendiente de la curva rápidamente por cada iteración.

Esta es la razón por la que se puede determinar una aceleración al diferenciar cuando esta se encuentra entre los dos patrones. A continuación, se presentan el análisis de los siguientes casos:

a) Si los $\Delta x_{n+1} \neq \Delta x_n$, el patrón se encontraría en los siguientes gráficos:

Figura 6

Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x)$

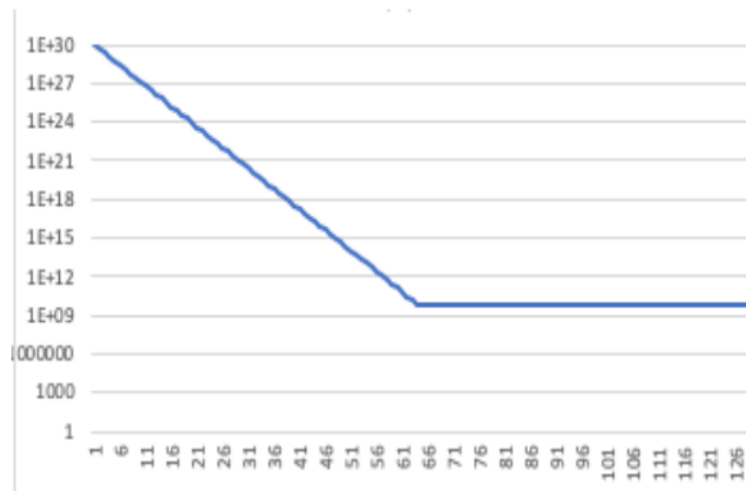


Figura 7

Curvas de Convergencia de Δx por número de iteración



Para acelerar la convergencia, se predice el valor de la distancia del valor de inicio con respecto a la raíz. Si se utiliza la expresión de la recta por número de iteración n :

$$y = m * n + b \quad (28)$$

$$\Delta x_1 = m * 1 + b \quad (29)$$

$$\Delta x_2 = m * 2 + b \quad (30)$$

Obtenemos la pendiente de la recta m :

$$m = \frac{\Delta x_2 - \Delta x_1}{2 - 1} \quad (31)$$

$$m = \Delta x_2 - \Delta x_1 \quad (32)$$

Obtenemos la intersección de la recta b :

$$\Delta x_1 = (\Delta x_2 - \Delta x_1) * 1 + b \quad (33)$$

$$b = 2 * \Delta x_1 - \Delta x_2 \quad (34)$$

Si se establece que $y=0$, se obtiene el número de iteraciones n :

$$0 = m * n + b \quad (35)$$

$$n = -\frac{b}{m} \quad (36)$$

$$n = -\frac{2 * \Delta x_1 - \Delta x_2}{\Delta x_2 - \Delta x_1} \quad (37)$$

Si se convierte n en entero:

$$n = \text{entero} \left(-\frac{2 * \Delta x_1 - \Delta x_2}{\Delta x_2 - \Delta x_1} \right) \quad (38)$$

De manera que se determina la distancia entre el valor inicial y la raíz al obtener $x_i + n * \Delta x_i$

a) Si los $\Delta x_{n+1} = \Delta x_n$, se encontraría el patrón en los siguientes gráficos.

Figura 8

Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x)$

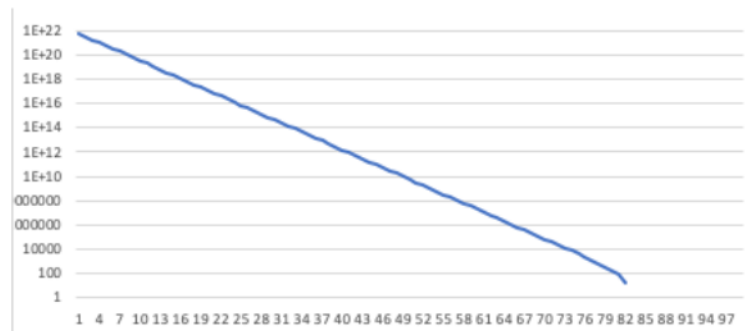
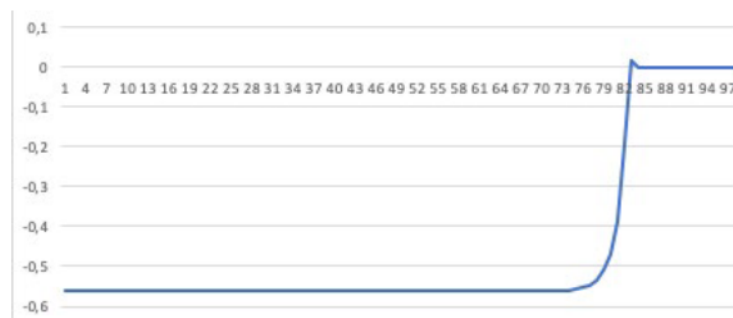


Figura 9

Curvas de Convergencia de Δx por número de iteración



Si este patrón se repite, el primer proceso de convergencia puede ser acelerado al considerar el patrón de las iteraciones, lo que, a su vez, puede determinar un procedimiento

para calcular la distancia entre el valor inicial y la raíz. Si para esto se utiliza la ecuación de la recta por iteración n:

$$y = m * n + b \quad (39)$$

$$\log (|f(x_0)|) = m * 1 + b \quad (40)$$

$$\log (|f(x_1)|) = m * 2 + b \quad (41)$$

Obtenemos la pendiente de la recta m:

$$m = \frac{\log (|f(x_1)|) - \log (|f(x_0)|)}{2 - 1} \quad (42)$$

$$m = \log (|f(x_1)|) - \log (|f(x_0)|) \quad (43)$$

Obtenemos la intersección de la recta b:

$$\log (|f(x_0)|) = (\log (|f(x_1)|) - \log (|f(x_0)|)) * 1 + b \quad (44)$$

$$\log \log (|f(x_0)|) - \log \log (|f(x_1)|) + \log (|f(x_0)|) = b \quad (45)$$

$$b = 2 * \log \log (|f(x_0)|) - \log \log (|f(x_1)|) \quad (46)$$

Por lo que el modelo sería:

$$y = \frac{(\log \log (|f(x_1)|) - \log \log (|f(x_0)|))}{\log \log (|f(x_1)|)} * x + 2 * \log \log (|f(x_0)|) - \log \log (|f(x_1)|) \quad (47)$$

Si se establece que $y=0$, se tendría a n:

$$0 = \frac{(\log \log (|f(x_1)|) - \log \log (|f(x_0)|))}{\log \log (|f(x_1)|)} * n + 2 * \log \log (|f(x_0)|) - \log \log (|f(x_1)|) \quad (48)$$

$$\begin{aligned} & -(\log \log (|f(x_1)|) - \log \log (|f(x_0)|)) * n \\ & = 2 * \log \log (|f(x_0)|) - \log \log (|f(x_1)|) \end{aligned} \quad (49)$$

$$n = - \frac{2 * \log \log (|f(x_0)|) - \log \log (|f(x_1)|)}{(\log \log (|f(x_1)|) - \log \log (|f(x_0)|))} \quad (50)$$

El valor de n debe ser entero y estar antes de la zona de convergencia, por lo que se le restará 3 iteraciones. Obteniendo:

$$n = \text{entero} \left(- \frac{2 * \log (|f(x_0)|) - \log (|f(x_1)|)}{(\log (|f(x_1)|) - \log (|f(x_0)|))} \right) - 3 \quad (51)$$

De manera que se pueda determinar la distancia entre el valor inicial y la raíz al obtener $x_i + n * \Delta x_i$. Esto mejora el algoritmo iterativo de la ecuación (27) y permite obtener la expresión:

$$x_{n+1} = x_n + n \left(\sqrt[3]{-\frac{q}{2} + \sqrt{\Delta}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\Delta}} - \frac{a}{3} \right) \quad (52)$$

- Si $\Delta x_{n+1} \neq \Delta x_n$, entonces $n = \text{entero} \left(-\frac{2 * \Delta x_1 - \Delta x_2}{\Delta x_2 - \Delta x_1} \right)$
- Si $\Delta x_{n+1} = \Delta x_n$, entonces $n = \text{entero} \left(-\frac{2 * \log(|f(x_0)|) - \log(|f(x_1)|)}{(\log(|f(x_1)|) - \log(|f(x_0)|))} \right) - 3$

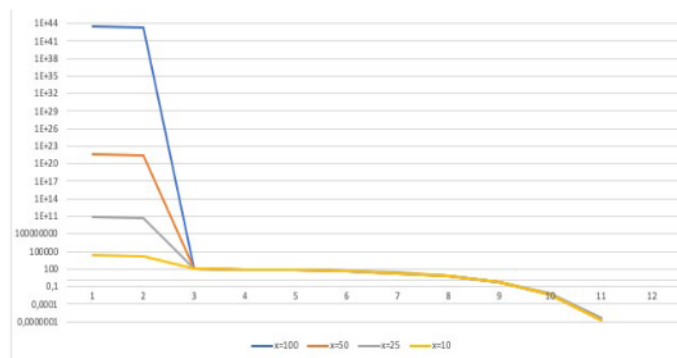
Validación de la convergencia del algoritmo iterativo utilizando un predictor de distancia entre el punto inicial y la raíz para funciones de una variable

Considerando los casos anteriores:

- a) Si $\Delta x_{n+1} = \Delta x_n$ en la función $f(x) = e^x - 100$, al aplicar el coeficiente de aceleración, se obtienen menos iteraciones.

Figura 10

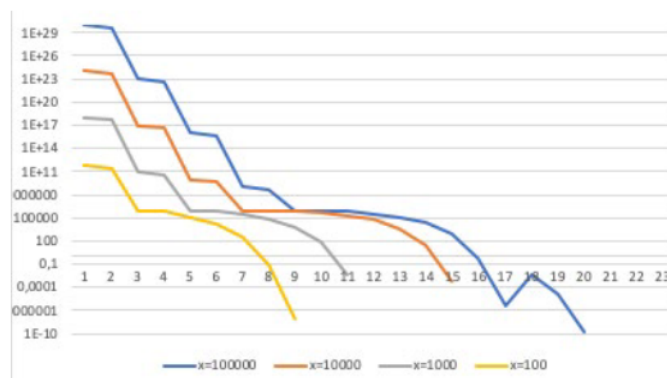
Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x) = e^x - 100$ con diferentes puntos de inicios alejados de la raíz



- a) Si $\Delta x_{n+1} \neq \Delta x_n$ en la función $f(x) = (x - 10)^6 - 10^6$, al aplicar el coeficiente de aceleración, se obtienen menos iteraciones.

Figura 11

Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x) = (x - 10)^6 - 10^6$ con diferentes puntos de inicios alejados de la raíz



Obsérvese cómo, sin importar la distancia del punto de inicio, se disminuyen las iteraciones con respecto a la Figura 5.

Validación de la convergencia del algoritmo iterativo para funciones de dos variables

Para $f(x, y) = e^{x*y} - 1$, se repiten las curvas de convergencia (ver Figura 12).

Figura 12

*Curvas de Convergencia de $|f(x)|$ en escala logarítmica por número de iteración para $f(x, y) = e^{x*y} - 1$ con diferentes puntos de inicios alejados de la raíz*



Conclusiones

Después de las aplicaciones, deducciones algebraicas y aplicaciones de funciones que ayudan a reducir el número de pasos, se puede obtener una generalización del algoritmo iterativo para algunas variables:

$$x_{j\ n+1} = x_{j\ n} + m_j \left(\sqrt[3]{-\frac{q_{j\ n}}{2} + \sqrt{\Delta_{j\ n}}} + \sqrt[3]{-\frac{q_{j\ n}}{2} - \sqrt{\Delta_{j\ n}}} - \frac{a_{j\ n}}{3} \right) \quad (53)$$

Donde j determina las variables y n el número de iteraciones.

El método de uso de resolución de una raíz cúbica es un proceso que converge mientras se cumplan las condiciones de los valores $\sqrt{\Delta_{j\ n}}$, y esto se obtiene si los valores de $\Delta_{j\ n} \geq 0$. Esto ayuda a definir intervalos de valores en donde no se puede iniciar.

Utilizar la memoria de los valores de las iteraciones, ayuda a definir modelos matemáticos que permitan acelerar ecuaciones iterativas.

Las funciones encontradas en las curvas de convergencias podrían convertirse en un patrón que determine ecuaciones iterativas para la búsqueda de una raíz en diferentes funciones.

Referencias

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

Arteaga, J., & Malakhaltsev, M. (2013). *Cálculo Vectorial*. Mexico: Cengage Learning Editores S.A.

- Burden, R., Faires, D., & Burden, A. (2017). *Análisis numérico*. México: Cengage Learning.
- Colley, S. (2012). *Vector Calculus*. New York: Pearson.
- Conte, S. D. (1980). *Elementary numerical analysis: algorithmic approach*. New York: McGraw-Hill.
- Marsden, J., & Tromba, A. (1991). *Cálculo vectorial*. México DF: Addison-Wesley Iberoamericana.
- Taylor, A. E., & Mann, W. R. (1983). *Advanced Calculus*. New York: Wiley.
- Weintraub, S. (1996). *Differential forms, a complement to vector calculus*. New York: Academic Press.
- Ivorra, C. (2020, 10 01). *Las fórmulas de Cardano-Ferrari*. Retrieved from Universidad de Valencia: <https://www.uv.es/ivorra/Libros/Ecuaciones.pdf>