

## **Otras variantes algorítmicas para la resolución de ecuaciones no lineales basándose en la técnica de iteración variacional.**

Other algorithmic variants for solving nonlinear equations based on the variational iteration technique.

Arnoldo Abraham Herrera Herrera  
Máster en Matemática Aplicada  
Estudiante del Doctorado en Matemática Aplicada  
UNAN – Managua, Nicaragua  
<https://orcid.org/0000-0003-3001-8861>  
[arnoldo.herrera@unan.edu.ni](mailto:arnoldo.herrera@unan.edu.ni)

Iván Augusto Cisneros Díaz  
Doctor en Matemática Aplicada  
UNAN – Managua, Nicaragua  
<https://orcid.org/0000-0003-2014-1946>  
[ivan.cisneros@unan.edu.ni](mailto:ivan.cisneros@unan.edu.ni)

## **RESUMEN**

Este trabajo aborda una continuidad de nuevas versiones algorítmicas sobre la técnica iteración variacional, la cual es un método iterativo para resolver ecuaciones no lineales de la forma  $f(x) = 0$ . En este sentido, el objetivo principal es generar nuevos algoritmos y esquemas iterativos que permitan obtener nuevas fórmulas y métodos iterativos.

También se estudia el desarrollo constructivo y la convergencia de cada uno de los métodos presentados bajo los cuales la técnica iteración variacional figura como eje fundamental para la resolución de diversos tipos de ecuaciones no lineales, por consiguiente, se crean nuevas fórmulas mediante procedimientos matemáticos basados en las variantes del método de Newton y las técnicas de iteración variacional.

La obtención de los principales esquemas iterativos de cada método mediante la deducción de su construcción, así como el análisis de convergencia mediante la aplicación computacional se realizaron en el lenguaje de programación Python. En efecto, se ejemplifican y se calculan raíces de ecuaciones no lineales de algunas funciones bases, utilizadas en los artículos científicos consultados, las cuales tienen características de ser continuas y diferenciables.

Por otra parte, se realiza una comparación entre algunos de los algoritmos existentes y los diseñados en esta investigación, utilizando los criterios de máximo y mínimo número de evaluaciones funcionales. Dichos aspectos son piezas fundamentales para la validez de las nuevas variantes algorítmicas para la resolución de ecuaciones no lineales basándose en la técnica de iteración variacional.

Según los resultados obtenidos después de las diversas comparaciones, los algoritmos presentan un excelente funcionamiento con respecto a los existentes en la literatura sobre esta área de conocimiento.

Palabras clave: algoritmos, convergencia, iterativos, métodos, comparación.

## **ABSTRACT**

This paper addresses a continuity of new algorithmic versions on the variational iteration technique, which is an iterative method for solving nonlinear equations of the form  $f(x) = 0$ . In this sense, the main objective is to generate new algorithms and iterative schemes that allow new iterative formulas and methods to be obtained.

It also studies the constructive development and convergence of each of the methods presented under which the technique variational iteration appears as a fundamental axis for solving various types of nonlinear equations, therefore, new formulas are created using mathematical procedures based on Newton's method variants and variational iteration techniques.

The obtaining of the main iterative schemes of each method by deducting its construction, as well as the convergence analysis by means of the computational application were done in the Python programming language. In fact, the roots of non-linear equations of some basic functions, used in the scientific articles consulted, which have characteristics of being continuous and differentiable, are exemplified and calculated.

On the other hand, a comparison is made between some of the existing algorithms and those designed in this research, using the criteria of maximum and minimum number of functional evaluations. These aspects are fundamental to the validity of the new algorithmic variants for solving nonlinear equations based on the variational iteration technique.

According to the results obtained after the various comparisons, the algorithms have an excellent function with respect to those existing in the literature on this area of knowledge.

Keywords: Algorithms, convergence, iteratives, methods, comparison.

## **Introducción**

El estudio continuo sobre la resolución de ecuaciones no lineales es cada día más interesante y sustancial, dado que se busca obtener soluciones aproximadas, de tal forma que permitan resolver problemas de la vida real y comprender en profundidad su proceso.

Entre los diversos tipos de métodos numéricos de carácter recursivo que se construyen y proponen siempre está latente una necesidad por mejorar el orden y rapidez de convergencia de los métodos iterativos empleados, así como la creación de nuevos métodos numéricos basados en otros ya existentes. Por lo antes expuesto, nuestro interés se centra en ¿Cómo determinar nuevos procedimientos matemáticos numéricos basados en las técnicas iterativas variacionales?

La intencionalidad de las diversas modificaciones (funciones auxiliares) que se pueden efectuar a un método, particularmente en la técnica de iteración variacional, es que permite acelerar aún más la convergencia de la solución, puesto que esta técnica en su versión original provee una serie convergente rápida y precisa, razón por la cual solo necesitan un número pequeño de términos para obtener una solución aproximada con alta precisión computacional. Este hecho aportó significativamente a la elección de este método para la elaboración de esta investigación, mismo que se podrá utilizar en la resolución de ecuaciones no lineales.

Otro aspecto de viabilidad que ofrece el trabajar con la técnica de iteración variacional es que se evidencia un proceso de transformación y adaptación de la modelización numérica computacional, mismo que se muestra de forma genuina y tangible durante el desarrollo de esta investigación.

La conveniencia del trabajo está definida en mostrar diferentes técnicas iterativas para formular nuevos algoritmos del Método de Newton, a su vez la relevancia y su impacto están reflejados en poder generalizar el método de Newton mediante la técnica iterativa variacional, por tanto, esta acción es un fiel reflejo sobre la estimulación de la creatividad matemática para desarrollar nuevas formas del pensamiento abstracto.

En cuanto a la clasificación de los esquemas iterativos para resolver ecuaciones no lineales del tipo  $f(x) = 0$ ; estos se basan en el número de pasos del proceso, en este sentido un método puede ser de un solo paso o multipaso. (Cisneros, 2017). El método más conocido de un solo paso es el de Newton, el cual en cada paso del proceso necesita evaluaciones funcionales  $(f, f')$ ; donde  $f$  representa la evaluación en la función original y  $f'$  la evaluación en su derivada, el esquema correspondiente según (Melan, 1997) es:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Los sucesivos intentos de mejorar su eficiencia, en términos de velocidad de convergencia, dieron sus frutos en forma de otros métodos de un solo punto: los esquemas de Halley, Chebyshev, etc. Haciendo referencia al primer esquema mencionado, (Noor & Noor, 2007) proponen una nueva variante donde muestra que se puede utilizar el método de Newton como un predictor y el esquema de Halley como corrector con la finalidad de obtener un mejor orden de convergencia.

Diversos autores señalan que los métodos de un solo punto antes mencionados tiene convexidad logarítmica, de acuerdo con (Diloné, 2013) el esquema correspondiente es

$$L_f(x_k) = \frac{f(x_k)f''(x_k)}{[f'(x_k)]^2}$$

El uso de este operador permite acelerar la convergencia hasta tercer orden, pero con la condición de evaluar la segunda derivada de la función no lineal en cada iterado, lo cual desde fines prácticos computacionales no es conveniente; esta y otras limitaciones de los métodos punto a punto llevan, en la segunda mitad del siglo XX, al desarrollo de métodos multipaso que, mediante expresiones iterativas sencillas y evaluaciones de derivadas de bajo orden de la función no lineal permiten obtener órdenes de convergencia elevados. (Cisneros, 2017).

A partir de las distintas limitantes que tienen los métodos punto a punto surge a finales de la segunda mitad del siglo XX; el desarrollo de métodos multipaso, los cuales según (Cisneros, 2017) mediante expresiones iterativas sencillas y evaluaciones de derivadas de bajo orden de la función no lineal permiten obtener órdenes de convergencia elevados. Esto se traduce a que existe una mayor factibilidad computacional para trabajar con este tipo de métodos.

Un esquema propuesto por (Traub, 1964), es el método de Potra-Ptak el cual posee la siguiente estructura

$$y_k = x_k - \frac{f(x_k)}{f'(x_k)}$$

$$y_{k+1} = y_k - \frac{f(y_k)}{f'(y_k)}$$

En su expresión iterativa este esquema parte del método de Newton como primer paso y un segundo paso consiste en aplicar Newton sobre el primero, pero evaluando únicamente la función en el primer paso y manteniendo la derivada evaluada en el iterado anterior. Este mismo autor demuestra que dicho método tiene orden de convergencia tres, es decir

$$\lim_{x \rightarrow \infty} \frac{|x_{k+1} - \varepsilon|}{|x_k - \varepsilon|} = \lambda, \quad \lambda > 0$$

Este método tiene orden de convergencia tres, posteriormente se desarrollaron los métodos clásicos de la familia Chebyshev-Halley, Jarratt, Ostrowski y los métodos de King, los cuales no hacían uso de la evaluación de la segunda derivada, todo ellos con orden de convergencia cuatro.

En relación a los métodos de la familia King, estos consideran tres y cuatro puntos para resolver ecuaciones no lineales y tienen un mejor desarrollo para funciones racionales. Además, en el trabajo realizado por (King, 1973) se presenta la estructura correspondiente para esta familia de métodos, la cual es:

$$w_i = x_i - \frac{f(x_i)}{f'(x_i)}$$

$$y_{k+1} = y_k - \frac{f(w_i)}{f'(w_i)} \frac{f(x_i) + \alpha f(w_i)}{f(x_i) + (\alpha - 2)f(w_i)}$$

En cuanto a la última década del siglo XX y la primera del siglo XXI aparecen numerosos trabajos en los que se diseñan tanto métodos como familias de métodos óptimos de órdenes crecientes (multipaso). Para ellos se utilizan diferentes técnicas, pero con una raíz común: la composición de métodos conocidos (básicamente el método de Newton) la posterior eliminación de algunas de las evaluaciones funcionales añadida en el proceso.

En los últimos años se analizan métodos iterativos multipunto óptimos con derivada para la solución de ecuaciones no lineales, donde dichos esquemas son construidos mediante la técnica de iteración variacional, la cual tienen la estructura del esquema de Potra-Ptak. Entre las principales investigaciones de la técnica de iteración variacional está el trabajo realizado por (Inokuti y otros, 1978) donde muestran que la principal ventaja de este método es la flexibilidad para dar soluciones aproximadas a problemas no lineales sin linealización ni discretización, en consecuencia, determinan que este método es simple y eficaz.

Entre los estudios más recientes y relevantes están los realizados por (Noor & Noor, 2007) en ellos se abordan nuevos métodos iterativos de cuarto orden de convergencia, modificaciones del método de House Holder y un método iterativo de quinto orden de convergencia para la resolución de ecuaciones no lineales. En todos estos trabajos figura como soporte principal e innovador modificaciones iterativas a esquemas ya construidos.

Posteriormente (Noor y otros, 2011) crean diferentes tipos de técnicas iterativas donde presentan el proceso constructivo y convergente de cada una, así mismo ejemplifican su eficiencia mediante la consideración de ecuaciones no lineales y su comparación con la solución de otros métodos iterativos. Además, enfatizan en la gran ventaja que proporciona trabajar con funciones auxiliares de tipo exponencial.

Otro documento que alude al mejoramiento y optimización de la técnica de iteración variacional es el creado por (Noor y otros, 2011), el cual aborda el análisis de algunos nuevos métodos iterativos para encontrar múltiples raíces de ecuaciones no lineales utilizando la técnica de iteración variacional. La técnica genera los métodos de orden superior. Aquí, los nuevos métodos son de segundo y tercer orden de convergencia. También muestran varios ejemplos para ilustrar la eficiencia de estos métodos. Luego (Noor y otros, 2012), aplican la técnica de iteración variacional para la resolución de sistemas de ecuaciones.

El trabajo de (Cisneros, 2017) permite mostrar a detalle la versatilidad computacional de la técnica de iteración variacional, así como diferentes procesos matemáticos para la obtención de esquemas iterativos principales y a su vez en la creación genuina e inédita de algoritmos iterativos para resolución de ecuaciones no lineales.

En dicho trabajo, se puede corroborar la versatilidad y el buen comportamiento de la aplicación de los polinomios de Adomian e iteración variacional, para la obtención de la generalización

del método de Newton en la solución de ecuaciones no lineales, a su vez la generación de nuevos métodos iterativos mediante la utilización de la serie de Taylor, la descomposición de Adomian y la técnica iterativa variacional.

Una investigación reciente es la creada por (Herrera & Cisneros, 2022) en la cual expresan resultados sustanciales sobre novedosos algoritmos para la resolución de ecuaciones no lineales mediante la técnica de iteración variacional. Además, enfatizan en la deducción, programación y comparación entre los algoritmos existentes y los nuevos algoritmos construidos, siendo estos aspectos primordiales para la demostración auténtica sobre la validez de estos.

Todo el abordaje expuesto permite confirmar que la técnica de iteración variacional tiene gran relevancia, por la rápida convergencia y la facilidad computacional para programarse, además que se tiene una alta precisión para el cálculo de las aproximaciones a las soluciones exactas para ecuaciones no lineales.

## **MATERIALES Y MÉTODOS**

En este acápite se muestra la metodología aplicada, así como una perspectiva general de la deducción y generalización del método de Newton mediante la técnica de iteración variacional para la resolución de ecuaciones no lineales, dicho desarrollo filosófico fue retomado del trabajo de (Herrera & Cisneros, 2022), permitiendo así obtener nuevos esquemas iterativos y fórmulas iterativas para el hallazgo de las raíces reales de ecuaciones no lineales consideradas en esta investigación.

La metodología empleada en este trabajo siguió las siguientes etapas:

- a. Revisión del estado del arte sobre los métodos iterativos existentes.
- b. Estudio de diversos procedimientos matemáticos de las técnicas de iteración variacional, así como sus diferentes aplicaciones a la teoría de los métodos iterativos y otros tópicos relacionados.
- c. Selección de funciones auxiliares que generaron diversos esquemas iterativos que permitieron obtener nuevos métodos iterativos, basados en las técnicas de iteración variacional.
- d. Programación Orientada a Objetos (POO) en Python de los algoritmos obtenidos por las técnicas de iteración variacional.

- e. La comparación realizada se basó en el número de iteraciones de cada método para determinar la solución real de la ecuación no lineal.

En la técnica iterativa variacional permite obtener métodos conocidos del tipo Ostrowsky, los cuales se desarrollan y retoman la investigación elaborada por (Shah, 2012). También se analizan sus criterios de convergencia y principales esquemas iterativos, por consiguiente, se muestran varios ejemplos donde se analizan la comparación entre dichos métodos. Otro aspecto a señalar es que todos los algoritmos son construcción genuina e inédita de esta investigación.

Por otra parte, para los dos casos de la técnica de iteración variacional se expresa en base a las funciones originales y sus derivadas, lo cual tiene por finalidad la obtención de nuevos algoritmos iterativos. Además, se menciona la importancia de utilizar ciertos valores numéricos y funciones para obtener una mejor aproximación a la solución de la ecuación no lineal.

### ***1.1. Caso 1 de Iteración Variacional***

$$H(x) = x + \lambda f(x)g(x)$$

Considerando

$$H(x) = x + \lambda f(x)g(x)$$

Donde  $g(x)$  es una función arbitraria y  $\lambda$  un multiplicador de Lagrange. Luego se tiene

$$\begin{aligned} H'(x) &= x' + \lambda[f(x)g(x)]' \\ &= x + \lambda f'(x)g(x) + f(x)g'(x) \end{aligned}$$

La condición de optimicidad, implica que

$$\lambda = -\frac{1}{f'(x)g(x) + f(x)g'(x)}$$

Resultando

$$H(x) = x - \frac{f(x)g(x)}{f'(x)g(x) + f(x)g'(x)}$$

Por tanto, el esquema principal correspondiente a este caso de la técnica de iteración variacional es:

$$\begin{aligned} x_n &= H(x_n) \\ x_{n+1} &= x_n - \frac{f(x_n)g(x_n)}{f'(x_n)g(x_n) + g'(x_n)f(x_n)} \end{aligned}$$

## 1.2. Caso 2 de Iteración Variacional

$$H_1(x) = \phi(x) + \lambda[f(x)g(x)]^p, \quad p = 1$$

Considerando

$$H(x) = x + \lambda f[(x)g(x)]$$

Donde  $\phi(x)$  es una función iterativa de orden  $p \geq 1$ ,  $g(x)$  es una función arbitraria y  $\lambda$  un multiplicador de Lagrange. Luego se tiene la condición de optimicidad definida por

$$\lambda = -\frac{\phi'(x)}{p[f(x)g(x)]^{p-1} + (f(x)g(x))'}$$

Sustituyendo en el esquema considerado inicialmente, se obtiene

$$H'(x) = \phi(x) - \frac{\phi'(x)(f(x)g(x))}{p[f'(x)g(x) + f(x)g'(x)]}$$

Ahora, tomando  $p = 1$

$$H'(x) = \phi(x) - \frac{\phi'(x)(f(x)g(x))}{f'(x)g(x) + f(x)g'(x)}$$

Considerando función iterativa

$$\phi(x) = x - \frac{f(x)}{f'(x)}$$

Derivando esta función, resulta

$$\phi'(x) = \frac{f(x)f''(x)}{f'(x)f'(x)}$$

$$\phi'(x) = \frac{f(x)f''(x)}{[f'(x)]^2}$$

Por consiguiente, el esquema principal correspondiente a este otro caso de la técnica de iteración variacional es:

$$H_1(x) = \phi(x) - \frac{\phi'(x)[f(x_n)g(x_n)]}{[f(x_n)g(x_n)]'}$$

Nótese que si  $\phi(x) = x$  se obtiene el método iterativo del caso I de la técnica de iteración variacional.

A su vez dicho esquema tiene un método iterativo asociado, definido de la siguiente forma

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{[f(x_n)]^2 f''(x_n) g(x_n)}{[f'(x_n)]^2 [f'(x_n)g(x_n) + g'(x_n)f(x_n)]}$$

### ❖ Convergencia de los Casos de Iteración Variacional

**Teorema:** Sea  $\phi(x)$  una función de iteración de orden  $p \geq 1$ ; sea  $r$  una solución de la ecuación no lineal  $f(x)$ . Supongamos que  $\phi(x)$  es continuamente diferenciable en  $r$ ; entonces el esquema iterativo anterior tiene orden de convergencia de al menos  $p + 1$ .

Demostración: Supongamos que

$$H'(x) = \phi(x) - \frac{\phi'(x)(f(x)g(x))}{p[f'(x)g(x) + f(x)g'(x)]}$$

Además, que  $r$  es una solución de  $f(x)$  y que  $\phi(x)$  es una función iterativa convergente de orden  $p$ , entonces se tiene que

$$f(r) = 0$$

$$\phi(r) = r$$

$$\phi'(r) = 0$$

⋮

$$\phi^{(p-1)}(r) = 0$$

$$\phi^p(r) \neq 0$$

Luego considerando a

$$\beta(x) = \frac{f(x)g(x)}{f'(x)g(x) + f(x)g'(x)}$$

Entonces

$$\beta(r) = 0$$

Así, usando el hecho que

$$f(r) = 0, \beta(r) = 0 \text{ y } \beta'(r) = 0$$

Resulta que

$$H_1(x) = \phi(x) - \frac{1}{p}\phi'(x)\beta(x)$$

De esto, se deduce que

$$H_1(r) = r$$

Ahora realizando derivadas de orden  $p$  e la función  $H(x)$ , se obtiene

$$H^{(p)}(x) = \phi^{(p)}(x) - \frac{1}{p} \sum_{n=0}^p \binom{p}{n} \phi^{(p-n+1)}(x) \beta^{(n)}(x)$$

y

$$H^{(p+1)}(x) = \phi^{(p+1)}(x) - \frac{1}{p} \sum_{n=0}^p \binom{p+1}{n} \phi^{(p-n+2)}(x) \beta^{(n)}(x)$$

De las igualdades anteriores se infiere que

$$H^{(p)}(r) = 0$$

Debido a que

$$\phi^{(1)}(r) = \phi^{(2)}(r) = \dots = \phi^{(p-1)}(r) = 0, \quad \beta(r) = 0, \quad \beta'(r) = 0$$

y

$$H^{(p+1)}(r) = -\frac{1}{p} \phi^{(p+1)}(r) \neq 0$$

porque

$$\phi^{(p+1)}(r) \neq 0$$

Este hecho demuestra que la función iterativa  $H(x)$  tiene orden de convergencia de al menos  $p + 1$ .

Como un corolario, se tiene que si

$$\frac{f(x)g(x)}{f'(x)g(x) + f(x)g'(x)} = x - H(x)$$

Entonces, la función iterativa  $H(x)$  toma la forma

$$H(x) = x - p \frac{x - \phi(x)}{p - \phi'(x)}$$

la cual tiene al menos orden de convergencia  $p + 1$ .

❖ **Nuevas Variantes Algorítmicas del Caso 1 de Iteración Variacional.**

**Algoritmo 1:** Sea la función auxiliar definida  $g(x) = e^{-\alpha \frac{f(x)}{f'(x)}}$ ,  $\alpha = \frac{1}{2} \frac{f''(x)}{f'(x)}$  entonces

$$g'(x) = -\frac{1}{2} e^{-\frac{1}{2} \frac{[f(x)]^2}{[f'(x)]^2}} \left( \frac{[f'(x)]^3 f''(x) + f(x)[f'(x)]^2 f''(x)}{[f'(x)]^4} \right)$$

Luego de efectuar las sustituciones de las funciones en el esquema principal y la realización de operaciones algebraicas correspondientes para su simplificación asociada, se obtiene el nuevo esquema iterativo, que toma la forma:

$$x_{n+1} = x_n - \frac{f(x_n)[f'(x_n)]^4}{[f'(x_n)]^5 - \frac{1}{2} [f(x_n)[f'(x_n)]^3 f''(x_n) + [f(x_n)]^2 [f'(x_n)]^2 f''(x_n)] + [f(x_n)]^2 f'(x_n)[f(x_n)]^2}$$

**Algoritmo 2:** Sea la función auxiliar definida  $g(x) = e^{-\frac{1}{f'(x)}}$  entonces

$$g'(x) = e^{-\frac{1}{f'(x)}} \left( \frac{f''(x)}{[f'(x)]^2} \right)$$

Realizando las simplificaciones correspondientes de las funciones involucradas se obtiene que el esquema iterativo toma la forma

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

❖ **Nuevas Variantes Algorítmicas del Caso 2 de Iteración Variacional.**

**Algoritmo 3:** Sea  $g(x) = e^{-\alpha f(x)}$  con  $\alpha = \frac{1}{2} f'(x)$  entonces

$$g'(x) = e^{-\frac{1}{2} f(x)f'(x)} \left( -\frac{1}{2} (f''(x)f(x) + [f'(x)]^2) \right)$$

Luego el esquema iterativo toma la siguiente forma

$$\left\{ \begin{array}{l} y_n = x_n - \frac{f(x_n)}{f'(x_n)} \\ x_{n+1} = y_n - \frac{[f(x_n)]^2 f''(x_n)}{[f'(x_n)]^3 - \frac{1}{2} [[f(x_n)]^2 [f'(x_n)]^2 f''(x_n) + f(x_n)[f'(x_n)]^4]} \end{array} \right.$$

**Algoritmo 4:** Sea  $g(x) = e^{-\alpha f(x)}$  con  $\alpha = \frac{1}{2f'(x)}$  entonces

$$g'(x) = e^{-\frac{1}{2f'(x)}f(x)} \left( -\frac{1}{2} \left( \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2} \right) \right)$$

Luego el esquema iterativo toma la siguiente forma

$$\begin{cases} y_n = x_n - \frac{f(x_n)}{f'(x_n)} \\ x_{n+1} = y_n - \frac{[f(x_n)]^2 [f'(x)]^2 f''(x_n)}{[f'(x_n)]^5 - \frac{1}{2} [f(x_n) [f'(x_n)]^4 - [f(x_n)]^2 [f'(x_n)]^2 f''(x_n)]} \end{cases}$$

## RESULTADOS

Los resultados obtenidos ilustran la eficiencia de los nuevos métodos desarrollados. La comparación se realiza entre el método de Newton y los algoritmos creados y tomando como base fundamental el número de iteraciones para encontrar dicha raíz. Además, funciones bases, aquellas funciones que fueron consideradas en las investigaciones de (Cisneros, 2017) y (Prieto, 2008) las cuales poseen características de ser funciones continuas y diferenciables.

| Ecuación               | Raíz (50 cifras decimales)                           |
|------------------------|--|
| $e^{-x} - x^3 = 0$     | 0.77288295914921012474962935812072828412055969238281 |
| $x - \cos x = 0$       | 0.73908513321516067229310920083662495017051696777344 |
| $e^{-x} + 2 \ln x = 0$ | 0.79851808532225987402597411346505396068096160888672 |
| $x + \sin x - 2$       | 1.10606015770627186256547247467096894979476928710938 |

Otro aspecto de trascendencia en cuanto a la selección de este tipo de funciones es que el tratamiento continuo de ellas permite mostrar de forma muy diferenciada las ventajas que poseen, entre ellas destaca la facilidad computacional al momento de programarse en Python o cualquier otro lenguaje de programación. De acuerdo con (Severance, 2020), Python es un lenguaje de alto nivel en el cual se han programado todos los algoritmos desarrollados

aplicando la filosofía de Programación Orientado a Objeto (POO) y en todos los criterios de parada de los diferentes códigos fuente se utiliza un nivel de tolerancia de  $10e^{-15}$ .

Es evidente que para obtener resultados concretos a partir de algoritmizaciones es trascendental tener en consideración el nivel de significancia para cada una de las ecuaciones no lineales, en palabras de (Herrera & Cisneros, 2022) el nivel de significancia en cuanto al criterio de parada, es traducido como la eficiencia de aproximación que se tiene para la solución real de la ecuación no lineal considerada. Este hecho cobra mayor auge cuando se busca minimizar el nivel de tolerancia, por eso es razonable que los distintos algoritmos construidos efectúen un número mayor de iteraciones para alcanzar el valor de tolerancia establecido.

Entre las principales ventajas que ofrece la POO es que se superan las limitaciones de la programación estructurada mediante una versión de programación donde las tareas se resuelven a través de la colaboración de objetos de clases, los cuales figuran como elementos principales para la creación de instancia. En este sentido, la estructuración y sistematización de la programación realizada en los distintos algoritmos constituyen el pilar fundamental de la POO aplicada, más aún al momento de ser llamadas o invocadas contribuye significativamente a mostrar resultados de forma óptima en un lapso de tiempo menor al que se llevaría realizándolo de modo usual.

En relación a los principales criterios que aseguran la funcionalidad y efectividad de los diversos algoritmos se encuentra el valor semilla considerado para alcanzar la aproximación deseada. En todos los casos dicho valor fue muy próximo a la solución real de cada ecuación, esto con la finalidad de garantizar una convergencia con un número menor de iteraciones, así como sus posibles modificaciones o variantes a partir de otros valores semillas.

En el proceso de comparación entre el algoritmo del método de Newton y los creados se utiliza el mismo valor semilla, así como los demás criterios anteriormente descrito, es decir todos funcionan bajo las mismas condiciones preestablecidas, en consecuencia, cuando los resultados alcanzados son iguales o superiores significa que las propuestas de nuevos algoritmos para ese tipo de ecuaciones no lineales son totalmente eficientes y óptimas.

En la siguiente tabla, a partir de las ecuaciones anteriormente descritas se muestra la comparación entre el método de Newton y los nuevos algoritmos construidos, basándose en el número de iteraciones que requiere cada método iterativo para alcanzar la aproximación establecida de acuerdo a la ecuación no lineal correspondiente.

| <b>Ecuación</b>        | <b>Valor Semilla</b> | <b>Método de Newton</b> | <b>Algoritmo 1</b> | <b>Algoritmo 2</b> | <b>Algoritmo 3</b> | <b>Algoritmo 4</b> |
|------------------------|----------------------|-------------------------|--------------------|--------------------|--------------------|--------------------|
| $e^{-x} - x^3 = 0$     | 0.7                  | 4                       | 3                  | 3                  | 4                  | 4                  |
| $x - \cos x = 0$       |                      | 4                       | 4                  | 3                  | 4                  | 4                  |
| $e^{-x} + 2 \ln x = 0$ |                      | 4                       | 3                  | 4                  | 4                  | 4                  |
| $x + \sin x - 2 = 0$   | 1.5                  | 5                       | 4                  | 5                  | 4                  | 5                  |

En los cuatro algoritmos diseñados el orden máximo de derivada es 3, debido a que desde la construcción y deducción que se tiene a partir de la función auxiliar establecida y el esquema principal, se concluye que desde fines computacionales - matemáticos trabajar con derivadas de mayor orden no es conveniente por el número de evaluaciones funcionales y el coste computacional de las mismas.

Por otra parte, el número de iteraciones de los algoritmos esta vinculado directamente al tipo de ecuación no lineal que se pretender resolver, este hecho queda manifestado en su totalidad a partir de los resultados que se muestran para cada ecuación. Específicamente, para las ecuaciones de tipo exponencial, las cuales desde diversas perspectivas son consideradas como ecuaciones bien portadas bajo la siguiente filosofía, que la  $n$  - ésima derivada de ella coincide consigo misma o bien se transforma en una expresión que involucra la ecuación original y esto es sumamente valioso debido a que cuando se expresa en el esquema principal permite múltiples simplificaciones.

Una nueva ecuación no lineal estudiada en esta investigación y que no es de tipo exponencial pero que sin duda posee características particulares y especiales es la ecuación:  $x + \sin x + 2 = 0$ . En síntesis, las distintas comparaciones realizadas entre el método de Newton y los algoritmos se sustentan mayormente en que todos fueron programados bajo las mismas condiciones y defiriendo únicamente en su propia naturaleza.

# CONCLUSIONES

Los resultados de relevancia alcanzados en este trabajo investigativos se detallan a continuación:

Demostración detallada sobre el proceso de convergencia asociado a cada uno de los casos de iteración variacional, el cual se determina directamente a partir de la construcción y deducción de cada esquema principal.

Generalización del método de Newton mediante la Técnica de Iteración Variacional para generar nuevos esquemas y métodos iterativos, por tanto, los 4 algoritmos creados para la resolución de ecuaciones no lineales se deducen a partir de 2 casos especiales de dicha técnica.

Se encontraron funciones auxiliares pertinentes, específicamente de tipo exponencial y una trigonométrica, las cuales contribuyeron a determinar los nuevos algoritmos que superan o son equivalentes al orden de convergencia del método de Newton.

Todos los algoritmos desarrollados fueron programados en el lenguaje de programación de alto nivel Python, bajo la filosofía de Programación Orientado a Objeto (POO) y en todos los criterios de parada de los diferentes códigos fuente se utiliza un nivel de tolerancia aproximadamente de  $10e^{-15}$ . Cada uno de los algoritmos construidos verificaron la condición de ejecutarse en un número menor o igual de iteraciones que el método de Newton. Esto se realizó con el objetivo de comparar los algoritmos óptimos y los que son equivalentes (número de iteraciones) al método de Newton.

# Bibliografía

- Cisneros, I. (2017). *Algoritmos basados en los Polinomios de Adomian e Iteración Variacional para la resolución de ecuaciones no lineales*. Managua: Unan - Managua.
- Diloné, M. (2013). *Métodos iterativos aplicados a la ecuación de Kepler*. España: Universidad de la Rioja.
- Herrera, A., & Cisneros, I. (2022). Algoritmización para la resolución de ecuaciones no lineales mediante la técnica de iteración variacional. *Revista de FAREM - Estelí. Medio Ambiente, Tecnología y Desarrollo Humano*(41), 60-76. <https://doi.org/https://doi.org/10.5377/farem.v11i41.13886>
- Inokuti, M., Sekine, H., & Mura, T. (1978). General Use of the Lagrange Multiplier in Nonlinear Mathematical Physics. *Nemat-Nasser*, pp. 156-162.
- King, R. .. (1973). A family of fourth-order methods for nonlinear equations. *SIAM* .
- Melan, A. (1997). Geometry and Convergence of Euler's and Halley's Methods. . *SIAM*, 728-735.
- Noor, K., & Noor, M. (2007). Iterative methods with fourth-order convergence for nonlinear equations. *Appl. Math. Comput*, 221-227.
- Noor, M., Shah, F., Noor, K., & Al-Said, E. (2011). Variational iteration technique for finding multiple roots of nonlinear equations. *Sci. Res. Essays*, 1344–1350.
- Noor, M., Waseem, M., Noor, K., & Al-Said, E. (2012). Variational iteration technique for solving a system of nonlinear equations. . *Optim. Lett.* DOI: 10. 1007/s11590-012-0479-3.
- Prieto, J. (2008). *Métodos Iterativos para Resolver Ecuaciones No Lineales*. Facultad de Ciencias, Universidad de los Andes.
- Severance, C. (2020). *Python para todos. Explorando la información con Python 3*. MI, USA: Ann Arbor.
- Shah, F. (2012). Variational iteration technique and Numerical Methods for solving nonlinear equations.
- Traub, J. (1964). *Iterative Methods for the Solution of Equations*. , , . New Jersey, USA.: Prentice-Hall Englewood Cliffs.

# Anexos

## 1. Declaración y sintaxis de funciones consideradas

### Ecuación 1

```
def f(self,x0):  
    return (self.x0-np.cos(self.x0))  
  
def fd1(self,x0):  
    return (1+np.sin(self.x0))  
  
def fd2(self,x0):  
    return (-np.cos(self.x0))  
  
def fd3(self,x0):  
    return (np.sin(self.x0))
```

### Ecuación 2

```
def f(self,x0):  
    return (self.x0-np.cos(self.x0))  
  
def fd1(self,x0):  
    return (1+np.sin(self.x0))  
  
def fd2(self,x0):  
    return (-np.cos(self.x0))  
  
def fd3(self,x0):  
    return (np.sin(self.x0))
```

### **Ecuación 3**

```
def f(self,x0):  
    return (np.exp(-self.x0)+2*np.log(self.x0))  
  
def fd1(self,x0):  
    return (-np.exp(-self.x0)+2/self.x0)  
  
def fd2(self,x0):  
    return (np.exp(-self.x0)-2/self.x0**2)  
  
def fd3(self,x0):  
    return (np.exp(-self.x0)+4*self.x0/self.x0**4)
```

### **Ecuación 4**

```
def f(self,x0):  
    return (self.x0+np.sin(self.x0)-2)  
  
def fd1(self,x0):  
    return (1+np.cos(self.x0))  
  
def fd2(self,x0):  
    return (-np.sin(self.x0))  
  
def fd3(self,x0):  
    return (-np.cos(self.x0))
```

## 2. Código fuente del Algoritmo de Newton

```
class Iterativos:
    def __init__(self):
        pass
    def Método_de_Newton(self, x0):
        self.x0=x0
        conta=0
        error=10e-25
        while True:
            xnn=self.x0-self.f(self.x0)/self.fd1(self.x0)
            if abs(xnn-self.x0)/xnn<=error:
                print("La solución es obtenida en ", conta, " iteracciones
{: .15f}".format(xnn))
                break
            conta+=1
            self.x0=xnn
        return ""
iterativos=Iterativos(); print("\nMÉTODO DE NEWTON")
print(iterativos.Método_de_Newton(0.7))
```

### 3. Código fuente del Algoritmo 1

```
def AAHH36(self,x0):
    conta=0
    self.x0=x0
    while True:
        #xnn=xn-(f(xn)*fd1(xn)**4)/(fd1(xn)**5-0.5*f(xn)*fd1(xn)**3*fd2(xn)-
0.5*f(xn)**2*fd1(xn)**2*fd3(xn)+f(xn)**2*fd1(xn)*fd2(xn)**2)

        xnn=self.x0-(self.f(self.x0)*self.fd1(self.x0)**4)/(self.fd1(self.x0)**5-
0.5*self.f(self.x0)*self.fd1(self.x0)**3*self.fd2(self.x0)
0.5*self.f(self.x0)**2*self.fd1(self.x0)**2*self.fd3(self.x0)+self.f(self.x0)**2*self.fd1(self.x
0)*self.fd2(self.x0)**2)

        if abs(xnn-self.x0)/xnn<=10e-15:
            print("La solución es obtenida en ", conta, " iteraciones {:.50f}".format(xnn))
            break
        conta+=1
        self.x0=xnn
    return ""
```

#### 4. Código fuente del Algoritmo 4

```
def AAHH54(self,x0):

    conta=0

    self.x0=x0

    while True:

        #xnn=yn-(f(xn)**2*fd1(xn)**2*fd2(xn))/(fd1(xn)**5-0.5*(f(xn)*fd1(xn)**4-
f(xn)**2*fd1(xn)**2*fd2(xn)))

        yn=self.x0-self.f(self.x0)/self.fd1(self.x0)

        xnn=yn-(self.f(self.x0)**2*self.fd2(self.x0))/(self.fd1(self.x0)**5-
0.5*(self.f(self.x0)*self.fd1(self.x0)**4-
self.f(self.x0)**2*self.fd1(self.x0)**2*self.fd2(self.x0)))

        if abs(xnn-self.x0)/xnn<=10e-15:

            print("La solución es obtenida en ", conta, " iteraciones {:.50f}".format(xnn))

            break

        conta+=1

        self.x0=xnn

    return ""
```